NONMEM USERS GUIDE
INTRODUCTION TO NONMEM 7.5.1

Robert J. Bauer
ICON Plc
Gaithersburg, Maryland

October 3, 2022

TABLE OF CONTENTS

I.1 What is updated/improved in NONMEM Version 7.5.1 versus NONMEM 7.5

The following bug fixes and minor adjustments have been made in NONMEM 7.5.1:

1) For $ABBR PROTECT, a base of 0 raised to a possitve power can result in INF and NAN, rather than 0. A work-around is to conditionalize the evaluation:

```
IF(BASE/=0.0) THEN
POW=BASE**(EXP1)
ELSE
POW=0.0
ENDIF
```

   Also, PROTECT functions within IF statements are not properly parsed, such as:
   IF(A/B .NE.0.0) …
   As a work-around, place parnetheses around the comparator:
   IF((A/B) .NE.0.0) …
   These two items now work properly in nm751.
2) For problems using analytical 2nd derivatives (such as $EST LAPLACE NOINTERACTION without NUMERICAL, or $SIM REQUESTSECOND), insufficient memory might be allocated in the PREDPP routine of ADVAN6, ADVAN8, ADVAN13, or ADVAN14. A work-around is to use -prdefault for small problems, and a larger than required $SIZES PC=xxx for larger problems. This bug has been in existence since NONMEM VI. NM751 now properly sizes these problems.
3) A new command line option, -do2test, tests if analytical 2nd derivative code is required for FSUBS construction. If required, NMTRAN will build FSUBS with analytical 2nd derivatives, otherwise it will build FSUBS without it, equivalent to the user entering $ABBR DERIV2=NO. See I.9 Dynamic Memory Allocation (NM72).
4) The manner in which default loss of degrees of freedom are handled for EM/BAYES has been modified for nm751 to be more consistent across estimation methods. Also, FOCE/ Lapalce methods may now incorporate loss of degrees of freedom adjustments as well. This has very little impact on problems with more than 10 subjects. See section I.45 Degrees of Freedom when Assessing Omegas.
5) An alternative evaluation of the objective function is provided for $LEVEL problems as an option (LEVOBJTYPE) in nm751. Also, while SLOW is still the default setting for $LEVEL FOCE problems, you may now also select NOSLOW or FAST for faster calculation. See *LEVOBJTYPE=(0 default) (NM75)* in section I.58 Adding Nested Random Levels Above Subject ID (NM75) for information and caveats.
6) The use of prior information in optimal design evaluation and estimation is now better integrated in nm751. Also, standard errors to thetas that are parameters to the residual variance (such as power terms to the predicted function, or serial intra-subject correlation parameters) may now be estimated when FIMDIAG=1. See I.72 Clinical Trial Design Evaluation and Optimization (NM75).
7) An option (INTERPTYPE) allows an additional interpolation and extrapolation algorithm for NPDE and NPD evaluation. See option *INTERPTYPE=0(default) (nm751)*, in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format
8) For ADVAN16/ADVAN17, RADAR5NM may use LAPACK routines for some of its core matrix algebra calculations. See the comments on R5ALG switch described in ..\pr\RADAR5U.f90

9) Even if a theta is not strict-mapping-mu-referenced but is associated with 0 valued OMEGA, nm751 will now have the FAST algorithm take effect for FOCE. (formerly, if omega is 0 valued, Mu-referencing had to be strictly a mapping relation for the theta). See I.48 The FAST Option for use with FOCE/ITS and Differential Equation ($DES) Models (NM74).
10) The DDE ODE's ADVAN16 and ADVAN18 have now greater internal coding flexibility to be coded with $ABBR DES=FULL or $ABBR DES=COMPACT.
11) New time delay algorithms and methods are presented in the following sections I.87 Modeling Discrete Delays on General Variables Using the ddexpand program (nm751); I.88 Solving Distributed Delayed Output Rate Prolems Using Repetition Variables RPTO/RPTI (nmVI); I.89 Transit Compartments for Gamma distributed Time Delays Using the ddexpand program (nm751); I.90 Creating Transit Compartments for Time Delays of any distribution, Using the ddexpand program (nm751); I.91 Using Discrete Delay Solvers for Time Delays of any distribution, Using the ddexpand program (nm751)
12) When $SIM TRUE=PRIOR is implemented, random thetas, omegas, and sigmas are generated for each SUBPROBLEM, and DV items are generated using these, but PRED, RES, and WRES values are calculated based on the initial $THETAS, $OMEGAS, $SIGMAS. This has been corrected, so that PRED, RES, and WRES are calculated using the randomly generated population parameters for that sub problem.
13) When ITYP>=1 for $NWPRI is set, the population parameters are incorrectly generated. This has been fixed.
14) Improvements to code have been made so that compiler errors from gfortran version 11 do not occur. This allows compilation of code into native M1 chip architecture for new Macintosh computers.
15) The lognormal pdf was not being calculated correctly. This has been corrected.
16) DOWHILE loops can now be nested, which is very useful for multiple independent variable integration, such as for distributed delay problems.

I.2 What is new in NONMEM Version 7.5 versus NONMEM 7.4

The following new features have been added in NONMEM 7.5:

1) Simulated etas and sigmas may retain separate random number sources during simulation. This is useful if you generate data from two separate data file templates which differ only in the number of data points among subjects, but you have the same number of subjects, and you wish to retain the same etas between the two data sets generated across subproblems. See *Extensions to Simulation Error Forgiveness (NM75)* in section I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)
2) Simulated eta samples may be rejected and call NONMEM to create a new sample with the EXIT statement. You can filter simulated values to create "normal truncated" distributions, for example. Please see *Extensions to Simulation Error Forgiveness (NM75)* in section I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)
3) A table, root.vpt, is created, which incorporates variance-covariances associated with etas as well as those associated with thetas (or omegas and sigmas). To present the

comparable total standard errors in a user defined table, set VARCALC=3. See *Requesting Standard Errors to User-Defined and PREDPP Variables (NM74)* in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

4) Variance-covariance matrix information can be imported from previous runs, and used to evaluate total standard errors of user defined items in tables, or to bring in variance-covariance matrices from alternative sources (IMP,SAEM,BAYES,SIR), as priors for TNPRI problems. See I.67 $RCOV and $RCOVI Record For Inputting Variance-Covariance information from another problem (NM75)
5) Increased flexibility has been added to $PRIOR TNPRI to allow additional sources to provide the prior information. See I.66 Improved Flexibility for using PRIOR information in TNPRI Problems (nm75).
6) Thetas may be simulated as a t-distribution. See *Simulating THETAS with t-Distribution (NM75)* in section I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74).
7) The option TBLN has been added to $EST METHOD=CHAIN and $CHAIN records to allow selecting a table within a raw output file. See I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records.
8) Additional records allow block specific setting of degrees of freedom information for LKJ decorrelation priors. Also, user-defined probability densities to the diagonal standard deviations to these blocks may be defined. See *TPU=0(default,NM75), $OLKJDF (0 default, NM75)* and subsequent items in section I.40 No U-Turn Sampling (NUTS) Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method (NM74).
9) A series of probability densities, their probability distributions, and random number generator routines are now available. SeeI.32 Probability Density Functions (NM742).
10) Delay Differential Equation solvers (ADVAN16, ADVAN18) are now available (ADVAN17 if there are also equilibrium compartments) is now available. For use of this ODE solver, please see Using the Delay differential equation Solvers with the ddexpand program for Discrete Delay Problems (nm75) in I.85 ddexpand Utility Program for Modeling Discrete Time Delays (NM74)
11) The IDFORMAT option in the $TABLE record allows a specialized format for ID (such as integer format) that is different from the other items in the table. See *IDFORMAT= I (NM75)* in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.
12) Thetas may be symbolically named at the $THETA record, in conjunction with the initial values specified. See *Symbolic Label Substitutions at $THETA, $OMEGA, and $SIGMA records (NM75)* in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items.
13) For MCMC Bayesian analysis the .phi table now contains the average of phi() values collected throughout the stationary distribution (positive iterations) phase, with variances. In addition, variances for phis/etc of levels above subject level (when using $LEVEL) are also now available. See *root.phi* and *root.phm (NM72)*) in section I.63 $EST: Additional Output Files Produced. In addition, shrinkage information is now available for MCMC Bayesian estimations.

14) Prepare a Single Burn-In for Multiple Stationary Chains. Because the burn-in of NUTS is computationally expensive, you can use the MSF file to perform a one-time burn-in in one control stream file, and then have subsequent control stream files use the burn-in information (including the mass density information) and evaluate their stationary phases in parallel (at different starting seeds). See *Prepare a Single Burn-In for Multiple Stationary Chains* in section I.40 No U-Turn Sampling (NUTS) Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method (NM74).
15) Optimal Design is now available. Please see I.72 Clinical Trial Design Evaluation and Optimization (NM75).
16) New options to replicate subjects in a data file are available, to facilitate simulations, avoiding the need to create replicate subjects data outside of NONMEM. See *$DATA REPL (NM75)* in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75)
17) Techniques to incorporate partial differential equations (PDE) into ODE problems are now available, using the doexpand utility. See *The DOPDE Method of Modeling PDE's*. in section I.84 doexpand Utility Program for Expanding Repetetive Code (NM74)
18) A new way of modeling steady state dosing is available. See I.24 An Empirical Method of Achieving Steady State (NM75).
19) A discussion on handling over-shoot when trying to capture extrema values within an ODE integration interval is available in I.21 ITASK_ and STOP_TIME: Avoiding overshoot in ADVAN9, ADVAN13, ADVAN14, and ADVAN15.
20) Individual parameter samples during BAYES analysis are stored in the .iph table when $EST …BAYES_PHI_STORE=1. See *root.iph (NM75)* in section I.63 $EST: Additional Output Files Produced.
21) Collect samples of individual BAYES parameters, while keeping the population parameters fixed. See *BIONLY=0 (default) (NM75)*in section I.39 Full Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method.
22) New format specifiers (FORMAT=q, c, QCSV, and CSV) offer compressed (no spaces) comma delimited file outputs. See *FORMAT=s1PE11.4 (default)* in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format; *FORMAT=s1PE12.5 (default)(NM75)* in section I.56 $COV: Additional Options and Behavior; and *DELIM=s or FORMAT=t or FORMAT=,* in section I.62 $EST: Format of Raw Output File
23) You may increase the number of significant digits displayed for population parameter results in the NONMEM report file. See *$FORMAT FMTN=3 (default) (NM75)*in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75)
24) A non-negative least squares algorithm is available for determining the support point probabilities in a non-parametric analysis. See *NPESTIM=0 (default, NM75)* in section I.33 Some Improvements in Nonparametric Methods (NM73).
25) Compartment names defined in $MODEL may be used as symbols in the control stream code. See *Symbolic Label Substitutions of Model Compartments (NM73,NM75)*in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75).
26) Additional options for SIR covariance of estimates sampling have been added to improve stability in sampling. See items *SIRMINWT=0.001, SIRMAXWT=1000.0 (default)*

(NM75); *SIR_CAPCORR=1.0 (default) (NM75)*; *SIRSAMPLE=0 (default) (NM74)* as list of values; *Using $RCOV with SIR sampling (NM75)* to input starting covariance of estimates from external sources, in I.56 $COV: Additional Options and Behavior.

27) When randomly selecting subjects using $SIML BOOTSTRAP, you may choose to analyze those subjects not selected. See *BOOTDATA=0 (Default) (NM75)*in section I.30 General New Options for $ESTIMATION Record (NM73).
28) Some improvements in assessing nested random effects above subject level. See I.58 Adding Nested Random Levels Above Subject ID (NM75).
29) Extensions to $DATA IGNORE exclusion of data records. See *PRED_IGNORE_DATA Feature (NM75)* in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75).

Bugs Fixed

I.3 What is new in NONMEM Version 7.4 versus NONMEM 7.3

The FAST option has been added to FOCE/ITS analysis, which can increase the speed of these analyses by up to 3-4 fold during the estimation of differential equation models, as well as the $COV step. The FAST method takes advantage of mapping analytical eta derivatives via MU referencing to evaluate theta analytical derivatives, increasing the speed and accuracy of derivatives required for FOCE assessment. See I.48 The FAST Option for use with FOCE/ITS and Differential Equation ($DES) Models (NM74).

An automated protection against floating point exceptions is now available. By setting $ABBR PROTECT, your code will be transposed to protect against floating point exceptions. See I.71 Stable Routines for Estimation Methods and Automated Protection Against Floating Point Exceptions (nm74).

The evaluation of weighted residuals to be outputted in tables can now be parallelized.
To turn off parallelization:

```
$TABLE … PARAFILE=OFF
```

See I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

The evaluation of final empirical Bayes estimates of etas (EBE's) after the estimation step (when FNLETA=1) can now be parallelized. To turn off parallelization during the FNLETA step,

```
$EST … FPARAFILE=OFF
```

Keep in mind that the PARAFILE option of $EST pertains to parallelization of the estimation step itself. See I.30 General New Options for $ESTIMATION Record (NM73).

The Simulation step can now be parallelized. By default, parallelization is not turned on, because simulation is very rapid anyway, and often does not need to be accelerated. To turn on parallelization during simulation

```
$SIML … PARAFILE=ON
```

or at the command line with the –simparon option:

```
nmfe74 … -simparon
```

And remember to permit constant seed patterns regardless of whether you choose parallelization or not with

```
$SIML … PARAFILE=ON RANMETHOD=P
```

See I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74) . When modeling with super-ID nested ETA levels ($LEVEL record is present), parallelization will not occur, since these etas are shared across individuals, and there is no guarantee that all subjects sharing the same etas will be simulated by the same process.

The noparametric analysis can now be parallelized. To turn off parallelization during the nonparametric step,

```
$NONP … PARAFILE=OFF
```

Negative times are now allowed in the data set. NM-TRAN has always allowed negative clock times when day-time translation is performed, and converted them to non-negative relative times for PREDPP. Now, any data set may contain negative values of time, and PREDPP will not consider this to be an error.

Specific table records may be excluded from being printed. A data item or defined variable may be identified on a $TABLE record as an EXCLUDE_BY variable, which if not 0, will exclude the record. For example:

```
$PK
...
EXCL=0
IF(ID.GE.45.AND.ID.LE.53) EXCL=1
...
$TABLE ID TIME DV IPRED CL V1 Q V2 ETAS(1:LAST) EXCLUDE_BY EXCL NOAPPEND FILE=exctable.par
NOPRINT
```

The table exctable.par will not list records from subjects 45 to 53. If more than one exclusion variable is listed, then if any of these have a non-zero value, the record will be excluded.

Furthermore, LASTONLY and FIRSTLASTONLY have been added as options to $TABLE, to request LAST record, or first and last records, of individual. Also, reserved variables may be accessed to determine whether the present record is first observation, last observation, first dose record, last dose record, etc. for refined decisions in modeling or table record outputting.

ONEHEADERALL or ONEHEADERPERFILE option may be used to have header information be written only at the beginning of the file, and never again.

See I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format

Standard errors of user-defined and PREDPP variables list in $TABLE records can be outputted. See I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format

Line -1000000007 in the .ext file has been added, indicating termination status information.

The first code is the termination status, and the subsequent codes are text message code numbers. See I.63 $EST: Additional Output Files Produced under *root.xml (NM72)* for interpretation termination status and text message codes.

Line -1000000008 in the .ext file has been added, partial likileihood with respect to the estimated parameters. Please see I.61 Format of NONMEM Report File.

Replacement code can map several variable names at once:

```
$ABBR REPLACE THETA(CL,V1,Q,V2)=THETA(1 TO 4)
```

In addition, these symbolic labels may be used in $TABLE references, and will appear in the NONMEM report file. Symbolic label substitutions will not be made in the additional output files *.ext, ,phi, etc), to maintain their third party software readability

See I.7 Expansions on Abbreviated and Verbatim Code under *$ABBR REPLACE feature for abbreviated code (NM73-NM75).*

Read in MSF files from previous versions. Choices are:

```
$MSFI myfile.msf VERSION=7.3.0
$MSFI myfile.msf VERSION=7.2.0
$MSFI myfile.msf VERSION=7.1.2
$MSFI myfile.msf VERSION=7.1.0
$MSFI myfile.msf VERSION=6.2
$MSFI myfile.msf VERSION=6.1
```

This allows you to use MSF files generated by an earlier version See I.29 Additional Control for $MSFI record (NM73).

More information in the xml file of file options. These are packaged in elements problem_options, sim_info, and estimation_options. See ..\util\output.xsd for the schema.

$SIM REWIND feature allows original data set to be used for all sub-problems. By default, if any data item is changed by a sub-problem, those data items remain changed for the start of the next sub-problem. If you want that each sub-problem start with using the values from the original data set, use the REWIND feature of $SIM. So, any changes to the data set made during simulation (when ICALL=4) of a sub-problem are used for that sub-problem only, and are not preserved for the next sub-problem. Keep in mind that any transgeneration you may have performed on the data set when using an $INFN when ICALL=1 will be considered original data set. For example:

```
$INFN
IF (ICALL==1) THEN
DOWHILE(DATA)
```

```
..modifying statements here
ENDDO
ENDIF
```

See *REWIND(NM74)* in I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)

$SIM NOSUPRESET feature allows the simulation seeds not to be reset with each iteration of a super-problem. By default, (SUPRESET), with subsequent iterations of a super-problem, the simulation seed is reset back to that listed in the $SIM record of the control stream file. It may be desirable that each iteration serves as a new random instance, so use NOSUPRESET. See *NOSUPRESET(NM74)* in I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)

A new kernel (mode) has been added to the Bayesian sampling of individual parameters, that reduces the correlation between Metropolis-Hastings generated samples at the individual parameter level, for BAYES and SAEM methods. See ISAMPLE_M1B, *ISAMPLE_M1B=2 (NM74)* in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method.

A Hamiltonian/No U-Turn Sampling algorithm has been implemented for BAYES analysis. This algorithm increases the efficiency of Markov-Chain Monte Carlo Bayesian sampling among population and individual parameters, by reducing the statistical correlation between samples. See I.40 No U-Turn Sampling (NUTS) Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method (NM74).

More flexible referencing of etas using the $TABLE ETAS() option. A list of etas may be referenced, or a TO/BY pattern may be given. See *Requesting a Range of Etas to be Outputted: Etas(x:y) (NM73)* in I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

More flexible referencing of $LEVEL information. Eta nestings may be given using TO/BY patterns. See I.58 Adding Nested Random Levels Above Subject ID (NM75).

Tools to expand code and data for problems with repetitive equations. A doexpand utility program (see I.84 doexpand Utility Program for Expanding Repetetive Code (NM74)) is available that allows repetitive code in control stream files to be expanded. Furthermore a EXTRADOSE feature has been added to the finedata utility program (see I.83 finedata Utility Program(NM73)**)** that replicates dose records.

NM-TRAN gives a new data warning. This warning appears for steady state dose records with AMT=RATE=II=0 when an analytic ADVAN routine is used. Such doses should only be present if there is an endogenous drug production term in the differential equations. With analytic ADVAN, this is not possible, hence the warning.

Additional EONLY option values are available to provide improved ways for evaluating stable objective function values after an SAEM analysis. See *EONLY=1* in I.36 Monte Carlo

Importance Sampling EM and *Obtaining the Objective Function for Hypothesis Testing After an SAEM Analysis* in I.38 Stochastic Approximation Expectation Maximization (SAEM) Method.

New algorithms for creating multivariate samples for importance sampling are offered. See *Note on the t-Distribution Sampling Density (DF>0), and its Use With Sobol Method (RANMETHOD=S)* in I.36 Monte Carlo Importance Sampling EM, and *IACCEPTL =0 (default) (NM74)* in I.36 Monte Carlo Importance Sampling EM.

The GRDQ option to provide a faster Importance sampling analysis when several thetas not mu-referenced need to be gradient assessed. See *GRDQ=0 (default) (NM74)* In section I.36 Monte Carlo Importance Sampling EM.

Degrees of freedom assignments for Omega priors and Sigma priors may be real numbers, not just integer. A gamma density that accepts non-integer arguments will create random samples for constructing inverse-Wishart distributed random Omega and Sigma matrices.

A new command line option, -nobuild, prevents a new nonmem executable from being built, particularly useful for a series of nonmem runs during bootstrap procedures. See I.9 Dynamic Memory Allocation (NM72).

A Monte Carlo based assessment of the variance-covariance after FO/FOCE/Laplace estimation can be obtained. See *Importance Sampling of the Variance-Covariance of the Parameter Estimates (NM74)*in section I.56 $COV: Additional Options and Behavior.

The EM algorithms will now report if a gradient to one of the thetas is zero, indicating improper model development.

A new output file, root.clt, is now constructed, that is the lower-triangular portion of the variance-covariance of the parameter estimates reported in root.cov. This is provided for easier pasting of the information as theta priors for a subsequent analysis. See I.63 $EST: Additional Output Files Produced.

The print iteration intervals to the parallelization log files can be controlled. The –parafprint option may be given at the command line, or parafprint option values may be given at specific $EST and $COV records. See I.73 Parallel Computing (NM72). Also see I.14 Interactive Control of a NONMEM batch Program, regarding the ctrl-F switch and sig paraf commands to interactively turn on and off parallelization log file printing.

The R Matrix of the variance-covariance of estimates for classical NONMEM methods may be preconditioned to improve precision and success rate of $COV step. See *Preconditioning the R Matrix to Improve Precision and Success Rate of $COV Step (NM74)* in section I.56 $COV: Additional Options and Behavior

A Saddle Point Reset may be conducted for FO/FOCE/Laplace to Improve Search for a Global Minimum. See *Resetting the Search to Circumnavigate Saddle Points and Detect*

Inestimable Parameters (NM74) in section I.25 $EST: Improvement in Estimation of Classical NONMEM Methods.

Priors to OMEGAS and SIGMAS may be specified as multi-variate Inverse Gamma densities, and priors to THETAS may be specified as multi-variate t-distributions. See *TTDF=0(default)* in I.40 No U-Turn Sampling (NUTS) Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method (NM74) and I.41 A Note on Setting up Prior Information.

The implementation of user-defined functions has been greatly expanded. See I.76 Expanded Syntax and Capacity for User-Defined Functions (FUNCA) (NM74).

The objective function now may include the constant term N*LOG(2pi). Specify LNTWOPI on the $EST record: See I.49 Options to Include Various Constants to the Objective Function (NM74).

The prior contribution has been added to the S matrix for classical methods of covariance assessment. Prior to nm74, it was recommended that MATRIX=R be used for classical NONMEM methods (FO/FOCE/ Laplace) when there are priors. As of nm74, the S matrix has a proper contribution component from the prior for the classical methods, as has been true for EM algorithms (ITS/MAP/SAEM) in earlier versions.

The CVODE solver system has been added as ADVAN14. Please see I.19 $SUBROUTINES: Yet Another New Differential Equation Solving Method: CVODES (ADVAN14) (NM74).

The IDA solver system has been added as ADVAN15. Please see I.20 $SUBROUTINES: Yet Another New Differential Equation Solving Method: IDAS (ADVAN15) (NM74)

Eta first derivative code creation can be turned off if not needed, and if the problem is very large. Please see I.77 First Derivative Assessments (NM72, NM74).

Advanced Relative Tolerance, Absolute Tolerance, and Steady State tolerences can be set in $SUBROUTINES. Please see I.18 $SUBROUTINES TOL, ATOL, SSTOL, and SSATOL: Additional control of relative and absolute tolerance (NM74).

When initial thetas are to be estimated, evaluations can now be done for FOCE and Laplace, not just for FO.

When all individual objective function values are zero for an EM/Bayes analysis, problem terminates with error rather than continuing.

Shrinkage information is presented in standard deviation and variance version. Please see I.61 Format of NONMEM Report File**,** ***Shrinkage and ETASTYPE (NM74)*****.**

Obtain random samples of individual etas, and use these for covariate and model diagnostics in accordance with Lavielle and Ribba (Pharmaceutical Research, 2016). See *ETASAMPLES=0 (default) (nm74)*in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method.

Conditional mean etas may be printed to the .phi and .phm table even for EM and BAYES methods. See *root.phi* in section I.63 $EST: Additional Output Files Produced for details.

Specified etas may be treated as if they are fixed effects when evaluating population diagnostics during the $TABLE step, particularly suitable for super-ID level etas. Please see *FIXEDETAS=(*number-list) *(NM74)* in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format

The $DATA record accepts a MISDAT value to allow NONMEM to report values with the missing data indicator. See *$DATA MISDAT (NM74)* in section I.7 Expansions on Abbreviated and Verbatim Code .

NPD May be evaluated for non-normal likelihood modeled data, if the user supplies the cumulative distribution function. See **Error! Reference source not found.** in Section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format**.**

The maximum number of characters for defining a compartment label in $MODEL has been expended to SD=30 (defined in SIZES.f90). For example:

```
$MODEL NCOMPARTMENTS=3
COMP(VERY_LONG_NAME_HERE)
COMP(VERY_LONG_NAME_HERE2)
COMP(VERY_LONG_NAME_HERE3)
```

Bugs found in version NONMEM 7.3.0, fixed in NONMEM 7.4.0

In addition, the following bugs discovered in NONMEM 7.3.0 have been fixed in NONMEM 7.4.0:

1. Two variables with names longer than six characters, and identical in the first six characters, defined in $PK, and used in $DES, will be seen as the same variable. Use variable names that differ in the first six characters. This occurs in NONMEM 7.1.0, 7.1.2, 7.2.0, and 7.3.0. A workaround is to move all assignment statements for variables whose first 6 characters match to $DES.

2. If more than one power operator is used within an IF/THEN block that defines random variables, this may cause 0 gradients to be produced for some Omega parameters, and no change in their values from the initial estimates for classical NONMEM methods (FO/FOCE/Laplace). For example:

```
IF(AGE>50) THEN
  TVCL=THETA(1)*(AGE/50)**THETA(3)
  CL=TVCL*EXP(ETA(1))
  TVV=THETA(2)*(AGE/50)**THETA(4)
```

```
  V=TVV*EXP(ETA(2))
ENDIF
```

OMEGA(1) will have a 0 gradient. Modifying the code so that random variables are not defined in IF/THEN blocks containing power operators will resolve this, for example:

```
IF(AGE.GT.50) THEN
TVCL=THETA(1)*(AGE/50)**THETA(3)
TVV=THETA(2)*(AGE/50)**THETA(4)
ENDIF
CL=TVCL*EXP(ETA(1))
V=TVV*EXP(ETA(2))
```

This bug occurs in NONMEM 7.2.0. and NONMEM 7.3.0. In NONMEM VI, NONMEM 7.1.and, NONMEM 7.1.2, the bug reveals itself only with LAPLACE no-interaction problems, and the result is typically a less accurate objective function value by 1 or 2 units.

3. When using $LEVEL, if Simulation is performed, followed by Estimation, an allocation error occurs. One work-around is to perform the Simulation (using ONLYSIMULATION option) with one control stream file, followed by Estimation in a subsequent control stream file. Another workaround is to have a non-useful $PROB with an $EST record, and no $SIM record, followed by a second $PROB performing the Simulation and Estimation problem you actually want to run, for example:

```
$PROB
...
$LEVEL
SID=(3[1],4[2])
CID=(5[3],6[4])
...
$EST METHOD=1 INTERACTION PRINT=1 NSIG=2 FNLETA=0 NOABORT MAXEVAL=0
...
$PROB
...
$LEVEL
SID=(3[1],4[2])
CID=(5[3],6[4])
...
$SIMULATION (567811 NORMAL) (2933012 UNIFORM) SUBPROBLEMS=1
$EST METHOD=1 INTERACTION PRINT=1 NSIG=3 NOABORT MAXEVAL=9999
...
```

4. When NSUBS > 9999 in $SIMULATION, the subproblem number is output as **** in the NONMEM report file. This affects only NONMEM 7.3.0, which allows NSUBS to be greater than 9999.

5. The $ETAS/PHIS statement does not always find the desired record in the .phi file for some compiler versions of gfortran. The bug is due to using an incorrect type variable in the NONMEM code, causing an internal clobber on another variable. Gfortran 4.4.0 is

susceptible to this bug. Versions of gfortran that are not susceptible to this bug are gfortran 4.6.0 and 4.6.3, and it is recommended that one uses these compiler versions.

6. After using $SIM BOOTSTRAP, if a new data file is to be used in a subsequent problem using $DATA, this file will be read into NONMEM incorrectly. To prevent this from happening, insert a dummy $PROB that runs a non-useful estimation record, then execute the problem that is using another data file. For example:

```
$PROB
$INPUT      ID AMT TIME DV WT EVID MDV
$DATA NMDATA1.CSV
...
$PK
...
...
   IF(NEWIND==0) NSUBJ=0
   IF(NEWIND/=2) NSUBJ=NSUBJ+1
...
...
$SIML (11234) BOOTSTRAP=-1
$EST METHOD=1 INTERACTION NOHABORT MAXEVAL=99999 PRINT=5  MSFO=msf1
$TABLE  NSUBJ  ID  AMT  TIME  DV  WT  EVID  MDV  NOAPPEND  NOPRINT  NOHEADER
NOFORWARD FILE=bsdata.tab

;This dummy estimation step set internal variable BOOTSTRAP_ON to 0
$PROBLEM dummy problem to reset BOOTSRAP setting
$INPUT      ID AMT TIME DV WT EVID MDV
$DATA NMDATA1.CSV REWIND
$THETA  (0,2.77) (0,0.0781) (0,0.0363)
$OMEGA .03 .03 .03
$SIGMA  .01
$EST METHOD=0 MAXEVAL=0

; Now, new data set, bsdata.tab can be read in correctly
$PROBLEM SIMULATION BASED ON PREVIOUS ESTIMATES
$INPUT ID PREV AMT TIME DV WT EVID MDV
$DATA bsdata.tab (8E12.0) NOOPEN ;REWIND
$MSFI msf1
$SIM(-1) ONLYSIM ; SUBPROBLEM=1 ; TRUE=FINAL
$TABLE  ID  PREV  AMT  TIME  DV  WT  EVID  MDV  NOAPPEND  NOPRINT  NOHEADER
FILE=what.tab
```

7. When using the $THETAR record, if you have final etas estimated (default FNLETA=1), or weighted residual components requested (WRES, CWRES, NPDE, etc) in the $TABLE record, the reported thetas (THETAR) will be used for their calculation, incorrectly, instead of the native thetas (THETA). This occurs when the last estimation step is an EM/Bayes estimation, but not when it is an FO/FOCE/Laplace estimation. To prevent this from occurring, perform the estimation and $TABLE steps in separate problems. Note the use of FNLETA settings and MSF files for this purpose in the following example:

```
$PROB Estimation and Variances of estimation
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1.csv IGNORE=C
```

```
...
$EST METHOD=ITS INTERACTION NOABORT CTYPE=3 PRINT=5 NOPRIOR=1 FNLETA=0
MSFO=thetair4.msf
$COV MATRIX=R PRINT=E UNCONDITIONAL

$PROB Table outputs
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1.csv IGNORE=C REWIND
$MSFI thetair4.msf
$EST METHOD=ITS INTERACTION NITER=0 FNLETA=2
$TABLE ID TIME DV IPRE TCLV ETACL CCL CL V1 Q V2 CWRES NOAPPEND
FILE=thetair4.tab
```

8. The NONMEM system and nmfe73 script for Linux and MACOSX do not work well from run directories with spaces in their names, and avoid using such names. However, should you wish to use NONMEM in this manner, the script may be easily modified. In the nmfe73 script for Linux and MAXOSX, change line 275 from
   `cd $nmcdir`
   to
   `cd "$nmcdir"`

   Even with this change in the script, the xml output file is not properly constructed when executing NONMEM in run directories with a space in its name. The elements in the xml file will not be correctly tagged with "nm:". This occurs in versions 7.2 as well as 7.3. Both of these problems are corrected in NONMEM 7.4.

9. The MCETA option for FOCE/Laplace works best using RANMETHOD=4P when desiring consistent results between single and parallelized analysis. Not setting RANMETHOD or setting RANMETHOD=P can cause inconsistent results for some problems.

10. For gfortran 4.9 or higher, when running nonmem 7.3, IEEE floating point errors may be issued to the console or terminal window. These error messages may be ignored, as they do not impact analysis. To prevent these messages from being issued, you must re-install NONMEM using the compiler switch –ffpe-summary=none.

    You can do this by adding the option to line 250 in SETUP73.bat:
    ```
    if "%f%" == "gfortran" set op=-O3 -ffast-math -ffpe-summary=none
    ```

    or line 368 and 376 in SETUP73:
    line 368:
    ```
    if [[ $ftail == gfortran ]]; then op="-w  -ffpe-summary=none"; fi
    ```
    line 376:
    ```
    if [[ $ftail == gfortran ]]; then op="-w -O3 -ffast-math
     -ffpe-summary=none "; fi
    ```

11. In NONMEM 7.2 and NONMEM 7.3, Request to estimate initial thetas using lower bound 0 information, such as:
    $THETA (0,,4.0)

results in an error message from NMTRAN. The work around is to provide lower bound that is slightly different from 0:
$THETA (0.0001,,4.0)

12. When population mixture model is used along with two Sigmas, a memory access violation can occur and the program will fail if IMP, SAEM, or BAYES is the first estimation. A work-around is to run an ITS estimation first (you can set NITER=0):

```
$EST METHOD=ITS NITER=0
$EST METHOD=IMP…
```

Or one may define THETAS as residual error coefficients, fixing the SIGMAS to 1.0:
For example, if you model

```
$ERROR
IF(TYPE==1) then
Y=Y+F*EPS(1)
Else
Y=Y+F*EPS(2)
Endif
…
$SIGMA 0.3 0.3
```

You can instead replace the two sigmas with two thetas:

```
$ERROR
IF(TYPE==1) then
Y=Y+THETA(21)*F*EPS(1)
Else
Y=Y+THETA(22)*F*EPS(2)
Endif
…
$SIGMA (1.0 FIXED) (1.0 FIXED)
…
$THETA
…
…
0.54
0.54
```

13. When a non-random variable is first defined and used within an IF/THEN block as part of a random variable, such as in the following code, where constant NR is used first defined in the IF/THEN block, and is then used to define random variable XNR:

```
B=1*EXP(ETA(1))
IF (1.EQ.1) THEN
 NR=1
 XNR=NR*THETA(1)*B
ENDIF
```

The derivatives may not be correctly constructed (for example, 0 valued gradients may appear), and can affect ITS/FO/FOCE/Laplace estimations. A work-around is to add $ABBR NOFASTDER just after $PROB. This bug can affect NONMEM 7.2 and NONMEM 7.3.

In addition, the following bug was fixed in version NONMEM 7.4.1:

The fix of bug 13 resulted in another bug:

14. When a non-random variable (one not associated with etas or epsilons) is introduced within an IF/THEN block, and utilized in a nested IF/THEN block or later used outside the block, the parsing of user code into intermediate variables by NMTRAN occurs incorrectly and can result in incorrect evaluations based on that non-random variable. For example, in the following code:

```
…
IF (AGE.GE.16) THEN
   SCRMM=SCR*10/MWTCR
   IF (SCRMM.GE.0.06) THEN
      CLCR=0.516*(112-AGE)/(112-40)*FSEX/SCRMM
   ELSE
      CLCR=0.516*(119-AGE)/(119-40)*FCPR/SCRMM
   ENDIF
ENDIF
…
```

The non-random variable SCRMM is introduced in the above IF/THEN block, and then used in the subsequent IF/THEN block. The result is an incorrect assessment of 0 for SCRMM, and a value of Infinity for CLCR. A work-around for this is to initialize the non-random variable unconditionally outside of all IF/THEN blocks:

```
…
SCRMM=0.0
IF (AGE.GE.16) THEN
   SCRMM=SCR*10/MWTCR
   IF (SCRMM.GE.0.06) THEN
      CLCR=0.516*(112-AGE)/(112-40)*FSEX/SCRMM
   ELSE
      CLCR=0.516*(119-AGE)/(119-40)*FCPR/SCRMM
   ENDIF
ENDIF
…
```

This bug has been fixed in NONMEM 7.4.1.

Bugs found in version NONMEM 7.4.1, fixed in NONMEM 7.4.2

15. When using the FAST option, and any Mu referenced thetas with OMEGA 0.0 FIXED are not at the end of the list of thetas to be estimated, then those thetas will not be moved during estimation.
16. If no thetas are to be estimated, and only OMEGAS and/or SIGMAS are to be estimated, then an error in the gradient setup occurs, and the estimation fails.
17. Because of inappropriate memory allocation, when applying the SIR algorithm, if the number of THETAS in the model is greater than the total number of parameters to be estimated (non-fixed thetas, omegas, and sigmas), an error can occur in assessing the variance from the SIR analysis.

18. ADVAN14 and ADVAN15 can have memory allocation problems, or it may report that initial advance from t to tout is too small.
19. An access violation sometimes occurs if the there are more than one SIGMA, and the first estimation method is IMP, SAEM, or BAYES. This bug has occurred since NONMEM 7.2
20. For ITS, IMP, SAEM, BAYES, and NUTS problems, if a theta is MU modeled but fixed, ordinary differential equations (ODE) problems may run inefficiently because it evaluates an unneeded derivative for it. This is now corrected in NONMEM 7.4.2.

    For ITS, MAP, SAEM, and BAYES, if the theta is MU modeled and estimated (not fixed), but has a 0 valued OMEGA associated with it, then the gradient is properly calculated. However in NONMEM 7.4.1, analytical first derivative assessment is turned on for all thetas, which may become much less efficient than evaluating by finite difference for the few thetas that are needed, as was done in earlier NONMEM versions. When there are only a few thetas that have no OMEGAS or zero valued OMEGAS associated with them, finite difference derivative evaluation may be more efficient. Therefore, for NONMEM 7.4.2, the default action for MU referenced Thetas associated with 0 valued OMEGAs has been reverted back to be finite difference evaluation, as was done in versions NONMEM 7.3 and earlier. The user may still turn on analytical evaluation of derivatives for MU referenced thetas with 0 valued Omegas by setting MUM=M(x) for Theta x.
21. The NPD and NPDE diagnostics do not perform properly on BQL data. This error has occurred since NONMEM 7.3.
22. During jobs that are parallelized, the CWRES and CWRESI values for some subjects may be those of WRES and WRESI. This is due to some of the worker processes not having available the empirical Bayes estimates for some subjects, necessary for evaluating CWRES and CWRESI.
23. Data files with recods longer than 1000 characters may prevent NMTRAN from completing the data input.
24. In defining the compartments in $MODEL, if the first compartment is initially set to OFF, such as:
    $MODEL
    COMP=(One,INITIALOFF,NODOSE)
    COMP=(PKCENT)

    NONMEM may issue an error statement:
    0DATA REC 1: COMPARTMENT ASSOCIATED WITH THE PREDICTION IS OFF

    This error has been present since NONMEM 7.1
25. Left-strings appearing in $ABBR REPLACE may not be replaced if there is a space preceding or trailing the left-string in the abbreviated code. For example:
    ```
    $ABBR REPLACE THINDEX=1
    $ABBR REPLACE THBASE=0.0
    …
    $PK
    …
    ```

```
CL=THETA(THINDEX ) + THBASE
```

THINDEX and THBASE in the abbreviated code will not be replaced. However, when the space is removed:

```
CL=THETA(THINDEX)+THBASE
```

THINDEX will be replaced with the right_string 1, and THBASE will be replaced with the right string 0.0.

The leading space bug has been present since NONMEM 7.3, the trailing space bug has been present since NONMEM 7.4.0, and these have been fixed for NONMEM 7.4.2.

Bug found in version NONMEM 7.4.2, fixed in NONMEM 7.4.3

A new bug was inadvertently introduced in NONMEM 7.4.2, and corrected in NONMEM 7.4.3:

1. If MSFO option is used with IMP, SAEM, ITS, or BAYES, the problem fails with an access violation at the $COV step.

Bugs found in version NONMEM 7.4.3, fixed in NONMEM 7.4.4

The following bugs found in version NONMEM 7.4.3 have been fixed in NONMEM 7.4.4:

1) The $PRIOR TNPRI method may not work correctly when there are fixed thetas and the prior information is brought in from an MSF file by a $MSFI record. This bug has been present since NONMEM VI. A work-around is to convert fixed thetas to fixed scalar parameters in the $PK record so they are not defined in the theta vector, in the control stream generating the MSF file, and in the control stream using it via the $PRIOR TNPRI record.

2) The CWRES and CPRED are incorrectly calculated as CIWRES and CIPRED when INTERACTION is not specified in the $EST record. This error has been present since NONMEM 7.4.0. As a work-around, always set INTERACTION (this is default for EM and Bayes methods) for the $EST record. Setting INTERACTION does not impact the analysis when the residual error is homoscedastic, and INTERACTION should always be used when residual error is heteroscedastic.

3) When $ABBR PROTECT is used and EXP(), PEXP() or X**Y expressions are used, the protect routines may occasionally not accurately evaluate eta derivatives due to a low threshold (100.0) placed on the argument to avoid floating overflow upon exponentiation. If this occurs, you will likely see poor objective function values when implementing $ABBR PROTECT, or PEXP(), or observe 0 gradients. A modified protect.f90 file is available with a higher threshold (350.0), can be downloaded from https://nonmem.iconplc.com/nonmem743 and can be used as a plug-in with the control stream statement $ABBR SUBROUTINES … other=protect.f90.

In general, the protect.f90 file is available for the user to modify the behavior of protection functions and their derivatives, if necessary. See the section **Stable Routines for Estimation Methods and Automated Protection Against Floating Point Exceptions** in intro7.pdf for further information.

4) Negative valued Lower bounds with an upper bound on thetas can result in an NaN valued theta for EM algorithms. Work-around is to remove the upper or lower bound, or scale theta so bounds are in positive region. This bug has been present since version NONMEM 7.1.0
5) When using VARCALC=1 in a $TABLE record, if a label exceeds 5 characters, an internal Fortran write error will occur while writing variance labels in the .vpd table. To prevent this from occurring, either enter a FORMAT statement in the $EST record that will define at least S=2X+2 characters, where X=largest label length in the $TABLE records and S is the character length that FORMAT is describing , or shorten the labels used in the $TABLE record.

   For example, the largest label is 10 characters long. The format could be set to
   $EST … FORMAT=s1PE23.16
   This format defines a 24 character length, which is more than 2*10+2=22.

   Or, shorten the labels in the $TABLE record to less than 6 characters:
   $TABLE … SHORT=LONGNAME…

6) During parallelization, the THIN setting is not transferred to workers during $EST METHOD=BAYES THIN=… method, and the BAYES_EXTRA setting is not transferred to workers during the $EST METHOD=NUTS method. When a user desires to control printing of additional individual parameters to a file using the BAYES_EXTRA switch, such as is shown in example8.ctl:

```
IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 .AND. TIME==0.0) THEN
"  WRITE(51,98) ITER_REPORT,ID,CL,V1,Q,V2
" 98 FORMAT(I12,1X,F14.0,4(1X,1PG12.5))
ENDIF
```

   The BAYES_EXTRA is never set to 1 for workers during METHOD=NUTS, so no records are placed in the file. When METHOD=BAYES THIN=…, the BAYES_EXTRA is 1 for every iteration, instead of every THIN iteration, so worker's files contain too many records.

   As a work-around, incorporate the source code file extrasend.f90 (located in https://nonmem.iconplc.com/nonmem743) into your control stream as an OTHER routine:

```
$SUBROUTINES … other=extrasend.f90
```

   In $PK, place the following line, after the last MU_ referencing declaration (or just before the first `IF(BAYES_EXTRA==1…` statement):

```
" CALL EXTRASEND()
```

The extrasend.f90 will transfer the appropriate BAYES_EXTRA and THINPRINT option values from manager to worker. The extrasend.f90 will transfer the appropriate BAYES_EXTRA and THINPRINT options from manager to worker.

7) For data sets with many items (columns), and when =DROP requests are made on several of these items, the FDATA file may not be correctly constructed. As a work-around, add the option WIDE to the $DATA record.
8) When using ADVAN13 or ADVAN14 with the -2LL options, such as:
```
$EST METHOD=1 LAPLACE -2LL
```

NMTRAN fails to create second derivatives in the $ERROR segment. The problem will fail immediately with an error message. As a work-around, add the option NUMERICAL.

9) In the Windows environment, when –tprdefault is used, and NMTRAN reports that certain predpp routines need to be recompiled so that some arrays are resized for the problem, the nmfe74.bat script should force a predpp recompile. However, this does not occur because of the ordering of certain lines in nmfe74.bat. The problem is likely to fail with access violations or some other error. As a remedy, download the updated nmfe74.bat located at https://nonmem.iconplc.com/nonmem743, and replace the values of dir, f, and op defined in the first several lines of the script with the appropriate values for your NONMEM installation.

10) In Windows and linux, if LVR is set to a value higher than 93 with $SIZES record, such as
$SIZES LVR=95
then diagnostic items such as CPRED, CWRES, will be incorrectly selected from what the user requested, due to NMTRAN not inserting the correct table ID's in the FCON file for NONMEM to use. As a remedy, download the updated nmfe74.bat (windows) or nmfe74 (Linux, MAC) file, located at https://nonmem.iconplc.com/nonmem743, and replace the values of dir, f, and op defined in the first several lines of the script with the appropriate values for your NONMEM installation. Furthermore, download also the tabindex_correct.* files, and place them in the ..\util directory of your nonmem installation. The tabindex_correct utility will be called by the updated nmfe74 scripts and correct the FCON table item ID's. This bug does not occur in versions of NONMEM earlier than 7.4.0.
11) A warning message was omitted. With FOCE/Laplace during the $COV step there is now a warning if the R matrix is forced to be positive definite.

A series of built-in probability densities have been introduced in NONMEM 7.4.2. Please see section I.32 Probability Density Functions (NM742).

A new parallel file option, called PARSE_PRESERVE, to improve efficiency of load distribution. See *The PARAFILE* in section I.73 Parallel Computing (NM72).

A new “Time after dose” (TAD) example has been added to on-line help and Guide VIII.

I.4 What is new in NONMEM Version 7.3.0 versus NONMEM 7.2.0

The main new features of NONMEM 7.3 compared to NONMEM 7.2.0 are as follows:

Execution script (nmfe73) offers more control in discerning location of compiler and mpi system. This option can facilitate execution of NONMEM in which there can be potential conflict with other software that may use alternative compilers and mpi systems. See section I.8 Invoking NONMEM, and the –locfile option.

Increased number of mixed effects levels. Random effects across groups of individuals, such as clinical site, can be modeled in NONMEM. Sites themselves may be additionally grouped, such as by country, etc. See section I.58 Adding Nested Random Levels Above Subject ID (NM75).

Easy to code inter-occasion variability. ETA’s to be referenced by an index variable related to the inter-occasion data item. See section I.7 Expansions on Abbreviated and Verbatim Code

Symbolic labels to thetas, etas, and epsilons. See section I.7 Expansions on Abbreviated and Verbatim Code

Priors for SIGMA matrix. A SIGMA prior matrix may be added (assumes inverse Wishart distributed) to provide prior information for SIGMAs. See section I.41 A Note on Setting up Prior Information.

Optimizing settings for some options in SAEM and Importance Sampling. User may request an optimal ISAMPLE setting be determined for each subject by NONMEM for SAEM and IMP, rather than relying on a pre-specified value. Similarly, user may request IACCEPT and DF settings be optimized for each subject by NONMEM when performing IMP. For BAYES and SAEM, user may request that most appropriate CINTERVAL be determined based on the degree of Markov chain correlation across iterations, rather than the user having to assess appropriate CINTERVAL by trial and error. See section I.36 Monte Carlo Importance Sampling EM and I.38 Stochastic Approximation Expectation Maximization (SAEM) Method

An AUTO option to allow NONMEM to determine the best options for Monte Carlo Expectation-Maximization (EM) and Bayesian Markov Chain Monte Carlo methods, instead of the user having to determine these settings for each problem. See section I.43 Some General Options and Notes Regarding EM and Monte Carlo Methods.

Perform a Monte Carlo search or select from a pre-existing list of initial thetas, omegas and sigmas that provide the lowest starting objective function for estimation. See section I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records.

Perform a Monte Carlo search for initial best estimates of etas for each subject. Together with a Monte Carlo search of best initial thetas, omegas, and sigmas, this provides a global search technique for the traditional, deterministic estimation methods, with less reliance on starting position for incidence of success. See MCETA in section I.30 General New Options for $ESTIMATION Record (NM73).

FOCE/Laplace and ITS to be assessed using only numerical eta derivatives for search of best etas and/or eta Hessian matrix assessment. This feature relaxes the requirement that analytic derivatives be computed for FOCE and Laplace by either NMTRAN or the user, which makes it easier to write user-supplied subroutines. Particularly useful for general stochastic differential equation analysis. See OPTMAP and ETADER in section I.30 General New Options for $ESTIMATION Record (NM73).

Conditional Individual Weighted Residual (CIWRES) added to residual variance diagnostics. While CIWRES for uncorrelated data is readily evaluated as (DV-iPRED)/W, CIWRES provides a proper individual weighted residual for L2 correlated data as well, which requires more extensive linear algebraic calculation. Furthermore, individual predicted and individual residual values, what are typically designated as IPRED and IRES and has often been inserted by hand into the control stream by users, is now assessed by NONMEM (called CIPRED, and CIRES, respectively) and can be requested in the $TABLES record. See section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

A range of Etas may be requested to be outputted. Instead of requesting for each eta to be outputted in a $TABLE record as ETA1, ETA2, ETA3, etc., a range of etas using the format of ETAS(x:y) may be requested. See I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

Boot-strap simulations to be performed in NONMEM. See section I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74).

Example control stream files demonstrating how to model population densities of individual parameters that are t-distributed. See section I.60 Model parameters as log t-Distributed in the Population (NM73).

Option to use Nelder-Mead optimization for obtaining best fit individual etas, particularly useful to improve robustness for importance sampling. See OPTMAP in section I.30 General New Options for $ESTIMATION Record (NM73).

Option to use either eigenvalue square root or Cholesky square root algorithms for assessing weighted residual diagnostics. See WRESCHOL in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

Option to have etabar and eta shrinkage information include only subjects which influence the etas. Furthermore, you may specify certain etas of particular subjects to be excluded, or specify certain etas of certain subjects to be included from the average eta shrinkage assessment by using a reserved variable (ETASXI) in the $PK or $PRED section. An alternative eta shrinkage evaluation using empirical Bayes variances (EBVs, or conditional mean variances) are now also reported. See information on shrinkage in section I.61 Format of NONMEM Report File, and information on the .shk and .shm files in I.63 $EST: Additional Output Files Produced.

Subscripted variables may be used in abbreviated code, with fewer restrictions on DOWHILE. See section I.7 Expansions on Abbreviated and Verbatim Code for and example on residual variance correlation, and see section I.58 Adding Nested Random Levels Above Subject ID (NM75) for another use.

Additional reserved variables may be declared in the control stream file not natively recognized by NMTRAN. Some useful but not often needed global variables may be accessed by listing them in an NMTRAN include file referenced in a control stream file, which can also be used in abbreviated code. See section I.7 Expansions on Abbreviated and Verbatim Code .

Enhanced non-parametric analysis methods, such as extended grid of support points, use of an outsize inter-subject variance to obtain support points that fit outlier subjects better, and built-in bootstrap analysis methods for obtaining empirical confidence ranges to non-parametric probability variables. See I.33 Some Improvements in Nonparametric Methods (NM73).

The TRANSLATE option of the $DATA record has been expanded. Now any value may be given for dividing time and II values, and any precision may be requested. Examples are:

```
TIME/1.0000
```

or

```
TIME/1/4
```

for formatting times in FDATA with 4 digits to the right of the decimal. Or

```
II/0.01/6
```

which divides II values by 0.01, and writes 6 digits to the right of the decimal for the II data item. See Help guide for more details.

Times may be optionally encoded as hh:mm:ss instead of just hh:mm. For example,

8:45:29

will be acceptable, and incorporates the seconds values.

The $ANNEAL record provides a means of SAEM simulated annealing to provide global search techniques for thetas that do not have Omegas associated with them. See I.55 $ANNEAL to facilitate EM search methods (NM73) for this additional annealing technique.

Population weighted residual diagnostic values can be calculated for normally distributed data even though there are also non-normally distributed data values in the same subject. See the MDVRES option in I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

When $TABLE values exceed 0.3E+39, a warning is issued, but the table is still produced.

A utility program to fill in extra records with small time increments, to provide smooth plots. This utility program can also fill in by various interpolation techniques missing covariate values for original records. Also, if an MDV is set to a value greater than or equal to 100, it is converted to that value minus 100 upon input, but will also not be used at all during estimation, only for table outputting. This option allows you to use a data file that was enhanced with extra records for both estimation as well as Table outputs, without significantly slowing down the estimation. See I.83 finedata Utility Program(NM73)**.** See also the examples section of on-line help and guide VIII on using the INFN routine to create interpolated values. The infn1 example has been completely rewritten. The infn2 and fine1 examples are new.

A utility program to fill in substitution variables in template control stream files. See I.97 nmtemplate Utility Program (NM73)

New command line options, -tprdefault, and -maxlim, are provided for more dynamic assessment of needed memory allocation. Furthermore, the dynamic memory allocation has been made even more efficient in assessing memory requirements. See I.9 Dynamic Memory Allocation (NM72) and I.10 Changing the Size of NONMEM Buffers.

The various random number generating techniques, including Sobol quasi-random sampling with scrambling have been expanded for use with SAEM, BAYES, simulations, and Monte Carlo assessed population diagnostics. See the descriptions on RANMETHOD in I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format, I.36 Monte Carlo Importance Sampling EM**,** and I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74). In addition, an option to have each subject retain their own random number sequence is available, so that near identical estimation results are obtained for Monte Carlo methods in single process or parallelized process problems. See the RANMETHOD item and the P descriptor in I.36 Monte Carlo Importance Sampling EM**.**

Initial etas may be introduced in the control stream file or from an external source. See I.65 $ETAS and $PHIS Record For Inputting Specific Eta or Phi values (NM73).

For the $DATA record, .EQN. may be used in the IGNORE/ACCEPT option to indicate a numerical comparison rather than a literal comparison as is done for .EQ. and .NE.. See *Numerical Equality Comparison for IGNORE option in $DATA Record (NM73)*in section I.7 Expansions on Abbreviated and Verbatim Code

Informative record names for prior information of thetas/omegas/sigmas provide easier entry of NWPRI prior information. See I.41 A Note on Setting up Prior Information.

Maximal number of numerical integration steps is now easy to modify for ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, and ADVAN18. See discussion on MXSTEP in I.17 $SUBROUTINES: New Differential Equation Solving Method.

Mu model checking by NMTRAN can be turned off. If you wish to turn this off (checking mu statements can take a long time for very large control stream files), then include the NOCHECKMU option on the $ABBR record:
$ABBR NOCHECKMU

NMTRAN will allow & as a continuation marker on abbreviated code lines. Furthermore, the total length of a control stream record, whether on a single line or continued on several lines using &, may be up to 67000 characters long. See *Continuation indicator is allowed in abbreviated code (non-verbatim) lines (NM73)* in section I.7 Expansions on Abbreviated and Verbatim Code

More user functions for use in abbreviated code may be defined, using FUNCA through FUNCI. See Guide VIII.

Additional functions MIN, MAX, MOD, and GAMLN may be used in abbreviated code. See *MIN,MAX Functions (NM73)*, *MOD Function (NM73)*, and
Also, ensure that if any data items are used in the MAX or MIN argument, that this data item appeared elsewhere in the code block. For example, consider a data item called AGE,

```
LAGE=LOG(AGE)
AGE2=MAX(AGE,0.0)
```

GAMLN Function (NM73) in section I.7 Expansions on Abbreviated and Verbatim Code .

ATOL now also acts on ADVAN9's differential equation solver, where by default absolute significant digits accuracy (absolute tolerance) is 12.

Enhanced selection methods from CHAIN records for use in multiple sub-problems. For each sub-problem, population parameters may be randomly (with or without replacement) or sequentially selected from a chain file. See SELECT option in I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records.

Total CPU time is reported in the NONMEM report file (Tag #CPUT:) and in the root.cpu file. See ***#CPUT: (nm73)*** in section I.61 Format of NONMEM Report File and *root.cpu (NM73)* in section I.63 $EST: Additional Output Files Produced

Analytical and numerical derivatives of predicted and residual variance values with respect to eta may be outputted. See *NUMDER=0 (default) (NM73)* in I.30 General New Options for $ESTIMATION Record (NM73).

The SUBP option in $SIML may be greater than 9999 (new limit is 2^{31}-1).

All EM/Bayes methods are now estimated with the INTERACTION option on by default, unless NOINTERACTION is specified.

When NOPRIOR=1 is set, the estimation will not use TNPRI prior information (TNPRI should only be used with FO/FOCE/Laplace estimations). In previous versions of NONMEM, NOPRIOR=1 did not act on TNPRI priors.

New elements are available in the NONMEM report xml file: termination_nfuncevals, termination_sigdigits, termination_txtmsgs which catalog termination text messages by number, which can be mapped to ..\source\txtmsgs.f90, etabarn, ebvshrink, np_objective_function, and total_cputime.

If inputted omega or sigma elements are not positive definite because of rounding errors, a value to the diagonal elements will be added to make it positive definite. A message in the NONMEM report file will indicate if this was done.

In root.ext, Iteration -100000006 indicates 1 if parameter was fixed in estimation, 0 otherwise. See I.62 $EST: Format of Raw Output File.

Thetas may be inputted and reported in their natural domain, even when linear MU referencing. See I.53 $THETAI ($THI) AND $THETAR ($THR) Records for Transforming Initial Thetas and Reporting Thetas (NM73).

Covariance assessment may be turned off for a particular estimation. See *NOCOV=[0,1] (NM73)*in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method.

If an interruption occurred during FOCEI/Laplace/FO during the $COV step, covariance analysis may be resumed where it left off. See *RESUME (NM73)* in section I.56 $COV: Additional Options and Behavior.

Bugs Fixed

The following bugs have been fixed that were in NONMEM 7.2.0:

1) Some operating systems do not like the word ‘nul’ for a file name for FNULL. Work-around for earlier versions of NONMEM: change ‘nul’ to ‘JUNK’ in ..\resource\nmdata.f90, rebuild NONMEM by running SETUP72 or SETUP72.bat in the installed NONMEM directory. For example, for Windows gfortran, if c:\nm72g is your installed NONMEM directory, then from c:\nm72g execute the following command in the command window:
   setup72 c:\nm72g c:\nm72g gfortran y ar same rec n
2) In parallelization, Windows 64, gfortran compiled, using population mixture model, a variable is not initialized and causes parallelization failure. Work-around for earlier versions of NONMEM is to add the gfortran compiler switch -finit-integer=0. To do this, edit setup72.bat (line 247) or setup72 (362), adding -finit-integer=0 just before –ffast-math (do not place it as the last optimizing option). Then, rebuild NONMEM. For example, if c:\nm72g is your installed NONMEM directory, then from c:\nm72g execute the following command in the command window:
   setup72 c:\nm72g c:\nm72g gfortran y ar same rec n
3) “BY USER INTERUPT” is misspelled.

4) SAEM terminates on some problems. Cause is access violation when CONSTRAIN is called. Work-around for earlier versions of NONMEM is to set CONSTRAIN=0. Or, set MAXOMEG using $SIZES such that they are at least (NEPS+1)*NEPS/2.
5) When defining compartments in $MODEL, NMTRAN does not always terminate DATA CMOD code lines properly with respect to continuation markers, resulting in a failed compilation of FSUBS. Work-around is to have more than an integer multiple of 6 compartments named (for example, if you have 24 compartments, define a 25th compartment).
6) When $CHAIN record is used, ISAMPLE may not be less than 1. Work-around for earlier versions of NONMEM is to change the index number (iteration number for a raw output file of a previous analysis) of the desired record in the file to a positive number.
7) When a simulation is desired using the results of a previous estimation using $MSFI, NONMEM sometimes prevents its use because of a flag indicating it was not properly estimated. Work-around for earlier versions of NONMEM: use the record $CHAIN FILE=file.ext ISAMPLE=xxxx, where file.ext is the name of the raw output file of the previous analysis, and xxxx is the iteration number, typically the last iteration.
8) During an estimation with FO or FOCE, and the last subject in the data set has non-influential etas (for example, with interoccasion variability, if the last subject had no data during the last inter-occasion, the eta for that last inter-occasion is non-influential), the estimation may become inefficient due to incorrect gradient assessments. This has been corrected for some types of problems, but this may still persist in other problems, which may be remedied with the SLOW option. For earlier versions of NONMEM another work-around, when possible, is to reorder the subjects so that the last subject does not have one or more non-influential ETA's.
9) When only thetas are in a problem, and there are single-subject data, then standard errors are printed out, but covariance, inverse covariance, and correlation matrices are reported as 0. Work-around for earlier versions of NONMEM: If possible, pose the problem as multi-subject, insert one eta as $OMEGA 0.0 FIXED
10) When using DOWHILE(DATA) in abbreviated NMTRAN code, there should be no comment on that line, such as DOWHILE(DATA) ; start of dowhile.
11) In abbreviated code, recursion code and $INFN DOWHILE(DATA) cannot both be present in the same control stream. The error message is MUST BE "DO WHILE (CONDITION) ...ENDDO" Workarounds for earlier versions of NONMEM: (1) avoid unnecessary recursive variables by defining them as COM(1), COM(2), etc. (2) use $MSF to put the $INFN block in another problem.
12) With large numbers of thetas and or omegas, the xml file may incorrectly print out the various variance matrices of estimates (covariance, correlation, inverse covariance, etc.). This has been corrected
13) When a series of $TABLE statements without FILE= specification is followed by $TABLE statements with FILE= specification, not all tables print out, and an error is issued in the NONMEM report file: “0ERROR IN WRITING FILE : TABLE FILE; USER FORMAT ERROR IN FORMAT_SWRITE”. Work-around is to set LFORMAT=NONE and RFORMAT=NONE on the first $TABLE record with a FILE= option.

14) Problems with temporally over-lapping dosing records and with $EST and $COV records may fail during a parallelization run at the $COV step. Work-around is to perform the $COV step without parallelization.
15) Repetition variables and data items (RPTI, RPTO, RPT_) useful for repeated records for convolution problems did not work properly for estimation methods other than FO. This has been corrected in NONMEM 7.3.
16) If the partial derivative of MTIME with respect to any eta is negative (such as MTIME(1)=THETA(5)-ETA(5)), then the predicted value of F and its derivatives will probably be incorrect. The bug exists in all versions of PREDPP from NONMEM VI to NONMEM 7.2. IT is corrected for NONMEM 7.3. A work-around is to use ALAG's in place of MTIME's, but this is somewhat complicated. A fix is to edit the file PRED.f90 (or PRED.f for older versions) in the pr directory. Locate the characters
DSUM=DSUM+GG(IMTGG(MTPTR),K+1)
Change to
DSUM=DSUM+ABS(GG(IMTGG(MTPTR),K+1))

I.5 What is new in NONMEM Version 7.2.0 versus NONMEM 7.1.2

The main new features of NONMEM 7.2 compared to NONMEM 7.1.2 are as follows:

Dynamic Memory Allocation: No need to modify SIZES for unusually large problems. Memory is automatically sized according to the number of parameters and number of subjects. User may override computer generated values using a $SIZES statement as the first executed line of the control stream. Often for moderate sized problems, this results in much smaller memory usage, compared to the standard memory usage in NONMEM 7.1. Particularly helpful for parallel computing when using multiple cores on a single computer. Please see section I.9 Dynamic Memory Allocation (NM72) and I.10 Changing the Size of NONMEM Buffers.

Parallel Computing: The computation of a single problem that can take many hours or days may be distributed over two or more cores and/or computers to complete in a shorter time. After the primary installation of standard NONMEM described below, parallel computing may require additional setup in order to implement, which can be very specific to the operating system and Fortran compiler used. In addition, you may need assistance from your IT administrator. Please read the installation notes below, and Section I.73 Parallel Computing (NM72)
MSF file system fully expanded to Monte Carlo Methods: Seamless resumption of expectation-maximization and Bayesian methods in case of sudden interruption, since the last print iteration.

XML Formatted Output: An XML markup version of the standard results output file is automatically produced.

Control Stream Files may be written in mixed case. User defined data labels and file names retain their case designation.

Stochastic Differential Equations (SDE): Additional data items have been added to facilitate SDE problems. Specialized data labels allow repeated PRED and ERROR calls for a single

record, but with different EVID values (XVID1, XVID2, XVID3, XVID4, XVID5). In addition, a plug in routine ("OTHER=SDE.f90") is available for Monte Carlo methods (but not for FOCE methods), that evaluates the stochastic differential equations, without requiring coding of these equations in the control stream file by the user. See sections I.74 Repeated Observation Records(NM72) and I.75 Stochastic Differential Equation Plug-In(NM72).

$CHAIN statement that is applicable to the entire $PROB, that allows incorporation of initial parameters from raw output files or randomization, and serves as parameters for simulations. The $EST METHOD=CHAIN supplies initial parameters from raw output files or randomizations only for the estimation method. See section I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records.

Both covariance and correlation matrices to OMEGAs and SIGMAs are now printed in the NONMEM report file. Also, all correlation matrices, whether to OMEGAS and SIGMAS, or pertaining to the correlation matrix of estimates, are printed out with diagonal elements equal to the square root of diagonal element of covariance matrix (standard error)

Allow user to input OMEGAs and SIGMAs as standard deviations and/or correlations, or Cholesky format. See *Alternative Inputs for $OMEGA and $SIGMA Values: VARIANCE/ CORRELATION/ CHOLESKY (NM72)* in section I.7 Expansions on Abbreviated and Verbatim Code .

New options for $EST: SIGLO, MAPINTER, MAPITER, NOHABORT, ORDER, METHOD=DIRECT, ISCALE_MIN, ISCALE_MAX, CONSTRAIN, FNLETA, ATOL. See the following sections:
I.26 Controlling the Accuracy of the Gradient Evaluation and Individual Objective Function Evaluation
I.27 The SIGLO level (NM72)
I.36 Monte Carlo Importance Sampling EM
I.37 Monte Carlo Importance Sampling EM Assisted by Mode a Posteriori (MAP) estimation
I.38 Stochastic Approximation Expectation Maximization (SAEM) Method
I.39 Full Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method
I.42 Monte Carlo Direct Sampling (NM72)
I.44 MU Referencing
I.46 Termination testing
I.47 Use of SIGL and NSIG with the new methods

New options for $COV: SIGLO, ATOL, NOFCOV. See section I.56 $COV: Additional Options and Behavior.

$TABLE has two new special output variables, OBJI and NPD OBJI is individual objective function (same as given in the root.phi file). NPD is the correlated (or non-decorrelated) NPDE value. Also, whole record format options are now available, LFORMAT and RFORMAT. See

section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format.

Native parameters are intermediately printed to the console during classical estimation, along with scaled parameters and gradients.

Alternative convergence criterion for FO/FOCE/Laplace: See Section **I.28 Alternative convergence criterion for FO/FOCE/Laplace (NM72)**.

S Matrix evaluation of Variance-covariance Allowed when NOPRIOR=1
If $EST NOPRIOR=1 is set and $COV MATRIX=S is set, NONMEM will evaluate the variance-covariance matrix, unlike in earlier versions of NONMEM 7.

Three digit limitation indexed Variables. The limitation of number of digits expressing the index to thetas, etas, Omegas, Mus, and Sigmas has been increased from 2 (1-99) to 3 (1-999).

Bugs Fixed

The following bugs have been fixed that were in NONMEM 7.1.2:
1) With very large problems of more than to 180 estimated parameters (thetas, omegas, and sigmas), the eigenvalues list with two sets of column labels.
2) When the number of records in a subject exceeds 250, a "stack overflow" in the Intel version of NONMEM may occur.
3) On occasion after an analysis with SAEM with a very complex problem, estimation of objective function with IMP or IMPMAP results in ever increasing objective function values without stabilization, even though the SAEM result is reasonable. The usual adjustment of options in nm 7.1.2 fails to correct the problem. In NONMEM 7.2, some internal scaling parameters have been adjusted. Also, the user can further adjust these scaling parameters.
4) For certain estimation problems, ADVAN 5 and ADVAN7 provide inaccurate prediction values, which are sensitive to the initial thetas. The work-around for earlier releases is to use ADVAN6 or ADVAN9.
5) During a simulation problem, if symmetric band matrix patterns are used in the OMEGA, including a block matrix which has all covariances of 0, the first simulated data set will be correct, but subsequent data sets will be incorrect. This occurs because the banding information is re-initialized after the first sub-problem simulation. This is corrected in NONMEM 7.2. As a work-around for earlier releases, during simulations, replace the 0 valued covariances with very small values of covariances (such as 1.0e-05).
6) During an estimation with FO or FOCE, and the last subject in the data set has non-influential etas (for example, with interoccasion variability, if the last subject had no data during the last inter-occasion, the eta for that last inter-occasion is non-influential), the estimation may become inefficient due to incorrect gradient assessments.
7) If DROP is used in $INPUT to not include a data item in any problem, this DROP attribute continues to the next problem. This is corrected in NONMEM 7.2. As a work-around with earlier releases, do not use DROP in control streams with more than one problem unless the same items are dropped in all problems.

I.6 Introduction to NONMEM 7 and higher

Many changes and enhancements have been made from NONMEM VI release 2.0 to NONMEM 7. In addition to code modification and centralization of common variables for easier access and revision, the program has been expanded to allow a larger range of inputs for data items, initial model parameters, and formatting of outputs. The choice of estimation methods has been expanded to include iterative two-stage, Monte Carlo expectation-maximization (EM) and Monte Carlo Bayesian methods, greater control of performance for the classical NONMEM methods such as FOCE and Laplace, and additional post-analysis diagnostic statistics.

Attention:
NONMEM 7 and higher produces a series of additional output files which may interfere with files specified by the user in legacy control stream files. The additional files are as follows:

root.agh
root.clt
root.cnv
root.coi
root.cor
root.cov
root.cpu
root.ets
root.ext
root.fgh
root.grd
root.imp
root.npd
root.npe
root.npi
root.npl
root.phi
root.phm
root.rmt
root.shk
root.shm
root.smt
root.vpd
root.xml

Where root is the root name (not including extension) of the control stream file given at the NONMEM command line, or root=”nmbayes” if the control stream file name is not given at the NONMEM command line.

Modernized Code

All code has been modernized from Fortran 77 to Fortran 90/95. The IMSL routines have also been updated to Fortran 90/95. Furthermore, machine constants are evaluated by intrinsic functions in FORTRAN, which allows greater portability between platforms. All REAL

variables are now DOUBLE PRECISION (15 significant digits). Error processing is more centralized.

I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75)

FORTRAN 95 Considerations

The greatest changes as of NONMEM 7.1 are the renaming of many of the internal variables, and their repackaging from COMMON blocks to Modules. Whereas formerly, a variable in a common block may have been referenced using verbatim code as:

```
COMMON/PROCM2/DOSTIM,DDOST(30),D2DOST(30,30)
```

Now, you would reference a variable as follows:

```
USE PROCM_REAL,ONLY: DOSTIM
```

And you may reference only that variable that you need, without being concerned with order.

In addition, FORTRAN 95 allows you to use these alternative symbols for logical operators:

Example:
Fortran 77:

```
IF(ICALL.EQ.3) THEN
WRITE(50,*) CL,V
ENDIF
```

Fortran 95:

```
IF(ICALL==3) THEN
WRITE(50,*) CL,V
ENDIF
```

The list of operators are

Name of logical operator	Fortran 77	Fortran 95
Equal to	.EQ.	==
Not equal to	.NE.	/=
Greater than	.GT.	>
Greater than or equal to	.GE.	>=
Less than	.LT.	<
Less than or equal to	.LE.	<=

In FORTRAN 95, the continuation marker & must be on the line to be continued, rather than at the sixth position of the continued line:

Fortran 77:

```
CL=THETA(6)*GENDER+
xTHETA(7)**AGE
```

Fortran 95:

```
CL=THETA(6)*GENDER+    &
THETA(7)**AGE
```

This affects verbatim code and user-written subroutines. For example, an NMVI version of CCONTR would be written as follows:

```
SUBROUTINE CCONTR (I,CNT,P1,P2,IER1,IER2)
PARAMETER (LTH=40,LVR=30,NO=50)
COMMON /ROCM0/ THETA (LTH)
COMMON /ROCM4/ Y
DOUBLE PRECISION CNT,P1,P2,THETA,Y,W,ONE,TWO
DIMENSION P1(*),P2(LVR,*)
DATA ONE,TWO/1.0D+00,2.D+00/
IF (I.LE.1) RETURN
W=Y
Y=(Y**THETA(3)-ONE)/THETA(3)
CALL CELS (CNT,P1,P2,IER1,IER2)
Y=W
CNT=CNT-TWO*(THETA(3)-ONE)*LOG(Y)
RETURN
END
```

Whereas in NM7, it would be written as:

```
SUBROUTINE CCONTR(I,CNT,P1,P2,IER1,IER2)
USE SIZES, ONLY: ISIZE,DPSIZE
USE ROCM_REAL,   ONLY: THETA=>THETAC,Y=>DV_ITM2
USE NM_INTERFACE,ONLY: CELS
IMPLICIT NONE
INTEGER(KIND=ISIZE), INTENT(IN OUT) :: I,IER1,IER2
REAL(KIND=DPSIZE),   INTENT(IN OUT) :: CNT,P1(:),P2(:,:)
REAL(KIND=DPSIZE) :: ONE,TWO,W
DATA ONE,TWO/1.00D+00,2.00D+00/
SAVE
IF (I.LE.1) RETURN
W=Y(1)
Y(1)=(Y(1)**THETA(3)-ONE)/THETA(3)
CALL CELS (CNT,P1,P2,IER1,IER2)
Y(1)=W
CNT=CNT-TWO*(THETA(3)-ONE)*LOG(Y(1))
RETURN
END
```

Continuation indicator is allowed in abbreviated code (non-verbatim) lines (NM73)

As of NONMEM 7.3.0, extra long lines may be continued using an & at the end of the line:

```
CL=EXP(THETA(1)*WERT &
+EPS(1))
```

The total number of characters in the resulting concatenated line may not exceed FSD (default set to 67000 in sizes.f90). In fact, the continuation marker & may be used on record lines as

well. If the ampersand at the end of a line is not to be interpreted as a continuation marker, but as a part of the record, then, place a ; after it. For example,
FORMAT=s1PE15.8:160& ;

Alternative Inputs for $OMEGA and $SIGMA Values: VARIANCE/ CORRELATION/ CHOLESKY (NM72)

In NONMEM 7.2.0, OMEGA and SIGMA elements may be entered in forms other than the default variance diagonal elements and covariance off-diagonal elements. Diagonal elements may also be entered as standard deviation, and off-diagonal elements may be entered as correlation values. Options are

VARIANCE/STANDARD to indicate form of diagonal elements
COVARIANCE/CORRELATION to indicate form of off-diagonal elements
CHOLESKY for inputting blocks of OMEGAS or SIGMAS in their Cholesky form.

Examples:

```
$OMEGA BLOCK(2) ; or $OMEGA VARIANCE COVARIANCE BLOCK(2)
0.64
-0.2402 0.58

$OMEGA STANDARD BLOCK(2)
0.8
-0.24 0.762

$OMEGA STANDARD CORRELATION BLOCK(2)
0.8
-0.394 0.762

$OMEGA VARIANCE CORRELATION BLOCK(2)
0.64
-0.394 0.58

$OMEGA CHOLESKY BLOCK(2)
0.8
-0.3 0.7

$SIGMA 0.3 STANDARD 0.8 STANDARD 0.3 VARIANCE
```

These input options do not affect how estimated OMEGAs and SIGMAs are outputted.

With NONMEM 7.3.0, there are new features for abbreviated code and the $ABBR record. Each is discussed in greater detail in the on-line help and Guide VIII:

Repeated SAME BLOCK for $OMEGA and $SIGMA Records (NM73)

No need to repeat multiple SAME block segments:
$OMEGA BLOCK(2) SAME(3)
Is equivalent to
$OMEGA BLOCK(2) SAME
$OMEGA BLOCK(2) SAME

$OMEGA BLOCK(2) SAME

The SAME(m) feature is also available for $SIGMA.
$SIGMA BLOCK(2) SAME(3)

Repeated Value Inputs for $THETA, $OMEGA, and $SIGMA (NM73)

As of NM73, repeated inputs of $THETA be entered as follows:
Long-hand:

```
$THETA 2 2 2 2 (0.001,0.1,1000) (0.001,0.1,1000) (0.001,0.1,1000)
      (0.5 FIXED) (0.5 FIXED)
```

Short-hand:

```
$THETA (2)x4 (0.001,0.1,1000)x3 (0.5 FIXED)x2
```

Where x*n* means to replicate *n* times. The item to be repeated must always be in parentheses, and the x*n* must always be immediately after the item, not before it (4x(0.2) is not permitted).

Repeated inputs of $OMEGA or $SIGMA may be entered as follows:

```
$OMEGA BLOCK(6)
0.1
0.01 0.1
(0.01)x2 0.1
(0.01)x3 0.1
(0.01)x4 0.1
(0.01)x5 0.1
```

The VALUES(*diag*,*odiag*) feature allows one to set up initial values with diagonals *diag* and off-diagonals *odiag*. The above example could have been entered as

```
$OMEGA BLOCK(6) VALUES(0.1,0.01)
```

For fixed block (such as for omega priors):

```
$OMEGA BLOCK(6) FIX VALUES(0.15,0.0)
```

$ABBR DECLARE feature for abbreviated code (NM73)

Integers and arrays may be declared and used in abbreviated code:

```
$ABBR DECLARE DOSE(100),DOSETIME(100)
$ABBR DECLARE INTEGER I
```

$ABBR REPLACE feature for abbreviated code (NM73-NM75)

Any character string may be replaced. In particular, this allows for symbolic labeling to thetas, etas, and epsilons. As an example, subscripts to THETAS and ETAS can be given symbolic names:

```
$ABBR REPLACE THETA(CL)=THETA(4)
$ABBR REPLACE ETA(CL)=ETA(5)
CL=THETA(CL)*EXP(ETA(CL))
```

Replacement with selection by data item and parameter is permitted:

```
$ABBR REPLACE THETA(OCC)=THETA(4,7,10)
$PK
```

```
KA=THETA(OCC)
which is equivalent to
$PK
IF (OCC==1) KA=THETA(4)
IF (OCC==2) KA=THETA(7)
IF (OCC==3) KA=THETA(10)
```

Another Example:

```
$ABBR REPLACE THETA(SID_KA)=THETA(4,6)
$ABBR REPLACE THETA(SID_CL)=THETA(5,7)
$PK
KA=THETA(SID_KA)
CL=THETA(SID_CL)
```

which is equivalent to

```
$PK
IF (SID==1) KA=THETA(4)
IF (SID==2) KA=THETA(6)
IF (SID==1) CL=THETA(5)
IF (SID==2) CL=THETA(7)
```

A list of numbers may be given as:

```
$ABBR REPLACE THETA(SID_KA)=THETA(4,7,10,13)
or by the short-hand
$ABBR REPLACE THETA(SID_KA)=THETA(,4 to 13 by 3)
```

At least one comma must appear, so NMTRAN knows it is a number list, not a variable name.

As of NM74, there is more flexibility of the BY variable:
The : may be used in place of TO. Also,

```
$ABBR REPLACE THETA(SID_KA)=THETA(10:4 by 3) ; order: 10,7,4
$ABBR REPLACE THETA(SID_KA)=THETA(4 to 10 by -3) ; order: 10,7,4
$ABBR REPLACE THETA(SID_KA)=THETA(10 to 4) ; order: 10,9,8,7,6,5,4
```

Another example:
Long-hand:

```
$ABBR REPLACE THETA(SID_KA)=THETA(4,7,10,13,25,29,33,37)
```

Short-hand:

```
$ABBR REPLACE THETA(SID_KA)=THETA(,4 to 13 by 3,25 to 37 by 4)
```

Also, a series of alias names may be conveniently defined (NM74):

```
$ABBR REPLACE THETA(CL,V1,Q,V2)=THETA(1 TO 4)
$ABBR REPLACE ETAQQ(CL,V1,Q,V2)=ETA(1 TO 4)
```

You may use them in abbreviated code, and, as of NM74, in $TABLE as well:

```
CL=EXP(THETA(CL)+ETAQQ(CL))

$TABLE ID TIME DV IPRED ETAQQ(CL) ETAQQ(V1) ETAQQ(Q) ETAQQ(V2)
```

The symbolic label substitutions will appear in the NONMEM report file and $TABLE outputs, for example (NM74):

```
THETA - VECTOR OF FIXED EFFECTS PARAMETERS   *********

         THETA(CL)  THETA(V1)  THETA(Q)   THETA(V2)

         1.68E+00   1.59E+00   8.13E-01   2.37E+00


 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

         ETAQQ(CL)  ETAQQ(V1)  ETAQQ(Q)   ETAQQ(V2)

 ETAQQ(CL)
+        1.65E-01

 ETAQQ(V1)
+        4.62E-03   1.34E-01

 ETAQQ(Q)
+        6.35E-03   1.69E-02   2.14E-01

 ETAQQ(V2)
+       -1.53E-02   1.26E-02   5.33E-02   1.63E-01
```

To turn off symbolic label substitution in the reporting of estimates, set $EST NOSUB=1 for a particular estimation problem. To turn off label substitution in $TABLE outputs, set $TABLE NOSUB=1 for that particular table. Also for $SCAT NOSUB=1 ETAQQ(V1) VS ETAQQ(CL) for example.

To set the default NOSUB for the entire problem, specify the following $DEFAULT (or $DEFAULTS) record:
$DEFAULT NOSUB=1

NOSUB also accepts a -1 to indicate revert to NONMEM default, which is to treat -1 as a 0. The NOSUB default setting is in effect throughout the control stream file, until another $DEFAULT record setting NOSUB is encountered.

Symbolic label substitutions will not be made in the additional output files *.ext, ,phi, etc), to maintain their third party software readability, However, see the utility lblr I.94 lblr: Insert User-Defined Labels in Additional Files (NM75), for implementing post-NONMEM-run label substitutions.

Symbolic Label Substitutions at $THETA, $OMEGA, and $SIGMA records (NM75)

A very easy way to apply symbolic label substitutions to Thetas is by defining the labels at the $THETA record:

```
$THETA
CL=(0.0,7.0)
V1=(5.0 fixed)
```

etc. This is equivalent to

```
$ABBR REPLACE THETA(CL)=THETA(1)
$ABBR REPLACE THETA(V1)=THETA(2)
```

Once defined, the $PK or $PRED code can use these definitions:

```
TVCL=THETA(CL)
```

To apply symbolic labels to OMEGAS:

```
$OMEGA BLOCK(4)
ECL=  0.3
EV1=  0.01 0.35
EQ=   0.01 0.01 0.54
EV2=  0.01 0.01 0.01 0.67
```

Or, for diagonals,

```
$OMEGA
ECL=  0.3
EV1=  0.35
EQ=   0.54
EV2=  0.67
```

Then, these take effect on etas and MU_'s:

```
MU_ECL=THETA(CL)
CL=EXP(MU_ECL+ETA(ECL))
```

Similar symbolic labels may be applied to SIGMAS/Epsilons. Consider the following example code:

```
$THETA
V1=(0.0,7.0) CL=(0.0,7.0) Q=(0.0,7.0) V2=7

$OMEGA BLOCK(4)
ECL=  0.3
EV1=  0.01 0.35
EQ=   (0.01)X2 0.54
EV2=  (0.01)X3 0.67

$SIGMA
RSW=(0.6 )   ;[P]

$THI
THETA(CL)=LOG(THETAI(CL))
THETA(V1)=LOG(THETAI(V1))
THETA(Q)=LOG(THETAI(Q))
THETA(V2)=LOG(THETAI(V2))

$THR
THETAR(CL)=LOG(THETA(CL))
THETAR(V1)=LOG(THETA(V1))
THETAR(Q)=LOG(THETA(Q))
THETAR(V2)=LOG(THETA(V2))


$PK
MU_ECL=THETA(CL)
```

```
MU_EV1=THETA(V1)
MU_EQ=theta(Q)
MU_EV2=THETA(V2)
CL=DEXP(MU_ECL+ETA(ECL))
V1=DEXP(MU_EV1+ETA(EV1))
Q=DEXP(MU_EQ+ETA(EQ))
V2=DEXP(MU_EV2+ETA(EV2))
S1=V1

$ERROR
Y = F + F*EPS(RSW)
```

Notice the $THETA, $OMEGA and $SIGMA records must be placed ahead of any records that use the symbolic label. So,these records must be placed before $THI, $THR, $PK, and $ERROR. Also, notice that the position of CL V1 in $PK in reverse order relative to the associated etas in $OMEGA. Thus, the inter-subject variability of CL is ETA(1), and its fixed effect is THETA(2). The user no longer needs to consider the numeric ordering of thetas and etas, however, and just utilize the symbols, and NMTRAN will map out everything correctly in the final FSUBS routine. If later, you wish to add a theta, you can place it anywhere in the $THETA record (its position in $THETA defines its numerical index), and not be concerned about renumbering the thetas in the code.

If specifying the Omega or Sigma values using VALUES(), then you can enter

```
$OMEGA BLOCK(4) NAMES(ECL,EV1,EQ,EV2) VALUES(0.03,0.01)
```

VALUES() must come after NAMES() in the $OMEGA record.

Similarly, you may find it more convenient to specify Theta symbols as follows:

```
$THETA NAMES(V1,CL,Q,V2)
(0.0,7.0) (0.0,7.0) (0.0,7.0) 7
```

For dynamic, or implicit mapping of labels, such as for various occasions, these still need to be done via the $ABBR REPLACE record.

Easier Inter-occasion variability modeling (NM73)

Abbreviated code Replacement Feature and Repeated Feature of $OMEGA may be combined for easier Inter-occasion variability modeling. For example,

```
$ABBR REPLACE ETA(OCC_CL)=ETA(4,7,10)
;when OCC=1, eta(4) to be used: when OCC=2, eta(7) to be used, etc.
$ABBR REPLACE ETA(OCC_V) =ETA(5,8,11)
$ABBR REPLACE ETA(OCC_KA)=ETA(6,9,12)
$PK
CL=TVCL*EXP(ETA(1)+ETA(OCC_CL))
V =TVV *EXP(ETA(2)+ETA(OCC_V))
KA=TVKA*EXP(ETA(3)+ETA(OCC_KA))
$OMEGA BLOCK(3) 0.1 0.01 0.1 0.01 0.01 0.1
$OMEGA BLOCK(3) 0.03 0.001 0.03 0.001 0.001 0.03
$OMEGA BLOCK(3) SAME(2); Repeat OMEGA BLOCK(3) SAME twice
```

In the above example, the NMTRAN parses the variable name OCC_CL at the underscore, and determines that there is a data item called OCC with which to associate the variable with the etas listed.

Symbolic Label Substitutions of Model Compartments (NM73,NM75)

As of NM73, one can readily substitute compartment designations by defining appropriate $ABBR REPLACE statements, such as:

```
$ABBR REPLACE A(CENTRAL)=A(1)
$ABBR REPLACE DADT(CENTRAL)=DADT(1)
```

The names designated in $ABBR REPLACE do not need to be the same as those designated in the $MODEL record, although this may cause reader's confusion if you do not do this.

As of NM75, compartment-name symbols defined in $MODEL are automatically available for substitution without requiring $ABBR RECORD definitions. For example:

```
$MODEL
   COMP=(GUT,DEFDOS)
   COMP=(CENTRAL,DEFOBS)
   COMP=(PERI)
```

allows susbsitutions to be made for A(GUT), DADT(GUT), A(CENTRAL),DADT(CENTRAL), etc, so you may use these symbols in your NMTRAN code, for example:

```
$DES
DADT(GUT)=-KA*A(GUT)
DADT(CENTRAL)=KA*A(GUT)-(KCP+KC0)*A(CENTRAL)+KPC*A(PERI)
DADT(PERI)=KCP*A(CENTRAL)-KPC*A(PERI)
…
$ERROR
IPRED=A(CENTRAL)/S2
```

$FORMAT FMTN=3 (default) (NM75)

You may now display any significant digits for thetas, omegas, and sigmas results, and variance-covariance of estimates, listed in the NONMEM report file. The FMTN is the number of significant digits (which is by legacy and default 3), between 3 and 23. This also applies to $TABLE outputs to the NONMEM report file. The FORTRAN format will be formed as 1PE{FMTN+6}.{FMTN-1}. For example FMTN=6 will be 1PE12.5. If you wish G field format, set the FMTN to a negative value. For example FMTN=-4 will be 1PG10.3.

It is recommended that you place the $FORMAT record immediately after the $PROB record:

```
$PROB My problem
$FORMAT FMTN=5
```

This FMTN precision format will be in effect for outputs of all problems, until another $FORMAT record is given for a new $PROB in the control stream.

DO WHILE enhancement (NM73)

DOWHILE may now be used in all blocks of abbreviated code. If a variable is used as a DOWHILE loop variable, it must be declared:

```
$ABBR DECLARE DOWHILE I
```

Recursive random variables ("dowhile recursive variables") may be computed in DOWHILE blocks, as well as in ordinary abbreviated code. A new example (..\examples\sumdosetn.ctl) uses DOWHILE for dose super-imposition in a transit compartment, and includes the following:

```
...
$abbr declare dosetime(100),dose(100)
$abbr declare dowhile i
$abbr declare dowhile ndose

$PK
CALLFL=-2
IF (NEWIND < 2) NDOSE=0

IF (AMT > 0 .and. cmt==1) THEN
 NDOSE=NDOSE+1
 dosetime(NDOSE)=TIME
 DOSE(NDOSE)=AMT
ENDIF
...
$DES
INPT=0
I=1
DOWHILE (I<=NDOSE)
IPT=0
IF (T>=dosetime(I)) IPT=DOSE(I)*(T-dosetime(I))**NN*EXP(-KTR*(T-dosetime(I)))
INPT=INPT+IPT
I=I+1
ENDDO
```

See also ssaddl.ctl, ssonedose.ctl, and ssmultidose.ctl for additional examples.

Subscripted Variables Enhancement (NM73)

Subscripts may be used with user-defined variables that are declared to be arrays using the $ABBR DECLARE record, and also with certain reserved variables such as THETA. Subscripts may be integer variables and expressions. For example,

```
$ABBR DECLARE INTEGER IND
$ABBR DECLARE X(10)
$PK
IND=1
X(IND)=THETA(IND+1)
```

In general random variables such as ETA may not have variables as subscripts. See section SUBSCRIPTED VARIABLES of the help entry for ABBREVIATED CODE for a more complete list.

Autocorrelation (CORRL2) (NM73)

Correlation of residual variables using CORRL2 may now be written in abbreviated code.

For example (..\examples\ar1mod.ctl):

```
$ABBR DECLARE T(NO)
$ABBR DECLARE DOWHILE J
$ABBR DECLARE INTEGER I
…
$ERROR
IF(NEWIND.NE.2)I=0
IF(MDV.EQ.0)THEN
I=I+1
T(I)=TIME
J=1
DOWHILE (J<=I)
CORRL2(J,1)=EXP(-THETA(4)*(TIME-T(J)))
J=J+1
ENDDO
ENDIF
```

Simulation with autocorrelation is also possible. A new example is provided (..\examples\ar1newsim.ctl).

MOD Function (NM73)

The Fortran intrinsic function MOD may now be used in abbreviated code:
k=MOD(i,j)
MOD returns the remainder when i is divided by j. The variables i and j must be either both integer or both real. For example,

```
TAD = MOD(TIME,24.0D+00)
```

will make the 24 a double precision variable, in keeping with the data type of TIME.

This function should not be involved in evaluation of the objective function.

MIN,MAX Functions (NM73)

The Fortran intrinsic functions MIN and MAX may now be used in abbreviated code:

```
DVALUE=MAX(VAL1,VAL2,VAL3…)
```

However, this function should not be involved in evaluation of the objective function. IF THEN statements should be used for those, for example:

```
DVALUE=VAL1
IF(VAL2>DVALUE) DVALUE=VAL2
IF(VAL3>DVALUE) DVALUE=VAL3
```

Also, ensure that if any data items are used in the MAX or MIN argument, that this data item appeared elsewhere in the code block. For example, consider a data item called AGE,

```
LAGE=LOG(AGE)
AGE2=MAX(AGE,0.0)
```

GAMLN Function (NM73)

The GAMLN function returns an accurate evaluation of the logarithm of the gamma function. It can be used in the evaluation the factorial:

```
FAC=exp(gamln(x+1.0))
```

Where

FAC=X!=X*(X-1)*(X-2)…*1

It is more accurate that the Stirling's approximation, and may be used in abbreviated code in the evaluation of the objective function.

RANDMT Function (NM75)

The RANDMT function returns a random variable, than can be used during estimation.

```
rval=RANDMT(N)
```

Where

N: integer Degrees of freedom

0: get uniform random variable

N=1 or N>101: get normal random variable

N>1 and N<=100: get Student-t random variable with N-1 degrees of freedom

The random source is the same as that used internally by NONMEM during estimation (ICALL=2), such as for Monte Carlo EM and BAYES methods, with starting seed and random generator algorithm defined by $EST SEED and RANMETHOD (see *RANMETHOD=[n|M|S|m|P] (default n=3) (NM72)*and *SEED=11456 (default)* in I.36 Monte Carlo Importance Sampling EM). During simulation (ICALL=4), RANDMT uses source defined by the first seed, and random generating algorithm specified by RANMETHOD, defined on the $SIML record. During NPDE generation for diagnostic assessment, the RANMETHOD and SEED specified on the first $TABLE record is the source (*RANMETHOD=[n|S|m|P] (default n=3) (NM72)SEED* in section I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format). Similarly, during the $CHAIN step, the $CHAIN SEED= and $CHAIN RANMETHOD= statements define the source (see *$CHAIN Record*).

During simulation, it is best to use the RANDOM(K,R) routine directly as described in the help guide (VIII.pdf), or via the probability density function generators (section I.32 Probability Density Functions (NM742)).

RANDMTU Function (NM75)

The RANDMTU function returns a random variable.

```
rval=RANDMTU(N,SOURCE,STARTSEED,RANM)
```

Where

N: integer Degrees of freedom

0: get uniform random variable
N=1 or N>101: get normal random variable
N>1 and N<=100: get Student-t random variable with N-1 degrees of freedom

SOURCE: Integer between 0 and 20, allowing the user to maintain up to 20 distinct seed sources. IF SOURCE=0, it uses the default source. The default source is that specified for $EST .. SEED= … RANMETHOD=… during estimation, or $SIML during simulation.

STARTSEED: Integer, which if not 0, causes initialization with the designated starting seed. STARTSEED should be set to 0 after this initialization. If SOURCE=0, this argument is ignored. If STARTSEED is <0, then the starting seed will be 10000*(seconds after midnight)+ABS(STARTSEED). The actual startseed is stored in RANDMTU_STARTSEED(SOURCE), module NM_BAYES_INT, accessible by using nonmem_reserved_general (see the example below).

RANM: Integer, indicating the type of random number generator algorithm to use (set at initialization when STARTSEED/=0), as follows:
0: ran0 of reference [5], minimal standard generator
1: ran1 of reference [5], Bays and Durham.
2: ran2 of reference [5].
3: ran3 of reference [5], Knuth.
4: NONMEM's traditional random number generator, used as default in $SIMULATION

If you wish to use the default random number generator relative to the present ICALL value, then set RANM to -1. This will use whatever random number generator has been specified in the $EST statement (ICALL=2), or $SIML statement (ICALL=4) to be used. If SOURCE=0, then this argument is ignored.

Random sources of RANDMTU are not properly preserved during Parallelization, unless SOURCE=0.

If you wish to use SOURCE=0, it is easier to simply call RANDMT, rather than RANDMTU(N,0,…):

```
rval=RANDMT(N)
```

Example:
In the following, 2 random sources are used, with different starting seeds

```
$ERROR ; may also be used in other records
include nonmem_reserved_general
…
…
; Make sure NMTRAN is aware of RVAL1 and RVAL2, so it is accessible
; for $TABLE output
  RAVL1=0.0
  RVAL2=0.0
  RVAL3=0.0
  IF(NEWIND==0) THEN
; Initialize only at beginning of the data set.
```

```
; 3=t-distribution with DF=2
  RVAL1=RANDMTU(3,1,-1234,-1)
  write(*,*) RANDMTU_STARTSEED(1)
  RVAL2=RANDMTU(3,2,5678,-1)
  ENDIF
  RVAL1=RANDMTU(3,1,0,-1)
  RVAL2=RANDMTU(3,2,0,-1)
  RVAL3=RANDMT(3)
```

Note that calling RANDMT(N) (or equivelantly RANDMTU(N,0,…)) did not require an initialization call (and would be ignored if you did), as this is under the control of $EST, $SIML, or $TABLE.

The RANDMTU random number generating system is independent of the CALL RANDOM() system. The RANMTU() routine has the flexibility to be used for estimation or simulation, and does not require its seed to be defined on the $SIML record, whereas RANDOM() can only be used during ICALL=4, and its seed must be defined on the $SIML record. During simulation, it is best to use the RANDOM(K,R) routine directly as described in the help guide (VIII.pdf), or via the probability density function generators (section I.32 Probability Density Functions (NM742)).

Declaring Reserved Variables (NM73)

Some useful reserved variables are explicitly recognized by NMTRAN that can be used by the user. There are however many other variables that are generally internal to NONMEM, and often are not needed by users except occasionally, which are not explicitly recognized by NMTRAN, and so cannot be used in abbreviated code, but must be used with verbatim code (“ at beginning of line). For example the variable ITER_REPORT is available that contains the present iteration number as reported to the console or NONMEM report file, that may be useful to be accessed within the $PK, $ERROR, or $PRED code. A convenient means of accessing this variable, as well as letting NMTRAN allow you to use that variable in abbreviated code is to place its MODULE definition in an include file that begins with the name NONMEM_RESERVED (case insensitive) at the beginning of the section you want to use it. For example, NONMEM_RESERVED_GENERAL in the ..\util directory has many quite useful variables listed, including ITER_REPORT, in the form of:

```
"C ITER_REPORT: Iteration number that is reported to output
"C (can be negative, if during a burn period).
"C BAYES_EXTRA, BAYES_EXTRA_REQUEST, used in example 8
" USE NMBAYES_REAL, ONLY: OBJI
" USE NMBAYES_INT, ONLY: ITER_REPORT,BAYES_EXTRA_REQUEST,BAYES_EXTRA
" USE PNM_CONFIG, ONLY: PNM_NODE_NUMBER
" USE NM_INTERFACE, ONLY: TFI,TFD
```

The user may use any one of these variables, such as shown in example 8:

```
$PK
include nonmem_reserved_general
BAYES_EXTRA_REQUEST=1
MU_1=THETA(1)
```

```
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 .AND. TIME==0.0) THEN
WRITE(50,*) ITER_REPORT,ID,CL,V1,Q,V2
ENDIF
```

Note the lack of needing to begin a line with " when using ITER_REPORT, BAYES_EXTRA_REQUEST, or BAYES_EXTRA, because NMTRAN "read" the nonmem_reserved_general file, and listed the variables declared in there as acceptable to use. A copy of the nonmem_reserved_general file is in the ..\util directory. It needs to be placed in the present run directory so NMTRAN has access to it. You could opt to copy only part of the list in nonmem_reserved_general according to need into any file with name starting with nonmem_reserved…

A list of useful variables and their meanings are also listed in ..\guides\useful_variables.pdf. However, the more complete list is in nonmem_reserved_general. Be careful in its use, as you have the ability to change the values of these reserved variables, and this could crash the system if you change the wrong thing.

Note also that the nonmem_reserved_general file may contain function declarations, such as TFI and TFD, which are convenient functions to easily convert an integer to text ("text from integer" TFI) or double precision value to text ("text from double" TFD) . This is quite useful so that the compiler can catch a misuse of that function's arguments.

If you wish to define your own function, and have the information about its proper use of arguments be conveyed upon its execution, so the compiler may detect errors, then one method is to package the definition of the function in a USE module, such as is done in the following example:

Myfuncmodule.f90 defines the functions mymin and mymax:

```
MODULE MYFUNCS
contains
function mymin(a,b,c,d,e)
integer mymin
integer a,b,c,d,e
mymin=min(a,b,c,d,e)
end function
function mymax(a,b,c,d,e)
integer mymax
integer a,b,c,d,e
mymax=max(a,b,c,d,e)
end function
END MODULE MYFUNCS
```

Nonmem_reserved_myfunc is the include file that declares its use:

```
" USE myfuncs, only: mymin,mymax
```

and the following control stream file uses the function:

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP

$SUBROUTINES  ADVAN2 OTHER=myfuncmodule.f90

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=MEAN ELIMINATION RATE CONSTANT (1/HR)
;THETA(3)=SLOPE OF CLEARANCE VS WEIGHT RELATIONSHIP (LITERS/HR/KG)
;SCALING PARAMETER=VOLUME/WT SINCE DOSE IS WEIGHT-ADJUSTED
include "nonmem_reserved_myfunc"
   CALLFL=1
   KA=THETA(1)+ETA(1)
   K=THETA(2)+ETA(2)
   CL=THETA(3)*WT+ETA(3)
   SC=CL/K/WT
I=mymin(1,2,3,4,5.0)
print *,'I ',I

$THETA  (.1,3,5) (.008,.08,.5) (.004,.04,.9)
$OMEGA BLOCK(3)  6 .005 .0002 .3 .006 .4

$ERROR
   Y=F+EPS(1)

$SIGMA  .4
```

If you use the wrong argument type (real instead of integer), or perhaps use the wrong number of arguments, the compiler will readily flag this.

Numerical Equality Comparison for IGNORE option in $DATA Record (NM73)

When the IGNORE option is used to filter records from the input file, the .EQ., =, .NE., and /= symbols perform literal string comparisons. To provide a numerical equality comparison, use .EQN. for numerical equals, and .NEN. for numerical not equals. For example
$DATA FILE=myfile.txt IGNORE=(OCC.EQN.1)
Will filter out all records for which the data item OCC is equal numerically to 1, even if it is stored as 1.0, or 1.00e+00, etc.

$DATA FILE=myfile.txt IGNORE=(OCC.EQ.1)
only filters out records for which OCC is literally ‘1’.

$DATA MISDAT (NM74)

You may wish to define a particular numerical value to indicate a missing data value in your data set, which is displayed on $TABLE table outputs, but is safely interpreted as 0 by other steps of

NONMEM. To do this, set MISDAT at the $DATA record, for as many misdat labels you may need (up to 20):

$DATA mydatafile MISDAT=1.0E-99 MISDAT=1.0E-102

Any values in the data set that are one of the MISDAT values will be interpreted as 0 during estimation, simulations, etc., but upon output to tables, will have the MISDAT value, to keep track of missing data values (or other markings in the original data set).

$DATA REPL (NM75)

For clinical trial simulations, typically one creates a data file containing template subjects, which are to be replicated a number of times. The replication of the template subjects was often done in R or Excel, as NONMEM could not expand the data file in this way. As of NM75, NONMEM will replicate each subject REPL=n times, and then utilize this expanded data set. (Note that the NONMEM data set is typically the file FDATA generated by NM-TRAN, unless there is nothing for NM- TRAN to change and the format is supplied on the $DATA record, in which case the file named on the $DATA record is the NONMEM data set. See NONMEM Users Guide, Part IV.) For example, suppose a data file, called template.csv, contains 3 subjects, each one representing a particular covariate type, dosing type, and/or sample time pattern. The desire is that each of these subjects be replicated 100 times in the data set. This is done with the following record.

```
$DATA template.csv ignore=@ REPL=100
```

Make sure that the ID values of the template are integer valued. If the number of replications is greater than the difference between two consecutive ID values in the template, NONMEM will add a fractional value to each replicate subject, so that the integer portion of the ID is the original ID, and the fractional portion represents the replication number (NONMEM does not require integer valued ID's, only unique ID numbers that have no more than 14 significant digits). Otherwise, the replicate ID's will be integer incremented. To assure only integer ID's are created, make sure the template ID's are spaced by an amount greater than the number of replications desired for each subject.

To specify a different replication for each subject, a reserved data item has been introduced in nm75, called REPL_, the value of which will be used as the replication number for that subject. If the value is fractional, it will be truncated to the nearest integer. Only the REPL_ value of the first record of each individual will be used to determine its replication. For example, in the following data set:

Control stream:

```
$INPUT ID TIME  AMT  RATE EVID MDV DV  REPL_
$DATA warfarin.dat ignore=C
```

Data file:

```
CID TIME    AMT  RATE EVID MDV DV    REPL_
1.0 0.0    70.0  0.0  1    1   0.0   2
1.0 0.5     0.0  0.0  0    0   1.0   .
1.0 1.0     0.0  0.0  0    0   1.0   .
```

```
1.0 2.0      0.0   0.0   0     0    1.0    .
1.0 6.0      0.0   0.0   0     0    1.0    .
1.0 24.0     0.0   0.0   0     0    1.0    .
1.0 36.0     0.0   0.0   0     0    1.0    .
1.0 72.0     0.0   0.0   0     0    1.0    .
1.0 120.0    0.0   0.0   0     0    1.0    .
2.0 0.0     70.0   0.0   1     1    0.0    3
2.0 0.5      0.0   0.0   0     0    1.0    .
2.0 1.0      0.0   0.0   0     0    1.0    .
2.0 2.0      0.0   0.0   0     0    1.0    .
2.0 6.0      0.0   0.0   0     0    1.0    .
2.0 24.0     0.0   0.0   0     0    1.0    .
2.0 36.0     0.0   0.0   0     0    1.0    .
2.0 72.0     0.0   0.0   0     0    1.0    .
2.0 120.0    0.0   0.0   0     0    1.0    .
```

data of subject 1 will be replicated 2 times within NONMEM, and data of subject 2 will be replicated 3 times. The ID values of the replicates will be 1.00, 1.01, 2.0, 2.01, 2.02.

You may use the $DATA REPL option along with the REPL_ data item such as:

```
$INPUT ID TIME  AMT  RATE EVID MDV DV  REPL_
$DATA warfarin.dat ignore=C REPL=100
```

which wil result in 200 replications for subject 1, and 300 replications for subject 2. The order in which they appear will be as follows. Two replicates of subject 1 followed by 3 replicates of subject 2, and this pattern is replicated 100 times. If one subject type takes much longer than another to compute during estimation, this pattern allows the best load balancing for parallelization.

Of course, the DV patterns will be identically replicated as well, so replicated data templates make sense only if you will be using $SIML or $DESIGN to subsequently create the unique DV patterns for each replicated subject.

PRED_IGNORE_DATA Feature (NM75)

The $DATA IGNORE=(list) and ACCEPT=(list) provide a limited means of filtering the input data set, which is performed by NMTRAN. To provide more elaborate filtering for excluding data, PRED can instruct NONMEM to filter out additional data records at the beginning of the run or problem.

This is done by creating a PRED_IGNORE_DATA_TEST==1 IF block presented in $INFN, $PK, or $PRED. For eample:

```
$INFN
IF(PRED_IGNORE_DATA_TEST==1) THEN
 PRED_IGNORE_DATA=0
 IF(AGE>35.0) PRED_IGNORE_DATA=1
 IF( ID>10.AND.ID<18.OR.ID>60.AND.ID<70 ) PRED_IGNORE_DATA=1
```

```
 RETURN ;Assures no additional code in INFN is executed (saves time)
ENDIF
```

Or:

```
$PRED
IF(PRED_IGNORE_DATA_TEST==1) THEN
PRED_IGNORE_DATA=0
 IF(AGE>35.0) PRED_IGNORE_DATA=1
 IF( ID>10.AND.ID<18.OR.ID>60.AND.ID<70 ) PRED_IGNORE_DATA=1
RETURN ;Assures no additional code in PRED is executed (saves time)
ENDIF
…
```

If PRED_IGNORE_DATA is set to a non-zero value, then the data record is ignored, and excluded from the internal data set. This allows the user to use more complex, multi-line and FORTRAN syntax based, logical operations on data record exclusions.

When PRED_IGNORE_DATA_TEST=1, then ICALL is set to -1. the following variables have properly defined values during this call:

NEWIND, NEWL2,IPROB, NPROB, S1NUM, S2NUM, S1NIT, S2NIT, S1IT, S2IT

So, it is possible to restrict PRED_IGNORE_DATA actions to a particular problem number:

```
IF(IPROB==2.AND.PRED_IGNORE_DATA_TEST==1) THEN
 PRED_IGNORE_DATA=0
 IF(AGE>35.0) PRED_IGNORE_DATA=1
 IF( ID>10.AND.ID<18.OR.ID>60.AND.ID<70 ) PRED_IGNORE_DATA=1
 RETURN
ENDIF
```

No other variables are properly defined during PRED_IGNORE_DATA_TEST=1, such as THETAS, data record information such as NIREC, NDREC, etc., and no calls to complicated functions that such as RANDOM() will be valid (simple functions, such as built-in FORTRAN functions, are fine). Furthermore, changes to data items may not be made during this call. Any other functions of $INFN, such as data modification, RANDOM() calls, etc., should be made with separate ICALL==0 or ICALL==1 blocks.

If the user write his own INFN routine in which PRED_IGNORE_DATA code is constructed, then NONMEM needs to be informed with the PRED_IGNORE_DATA option in $DATA. For example:

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP PRED_IGNORE_DATA
$SUBROUTINES  ADVAN2 INFN=myinfn.f90
```

myinfn.f90:

```
      SUBROUTINE INFN(ICALL,THETA,DATREC,INDXS,NEWIND)
      USE SIZES,     ONLY: DPSIZE,ISIZE
```

```
USE NMPRD_INT, ONLY: PRED_IGNORE_DATA,PRED_IGNORE_DATA_TEST
INTEGER(KIND=ISIZE) :: ICALL,INDXS,NEWIND
REAL(KIND=DPSIZE) :: THETA
REAL(KIND=DPSIZE) :: DATREC
DIMENSION :: THETA(*),DATREC(*),INDXS(*)
IF(PRED_IGNORE_DATA_TEST==  1)THEN
IF (DATREC(3)>3) THEN
PRED_IGNORE_DATA=1
ENDIF
ENDIF
RETURN
END
```

If PRED_IGNORE_DATA_TEST and PRED_IGNORE_DATA are used only in verbatim code, it
is also necessary to code the PRED_IGNORE_DATA option in $DATA.

$DATA CREATES FILE FDATA.csv (NM75)

An additional file, FDATA.csv is produced that outputs the contents of its input data file (typically FDATA) in a comma delimited file format, so you can check how NONMEM interprets the input data. The records in FDATA.csv may differ from those in FDATA in the following cases. If REPL/REPL_ is used, the replicated form of the data will appear in FDATA.csv. Also, records excluded by PRED_IGNORE_DATA will not be present in FDATA.csv.

I.8 Invoking NONMEM

NONMEM 7.5 can be invoked using one of the supplied scripts:

nmfe75.bat for Windows
nmfe75 for Linux/Unix

These script files take at least two arguments, the control stream file name, and the main report file name, such as:
Windows:

```
nmfe75 mycontrol.ctl myresults.res
```

Unix:

```
./nmfe75 mycontrol.ctl myresults.res
```

The control stream file name is passed to NONMEM as its first argument. Write and print statements supplied by the user in verbatim code will be routed as follows:

Unit * prints to console
Unit 6 prints to report file
WRITE(*,… or PRINT *,… : to console
WRITE(6,… to report file.

If you wish to reroute all console output to a file, the execution statement could have a redirection added to it:
Windows:
```
nmfe75 mycontrol.ctl myresults.res >console.txt
```
Linux:
```
./nmfe75 mycontrol.ctl myresults.res >console.txt
```

To prevent NONMEM from polling the standard input for ctrl key characters (a new feature described later):
Windows:
```
nmfe75 mycontrol.ctl myresults.res -background>console.txt
```
Linux:
```
./nmfe75 mycontrol.ctl myresults.res -background>console.txt
```

In Unix/Linux, you can additionally append & to the command to execute it in the background (you must also use –background option when using &):
```
./nmfe75 mycontrol.ctl myresults.res –background >& console.txt &
```

And periodically monitor the rerouted file:
```
tail –f  console.txt
```

For the more adventurous user, you may modify the nmfe75 scripts for alternative behaviors.

Additional options are available to make execution of the nmfe75 script more flexible. From the nmfe75 command line, the user may enter a run directory that is different from the directory in which the nmfe75 script is launched:
```
-rundir=c:\my_favorite_dir
```

Where rundir is the run directory if it is different from the present working directory (you must make sure all user dependent input files, control stream file, msf files, and data files, are available in that run directory).

The user may also enter an alternative name for the constructed executable:

```
-nmexec=nonmem2
```

specifies an alternative executable name, than the default nonmem.exe (windows) or nonmem (Linux).

To turn off production of the XML output file root.xml, where root is the root name of the control stream file, use the option –xmloff.

Activities that take a short period of time (less than a minute, for example) such as simulations of simple models or small data sets, particularly when only simulations are performed ($SIM ONLYSIM) can benefit from setting –xmloff. Also, new in nm74, -flushtime=x will cause file updating (flushing) to occur not more frequently than every x seconds. So, using the command line options
-xmloff -flushtime=10.0

may increase the speed considerably for simulations. File flushes for the MSF system are not affected, and continue to be updated every PRINT iterations during estimation. The default setting of flushtime is 1 second, suitable for most problems, but you can change this as needed (fractional seconds are also allowed). If you prefer file updates occurring as frequently as possible, you can set flusthime to 0.0, but this could cause considerable slowing of execution for small problems and data sets.

Beginning in NM73, an additional feature of the execution script file is that the path to the fortran compiler system and MPI system that is appropriate for NONMEM may be retrieved from a script file that could have the following environment variables defined:
compilerpath
mpibinpath
mpilibpath
mpilibname
Comments in these files are provided for instructions about each of these environment variables. These paths will be temporarily added to the front of the PATH environment variable, so that the appropriate compiler or MPI system is called to service NONMEM. In the past, conflicts with other installed fortran compilers from other applications would prevent the appropriate compiler from being used for the NONMEM system. This location file method allows NONMEM to be forced to look in a particular location.

The location file should be called nmloc.bat or nmloc by convention (see ..\util\nmlocoriginal.* as templates). It may be specified at the nmfe7x command line by the -locfile option, for example:
nmfe75 myfile.ctl myfile.res -locfile=nmloc.bat

If –locfile is not specified, the nmfe7x script looks in the present working directory for nmloc.bat (windows) or nmloc (linux). If this file is not found, it looks in the top directory of the NONMEM installed directory. Thus, the file nmloc.bat (Windows) or nmloc (Linux) in the top nonmem installed directory serves as the default location file, and may be modified, or used as a template and placed in the working directory or specified in the –locfile option on the command line. If a particular environment variable in the above list is not found or is not defined, then nmfe7x will behave as in earlier versions, and rely on the presently existing PATH for finding the compiler and MPI system. The nmfe7x script will display a statement as to what path it will use.

To find environment variables for your fortran compiler, you can use the “which” command in Linux based operating systems. Open a terminal window and type the command

which gfortran
or
which ifort
depending on your compiler.

As an example, this may return
/usr/local/bin/gfortran

Next, locate the file nmloc in the top directory of your NONMEM installation. The environment variable "compilerpath" needs to be modified to point to this directory, for example:
compilerpath=/usr/local/bin:/usr/bin

You can do the same for finding the path to your MPI system:

which mpiexec

or whatever the executable name of your MPI system is, and populate the enveironment variable "mpibinpath" in the nmloc file.

The Inel fortran compiler system on Windows can have an elaborate setup, and typically creates an icon on the desktop for a specialized command window, and this window must be opened and from which NONMEM should be run. However, some GUI software, such as PDx-Pop, cannot make use of this special command window. In such cases, part of the command in the properties section of the desktop icon, which is usually something like:
C:\Windows\System32\cmd.exe /K {location of ifortvars.bat}\ifortvars.bat intel64 vs2010

can be placed in the nmloc.bat file, in the following manner:

call {location of ifortvars.bat}\ifortbvars.bat intel64 vs2010
and setting the "compilerpath" environment variable is not necessary:
rem set compilerpath=

I.9 Dynamic Memory Allocation (NM72)

With NONMEM 7.2.0 and higher versions, the user need no longer specify "big" or "reg" when using SETUP72 (or SETUP73) to install NONMEM. (The reg/big/same choice is ignored. It is in effect always "same" and is shown as "same" in all examples. However, some constants in SIZES are not dynamically allocated, for example, LSTEXT or PNM_MAXNODES. See help entry for sizes, or see comments regarding the various parameters in resource\SIZES.f90). NMTRAN sizes each NONMEM executable only as large as it needs to be for the specific control stream run. NONMEM 7.2.0 has the ability to dynamically size the main arrays in NONMEM, according to the number of subjects, and number of parameters described in the control stream file, etc. To do this, NMTRAN determines the appropriate sizes for arrays, and puts this information in a subroutine called FSIZESR in the FSUBS file. NONMEM dynamically allocates the sizes of arrays at run-time, based on the values in FSIZESR. Although unnecessary for most problems, the user may override the size that NMTRAN assesses for a select number of arrays, by including a $SIZES statement as the first non-comment line of the control stream file. For example:

$SIZES MAXIDS=230 NO=300 LTH=50 LVR=30

The following is an example of FSIZESR information from a run with CONTROL5. All parameters can be changed with $SIZES (see resource/sizes.f90 for descriptions and default

values), except NTT, NOMEG, NSIGM, PPDT, which are always evaluated properly by NMTRAN and should not be over-ridden.

```
LTH             3
LVR             4
LVR2            0
LPAR            10
LPAR3           0
NO              0
MMX             1
LNP4            0
LSUPP           1
LIM7            0
LWS3            0
MAXIDS          12
LIM1            0
LIM2            0
LIM3            0
LIM4            0
LIM5            0
LIM6            0
LIM8            0
LIM10           0
LIM11           0
LIM13           0
LIM15           0
LIM16           0
MAXRECID        0
PC              0
PCT             1
PIR             1
PD              7
PAL             0
MAXFCN          0
MAXIC           0
PG              0
NPOPMIXMAX      0
MAXOMEG         3
MAXPTHETA       4
MAXITER         20
ISAMPLEMAX      0
DIMTMP          0
DIMCNS          0
DIMNEW          0
PDT             4
LADD_MAX        0
MAXSIDL         0
```

```
NTT             3
NOMEG           3
NSIGM           1
PPDT            3
```

The file FSIZES is also produced that contains the same contents as the FSIZESR routine in FSUBS. The FSIZES file is produced for easy reading for the user, and is not used by the NONMEM system. Those parameters with a 0 cannot be determined or are not given by NMTRAN and will default to the values hard-coded in resource\SIZES.f90. See the file SIZES.f90 itself, or on-line help entry for sizes, for these values. On occasion, NMTRAN mis-interprets the true scope of the run, and NONMEM may stop the run because one of the sizing parameters was too low. The user should then insert a $SIZES record in the control stream file, set the offending sizing parameter to the appropriate value, and run the problem again.

SIZES.f90 no longer contains parameters DIMPKS and DIMRHS and DIMRV for NMTRAN. The arrays sized by these parameters are dynamically allocated to whatever size is necessary for the abbreviated code in the current control stream. All other arrays for NMTRAN can be increased in size if necessary with $SIZES.

As of NM73, NMTRAN determines the maximum number of observation records (MDV=0) that occur in any subject, among all data files used in the entire control stream file. If this value is greater than the NO value listed in SIZES.f90, it will set NO to this larger size. Thus, users no longer have to be conscientious of sizing the NO parameter. However, there is no guarantee that NMTRAN will correctly assess NO for the entire scope of the control stream file for all types of problems. Should this occur, NONMEM may issue an error, and the user will need to set the NO value with a $SIZES record.

When PREDPP ($PK, $ERROR, $INFN, etc.) is used, NMTRAN also creates a sizes file called prsizes.f90. This file contains sizing and other parameters needed by PREDPP. Some parameters (PD, LVR which sets the prsizes parameter PE) are the same as in FSIZES and have the same values. Some (PC, PCT, PIR, PAL, MAXFCN, MAXRECID) are unique to PREDPP and prsizes.f90. All may be changed with $SIZES. For example, $SIZES MAXFCN=9000000 might be used with General Non-Linear models ADVAN6, ADVAN8, ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, ADVAN18) to request more function evaluations than the default value in resource\SIZES.f90, which is MAXFCN=1000000. As of NM73, PCT and PIR are assessed by NMTRAN and submitted to NONMEM, if –prdefault is not used.

Usually a parameter value needs to be specified in $SIZES when the problem is bigger than what is specified in sizes.f90. For example, if LTH=40 in sizes.f90, and your problem needs only 35 thetas, then NONMEM executable will be built to size for 35 thetas, and $SIZES was not needed. If, however, the problem requires 45 thetas, then
$SIZES LTH=45
or greater needs to be specified, and then NONMEM will be set to a size of LTH=45 as well.

For the following parameters LTH, LVR, PD, PC, DIMTMP, MMX, DIMCNS, and/or PDT, NMTRAN must anticipate a maximum size, because it needs to set up internal arrays that stores

the information it will gather from the control stream file. It will get this maximum size from the values in sizes.f90, or from the user specifying the required size in $SIZES. If the user does not specify in $SIZES, then NMTRAN will determine the best size for the problem and construct the NONMEM executable accordingly. But if the user specifies a size in $SIZES, then this is also the size by which the NONMEM executable will be constructed.

To anticipate large sizes without needing to specify values in $SIZES, then set LTH, LVR, PD, PC, DIMTMP, MMX, DIMCNS, and/or PDT in sizes.f90 to the maximum you think you will ever need. NTMRAN will still create a NONMEN executable that is sized to fit the problem. Be aware, however, that if parameter values are set too large, NMTRAN may not run, as it uses sizes.90 to set its array sizes at the beginning, before it knows the actual size of the problem.

As of NM73, as an alternative to modifying sizes.f90 to very large maximum sizes, you can tell NMTRAN the maximum size that may be needed by specifying a $SIZES parameter as a negative value. Thus, a user can give NMTRAN permission to deal with all problems that have data input files that have up to 1000 data items, and up to 150 omegas, and up to 200 thetas, by the following:
$SIZES PD=-1000 LVR=-150 LTH=-200

but the size of these parameters when the NONMEM executable is constructed will be only what is needed for the particular problem. In contrast,
$SIZES PD=1000 LVR=150 LTH=200

will result in sizing the NONMEM executable with these values, and won't make a "tailor fit". This would result in a very large executable regardless of the model size. Thus, $SIZES PD=-1000 tells NMTRAN that you may need as many as 1000 data items in a data file, whereas $SIZES PD=1000 tells NMTRAN that you need exactly that size.

With nonmem 7.1.2 and earlier releases, only FSUBS is compiled at run time. With NONMEM 7.2.0 and higher (nmfe72 and higher) or certain of the PREDPP files in the ..\pr directory are also compiled at run time, with the sizes and values given in prsizes.f90. Thus, arrays internal to PREDPP are statically allocated. In contrast, the NONMEM source code in ..\nm are precompiled and the main NONMEM arrays are allocated dynamically. PREDPP source code is not pre-compiled and dynamically allocated due to significant increase in run times. Many compilers produce a much more elaborate binary code in order to deal with variables that are dynamically shaped, which occurs with dynamically sized variables that have more than one dimension to them, and this slows down execution considerably with routines that are accessed very frequently, such as PREDPP routines.

The nmfe75 script file copies the required PREDPP routines from the nonmem ..\pr directory into a temporary folder (called temp_dir) under the user's run directory, and compiles the routines there. The resulting object files are then linked with NONMEM, and the nonmem executable is created. The compilation of the PREDPP routines may take some time (about 10 to 50 seconds). If you are repeatedly running the same problem, by default the nmfe75 script will skip the PREDPP recompilation. It does this by testing that all of the PREDPP files listed in the

file LINK.LNK from the previous run are appropriate for the present run, and testing that the present prsizes.f90 is not different from the present run.

Typically, you can expect that the nmfe75 script will do a PREDPP recompile when any of the following sizes change LVR,PD, PC, PCT, PIR, PAL, MAXFCN. This could happen if the user changes the values via $SIZES. Also, NMTRAN will resize LVR if the number of $OMEGA entries changes, and it will resize PD if the number of data items listed in $DATA changes. Size changes are all listed in prsizes.f90 in the PREDPP temporary recompile directory. The PREDPP files selected for linking (listed in LINK.LNK) can change if the $SUBROUTINES statement, which specifies ADVAN/TRAN, is changed.

You may force PREDPP recompilation, in case the run does not appear to execute properly when no recompilation occurs, by setting the –prcompile switch:

```
nmfe75 mycontrol.ctl myresults.res -prcompile
```

On the other hand, if the nmfe75 script for some reason believes there is a change in the previous run from the present run, but you are convinced there is not a change, you may force the skipping of the PREDPP compilation step and use the compiled files from the previous run by adding the argument –prsame, at the end of the command line. For example,

```
nmfe75 mycontrol.ctl myresults.res -prsame
```

If you are repeatedly going between two or more problems, so that often they need to be PREDPP recompiled, and you want to save time, you can specify a unique temporary directory for the PREDPP compilation for a given problem, by using –runpdir option at the nmfe75 command line. For example,

You may run problem A as

```
nmfe75 mycontrolA.ctl myresults.res -runpdir=mycontrolA
```

and then follow with problem B as

```
nmfe75 mycontrolB.ctl myresults.res -runpdir=mycontrolB
```

When you return to rerunning problem A at some later time:

```
nmfe75 mycontrolA.ctl myresults.res -runpdir=mycontrolA
```

it won’t need to recompile (assuming your PREDPP sizings and PREDPP model did not change for problem A), as its PREDPP recompile directory was not overwritten by the intervening call to problem B.

Finally, if you feel that it is sufficient to use default sizes in sizes.f90 for the various PREDPP parameters, and therefore use the precompiled routines in ..\pr of the NONMEM installed directory, you may use the –prdefault option:

```
nmfe75 mycontrol.ctl myresults.res –prdefault
```

As of nm73, you may also use the –tprdefault option , which tests if –prdefault is acceptable, and if so, will use it, otherwise, it will perform a PREDPP recompile:
```
nmfe75 mycontrol.ctl myresults.res –tprdefault
```

If you enter
```
nmfe75 mycontrol.ctl myresults.res –tprdefault –prcompile
```
then if –prdefault is not acceptable, and will act on the –prcompile option.

If you enter
```
nmfe75 mycontrol.ctl myresults.res –tprdefault –prsame
```
then if –prdefault is not acceptable, and will act on the –prsame option.

As of nm7.5.1, the option -do2test is available that has NMTRAN check the control stream file for requirement of analytical 2nd derivatives. If required, NMTRAN will build FSUBS with analytical 2nd derivatives, otherwise it will build FSUBS without it, equivalent to the user entering $ABBR DERIV2=NO. For complicated control streams, inserting analytical 2nd derivative equations can take a long time to produce the FSUBS code. Problems use 2nd derivatives when $EST LAPLACE NOINTERACTION without NUMERICAL, or $SIM REQUESTSECOND.

As of nm75, the option –prseq is available to cause the PREDPP files to be compiled sequentially, rather than in parallel (default). Sometimes the parallel method of compiling may actually take longer, or the compiler license may limit the number of simultaneous instances of the compiler that are permitted to be operating at once.

You may skip the NMTRAN step using the –trskip switch:
```
nmfe75 mycontrol.ctl myresults.res –background -trskip
```

The –trskip option is useful if you wish to modify FSUBS created by a previous run, and insert extra debug lines into FSUBS, and prevent your modified FSUBS from being over-written by NMTRAN (it will still be compiled). The trskip and any one of prsame, prcompile, or prdefault switches may be used together.

Conversely, as of nm74, you may skip recompiling and rebuilding a new nonmem executable with the option –nobuild:
```
nmfe75 mycontrol.ctl myresults.res -nobuild
```

This is particularly useful if you are performing a series of bootstrap nonmem runs, which only differ in a starting seed, or initial parameters, etc., but otherwise, the model and the sizing remains the same as the previous run.

The -nobuild option causes nmfe75 script to rerun NMTRAN so it can make fresh FDATA, FCON, FMSG, and FSTREAM, and FSIZES files. None of these should change with each run except FDATA (for new data, but which must have identical data column structure with previous

runs) and FCON (new theta, omega, sigma inputs). Therefore do not change $SIZES, or model code ($PK, $ERROR, $DES, etc.), between –nobuild calls.

For the Pirana environment as an example (courtesy of Devin Pastoor), don't delete the nonmem executable, and point to existing folder for subsequent calls:

```
# first run
execute run007.mod -mod -clean=0

# subsequent runs
execute run007.mod -directory=run007.dir1/ -nmfe_options="-nobuild"
```

I.10 Changing the Size of NONMEM Buffers

The entire data set is not necessarily stored in memory at one time. It may be stored in a temporary disk file, and parts of it are brought into a memory buffer as needed. Some other large arrays are also stored on disk files. Of course, memory-file swapping of data set information leads to increased computer run-time. So the bigger the buffer size, the shorter may be the run time. The sizes of the NONMEM buffers are set by constants LIM1 to LIM16. The default settings of these constants are set in SIZES.f90. If these constants are not adequate, NONMEM will produce error messages such as the following.

```
TOT NO. OF DATA RECS IN BUFFER 1 IS LESS THAN
NO. OF DATA RECS IN INDIVIDUAL REC NO. 1 (IN INDIVIDUAL REC ORDERING)
```

Unlike most of the other dynamically changeable parameters, NMTRAN does not determine the most appropriate LIM value for the problem, but instructs NONMEM to use the default value specified in resource\SIZES.f90 by default. For many problems, the default LIM values are high enough that all of the data may reside in memory without resorting to the buffer files. For large data sets, buffer files are likely to be used. The user may, however, select a LIM value that is different from that specified in sizes.f90, via the $SIZES record in the control stream file, e.g.:

$SIZES LIM1=20000

It is not necessary to recompile NONMEM, just rerun the nmfe75 script, and the appropriate arrays will be allocated according to the user specified LIM value.

It is most desirable to set the LIM value that is the proper size for the run, so that the buffer file does not have to be used. With today's very large memory computers, this should usually be alright to do without running out of memory. Below is a table describing the minimal allowable value for each LIM, and the value needed to prevent using the buffer file for a particular problem:

LIM	Minimum Value	Maximum Value needed to prevent buffer file usage	Buffer files used (FILExx)
1	MAXDREC	TOTDREC	10,13,20,33
2	MAXDREC	TOTDREC	39,14
3	2	MAXIDS	12

LIM	Minimum Value	Maximum Value needed to prevent buffer file usage	Buffer files used (FILExx)
4	2	MAXIDS	15,16
5	2	MAXIDS	17,18
6	MAXDREC	TOTDREC	7,19
7	2	MAXDREC	21,22
8	2	MAXIDS	23,24
9 (nm74, uses LIM1)	MAXDREC	TOTDREC	For simulation REWIND: 41,42,43,44
10 (nm74)	10	MAXIDS*(NPNIND+1)	45,46
11	2	NPROB	31,32
12 (nm74, uses LIM6)	MAXDREC	TOTDREC	47,48
13	2	MAXIDS	11
14	NOT USED		
15	2	MAXIDS	26,27
16	MAXDREC	TOTDREC	26,27

MAXIDS=Largest total number of individual records (subjects) in a data set used in the run
MAXDREC= Largest number of data records in any one individual record (in any one subject)
TOTDREC=total number of data records (lines) in largest data set to be used.
NPNIND=Larger of MAXIDS and NPSUPP/NPSUPPE given on $NONP record.
NPROB=Total number of problems in the control stream.
LVR=Largest number of etas in any problem (including those listed in $PRIOR)

As of NM73, the values for MAXDREC and TOTDREC are assessed by NMTRAN, and the user may take advantage of NMTRAN's evaluation by using the –maxlim option to the nmfe75 script (see below). But NMTRAN may not always correctly assess these values. Thus, it is best if the user ascertains these values ahead of time by inspection of his largest data set among all of the problems to be used by the control stream file, and the largest number of parameters to be used. Then set the LIM values accordingly via the $SIZES record.

One can alternatively assess empirically whether file buffers are used, by beginning the run, allowing perhaps one iteration to transpire, then from another command window do a directory search for FILE*, (or WK* for worker files in parallelization problems, section I.73 Parallel Computing (NM72)
If any of the FILExx do not have 0 size, then they are being used. Interrupt the analysis, then increase the appropriate LIM value with the $SIZES record, delete the FILE* in case some remain due to a ctrl-C interrupt, rerun the problem, and look again for any non-zero sized FILE* again. Repeat as needed.

By default (-maxlim=0), NMTRAN will set the LIM values to those listed in sizes.f90, or to the minimum required, whichever is larger. As of NM73, if you set –maxlim=1 on the command line, then LIM1, LIM3, LIM4, LIM13, and LIM15 (those used during estimation, and therefore by workers in a parallelization problem), will be set to the size needed to assure no buffer files are used, and everything is stored in memory, for the particular prolem. If you set –maxlim=2,

then LIM1, LIM2, LIM3, LIM4, LIM5, LIM6, LIM7, LIM8, LIM10, LIM11, LIM13, LIM15, and LIM16 are also sized to what is needed to assure that buffer files are not needed.

If you set -maxlim=3, then MAXRECID will also be sized, to MAXDREC, the largest number of records in any individual. MAXRECID sizes arrays involved in storing state variables during partial derivative estimates of sigmas and sigma like thetas, to improve efficiency of the EM and Monte Carlo methods. When setting -maxlim=3, it is preferred to also use –tprdefault, or –prcompile, but not –prdefault, as NMTRAN's optional resizing of the PREDPP size parameter MAXRECID may conflict with the –prdefault option.

To specify only a subset of LIM's to be sized by NMTRAN, set –maxlim to a number list enclosed within parantheses, such as –maxlim=(1,2,3,11-16), which will have NMTRAN find size requirements for LIM1, LIM2, LIM3, LIM11, LIM13, LIM15, and LIM16 (LIM12 and LIM14 are not used). Enclosing the option in quotes "–maxlim=(1,2,3,11-16)" is required for some operating systems. For sizing MAXRECID, use the number 17. Setting maxlim=(1-17) is equivalent to –maxlim=3, whereas –maxlim=(3) means to have NMTRAN size only LIM3.

Description of Buffers

A number of contiguous data records are stored in memory at any one time in buffers. If a large enough memory area can be made available for this purpose, then the entire data set can be stored in memory throughout the NONMEM run, and computing costs can be decreased. The following discussion of NONMEM buffers should not be confused with I/O buffers which are used by the operating system.

The size of buffer 1 is related to the number, LIM1, of data records stored in memory at any one time. A large proportion of data sets will consist of no more than 10000 data records. Consequently, the size of buffer 1 has been set to allow LIM1=10000 data records. The least number of data records allowable must exceed the largest number of data records used with any one subject, which rarely will be as large as 10000. Each data record consists of PD 8 byte double precision computer words, and the allocation of memory for buffer 1 is PD*(LIM1+3)*8 bytes.

Buffer 2 holds a number of contiguous residual records. For each data record, NONMEM generates prediction, residual and weighted residual data items, NPDE, EWRES, etc.; these data items comprise the residual record. The default size of buffer 2 is related to the number, LIM2, of residual records, stored in memory at any one time. The size of buffer 2 has been set to allow LIM2=100,000 residual records, for up to 100,000 data records. The least number of residual records allowable must exceed the largest number of data records used with any one subject. Each residual data record consists of 19 eight byte double precision computer words. The allocation of memory for buffer 2 is 19*(LIM2+3)*8 bytes.

Buffer 3 holds a number of contiguous subject header records for input data. The size of buffer 3 is related to the number, LIM3, of subject header records stored in memory at any one time. The default size of buffer 3 has been set to allow LIM3=1000 subject header records. Each subject header record consists of four 8 byte computer words. The allocation of memory for buffer 3 is 4*(LIM3+1)*8 bytes.

Buffer 4 holds a number of contiguous ETA records. For each subject, NONMEM generates values for ETA variables. The size of buffer 4 is related to the number, LIM4, of ETA records stored in memory at any one time. The size of buffer 4 has been set to allow LIM4=1000 ETA records. Each ETA record consists of MMX*LVR 8 byte double precision computer words. The allocation of memory for buffer 4 is MMX*LVR*(LIM4+3)*8.

Buffer 5 holds a number of contiguous mixture model records. For each subject record, NONMEM generates information about the component models of a mixture model; this information constitutes the mixture model record. The size of buffer 5 is related to the number, LIM5, of mixture model records stored in memory at any one time. The default size of buffer 5 has been set to allow LIM5=200 mixture model records. Each mixture model record consists of five 8 byte single precision computer words. The allocation of memory for buffer 5 is (MMX+1)*(LIM5+3)*8 bytes.

Buffer 6 holds a number of contiguous PRED-defined records. For each data record of a given subject record, NONMEM stores the values found in module NMPRD4; these values comprise the NMPRD4 record. The size of buffer 6 is related to the number, LIM6, of PRED-defined records stored in memory at any one time. The size of buffer 6 has been set to allow LIM6=400 PRED-defined records. The least number of PRED-defined records allowable must exceed the largest number of data records used with any one subject, which rarely will be as large as 400. Each PRED-defined record consists of PDT 8 byte double precision computer words. The allocation of memory for buffer 6 is PDT*(LIM6+3)*8 bytes.

Buffer 7 holds a number of contiguous NMPRD4 records *for a single individual only*. For each problem in a NONMEM run, NONMEM generates information about the problem; this constitutes the problem header record. The size of buffer 7 is related to the number, LIM7, of NMPRD4 records stored in memory at any one time. The size of buffer 7 has been set to allow LIM7=2 NMPRD4 records, which is generally fewer than the number of NMPRD4 records existing for any given subject. Each NMPRD4 record consists of (LIM7+2)*LNP4 8 byte double precision computer words. The default allocation of memory for buffer 7 is 4*LNP4*8 bytes.

The memory allocation of Buffer 8 is (LVR+1)*(LIM8+3) double precision values.
The memory allocation of Buffer 10 is (LIM10+3) double precision values.

Buffer 11 holds a number of contiguous problem header records. The size of buffer 11 is related to the number, LIM11, of problem header records stored in memory at any one time. The size of buffer 11 has been set to allow LIM11=25 problem header records. Each problem header record consists of forty-two 8 byte integer computer words. The allocation of memory for buffer 11 is 42*(LIM11+3)*8= 9408 bytes.

The memory allocation of Buffer 13 is 404*(LIM13+3) double precision values.

After NONMEM VI, there are also buffers 15 and 16. The sizes of these buffers are related

to constants LIM15 and LIM16. These buffers are used in DAT15 and DAT16. If LIM16 is , not adequate, NONMEM will produce error messages such as the following.

```
TOT NO. OF RESIDUAL RECS IN BUFFER 16 IS LESS THAN
NO. OF DATA RECS WITH SOME INDIVIDUAL
```

The memory allocation of Buffer 15 is LCM110*(LIM15+3) double precision values.
The memory allocation of Buffer 16 is MMX*4*(LIM16+3) double precision values.

Buffers 1, 3, 4, 13, and 15 are used during an estimation step. To obtain the fastest analysis, even when the estimation is parallelized, you may want to optimize their LIM sizes.

I.11 Multiple Runs

As of NONMEM 7, there is decreased likelihood of early termination of runs using multiple problems and/or the "Super Problem" feature.

I.12 Improvements in Control Stream File input limits

1. By default, there may be up to 50 data items per data record. In NM72, set PD in $SIZES record to change this.
2. Data labels may be up to 20 characters long
3. Numerical values in the data file may now be up to 24 characters long.
4. ID values in the data file may be up to 14 digits long.
5. The numerical values in $THETA, $OMEGA, and $SIGMA may be each up to 30 characters long, and may be described in E field notation.
6) By default, you may have up to 50 items printed in tables. In NM72, set PDT in $SIZES record to change this.

I.13 Issuing Multiple Estimations within a Single Problem

A sequence of two or more $EST statements within a given problem will result in the sequential execution of separate estimations. This behavior differs from NONMEM VI, where two sequential $EST statements acts as the continuation of defining additional options to a single estimation. For example:

```
$THETA 0.3 0.5 6.0
$OMEGA 0.2 0.2 0.2
$SIGMA 0.2
; First estimation step
$EST METHOD=0 MAXEVAL=9999
     PRINT=5 NSIG=3
; Second estimation step
$EST METHOD=CONDTIONAL
     NSIG=4
```

will first result in estimation of the problem by the first order method, using as initial parameters those defined by the $THETA, $OMEGA, and $SIGMA statements. Next, the first order

conditional estimation method will be implemented, using as initial parameters the final estimates of THETA, OMEGA, and SIGMA from the previous analysis. Up to 20 estimations may be performed within a problem. For all intermediate estimation steps, their final parameter values and objective function will be printed to the raw output file.

Many settings to options specified in a $EST method will by default carry over to the next $EST method, unless a new option setting is specified. Thus, in the example above, PRINT will remain 5 and MAXEVAL will remain 9999 for the second $EST statement, whereas NSIG will be changed to 4 and METHOD becomes conditional. An exception to this rule are NOTHETABOUND, NOOMEGABOUND, and NOSIGMABOUND, in which these options pertain to all of the estimations in the series within a $PROB. In NM710, NM712, and NM720, these options must be given with the very first $EST record in the problem. With NM73, these options may be placed with any of the $EST records, but will still apply to all $EST records in the problem.

The EM and Monte Carlo estimation methods particularly benefit from performing them in sequence for a given problem. Even the classical NONMEM methods can be facilitated using an EM method by first having a rapid EM method such as iterative two stage be performed first, with the resulting parameters being passed on to the FOCE method, to speed up the analysis:

```
$EST METHOD=ITS INTERACTION
$EST METHOD=CONDITIONAL INTERACTION
```

More information on this is described in the Composite Methods section.

I.14 Interactive Control of a NONMEM batch Program

A NONMEM run can now be controlled to some extent from the console by issuing certain control characters.

Console iteration printing on/off during any Estimation analysis (ctrl-J from console NONMEM, Iterations button from PDx-POP).
Exit analysis at any time, which completes its output, and goes on to next mode or estimation method (ctrl-K from console, or Next button in PDx-POP).
Exit program gracefully at any time (ctrl-E or Stop button).
Monitor the progress of each individual during an estimation by toggling ctrl-T. Wait 15 seconds or more to observe a subject's ID, and individual objective function value. It is also good to test that the problem did not hang if a console output had not been observed for a long while.

Control console printing of parallelization information with Ctrl-B. Control log file parallelization information with ctrl-F.

If you run NONMEM from PDx-POP, you can get graphical view of objective function or any model parameter progress during the run. The parameter and objective function progress is

written in a root.ext file (where root is base name of control stream file), which may also be monitored by a text editor during the run.

If you run NONMEM from PDx-POP, Bayesian sample histories of the population parameters can be viewed after analysis is done. The sample history file is written to that specified by the $EST FILE= option, which can be also monitored by a text editor during or after the run.

Sometimes NONMEM does not respond to user input. This may occur during a parallel distribution run using MPI, or if the user began NONMEM with the –background switch. The user may open another console window, copy the program sig.exe from the NONMEM installed ..\util directory to your run directory, then enter any one of these commands:

Print toggle (monitor estimation progress):
Sig J
Sig R
Sig P

Console paraprint toggle (monitor parallel processing traffic):
Sig B
Sig A
Sig P
Sig PA
Sig PP

Next (move on to next estimation mode or next estimation):
sig K
sig N

Stop (end the present run cleanly):
Sig E
Sig S

Subject print toggle:
sig T
sig U
sig SU

Parallelization log-file parafprint toggle:
Sig PARAF
Sig PF

Alternatively, you may execute the sig program from another directory if you specify the run directory in which you want the signal file created:

sig next \nonmem\run\

Make sure you terminate the directory name with a directory parse symbol appropriate for the operating system.

I.15 $COV: Unconditional Evaluation

The $COV step can be performed unconditionally even when an estimation terminates abnormally, by specifying:

```
$COV UNCONDITIONAL
```

I.16 $TABLE: Additional Statistical Diagnostics, Associated Parameters, and Output Format

Requesting a Range of Etas to be Outputted: Etas(x:y) (NM73)

Instead of requesting each ETA specifically in a $TABLE item list, a range of etas may be requested:

ETAS(2:4)
is equivalent to requesting ETA2, ETA3, and ETA4.

ETAS(5)
or
ETAS(5:LAST)
is equivalent to requesting ETA(5), ETA(6), … to ETA(NETAS).

As of nm74, if NPOPETAS is specified on a $MSFI record, then the value of NPOPETAS is used as LAST.

As of NM74, more flexible syntax is available:
The word TO may be used in place of the semi-colon :
The BY expression may be used:
ETAS(1 TO 10 by 3) prints out etas 1,4,7,10
ETAS(LAST TO 1 by -3) prints out etas 10,7,4,1 (assuming LAST=10)
A number list may be given:
ETAS(1,5,12,4) prints out etas 1, 5, 12, 4.
ETAS(4:1) prints etas 4, 3, 2, 1
ETAS(4:1 by -2) prints etas 4, 2
ETAS(1:4 by -1) prints etas 4, 3, 2, 1 (the by value sets the direction).

The $SCAT will also interpret this syntax, for example,

$SCAT ETAS(1:2) VS ETA3
is equivalent to
$SCAT ETA1 ETA2 VS ETA3

However, unlike $TABLE, $SCAT will ignore implied endings, such as

$SCAT ETAS(1:LAST) VS ETA3

And just interpret it as

$SCAT ETA1 VS ETA3

Excluding Records from Being Outputted: The EXCLUDE_BY option (NM74)

A data item or defined variable may be identified on a $TABLE record as an EXCLUDE_BY variable, which if not 0, will exclude the record. For example:

```
$PK
...
EXCL=0
IF(ID.GE.45.AND.ID.LE.53) EXCL=1
...
$TABLE ID TIME DV IPRED CL V1 Q V2 ETAS(1:LAST) EXCLUDE_BY EXCL NOAPPEND FILE=exctable.par
NOPRINT
```

The table exctable.par will not list records from subjects 45 to 53. If more than one exclusion variable is listed, then if any of these have a non-zero value, the record will be excluded.

To further assist in selecting records, the following variables are available, against which you may compare with NDREC, the present record number for an individual:

FIRSTREC: first record of subject
LASTREC: last record of subject
FIRSTOBS: first observation record of subject (for which MDV=0 or 100)
LASTOBS: last observation record of subject (for which MDV=0 or 100)
FIRSTDOS: First record of subject with EVID=1 or EVID=4. FIRSTDOS=-1 when there are no dose records, or PREDPP is not used.
LASTDOS: Last record of subject with EVID=1 or EVID=4. LASTDOS=-1 when there are no dose records, or PREDPP is not used.
EFIRSTREC: first record of subject during estimation (so, among records for which MDV=0 or MDV=1)
ELASTREC: last record of subject during estimation (so, among records for which MDV=0 or MDV=1)
EFIRSTOBS: first observation record of subject during estimation (so, among records for which MDV=0)
ELASTOBS: last observation record of subject during estimation (so, among records for which MDV=0)
EFIRSTDOS: First record of subject with EVID=1 or EVID=4 during estimation (so, among records for which MDV=1). EFIRSTDOS=-1 if no dose record PREDPP is not used.
ELASTDOS: Last record of subject with EVID=1 or EVID=4 during estimation (so, among records for which MDV=1). ELASTDOS=-1 if no dose record PREDPP is not used.
IRECIDX: IRECIDX+1 is starting absolute record number in the data set for the present subject (so while NDREC always sets to 1 for the first record of each subject, IRECIDX accumulates

record number positions from previous subjects. For subject 1, IRECIDX+1=1, for subject 2, IRECIDX+1=LASTREC of previous subject +1).

During ICALL=1 (INFN initialization), ICALL=3 (INFN finalization), or ICALL=4 (simulation) only FIRSTREC and LASTREC are available. The other values will be set to -1.

To refer to these variables in $PK, for example, insert include the nonmem_reserved_general file statement, which contains their location information:

```
$PK
include nonmem_reserved_general
…
EXCL=1
IF(NDREC==LASTOBS) EXCL=0
IF(NDREC==FIRSTOBS) EXCL=0
…

$TABLE ID TIME IPRED EXCLUDE_BY EXCL NOPRINT NOAPPEND file=mytable.TAB
```

Then, only the record containing first and last obervations will be printed to the table.

Selecting LASTONLY and FIRSTLASTONLY Records (NM74)

In addition to FIRSTONLY, you may select LASTONLY to just report the last record of each subject, or FIRSTLASTONLY, to report just first and last records. The FIRSTONLY, LASTONLY, and FIRSTLASTONLY options are mutually exclusive, and may not be used in combination.

Requesting Standard Errors to User-Defined and PREDPP Variables (NM74)

With the variance-covariance of the etas, and evaluation of the partial derivative of the particular table item with respect to eta by central finite difference method, and by principle of propagation of errors, the standard errors of the table items are evaluated. To report standard errors associated with etas (individual variables) in the tables for user defined variables, set

$TABLE …VARCALC=1.

See setest.ctl in the examples directory. If using RFORMAT formatting system, make sure to allow enough format fields to include reported standard errors. In addition, full variances-covariances among all user-defined and PREDPP variables will be outputted to root.vpd (the FORMAT used for this file is that defined in the $EST statement). To only print to the vpd file, and not report SE's to the table, set VARCALC=2. The variances are produced by using the variance-covariance of the etas/phis reported in the root.phi file, centered about the values reported in the table files. For classical NONMEM methods, this means variables are centered at the eta-mode assessed variable values, and approximate variance-covariance from the MAP estimation step. For EM methods:

FNLETA=1,3: variance-covariance of etas from last iteration of last estimation method is used, and centered about FNLETA basedevaluations of variables (MAP, eta-mode).

FNLETA=0: variance-covariance of etas from last iteration of last estimation method is used, and centered about method based evaluations of variables (conditional means for EM methods, conditional modes for classical and ITS methods, MCMC posterior means for BAYES).

FNLETA=2: variance-covariance, and etas used for assessing variables are obtained from external source.

As of nm75, table root.vpt is also created, which incorporates variance-covariances associated with thetas (or omegas and sigmas, listed in the .cov file) as well as those associated with etas/individual variables (phc() listed in the .phi file). To present the comparable total standard errors in a user defined table, set $TABLE …VARCALC=3. The standard errors in the .vpt table are most accurately assessed if the problem is MU modeled. If it is not MU modeled, then only standard errors of variables that do not vary with etas (variables derived exclusively from thetas, for example), are assured to be correctly calculated. It also takes into account that the .phi tables contain variances to phi's and etas under fixed theta conditions for all methods except BAYES, where the variances to phis or etas are sampled under varying theta conditions.

Note that to obtain the PRED and its error, you should define a variable that is based on IPRED, but evaluated only when COMACT=1 (NONMEM passes through the model system with ETA=0):

```
IPRED=F
If(COMACT==1) PREDU=IPRED
```

So, PREDU is IPRED evaluated with ETA=0. Then, list this in a table:

```
$TABLE …  PREDU … VARCALC=3
```

Keep in mind that PREDU has only stastical errors associated with THETAS (and SIGMAS, depending on how PREDU is evaluated, based on the variance-covariance in the .cov file), and not etas (the phc() variances in the .phi file do not contribute to PREDU, since etas=0 when COMACT=1), so VARCALC=3 is required. No statistical errors are evaluated for weighted residual diagnostics (WRES, NPD's, etc).

The vpd and vpt files contain entries for each COMACT the NONMEM used to calculate the table items. For the vpd file, variance-covariance among user-defined variables and PREDPP variables are placed in this file, associated with individual variances of the .phi file. The COMACT column value is usually 2, or 3 if $NONPARAMETRIC ETAS was implemented. The subsequent columns contain the values of user-defined table items, followed by variance-covariance of these items (labled VAR1_VAR2, wher VAR1 and VAR2 are the particular table items).

For the vpt file, the total variance-covariance among user-defined variables and PREDPP variables are placed in this file, associated with individual variances of the .phi file, as well as the variance-covariance of the fixed effects variables (THETAS, SIGMAS, OMEGAS) of the .cov

file. The COMACT column value is 1 to indicate the record contains the values during a COMACT=1 pass, that is with ETAs set to 0. Thus, an individual predicted value (IPRED) on this record would in fact by the population predicted value (PRED). The subsequent columns contain the values of user-defined table items, followed by variance-covariance of these items (labled VAR1_VAR2, wher VAR1 and VAR2 are the particular table items). The variance IPRED_IPRED for a record with COMACT=1 would then have essentially the statistical error associated with the variance-covariance of the appropriate thetas. If you are satisfied to pick up the PRED values and their errors from the vpt table, then it is not necessary to define a COMACT=1 evaluated variable in the control stream, as you can get it from IPRED on the COMACT=1 rows in the .vpt table. But if you want the appropriate PRED variable with standard error presented in the user-requested $TABLE table, then the PREDU must be defined as above. For non-parametric estimated etas, when $NONPARAMETRIC ETAS is used, then a COMACT 3 pass will be made, and their values will be presented in the vpt table with the COMACT set to 3. See Guide VIII for more on COMACT.

See the example ..\examples\predu.ctl, which prints out IPRED, PRED, and PREDU in the various tables (predu.tab, predu.phi, predu.cov, predu.vpt, predu.vpd), and you can compare their values and associated standard errors.

New diagnostic items

Additional types of pred, res, and wres values may be requested than the usual set available in NONMEM VI. They may be specified at any $TABLE command or $SCATTER command, as one would request PRED, RES, or WRES items. If $TABLE statements succeed multiple $EST statements within a run, the table results (as well as scatter plots if requested via $SCATTER) will pertain to the last analysis.

OBJI

These are objective function values for each individual. The sum of the individual objective function values is equal to the total objective function.

NPRED, NRES, NWRES

These are non-conditional, no eta-epsilon interaction, pred, res, and wres values. These are identical to those issued by NONMEM V as PRED, RES, and WRES.

PREDI, RESI, WRESI

These are non-conditional, with eta-epsilon interaction, pred, res, and wres values. These are identical to those issued by NONMEM VI as PRED, RES, and WRES. The WRESI will not differ from NWRES if INTERACTION was not selected in the previous $EST command.

CPRED, CRES, CWRES

These are conditional, no eta-epsilon interaction, pred, res, and wres values as described in [1].

The conditional mode etas (from FOCE or ITS, also known as conditional parametric etas (CPE), empirical bayes estimates (EBE), posthoc estimates of etas, or mode a posteriori (MAP) estimates) or conditional mean etas (from Monte Carlo EM methods) will be referred to as $\hat{\boldsymbol{\eta}}$ (eta hat), must be available from a previous \$EST MAXEVAL>0 command. The conditional weighted residuals are estimated based on a linear Taylor series approximation that is extrapolated from the conditional mean or mode (or posthoc) eta estimates, rather than about eta=0:

$$CPRED_{ij} = f_{ij}(\hat{\boldsymbol{\eta}}) - \mathbf{g}'_{ij}(\hat{\boldsymbol{\eta}})\hat{\boldsymbol{\eta}}$$

using the nomenclature of Guide I, Section E2. Then

$$CRES_{ij} = y_{ij} - CPRED_{ij}$$

The population variance covariance of observed data described in Guide I, E.2 is also evaluated at eta_hat: $C_i(\hat{\boldsymbol{\eta}})$:

$$\mathbf{CWRES}_i = \mathbf{C}(\hat{\boldsymbol{\eta}})_i^{-1/2}(\mathbf{y}_i - \mathbf{CPRED}_i(\hat{\boldsymbol{\eta}}))$$

Because of the linear back extrapolation, it is possible for some CPRED values to be negative. Users may prefer to request NPRED CRES CWRES, or NPRED RES CWRES. The conditional weighted residual will not differ from the non-conditional weighted residual if FO was selected in the previous \$EST command.

In NM72, if \$EST INTERACTION was not specified prior to requesting \$TABLE CWRES, then the population variance-covariance is evaluated at eta=0: $C_i(\boldsymbol{\eta} = 0)$. In NONMEM 7.1.0 and 7.1.2, regardless of INTERACTION setting in a previous \$EST statement, $C_i(\hat{\boldsymbol{\eta}})$ is used.

CPREDI, CRESI, CWRESI

These are conditional, with eta-epsilon interaction, pred, res, and wres values. The conditional mode or conditional mean etas must be available from a previous \$EST MAXEVAL>0 command.

EPRED, ERES, EWRES

The EPRED, ERES, EWRES are Monte-Carlo generated (expected, or exact) pred, res, and wres values, and are not linearized approximations like the other diagnostic types.

The expected diagnostic items are evaluated using predicted function and residual variances evaluated over a Monte Carlo sampled range of etas with population variance Omega, and assuming asymptotic values for averaging over the *y* domain. Define

$$EPRED_{ij} = \int_{-\infty}^{\infty} f_{ij}(\boldsymbol{\eta})\, p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

is the expected predicted value for data point *j* of subject *i* for a given subject, evaluated by Monte Carlo sampling, overall possible eta. The probability density of eta:

$$p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

is a multivariate normal distribution with eta variance $\boldsymbol{\Omega}$. The 1xn_i vector of EPRED for a given subject, where n_i is the number of data points to that subject, is then:

$$\mathbf{EPRED}_i = \int_{-\infty}^{\infty} \mathbf{f}_i(\boldsymbol{\eta}) p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

Then the corresponding residual vector for observed values $\mathbf{y}_i$ is

$\mathbf{ERES}i=\mathbf{y}_i-\mathbf{EPRED}_i$

The residual (epsilon) variance matrix using the nomenclature in Guide I, Sections E.2 may be

$$\mathbf{V}_i(\boldsymbol{\eta}) = diag(\mathbf{h}_i(\boldsymbol{\eta})\boldsymbol{\Sigma}\mathbf{h}_i(\boldsymbol{\eta}))$$

or it may be the more complicated form described in section of E.4 in the case of L2 data items. Then, the expected residual (epsilon) variance (assessed by Monte Carlo sampling) is

$$\mathbf{EV}_i = \int_{-\infty}^{\infty} \mathbf{V}_i(\boldsymbol{\eta}) p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

The full variance-covariance matrix of size n_ixn_i, that includes residual error (epsilon) and inter-subject (eta) variance contributions is:

$$\mathbf{EC}_i = \mathbf{EV}_i + \int_{-\infty}^{\infty} (\mathbf{f}_i(\boldsymbol{\eta}) - \mathbf{EPRED}_i)(\mathbf{f}_i(\boldsymbol{\eta}) - \mathbf{EPRED}_i)' p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

And is the expected population variance, Monte Carlo averaged over all possible eta. Then, following the Guide I, section E nomenclature, the population weighted residual vector for subject *i* is:

$$\mathbf{EWRES}_i = \mathbf{EC}_i^{-1/2}\mathbf{ERES}_i$$

where the square root of a matrix is defined here by default as evaluated by diagonalizing the matrix, and multiplying its eigenvector matrices by the square roots of the eigenvalues. Selecting the WRESCHOL option obtains the square root of the matrix by Cholesky decomposition.

ECWRES

ECWRES is a Monte Carlo assessed expected weighted residual evaluated with only the predicted function evaluated over a Monte Carlo sampled range of etas with population variance Omega, while residual variance **V** is always evaluated at conditional mode (from the most recent FOCE/ITS estimation) or conditional mean (from the most recent IMP/IMPMAP/SAEM analysis) eta ($\hat{\boldsymbol{\eta}}$), so that

$$\mathbf{ECC}_i = \mathbf{V}_i(\hat{\boldsymbol{\eta}}) + \int_{-\infty}^{\infty} (\mathbf{f}_i(\boldsymbol{\eta}) - \mathbf{EPRED}_i)(\mathbf{f}_i(\boldsymbol{\eta}) - \mathbf{EPRED}_i)' p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

and

$$\mathbf{ECWRES}_i = \mathbf{ECC}_i^{-1/2}\mathbf{ERES}_i$$

As with CWRES, the eta_hat (conditional mode or mean) values must be available from a previous $EST MAXEVAL>0 command.

Thus, ECWRES is the Monte Carlo version of CWRES, while EWRES is the Monte Carlo version of CWRESI.

In NM72, if $EST INTERACTION was not specified prior to requesting $TABLE CWRES, then the residual variance is evaluated at eta=0: $\mathbf{V}_i(\boldsymbol{\eta} = 0)$. In NONMEM 7.1.0 and 7.1.2, regardless of INTERACTION setting in a previous $EST statement, $\mathbf{V}_i(\hat{\boldsymbol{\eta}})$ is used.

NPDE

The NPDE is the normalized prediction distribution error (reference [2]: takes into account within-subject correlations), also a Monte Carlo assessed diagnostic item. For each vector of simulated data $\mathbf{y}_{ki}$:

$\mathbf{ESRES}_{ki}=\mathbf{y}_{ki}-\mathbf{EPRED}_i$

its decorrelated residual vector is calculated:

$$\mathbf{ESWRES}_{ki} = \mathbf{EC}_i^{-1/2}\mathbf{ESRES}_{ki}$$

and compared against the decorrelated residual vector of observed values $\mathbf{EWRES}_i$ such that

$$\mathbf{pde}_i = \frac{1}{K}\sum_{k=1}^{K}\delta(\mathbf{EWRES}_i - \mathbf{ESWRES}_{ki})$$

For K random samples, where

$\delta(x) = 1 \; for \; x >= 0$

$= 0 \; for \; x < 0$

For each element in the vector. Then, an inverse normal distribution transformation is performed:

$$\mathbf{npde}_i = \Phi^{-1}(\mathbf{pde}_i)$$

NPD

The NPD is the correlated normalized prediction distribution error (reference [3]: does not take into account within-subject correlations), also a Monte Carlo assessed diagnostic item. For each vector of observed data $\mathbf{y}_i$ of subject i, and vector of simulated etas $\boldsymbol{\eta}_k$:

$\mathbf{IWRES}_{ki}= \mathbf{V}(\boldsymbol{\eta}_k)_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\boldsymbol{\eta}_k))$

These are then averaged over all the random samples;

$$\mathbf{pd}_i = \frac{1}{K}\sum_{k=1}^{K}\Phi(\mathbf{IWRES}_{ki})$$

Then, an inverse normal distribution transformation is performed:

$$\mathbf{npd}_i = \Phi^{-1}(\mathbf{pd}_i)$$

The default PRED, RES, and WRES will be given the same values as PREDI, RESI, and WRESI, when INTERACTION in $EST is specified, or NPRED, NRES, and NWRES when INTERACTION in $EST is not specified.

As the PRED, RES, and WRES, may be referenced in a user-supplied $INFN routine, or in $PK or $PRED (when ICALL=3) as PRED_, RES_, WRES_, so the additional variables may be referenced by their names followed by _ (for example EWRES_).

NPDTYPE=0(default)

If NPDTYPE=1 is set as a $TABLE option, then the strict stochastic (Monte Carlo) method over the data y domain as well as etas, as referenced in [2] and [4], is implemented for NPD diagnostics:

$$\mathbf{pd}_i = \frac{1}{K}\sum_{k=1}^{K}\delta(\mathbf{y}_{ki} - \mathbf{y}_i)$$

$$\delta(x) = 1 \; for \; x >= 0$$

$$= 0 \; for \; x < 0$$

Where $\mathbf{y}_{ik}$ is the kth simulated vector of data, and $\mathbf{y}_i$ is the observed data. Similarly for npde data:

$$\mathbf{EPRED}_i = \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}_{ki}$$

$$\mathbf{ERES}_i = \mathbf{y}_i - \mathbf{EPRED}_i$$

$$\mathbf{ESRES}_{ki} = \mathbf{y}_{ki} - \mathbf{EPRED}_i$$

$$\mathbf{EV}_i = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{y}_{ik} - \mathbf{EPRED}_i)(\mathbf{y}_{ik} - \mathbf{EPRED}_i)'$$

$$\mathbf{EWRES}_i = \mathbf{EV}_i^{-1/2}\mathbf{ERES}_i$$

$$\mathbf{ESWRES}_i = \mathbf{EV}_{ki}^{=1/2}\mathbf{ESRES}_i$$

$$\mathbf{pde}_i = \frac{1}{K}\sum_{k=1}^{K}\delta(\mathbf{EWRES}_i - \mathbf{EWRES}_{ik})$$

INTERPTYPE=0(default) (nm751)

By default the interpolation for NPD/NPDE evaluations type is 0, that is, the nearest value in the quantile area is chosen, without interpolation. If INTERPTYPE=1, then an interpolation is performed using an inverseNORM linearized interpolator. In addition, for NPDE it will extrapolate quantile positions that are outside the Monte Carlo sampled range, allowing severe outliers to be less truncated. For example, by default, if the EXAMPLE=300, and a quantile position is 0.001, then the 1/300 value is returned, the lowest value from 300 samples. With INTERPTYPE=1, an extrapolation is performed to better represent the 0.001 value, and will be less than the lowest value among 300 samples simulated. The INTERPTYPE has less impact with increasing ESAMPLE.

CIWRES, CIPRED,CIRES, CIWRESI (NM73)

The CIWRES is the conditional individual weighted residual as evaluated during the estimation, equivalent to (DV-F)/(F*SQRT(SIGMA(1,1))) for simple problems with proportional residual error. With L2 data or CORRL2 data, the individual weighted residuals are in their decorrelated forms:

$$\mathbf{CIWRES}_i = \mathbf{V}(\hat{\boldsymbol{\eta}})_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\hat{\boldsymbol{\eta}}))$$

when INTERACTION in the previous $EST record is set, and a conditional analysis (non-FO) was performed. For individual *i*, where individual residual variance matrix $\mathbf{V}_i$ and individual vector of predicted $\mathbf{f}_i(\hat{\boldsymbol{\eta}})$ are evaluated at the conditional mode or mean eta (designated as eta hat). The square root of the matrix $\mathbf{V}_i$ may be evaluated by using the square root of the eigenvalues, or by Cholesky decomposition when WRESCHOL option is used (see below). Similarly, the CIPRED is the individual predicted value $\mathbf{f}_i(\hat{\boldsymbol{\eta}})$ at the conditional mode or mean eta, and CIRES=DV-$\mathbf{f}_i(\hat{\boldsymbol{\eta}})$.

When INTERACTION is not set, then

$\mathbf{CIWRES}_i = \mathbf{V}(\boldsymbol{\eta}=0)_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\hat{\boldsymbol{\eta}}))$

is evaluated, that is, the variance portion is evaluated using $\mathbf{f}_i(\boldsymbol{\eta}=0)$. However CIWRESI (conditional individual weighted residual with interaction) is always evaluated as (except for FO, see below)

$\mathbf{CIWRESI}_i = \mathbf{V}(\hat{\boldsymbol{\eta}})_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\hat{\boldsymbol{\eta}}))$

regardless of the INTERACTION setting.

For FO, the conditional individual weighted residual will not differ from the non-conditional weighted residual. That is, for FO, the CIWRES and CIPRED are evaluated using F(eta=0) for numerator and denominator terms, since this is what is done during estimation, and no EBE (eta-hat) is evaluated:

$\mathbf{CIWRES}_i = \mathbf{V}(\boldsymbol{\eta}=0)_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\boldsymbol{\eta}=0)) = \mathbf{CIWRESI}_i$

Even for FO with interaction, the predicted function (numerator) and residual variance (denominator) is still evaluated at eta=0, so CIWRESI=CIWRES. The interaction contribution is accounted for with additional first-order Taylor terms to make a linear projection of the contribution of eta-eps interaction. While it would be inappropriate to add these Taylor terms to CIWRESI, these Taylor terms *are* added to the population residual assessment WRESI, hence WRESI will differ from NWRESI with FO INTERACTION.

There are other individual residual values available, mostly as place holders in the system, but these have no additional statistical value. They are:

NIPRED=IPREDI=IPRD=NPRED
CIPREDI=CIPRED
EIPRED=EPRED

NIRES=IRESI=NRES=IRS
CIRESI=CIRES
EIRES=ERES

$\mathbf{NIWRES}_i = \mathbf{V}(\boldsymbol{\eta}=0)_i^{-1/2}(\mathbf{y}_i - \mathbf{f}_i(\boldsymbol{\eta}=0))$
IWRESI=NIWRES=IWRS

$$\mathbf{EIWRES}_i = \int_{-\infty}^{+\infty} \mathbf{V}(\boldsymbol{\eta})_i^{-1/2} (\mathbf{y}_i - \mathbf{f}_i(\boldsymbol{\eta})) \, p(\boldsymbol{\eta} \mid 0, \boldsymbol{\Omega}) d\boldsymbol{\eta}$$

MDVRES=0 (NM73) (default)

Set MDVRES to 1 in the $ERROR or $PRED routine if you do not want to include a particular value for weighted residual assessment. This may be useful when, for example, this data point is assessed by a non-normal distribution likelihood such as the PHI() function for below detection limit values, in which F_FLAG is set. By default, if at least one data value of a given subject is fitted with a non-normal distribution likelihood, then population weighted residual diagnostics are not assessed for any of the data for that subject. By setting MDVRES=1 to these particular below detection values, the weighted residual algorithm can assess the remaining normally distributed values for that subject. For example,

```
$ERROR
SD = THETA(5)
IPRED = LOG(F)
DUM = (LOQ - IPRED) / SD
CUMD = PHI(DUM)
IF (TYPE .EQ. 1) THEN
      F_FLAG = 0
      Y = IPRED + SD * ERR(1)
ENDIF
IF (TYPE .EQ. 2) THEN
      F_FLAG = 1
      Y = CUMD
      MDVRES=1
ENDIF
```

MDVRES stands for missing data value (MDV) for residual (RES) assessment. Setting MDVRES to 1 is equivalent to temporarily declaring that data point as missing during the weighted residual assessments.

Handling Level of Quantitation (LOQ) and Level Above Quantitation (LAQ) Data, and their Incorporation in to NPDE and NPD Assessments (nm73, nm74)

To incorporate LOQ data into NPDE assessments [4], use the following method (as an example):

Here, TYPE and LOQ are user-defined in previous code, or data item (..\examples\loq\ad3tr4_loq0).

```
$ERROR
SD = THETA(5)
IPRED = LOG(F)
DUM = (LOQ - IPRED) / SD
CUMD = PHI(DUM)
IF (TYPE .EQ. 1.OR.NPDE_MODE.EQ.1) THEN
      F_FLAG = 0
      Y = IPRED + SD * ERR(1)
ENDIF
IF (TYPE .EQ. 2.AND.NPDE_MODE.EQ.0) THEN
      F_FLAG = 1
      Y = CUMD
```

```
      MDVRES=1
ENDIF
IF(TYPE.EQ.2) DV_LOQ=LOQ
```

By default, DV_LOQ is set to -1.0d+300 by the NONMEM routine that calls ERROR/PRED. If the user's ERROR/PRED sets DV_LOQ to some other value and NPDE_MODE=1, then the NPDE is being evaluated during that time, and this censored value is to be treated as if it is a non-censored datum with value of LOQ (DV_LOQ=LOQ), in accordance with [4], utilizing a standard F_FLAG=0 definition for Y. Note that during estimation of the objective function (when NPDE_MODE=0), NPDE is not being evaluated, and censored values should be treated using F_FLAG=1, and Y must be defined as the integral of the normal density from –inf to LOQ.

New in nm743, you can specify an above quantifiable limit with the reserved variable DV_LAQ as well. For example, (..\examples\loq\ad3tr4a_loq0):

```
$ERROR
SD = THETA(5)
IPRED = LOG(F)
DUM = (LOQ - IPRED) / SD ; LOQ=lower level of quantifiable level
CUMD = PHI(DUM)+1.0E-10
DUMA = (LAQ - IPRED) / SD ; LAQ=Upper (above) quantifiable level
CUMDA = PHI(DUMA)-1.0E-10
IF(TYPE.EQ.2) DV_LOQ=LOQ ; Type=2 is concentration below LOQ
IF(TYPE.EQ.3) DV_LAQ=LAQ ; Type=3 is concentration above LAQ
IF (TYPE .EQ. 1.OR.NPDE_MODE==1) THEN
      F_FLAG = 0
      Y = IPRED + SD * ERR(1)
ENDIF
IF (TYPE .EQ. 2.AND.NPDE_MODE==0) THEN
      F_FLAG = 1
      Y = CUMD
      MDVRES=1
ENDIF
IF (TYPE .EQ. 3.AND.NPDE_MODE==0) THEN
      F_FLAG = 1
      Y = (1.0-CUMDA)
      MDVRES=1
ENDIF
```

New in nm74, for use with NPD, the user may supply the cumulative distribution function using the reserved variable CDF_L. For example, in a general likelihood modeled problem, essentially the previous example, but the Y values of all data are returned in their -2LL form (..\examples\loq\ad3tr4_loq6):

```
$ERROR
SD = THETA(5)
IPRED = LOG(F)
DUM2 = (DV - IPRED) / SD
DUM = (LOQ - IPRED) / SD
CUMD = PHI(DUM)+1.0E-30
CUMD2 = PHI(DUM2)+1.0E-30
IF(TYPE.EQ.1) THEN
Y=2.0*LOG(SD)+DUM2*DUM2
```

```
CDF_L=CUMD2
ENDIF
IF(TYPE.EQ.2) THEN
Y = -2.0*LOG(CUMD)
CDF_L=CUMD
DV_LOQ=LOQ
ENDIF
…
$EST METHOD=COND LAPLACE -2LL MAXEVAL=9999 NSIG=3 SIGL=9 SIGLO=9 PRINT=5
NOABORT MCETA=5
$TABLE ID TIME DV IPRED NPD NOAPPEND ONEHEADER ESAMPLE=1000
 FILE=ad3tr4_loq6.TAB NOPRINT
```

Note that only NPD can be evaluated without consideration of EWRES and EPRED constructs. NPDE, EWRES and EPRED cannot be evaluated for general non-normal likelihood data, or when normal data are modeled in a general log-likelihood manner, rather than assumed normal in the usual way. The CDF reserved variable associated with above quantitation level DV_LAQ is CDF_LA.

The DV_LOQ/CDF_L and DV_LAQ/CDF_LA reserved variables may be also used for evaluating NPD diagnostics for categorical data. In such cases, the DV_LAQ/CDF_LA variables must define the lower bound of a caterorical level, while the DV_LOQ/CDF_L defines the upper boundary of a level, which is counter-intuitive, and its use is demonstrated as follows. In the following example, some data are normally distributed, and others are binomial (categorical). The NPDE will be evaluated only for those that are normal, while NPD are evaluated for both types of data. The CDF_L indicates to the diagnostics routine that this datum is to be treated as non-normal, with a cumulative distribution value of CDF_L, which it can use for evaluating the NPD. Because the probability is categorical, the lower bound CDF_LA needs also to be given, to map the probability of having data value DV (=0 or 1) be between CDF_LA and CDF_L for a random uniform variable (..\examples\loq\example10lcdf).

```
$ERROR
EXCL2=1.0-TYPE
EXCL=TYPE
EXCL3=0.0
IF(EVID/=0) EXCL=1.0
IF(EVID/=0) EXCL2=1.0
IF(EVID/=0) EXCL3=1.0
    EXPP=THETA(4)+F*THETA(5)
IPRED=F
; Use protected exponent PEXP, to avoid numerical overflow
A=PEXP(EXPP)
B=1.0+A
IF (TYPE.EQ.0.OR.NPDE_MODE==1) THEN
; PK Data
    F_FLAG=0
    Y=F+F*ERR(1) ; a prediction
 ELSE
; Categorical data
    F_FLAG=1
    Y=DV*A/B+(1.0-DV)/B      ; a likelihood
    MDVRES=1
 ENDIF
```

```
IF(TYPE==1)  THEN
CDF_L=(1.0-DV)*1.0/B + DV
CDF_LA=DV*1.0/B
DV_LOQ=DV
DV_LAQ=DV-1.0
ENDIF
```

For DV=0 the lower bound of its value is DV_LAQ=-1 with a cumulative probability of CDF_LA=0 (that is, it is impossible for DV to have a value of -1), and upper bound DV_LOQ=0 has probability of CDF_L of 1/B (that is, the probability of it being 0 is 1/B, which is consistent with how the likelihood Y is defined). When DV=1, then range of its value is from lower bound DV_LAQ=0 with cumulative probability CDF_LA=1/B (which is consistent with the upper bound interpretation for DV=0), to upper bound DV_LOQ=1 with cumulative probability CDF_L= of 1 (consistent with the idea that the highest DV can be is 1).

When tabulated, we have the following:

DV	Lower Bound (DV_LAQ)	Upper bound (DV_LOQ)	Lower Cumulative (CDF_LA)	Upper Cumulative (CDF_LOQ)
0	-1	0	0	1/B
1	0	1	1/B	1

The common thread of logic is to direct the NPD algorithm to create random positioned NPD values for any data that are below DV_LOQ, and or above DV_LAQ. For continuous data, values are accurately known as the observed value DV between DV_LOQ and DV_LAQ, but are not known below DV_LOQ, or above DV_LAQ, and so, the NPD should create a scrambled DV value whenever it is below DV_LOQ and/or above DV_LAQ. Categorical data have discrete values, so in order to turn them into continuous, normally distributed equivalents for NPD, scrambled values must be created, for values above DV_LAQ, and/or below DV_LOQ, that is, for all legitmate discrete values. Thus, when DV_LOQ<DV_LAQ, the DV_LOQ and DV_LAQ serve as the boundaries for the outer distribution region for scrambling, and when DV_LAQ<DV_LOQ, the DV_LAQ and DV_LOQ serve as boundaries for the inner distribution region for scrambling.

ESAMPLE=300

Number of random samples to be used to generate a Monte-Carlo based set of EPRED, ERES, ECWRES, NPDE (via EC), and EWRES. ESAMPLE should be specified only on the first $TABLE command. By default, ESAMPLE=300.

The ESAMPLE size should be at least the size of the largest number of individual observations (N) to be encountered among the subjects. It would be good to have ESAMPLE>=N+300, for example. This is because N equals the dimension size of the empirical random variance components (EC, EWRES) used for decorrelating, which will have a rank of min(ESAMPLE,N):

WRESCHOL (NM73)

Normally, population and individual weighted residuals are evaluated by square root of the eigenvalues of the population or individual residual variance. However, an alternative method is to Cholesky decompose the residual variance (suggested by France Mentre, personal communication), by entering the WRESCHOL option. This should be specified only on the first $TABLE command. The Cholesky form has the property of sequentially decorrelating each additional data point in the order of the data set.

SEED

Specify starting seed for Monte Carlo evaluations of EPRED, ERES, EWRES, ECWRES, and NPDE. The default seed is 11456. SEED should be specified only on the first $TABLE command.

CLOCKSEED=0 (default) (nm75)

As of nm75, the actual starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if CLOCKSEED=1. This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication.

RANMETHOD=[n|S|m|P] (default n=3) (NM72)

By default, the random number generator used for Monte Carlo simulations of weighted residual items is ran3 of reference [5]. We feel this is the best random number generator for many purposes. However, you may choose alternative random number generators as follows:
0: ran0 of reference [5], minimal standard generator
1: ran1 of reference [5], Bays and Durham.
2: ran2 of reference [5].
3: ran3 of reference [5], Knuth.
4: NONMEM's traditional random number generator, used as default in $SIMULATION

RANMETHOD should be specified only on the first $TABLE command. The RANMETHOD set in the $TABLE command does not propagate to $EST or $CHAIN.

As of NM73, the Sobol sequences with scrambling may be requested:

RANMETHOD=[n|S|m|P]

where n is the random number generator type, S is Sobol sequence, and m is the Sobol scrambler, and P may be specified to retain random number sequences for each subject, so that the random number sequence is retained regardless of single or parallel processing. See the description of RANMETHOD under I.36 Monte Carlo Importance Sampling EM.

Among the Sobol sequence methods, the S2 method appears to provide the least biased random samples, that is nearly uniform distribution, with good mixing in multi-dimensional spaces.

NOLABEL (NM73)
Do not print column labels. It may be combined with ONEHEADER to print only the title at the beginning of each table.

NOTITLE (NM73)
Do not print table titles. It may be combined with ONEHEADER to print only the column labels at the beginning of each table. NOLABEL NOTITLE is equivalent to NOHEADER.

ONEHEADERALL or ONEHEADERPERFILE (NM74)
Print header information only once per file. If a file already exists when the problem was started, and the FORWARD option was used, results are appended. In such cases, NONMEM assumes that no additional headers should be added to this file.

FORMAT=s1PE11.4 (default) (NM75)
This option defines the delimiter and number format for the present table, and subsequent tables, until a new FORMAT is specified. The first character defines the delimiter, which may be s for space, t for tab, comma (,) for a comma delimited file with aligned fields (so, padded with spaces), c for comma delimited file with no spaces, or q for comma delimited file with no spaces and double quotes around column names that have commas in their names (such as "OMEGA(2,1)").

The syntax for the number format is Fortran based, as follows:

For E field:
xPEw.d
indicates **w** total characters to be occupied by the number (including decimal point, sign, digits, E specifier, and 2 digit magnitude), **d** digits to the right of the decimal point, and **x** digits to the left of the decimal point.
Examples:
E12.5: -0.12345E+02
2PE13.6: -12.12345E+02

If you are outputting numbers that are less than 1.0E-99, such as 1.22345E-102, there will be one less significant digit displayed to make room for the extra digit in the exponent. To make room for a three digit exponent, you may set the format as follows:

xPEw.dEe

where e is the number of digits to be provided for the exponent. For example

1PE12.4E3: -2.3456E+002

For F field:
Fw.d
indicates **w** total characters to be occupied by the number (including decimal point, sign and digits), **d** digits to the right of the decimal point.
Examples:
F10.3: -0.012, 234567.123

For G field:
xPGw.d
For numbers >=0.1, will print an F field number if the value fits into w places showing d digits, otherwise will resort to xPEw.d format. For numbers <0.1, will always use xPEw.d format.

If the user-defined format is inappropriate for a particular number, then the default format will be used for that number.

An example $TABLE record could be:

```
$TABLE ID CMT EVID TIME NPRED NRES PREDI RESI WRESI CPRED CRES CWRES CPREDI
       CRESI CWRESI=ZABF EPRED ERES EWRES PRED RES WRES NPDE=PDERR ECWRES
       NOPRINT NOAPPEND FILE=myfile.tab ESAMPLE=1000 SEED=1233344
```

If you specify FORMAT=QCSV then this is equivalent to FORMAT=q1PG23.16
If you specify FORMAT=CSV then this is equivalent to FORMAT=c1PG23.16

IDFORMAT= I (NM75)

This secifies the format (do not include the delimiter) for the ID column. By default the ID column has the same format as specified by FORMAT. However, sometimes you wish the ID to appear as an integer, in which case, you may set IDFORMAT as I.

Some examples:
IDFORMAT=I
Integer value, left adjusted in the field.
IDFORMAT=I6
Integer value, right adjusted in the first 6 characters of the field
IDFORMAT=F6.1
Floating value, with single digit to the right of the decimal.

If an improper format is given, it defaults to that of FORMAT.

LFORMAT, RFORMAT (NM72)

An alternative format description to FORMAT is RFORMAT and LFORMAT. RFORMAT (where R=real numbers) describes the full numeric record of a table, so that formats for specific columns may be specified. LFORMAT (where L=label) specifies the format of the full label record of a table. The formats must be enclosed in double quotes, and (), and have valid Fortran

format specifiers. The RFORMAT and LFORMAT options can be repeated if the format specification is longer than 80 characters. Multiple RFORMAT and LFORMAT entries will be concatenated to form a single format record specification. For example,

```
       LFORMAT="(4X,A4,4(',',4X,A8))"
       RFORMAT="(F8.0,"
       RFORMAT="4(',',1PE12.5))"
```

Will result in the following formats submitted to a Fortran write statement:
`LFORMAT=(4X,A4,4(',',4X,A8))`
for the table's label record, and
`RFORMAT=(F8.0,4(',',1PE12.5))`
For the table's numeric records. If RFORMAT and LFORMAT are given, then the FORMAT option will be ignored. By default, FORMAT, RFORMAT, LFORMAT specifications will be passed on to the next $TABLE record in a given problem unless new ones are given. To turn off an RFORMAT/LFORMAT specification in a subsequent table (and therefore use FORMAT instead), set

```
       LFORMAT="NONE"
       RFORMAT="NONE"
```

Here is an example of $TABLE statements designated in a control stream file:

```
$TABLE ID TIME PRED RES WRES CPRED CWRES EPRED ERES EWRES NOAPPEND ONEHEADER
 FILE=tabstuff.TAB NOPRINT,FORMAT=,1PE15.8
$TABLE ID CL V1 Q V2 FIRSTONLY NOAPPEND NOPRINT FILE=tabstuff.PAR
       LFORMAT="(4X,A4,4(',',4X,A8))"
       RFORMAT="(F8.0,"
       RFORMAT="4(',',1PE12.5))"
$TABLE ID ETA1 ETA2 ETA3 ETA4 FIRSTONLY NOAPPEND NOPRINT
 FILE=tabstuff.ETA,FORMAT=",F12.4"
       LFORMAT="NONE"
       RFORMAT="NONE"
```

There is no NMTRAN error checking on the RFORMAT and LFORMAT records, so the user must engage in trial and error to obtain a satisfactory table output. You should set MAXEVAL=0 or MAXEVAL=1 for the $EST step to do a quick check, so you don't spend hours on estimation only to find the RFORMAT/LFORMAT were not appropriate, or use the model specification file system (MSFO/$MSFI) to preserve the estmation results and resume table output at the final estimates in a new control stream file.

A word of caution. The FORMAT descriptor 1P, which means move the decimal point to the left by 1, will be in effect for all remaining FORMAT components. For example, in

```
RFORMAT="(F8.0,37(',',1PE13.6),24(',',F7.2))"
```

the F field format that follows an E field format, in which 1P was used, will also have the decimal placed to the left, and a 1.00 would appear as a 10.00. To prevent this from occurring, revert to no decimal shift with 0P:

```
RFORMAT="(F8.0,37(',',1PE13.6),24(',',0PF7.2))"
```

PARAFILE (NM74)

As of NONMEM 7.4, computations of weighted residual diagnostics that are evaluated for table output are performed in a parallel computing setting, if parallel computing was requested, by the –parafile option on the command line, for example (see section I.73 Parallel Computing (NM72)). If you wish to turn parallel computing off for the weighted residual computation, then set parafile to off on the first $TABLE record in the problem:

```
$TABLE PARAFILE=OFF
```

Remember that parallelization remains OFF until you turn it back ON with a PARAFILE option in a $EST, $COV or $TABLE record in subsequent problems. Also, use $TABLE RANMETHOD=P to assure that the sequence of generated random numbers remain consistent for evaluation of NPDE, etc., for repeated executions of the control stream file

NOSUB=0 (Default) (NM74)

Subscripts of etas may be replaced with meaningful aliases, such as ETA(CL) in place of ETA(1). See I.7 Expansions on Abbreviated and Verbatim Code under *$ABBR REPLACE feature for abbreviated code (NM73-NM75)*. for how the $ABBR REPLACE command can be used to do this. These aliases will also be used as labels for the column identifiers in tables produced by $TABLE.. If you like the convenience of using meaningful labels in your abbreviated code, but do not wish to see them expressed in the table files for a given table, then set

```
$TABLE NOSUB=1
```

to turn this symbolic label substitution off. Scatter plots also are labeled with the aliases, which can be turned off or on for a given plot such as:
$SCAT NOSUB=1 ETA(V1) VS ETA(CL)
$SCAT NOSUB=0 ETA(V2) VS ETA(CL)

To set the default NOSUB for the entire problem, specify the following $DEFAULT (or $DEFAULTS) record:
$DEFAULT NOSUB=1

NOSUB also accepts a -1 to indicate revert to NONMEM default, which is to treat -1 as a 0. The NOSUB default setting is in effect throughout the control stream file, until another $DEFAULT record setting NOSUB is encountered.

FIXEDETAS=(*number-list)* (NM74)

It may be desired to treat certain etas, particularly $LEVEL etas that span groups of subjects, as if they were a fixed effect (theta) when evaluating populations characteristics during the $TABLE step, such as PRED, CWRES, NPDE, etc. In this way, the PRED evaluated will be, not of the total population, but of a given site level for that subject. For example, FIXEDETAS=(3-6,10-12) indicates etas 3 through 6, and 10 through 12 are to be placed at the

Empirical Bayes position, while the other etas are set to 0, during evaluation of PRED, RES, WRES, CWRES, NPDE, EWRES, etc. Consider example ..\example\fixedetas.ctl:

```
$PK
…
IF(COMACT==1) THEN
PREDCL=CL
PREDV=V
ENDIF
…
$LEVEL
SID=(3[1],4[2])
CID=(5[3],6[4])

$EST METHOD=ITS INTERACTION PRINT=1 NSIG=3 NITER=10 SIGL=6 FNLETA=0 MCETA=3
     LEVCENTER=1
$COV MATRIX=R UNCONDITIONAL
$TABLE  ID PREDCL PREDV DOSE RATE TIME CONC IPRED PRED IRES RES WRES CWRES
 EWRES NPDE NOAPPEND ONEHEADER FILE=FIXEDETAS.tab  NOPRINT
   ESAMPLE=1000 SEED=1115678
$TABLE  ID PREDCL PREDV DOSE RATE TIME CONC IPRED PRED IRES RES WRES CWRES
 EWRES NPDE NOAPPEND ONEHEADER FIXEDETAS=(3-6) FILE=FIXEDETAS2.tab  NOPRINT
 ESAMPLE=1000 SEED 1115678
```

Notice that FNLETA=0 so that the super-individual level etas (3,4, 5, and 6) evaluated during the estimation remain in memory (and not replaced by the FNLETA step), and so they are available for the table step. Notice also that the first table produces the standard population diagnostics, but the second table will produce those for which only Etas 1 and 2 (the inter-individual etas) are set to 0. Notice also that PREDCL and PREDV, because they are evaluated conditionally when COMACT=1, will be the population values (with their respective level interprations for each table) of CL and V. If you apply VARCALC=3 (see *Requesting Standard Errors to User-Defined and PREDPP Variables (NM74)* above), then you will obtain standard errors that incorporate errors associated with the super-individual level etas that are in PREDCL and PREDV.

I.17 $SUBROUTINES: New Differential Equation Solving Method: LSODA (ADVAN13)

As of NM7, A differential equation solver has been introduced, called LSODA, and is accessed using ADVAN=13 or ADVAN13. This routine is useful for stiff and non-stiff equations. This is similar to the LSODI routine used by ADVAN9, except that ADVAN13 can at times execute more quickly than ADVAN9. The ADVAN 13 differential equation solver has been shown to solve problems more quickly with the new estimation methods, whereas for classical NONMEM methods, selecting ADVAN 6 or 9 may still be of greater advantage.
Example:

```
$SUBROUTINES ADVAN13 TRANS1 TOL=5
```

Where TOL is the number of digits accuracy desired to integrate the differential equations (accuracy to within 10^{-TOL}). The code to the differential equation solver is found in ..\source\LSODA.f90. On occasion, coded errors will be displayed if the algorithm is having

trouble integrating the equations. These errors may usually be ignored, unless the error shows up frequently, and ultimately results in failure for the problem to complete. Typically the remedy is to increase or decrease TOL, but for those who desire to understand what the error codes mean, there are well documented comments on these at the beginning of LSODA.f90. They are printed here for convenience:

```
! ISTATE=An index used for input and output to specify the the state of the calculation.
!
!        On input,the values of istate are as follows.
!        1  Means this is the first call for the problem (initializations will be done).
!           See note below.
!        2  Means this is not the first call,and the calculation is to continue
!           normally, with no change in any input parameters except possibly TOUT
!           and ITASK. (If ITOL,RTOL,and/or ATOL are changed between calls with
!           ISTATE=2,the new values will be used but not tested for legality.)
!        3  Means this is not the first call,and the calculation is to continue
!           normally,but with a change in input parameters other than TOUT and ITASK.
!           changes are allowed in NEQ,ITOL,RTOL,ATOL,IOPT,LRW,LIW,JT,ML,MU and any
!           optional inputs except H0,MXORDN,AND MXORDS.
!           (see IWORK description for ML and MU.)
!        Note: A preliminary call with TOUT=T is not counted as a first call here,as
!        no initialization or checking of input is done. (Such a call is sometimes
!        useful for the purpose of outputting the initial conditions.) Thus the first
!        call for which TOUT /= T requires ISTATE=1 on input.
!
!        On output,istate has the following values and meanings.
!         1  Means nothing was done; TOUT=T and ISTATE=1 on input.
!         2  Means the integration was performed successfully.
!        -1  Means an excessive amount of work (more than MXSTEP steps) was done on
!            this call,before completing the requested task,but the integration was
!            otherwise successful as far as T. (MXSTEP is an optional input and is
!            normally 500.) TO continue,the user may simply reset ISTATE to a value > 1
!            and call again (the excess work step counter will be reset to 0).
!            In addition,the user may increase MXSTEP to avoid this error return
!            (see below on optional inputs).
!        -2  Means too much accuracy was requested for the precision of the machine
!            being used. This was detected before completing the requested task,but
!            the integration was successful as far as T. To continue,the tolerance
!            parameters must be reset,and ISTATE must be set to 3. The optional output
!            TOLSF may be used for this purpose. (Note: If this condition is detected
!            before taking any steps,then an illegal input return (ISTATE=-3) occurs
!            instead.)
!        -3  Means illegal input was detected,before taking any integration steps.
!            See written message for details.
!            Note: If the solver detects an infinite loop of calls to the solver with
!            illegal input,it will cause the run to stop.
!        -4  Means there were repeated error test failures on one attempted step,before
!            completing the requested task,but the integration was successful as far as T.
!            The problem may have a singularity,or the input may be inappropriate.
!        -5  Means there were repeated convergence test failures on one attempted step,
!            before completing the requested task,but the integration was successful as
!            far as T. This may be caused by an inaccurate jacobian matrix, if one is
!            being used.
!        -6  Means EWT(I) became zero for some I during the integration. Pure relative
!            error control (ATOL(I)=0.0) was requested on a variable which has now
!            vanished. The integration was successful as far as T.
!        -7  Means the length of RWORK and/or IWORK was too small to proceed,but the
!            integration was successful as far as T. This happens when DLSODA chooses
!            to switch methods but LRW and/or LIw is too small for the new method.
!
!        Note: Since the normal output value of ISTATE is 2, it does not need to be
!        reset for normal continuation. Also,since a negative input value of ISTATE
!        will be regarded as illegal, a negative output value requires the user to
!        change it, and possibly other inputs,before calling the solver again.
```

ATOL (NM72)

A $EST option when using ADVAN13 is the absolute tolerance. The ATOL for ADVAN13 by default is 12 (that is, precision is 10^{-12}). Usually the problem runs quickly when using ADVAN13 with this setting. On occasion, however, you may want to reduce ATOL (usually set it equal to that of TOL), and improve speeds of up to 3 to 4 fold. ATOL may be set at the $EST or $COV command. The absolute tolerance is set to the same ATOL for all compartments.

As of NM73, ATOL also acts on ADVAN9's differential equation solver, where by default absolute significant digits accuracy (absolute tolerance) is 12. As of NM74, for ADVAN9, ATOL=99 specifies that variable, calculated ATOL values are to be derived, in accordance with an algorithm that is present in ADVAN9. Do not use ATOL=99 for ADVAN13, 14, and 15, as it does not work properly for these algorithms.

The relative tolerance is still set by TOL by the $SUBROUTINES, $COV, or $TOL record.
As of NM74, ATOL acts on ADVAN14, ADVAN15, ADVAN16, ADVAN17,and ADVAN18 as well.

MXSTEP (NM73)

Additional control may be obtained by setting the maximum number of integration steps (default is 10000 for ADVAN13, 14, 15, 16, and 17 and 2147483647 for ADVAN9)

```
$PK
MXSTEP=5000
```

Suitable for ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, and ADVAN18 maximum integration steps can be controlled by this variable.

I.18 $SUBROUTINES TOL, ATOL, SSTOL, and SSATOL: Additional control of relative and absolute tolerance (NM74)

Additional TOL type options may be set at $SUBROUTINES, which will allow these settings to be in effect throughout the $PROB, including during simulation.

ATOL=n

The absolute tolerance can be set at the $SUBROUTINES record, similar to TOL, the relative tolerance. ATOL is used for ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, and ADVAN18. If not set, ATOL is set to 1.0E-12 by default. Note an ATOL set at the $EST record will override that set at the $SUB record, during the estimation. An ATOL set at the $COV record will override that set at the $SUBR record. Similarly, a relative TOL set at the $COV record will override a TOL set at the $SUBR record.

SSTOL=n

The relative tolerance for steady state evaluations can be set at the $SUBROUTINES record. If not specified, the SSTOL is set to TOL.

SSATOL=n

The absolute tolerance for steady state evaluations can be set at the $SUBROUTINES record. If not specified, the SSATOL is set to ATOL.

TOLC=n

The relative tolerance for the FOCE/LAPLACE $COV step can be set at the $SUBROUTINES record. TOLC is used for ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, and ADVAN18. If not set TOLC defaults to the value of TOL. Note a TOL set at the $COV record will override TOLC set at the $SUB record, during the estimation.

ATOLC=n

The absolute tolerance for the FOCE/LAPLACE $COV step can be set at the $SUBROUTINES record. ATOLC is used for ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, and ADVAN18. If not set, ATOLC is set to ATOL. Note an ATOL set at the $COV record will override the ATOLC set at the $SUB record.

SSTOLC=n

The relative tolerance for steady state evaluations during the FOCE/Laplace $COV step can be set at the $SUBROUTINES record. If not specified, the SSTOLC is set to SSTOL.

SSATOLC=n

The absolute tolerance for steady state evaluations during the FOCE/Laplace $COV step can be set at the $SUBROUTINES record. If not specified, the SSATOLC is set to SSATOL.

$TOL

An alternative method to setting TOL, ATOL, SSTOL, and SSATOL, TOLC, ATOLC, SSTOLC, and SSATOLC at the $SUBR record is by using the $TOL record, and specific ones for each compartment may be entered. For example, following the $TOL record, the following may be entered, one on each line:

```
NRD(0)=4 ; set SSTOL
NRD(1)=5 ; set TOL for compartment 1
NRD(2)=6 ; set TOL for compartment 2
…
ANRD(0)=4; set SSATOL
ANRD(1)=7; set ATOL for compartment 1
ANRD(2)=6; set ATOL for compartment 2
…
NRDC(0)=8 ; set SSTOLC
```

NRDC(1)=9 ; set TOLC for compartment 1
NRDC(2)=10 ; set TOLC for compartment 2
…
ANRDC(0)=3; set SSATOLC
ANRDC(1)=4; set ATOLC for compartment 1
ANRDC(2)=5; set ATOLC for compartment 2

Note that if NRD(0) and ANRD(0) are not defined, the values for NRD(1) and ANRD(1) are used, respectively. Similarly, if NRDC(0) and ANRDC(0), the values for NRDC(1) and ANRDC(1) are used, respectively For additional compartments not specified, the tolerance of the last compartment specified will be used. So in the above example, NRD(3)=NRD(2), NRD(4)=NRD(2), etc, and ANRD(3)=ANRD(2), ANRD(4)=ANRD(2), etc.

ATOL set at the $EST record and TOL and ATOL set at the $COV record will temporarily override these values set in $TOL, for all compartments (except the SS tolerances). These overrides at the $EST and $COV records are a legacy of pre-nm74 versions, and should no longer be used, now that $SUBR tolerance options and $TOL statements can allow the user to provide detailed settings to various compartment, steady state, estimation, and covariance tolerances.

As of nm75, you may specify compartment by its name defined in $MODEL. For example:

```
$MODEL COMP=(DEPOT,INITIALOFF,DEFDOSE) COMP=(CENTRAL,DEFOBS,NOOFF)
…
$TOL
NRD(DEPOT)=5
NRD(central)=6
```

Finally, you may supply a TOL routine that assigns values of NRD and ANRD specifically for each NONMEM step (estimation, covariance, simulation, table/scatter step, simulation, initial parameters estimate, nonparametric). For example, create a toluser.f90 file, as

```
      SUBROUTINE TOL(NRD,ANRD,NRDC,ANRDC)
      USE NMPRD_INT, ONLY: IPROB
      USE NM_BAYES_INT, ONLY: NM_STEP,BASE_STEP,EST_STEP,COV_STEP, &
      TABLE_STEP,SIML_STEP,INE_STEP, NONP_STEP
      INTEGER :: NRD(0:*), ANRD(0:*), NRDC(0:*), ANRDC(0:*)

      IF(NM_STEP==EST_STEP) THEN
      NRD(1)=6
      ANRD(1)=10
      ELSE IF (NM_STEP==COV_STEP) THEN
      NRD(1)=7
      ANRD(1)=8
      ELSE IF (NM_STEP==TABLE_STEP) THEN
      NRD(1)=8
      ANRD(1)=7
      ELSE
      NRD(1)=9
      ANRD(1)=12
      ENDIF
```

```
IF(IPROB>1) THEN
  NRD(1)=NRD(1)+1
  ANRD(1)=ANRD(1)+1
ENDIF
RETURN
END
```

and incorporate using $SUBR:

```
$SUBROUTINES ADVAN13 TRANS1 TOL=toluser.f90
```

You can have tolerances settings defined for each NONMEM step. Notice that NRD and ANRD defined with the XXX_STEP condition takes the place of having to define NRDC and ANRDC. You may even define tolerances for specific problems, etc., as shown above. More of the USE declarations for relevant parameters are listed in ..\util\nonmem_reserved_general. Compartment names defined in $MODEL are not available in user-defined TOL subroutines.

I.19 $SUBROUTINES: Yet Another New Differential Equation Solving Method: CVODES (ADVAN14) (NM74)

The CVODES ordinary differential equation (ODE) solver system allows advanced control of how the ODE's are solved. This system is for users with large sets of ODE's to solve, and who have some understanding of ODE solving methods. The CVODES is a subset of the Sundials system from Lawrence Livermore National Laboratory, by Alan Hindmarch and Radu Serban, and is a descendent of the LSODA (ADVAN13) system. For many problems, the LSODA (ADVAN13) method works just as efficiently as CVODES (ADVAN14), but there may be cases of very complicated models where the user may benefit from using ADVAN14.
ADVAN14 can be used with its default settings, by simply setting
$SUBROUTINES ADVAN14 …

If the user wishes to change the settings of the CVODES system, he should make a copy of ..\pr\CVODEU.f90 to the run directory (or other location), modify its contents, preferably rename it, such as CVODEU2.f90, so you know it is modified, then the $SUBROUTINES statement in the control stream file should include the OTHER statement:
$SUBROUTINES ADVAN14 .. OTHER=CVODEU2.f90

Include a path if your modified CVODEU file is not in your run path, e.g.:
$SUBROUTINES ADVAN14 .. OTHER=\my_favoriate_settings\CVODEU2.f90

While there are some comments in the CVODEU.f90 routine to identify the options, a thorough understanding requires consulting the SUNDIALS manual ..\guides\cvs_guide.pdf. Furthermore, the user may develop their own Jacobian evaluation and other routines, etc. Using ADVAN14 in a customized manner is not for the faint-hearted.

Options ATOL and MXSTEP are available to ADVAN14, as with ADVAN13.

ADVAN14 also has a root-finding algorithm. See the comments in CVODEU.f90 for its use. See also ..examples\cmax14.ctl and ..\examples\cmax14u.f90 for using the root-finder to the find the exact tmax and cmax in an asorption-central compartment model.

For the c routines to have been compiled, the proper c compiler (variable ccc) and compiler options (variable ccop) needed to have been set in the SETUP74 script at the time of building NONMEM. If this was not done at the time of building, you can modify the compiler and options in the appropriate cvode_build* script located in the ..\pr directory, and execute it from a terminal window.

I.20 $SUBROUTINES: Yet Another New Differential Equation Solving Method: IDAS (ADVAN15) (NM74)

The IDAS differential-algebraic equation (DAE) solver system allows advanced control of how the ODE's with equilibrium systems (algebraic equations) are solved. This system is for users with large sets of ODE's to solve, and who have some understanding of ODE solving methods. The IDAS is a subset of the Sundials system from Lawrence Livermore National Laboratory, by Alan Hindmarch and Radu Serban, and is a descendent of the LSODI1 (ADVAN9) system. For the few tests performed, the IDA system can work more efficiently than LSODI1 (ADVAN9) depending on the settings of parameters in IDAU.f90 (see below). There may also be cases of very complicated models where the user may benefit from using ADVAN15.
ADVAN15 can be used with its default settings, by simply setting
$SUBROUTINES ADVAN15 …

If the user wishes to change the parameters of the IDAS system, he should make a copy of ..\pr\IDAU.f90 to the run directory (or other location), modify its contents, preferably rename it, such as IDAU2.f90, so you know it is modified, then the $SUBROUTINES statement in the control stream file should include the OTHER statement:
$SUBROUTINES ADVAN15 .. OTHER=IDAU2.f90

Include a path if your modified IDAU file is not in your run path, e.g.:
$SUBROUTINES ADVAN15 .. OTHER=\my_favoriate_settings\IDAU2.f90

While there are some comments in the IDAU.f90 routine to identify the options, a thorough understanding requires consulting the SUNDIALS IDA manual ..\guides\idas_guide.pdf. Furthermore, the user may develop their own Jacobian evaluation and other routines, etc. Using ADVAN15 in a customized manner is not for the faint-hearted.

Options ATOL and MXSTEP are available to ADVAN15, as with ADVAN9.

ADVAN15 also has a root-finding algorithm. See the comments in IDAU.f90 for its use.

For the c routines to have been compiled, the proper c compiler (variable ccc) and compiler options (variable ccop) needed to have been set in the SETUP74 script at the time of building NONMEM. If this was not done at the time of building, you can modify the compiler and

options in the appropriate cvode_build* script located in the ..\pr directory, and execute it from a terminal window (cvode_build* scripts build CVODE and IDA c routines).

The default SS routine used for steady state assessments for ADVAN13 in nm710 to nm73 had been SS13. Also, ADVAN13 used FULL arrays internally, and could not use analytical second derivatives for the Laplace method.

As of nm74, the default SS routine for ADVAN13 and ADVAN14 is SS6, COMPACT or FULL arrays may be used (COMPACT by default, unless $ABBR DES=FULL is specified), and analytical second derivatives can be used for the Laplace method.

ADVAN9 and ADVAN15 utilize SS9 for the SS routine, use FULL array structure, can be used for equilibrium equations ($AES), and analytical second derivatives cannot be used for the Laplace method.

Before NM75, Sundials 2 version 2.6.2 was used for the source code underlying the ADVAN14 and ADVAN15 algorithms. As of NM75, ADVAN14 and ADVAN15 have been upgraded to SUNDIALS 5.1.0. The interface architecture between Fortran and C code has been modified to be systematically based on INTERFACE and MODULE definitions in accordance with Fortran 95 (2003), which allows Fortran programmers to access any C routine in CVODES and IDAS. The file ..\pr\cvode\fcvodes.f90 contains FORTRAN routines that properly connect the NONMEM ..\pr\CVODE.f90 and ..\pr\IDA.f90 routines with the sunfials routines. Study these files to understand how this interfacing works, if you desire to add additional functionality to the NONMEM system. Please read the chapter on Fortran interfacing in guides cvs_guide.pdf and idas_guide.pdf if you wish to understand this. Most users will not need to deal with this matter. However, the IDAU and CVODEU routine structures in nm75 differs from that of nm74 because some of the hook-in routines have changed from Sundials 2.6.2 to 5.1.0, so if you made custom IDAU or CVODEU files for nm74, please transfer your custom components into the new formats by using nm75's IDAU.f90 and CVODEU.f90 files.

I.21 ITASK_ and STOP_TIME: Avoiding overshoot in ADVAN9, ADVAN13, ADVAN14, and ADVAN15

These LSODA based routines use an algorithm for integration that overshoots the integration interval during calls to DES, but still accurately evaluates at the end of the integration interval when all calculations are completed. However, you may wish to capture a maximal or minimal value during $DES, and the overshoot should be turned off for this purpose. This is readily done by setting ITASK_=4 in $PK:

```
$PK
ITASK_=4
```

You may also specify a STOP_TIME (Tcrit) past which it should not integrate, if it is different from the end of the normal integration interval:

```
IF(TIME==4.0) STOP_TIME=5.0
```

To set back to default (end of normal integration interval),
STOP_TIME=-1.0d+300

ITASK_=3 will also work for ADVAN14 and ADVAN15. Do not use ITASK_=2 for any of the routines, or ITASK_=3 for ADVAN9 or ADVAN13, as inaccurate evaluations of the state variables will result. You can read about ITASK by searching for the word in ..\pr\LSODI1.f90 (ADVAN9), ..\pr\LSODA.f90 (ADVAN13), (..\pr\CVODE.f90, ..\pr\cvode\fcvodes.c, ..\pr\cvode\cvodes.c, ..\guides\cv*.pdf)=ADVAN14, and (..\pr\IDA.f90, ..\pr\cvode\fidas.c, ..\pr\cvode\idas.c, ..\guides\ida*.pdf)=ADVAN15.

I.22 $SUBROUTINES: Yet Another New Differential Equation Solving Method: Delay Differential Equations with no Equilibrium Compartments (ADVAN16, ADVAN18) or with Equilbrium Compartments (ADVAN17) (NM75)

For use of these delay differential equation (DDE) solvers, please see Using the Delay differential equation Solvers with the ddexpand program for Discrete Delay Problems (nm75) in I.85 ddexpand Utility Program for Modeling Discrete Time Delays (NM74)

I.23 Updating Amounts in Compartments at any Time: The A_UFLG Flag (NM75)

Previous to NM75, the amounts in compartments could be set by the use of the A_0() array only when EVID=3, EVID=4, or for the first record of an individual, that is, only at the “initial” state. The A_0FLG is set to 1 by PREDPP at those times, which then executes code that sets the A_0() values. For example:

```
$PK
If (A_0FLG==1) then
A_0(3)=k03/k30
Endif
```

Any A_0(x) not explicitly defined are set to 0. However, the A_0FLG may not set by the user in the $PK record, preventing the user from updating compartment amounts at any time. As of nm75, the modeler may update the compartment values at any time, using the flag A_UFLG (the ending U stands for update) with the desired compartment values set in the array A_U(). By use of MTIME to designate a variable time position at which an abrupt change in compartment amounts occurs, one could input a dose as follows:

```
MTIME(1)=wtime
MTDIFF=1
AZTEST=A_0FLG
IF(TSTATE==MTIME(1).AND.AZTEST==0) A_UFLG=1
IF(A_UFLG==1) THEN
A_U(1)=A(1)+wdose
A_U(2)=A(2)
A_U(3)=A(3)
ENDIF
```

The IF(A_UFLG==1) block is optional, as NMTRAN will insert one if not present. So the code could also be written as:

```
MTIME(1)=wtime
MTDIFF=1
AZTEST=A_0FLG
IF(TSTATE==MTIME(1).AND.AZTEST==0) A_UFLG=1
A_U(1)=A(1)+wdose
A_U(2)=A(2)
A_U(3)=A(3)
```

The A_UFLG event must be triggered with an IF(TSTATE==MTIME()) condition as indicated in the above example. Care should be taken that all the A_U(x) for all compartments x be properly defined. NMTRAN converts A_U() to A_0() during FSUBS code construction. You may wish to ensure A_0FLG is 0 whenever A_UFLG is set to 1, as shown in the example above (..\examples\a_uflg.ctl).

I.24 An Empirical Method of Achieving Steady State (NM75)

The usual way of inserting steady state dosing is to use the SS data item, and allowing NONMEM's PREDPP component to determine the starting state variables that will result in no change in values in state from one inter-dose interval (date item II) to the next. However, such a technique cannot be used with some systems of ODE's. This includes delay differential equations and also other systems, for example, those in which there is a change in kinetics outside the dosing interval. Also, use of SS requires that at least first derivatives be used, even when using EM/Bayes methods that do not normally require first derivatives.

An alternative method of steady state assessment is to do this empirically, by using the usual II, and ADDL (additional doses) data item method, where a pre-prescribed large number of doses (ADDL+1) are given, which are presumed to be a sufficient number such that near steady state is reached for all subjects and parameters, and thereafter records are added for monitoring after the last dose. One could decide to have 100 doses, say, followed by time samples after the last dose. But this would mean that 100 doses are always given, even though perhaps just 20 were needed to reach the steady state to an appropriate SSTOL/SSATOL tolerance.

As of nm75, the user may enter a negative valued ADDL to the data set, whose absolute value represents the maximum number of dose inputs to be given, followed by records with times relative to this last maximum dose. PREDPP will then determine, as it simulates each added dose, when the state variables no longer change according to the SSTOL/SSATOL tolerance specified in $SUBROUTINES, and stop adding doses when this is reached, up to the maximal ABS(ADDL). In addition, PREDPP will adjust the times of the subsequent records so it retains the same time after last dose relationship according to the actual number of doses given. The actual additional doses used is recorded in a reserved variable called ADDL_ACTUAL, accessible from $PK, $ERROR, $DES, and $AES. If ADDL_ACTUAL remains 0 for all records, this means the maximal number of doses ABS(ADDL) was reached before steady state occurred. The adjusted times are recorded in reserved variable ADDL_TIME, and the time

difference between TIME (which is not altered) and ADDL_TIME is recorded in reserved variable ADDL_TIMEDIFF. The time of integration T for the ODE's, TSTATE, DOSTIM, MTIME() are in reference to ADDL_TIME, not TIME, and the user's model should be aware of this adjustment. When outputting results to a table, the relevant times displayed should be ADDL_TIME, not TIME. A user defined variable defined in $PK or $ERROR may store the ADDL_TIME value, such as

```
ADDLTIME=ADDL_TIME
```

and then ADDLTIME can be outputted to the table. IF $PK is given limited calls (such as with CALLFL=1), then ADDLTIME should be defined in $ERROR. Similarly, if $ERROR is given limited calls (such as OBS ONLY or CALLFL=0), then ADDLTIME should be defined in $PK. In this way, the user-defined variable ADDLTIME is updated for each outputted record. There is no harm in defining ADDLTIME in both $PK and $ERROR.

Consider the following example, ..\examples\simpledii16_2d, which consists of a DDE problem, using ADVAN16, and records are set up for a maximum of 100 additional doses (designated as ADDL=-100) given every II=8 hours (data file simpledii16_2d.csv):

CID	TIME	AMT	RATE	II	ADDL	CMT	EVID	MDV	DV
100	0	100	0	8	-100	1	1	1	0
100	799.9	0	0	0	0	1	0	0	1
100	800	0	0	0	0	1	0	0	1
100	801	0	0	0	0	1	0	0	1
100	802	0	0	0	0	1	0	0	1
100	803	0	0	0	0	1	0	0	1
100	804	0	0	0	0	1	0	0	1
100	805	0	0	0	0	1	0	0	1
100	806	0	0	0	0	1	0	0	1
100	807	0	0	0	0	1	0	0	1
100	807.9	0	0	0	0	1	0	0	1
100	808	0	0	0	0	1	0	0	1
100	809	0	0	0	0	1	0	0	1
100	810	0	0	0	0	1	0	0	1
100	811	0	0	0	0	1	0	0	1
100	812	0	0	0	0	1	0	0	1
100	813	0	0	0	0	1	0	0	1
100	814	0	0	0	0	1	0	0	1
100	815	0	0	0	0	1	0	0	1
100	816	0	0	0	0	1	0	0	1
100	818	0	0	0	0	1	0	0	1
100	820	0	0	0	0	1	0	0	1
100	822	0	0	0	0	1	0	0	1
100	824	0	0	0	0	1	0	0	1
100	826	0	0	0	0	1	0	0	1

100	828	0	0	0	0	1	0	0	1
100	830	0	0	0	0	1	0	0	1

Notice that the times after the multi-dose record are in accordance to the maximum doses to be given (ABS(ADDL)+1=101), and that the observation records begin just before the last maximum number of doses (at 799.9 hours), to capture the trough value, followed by the peak of the last maximum number of bolus doses (at 800 hours), with hourly observations thereafter.

The pertinent information is captured for table output as follows (..\examples\simpledii16_2d.ctl):

```
$SUBROUTINES ADVAN16 TOL=10 ATOL=10 SSTOL=6 SSATOL=6
…
…
$PK
ADDLA=ADDL_ACTUAL ; may be set in $PK and/or $ERRROR
ADDLTIME=ADDL_TIME; may be set in $PK and/or $ERRROR
…
…
$TABLE ID TIME ADDLTIME A1 A4 A5 ADDL ADDLA
NOAPPEND NOPRINT ONEHEADER FILE=simpledii16_2d.tab
…
…
$ERROR
ADDLA=ADDL_ACTUAL  ; may be set in $PK and/or $ERRROR
ADDLTIME=ADDL_TIME ; may be set in $PK and/or $ERRROR
…
…
```

The resulting output (simpledii16_2d.tab) is shown as follows:

```
ID          TIME        ADDLTIME    A1          A4          A5          ADDL        ADDLA
 1.000E+02   0.000E+00   0.000E+00   1.000E+02   0.000E+00   0.000E+00  -1.000E+02   0.000E+00
 1.000E+02   7.999E+02   3.199E+02   4.867E+01   2.983E+02   6.525E+01   0.000E+00   4.000E+01
 1.000E+02   8.000E+02   3.200E+02   1.484E+02   2.981E+02   6.461E+01   0.000E+00   4.000E+01
 1.000E+02   8.010E+02   3.210E+02   1.068E+02   2.956E+02   5.919E+01   0.000E+00   4.000E+01
 1.000E+02   8.020E+02   3.220E+02   8.484E+01   2.923E+02   5.499E+01   0.000E+00   4.000E+01
 1.000E+02   8.030E+02   3.230E+02   7.238E+01   2.884E+02   5.148E+01   0.000E+00   4.000E+01
 1.000E+02   8.040E+02   3.240E+02   6.461E+01   2.841E+02   4.838E+01   0.000E+00   4.000E+01
 1.000E+02   8.050E+02   3.250E+02   5.919E+01   2.946E+02   1.068E+02   0.000E+00   4.000E+01
 1.000E+02   8.060E+02   3.260E+02   5.499E+01   2.987E+02   8.484E+01   0.000E+00   4.000E+01
 1.000E+02   8.070E+02   3.270E+02   5.148E+01   2.994E+02   7.238E+01   0.000E+00   4.000E+01
 1.000E+02   8.079E+02   3.279E+02   4.867E+01   2.983E+02   6.525E+01   0.000E+00   4.000E+01
 1.000E+02   8.080E+02   3.280E+02   4.838E+01   2.981E+02   6.461E+01   0.000E+00   4.000E+01
 1.000E+02   8.090E+02   3.290E+02   4.555E+01   2.956E+02   5.919E+01   0.000E+00   4.000E+01
 1.000E+02   8.100E+02   3.300E+02   4.294E+01   2.923E+02   5.499E+01   0.000E+00   4.000E+01
 1.000E+02   8.110E+02   3.310E+02   4.049E+01   2.884E+02   5.148E+01   0.000E+00   4.000E+01
 1.000E+02   8.120E+02   3.320E+02   3.820E+01   2.841E+02   4.838E+01   0.000E+00   4.000E+01
 1.000E+02   8.130E+02   3.330E+02   3.604E+01   2.794E+02   4.555E+01   0.000E+00   4.000E+01
 1.000E+02   8.140E+02   3.340E+02   3.400E+01   2.744E+02   4.294E+01   0.000E+00   4.000E+01
 1.000E+02   8.150E+02   3.350E+02   3.208E+01   2.691E+02   4.049E+01   0.000E+00   4.000E+01
 1.000E+02   8.160E+02   3.360E+02   3.027E+01   2.637E+02   3.820E+01   0.000E+00   4.000E+01
 1.000E+02   8.180E+02   3.380E+02   2.695E+01   2.523E+02   3.400E+01   0.000E+00   4.000E+01
 1.000E+02   8.200E+02   3.400E+02   2.400E+01   2.405E+02   3.027E+01   0.000E+00   4.000E+01
 1.000E+02   8.220E+02   3.420E+02   2.137E+01   2.285E+02   2.695E+01   0.000E+00   4.000E+01
 1.000E+02   8.240E+02   3.440E+02   1.903E+01   2.164E+02   2.400E+01   0.000E+00   4.000E+01
 1.000E+02   8.260E+02   3.460E+02   1.694E+01   2.044E+02   2.137E+01   0.000E+00   4.000E+01
 1.000E+02   8.280E+02   3.480E+02   1.508E+01   1.927E+02   1.903E+01   0.000E+00   4.000E+01
 1.000E+02   8.300E+02   3.500E+02   1.343E+01   1.812E+02   1.694E+01   0.000E+00   4.000E+01
```

Notice that ADDLTIME is the appropriate adjusted time when relating to the actual number of additional doses (ADDLA=40) given.

If you have code that involves MTIME(), the MTIME should be adjusted with ADDL_TIMEDIFF. For example, if you want MTIME(1) to be at TIME=812.5 hours (relative to maximal number of doses), then MTIME should be set to the adjusted time value, after ADDL_TIMEDIFF becomes non-zero (which is after PREDPP has assessed the appropriate number of additional doses):

```
IF(ADDL_TIMEDIFF/=0.0.AND.MTIME(1)==0.0) THEN
MTIME(1)=812.5+ADDL_TIMEDIFF
MTDIFF=1
ELSE
MTIME(1)=MTIME(1)
ENDIF
```

The above code is written for efficiency, so that MTIME(1) is set once.

Only the first negative ADDL is recognized as an indicator to adjust the number of doses. However, if you have a cluster of doses together, with the same starting time (althouth they may have different lag times), II, and ADDL, these will be treated as a composite steady state dosing strategy. For example, in method of steps problems, you may want a second dose record for the delay compartment (which may have an ALAG modeled in the control stream):

CID	TIME	AMT	RATE	II	ADDL	CMT	EVID	MDV	DV
100	0	100	0	8	-100	1	1	1	0
100	0	100	0	8	-100	5	1	1	0
100	799.9	0	0	0	0	1	0	0	1
100	800	0	0	0	0	1	0	0	1
100	801	0	0	0	0	1	0	0	1

A steady state continuous infusion problem would also be modeled with II and negative valued ADDL. For this case, set the II equal to AMT/RATE, so that when one infusion period ends, the next one begins. See ..\examples\simpledii16_2e.ctl. Do no use the SS item with the negative valued ADDL method.

I.25 $EST: Improvement in Estimation of Classical NONMEM Methods

In pre-NM7 NONMEM installations, the classical first order conditional estimation methods tended to be particularly sensitive to the formation of a non-positive definite Hessian matrix during the estimate of etas. In NONMEM 7, if the user selects NOABORT as a $EST option, most Hessian matrices will be forced to be positive definite if not already, allowing the program to continue, and abnormal termination of an estimation will occur less often. The occasional occurrence and correction of non-positive definite Hessian matrices during the intermediate steps does not typically result in erroneous results. Even with the NOABORT option, there is one remaining component in the NONMEM algorithm for which positive definite correction is not performed, which can still cause problems at the beginning of an estimation. It remains so the user may diagnose a serious problem in the setup of the estimation. Should this still be a nuisance, as of NONMEM 7.2.0 the user may select the NOHABORT option, which will

perform positive definite correction at all levels of the estimation, but it can hide a serious ill-posed problem, so use with care. Also, NOHABORT will cause NONMEM to ignore any errors (and not print them after 10 occurrences as of NONMEM 7.4) that occur during the evaluation of variables for the $TABLE step (when NP4F is called), and the variables printed in tables should therefore be cautiously assessed.

Resetting the Search to Circumnavigate Saddle Points and Detect Inestimable Parameters (NM74)

Sometimes the variable metric search algorithm used for FO/FOCE/Laplace ends near a local minimum with an eigenvalue that is near zero, suggesting a saddle point or inestimable or non-identifable parameters. You can request the saddle point reset, which repositions the values about 1 OFV unit away, and resumes the search, in hopes of continuing toward a minimum with a smaller OFV. This is based on the method by Yasunori and Nyberg, in the Perl Speaks NONMEM software. If the final OFV results is nearly the same value as just before the saddle point reset, and one or more of the final parameters differ from those just before the saddle reset (see .ext file, or Nparameters on the iteration just before the saddle reset mark), then this suggests that those parameters may be inestimable or non-identifiable.

SADDLE_RESET=0 (default) (NM74)

Saddle_reset is the number of times that you wish a reset to occur in the course of the search. Normally, you should request just 1.

SADDLE_HESS=0 (default) (NM74)

Saddle_hess=0 selects the Hessian matrix last generated by the variable metric method. This Hessian is not the true second derivative, but is a guaranteed positive definite matrix. Perturbing the estimates using this matrix requires very little computation, and is often sufficient to reposition the problem away from a saddle point. If SADDLE_HESS=1, then the full second derivative information matrix (identical to R matrix in the $COV step) will be evaluated, and used to reposition the estimates. This may work better than the SADDLE_HESS=0 setting, but the computational expense is high, equivalent to that of a $COV step.

I.26 Controlling the Accuracy of the Gradient Evaluation and Individual Objective Function Evaluation

In classical NONMEM methods (First order, First order conditional, Laplace), the user specifies SIGDIGIT or NSIG to indicate the number of significant digits that population parameters are to be evaluated at the maximum likelihood. If NSIG=3 (the default), then the problem would be optimized until all of the parameters varied by less than 3 significant digits. This same NSIG value would also be used to specify relative step size (h) to each THETA, SIGMA, and OMEGA, for evaluating the partial derivative of the objective function with respect to the parameter. Such partial derivative evaluations are needed to set up gradients to determine the direction the search algorithm must travel to approach the minimum of the objective function.

The forward finite difference of the partial derivative of O (the objective function) with theta(1) would be evaluated as

$$\frac{O(\theta_1(1+h))-O(\theta_1)}{\theta_1 h}$$

Numerical analysis of forward finite difference methods [6] recommends that the ideal relative step size h for the parameter theta(1) should be no greater than SIGL/2, where SIGL is the significant digits to which the objective function is evaluated. If h is set to a precision of SIGL/2 (which for the present discussion we mean it is set to $10^{-SIGL/2}$), then the resulting derivative itself will have approximately SIGL/2 precision as well.

In the main search algorithm, finite central difference methods are also used. These are evaluated as:

$$\frac{O(\theta_1(1+h))-O(\theta_1(1-h))}{2\theta_1 h}$$

Numerical analysis of central finite difference methods recommend that the ideal relative step size h for the parameter theta(1) should be no greater than SIGL/3. If h is set to SIGL/3, then the resulting finite difference value itself will have approximately 2*SIGL/3 precision.

The main search algorithm also utilizes pseudo-second derivative type evaluations using forward difference methods. For these calculations, an ideal h would be $10^{-SIGL/3}$, resulting in precision of second derivative constructs of about SIGL/3. Thus, it is safest to set the step size h, as specified by NSIG, to be no more than SIGL/3.

An internal SIGL in NONMEM specifies the precision to which the objective function itself (actually, the individual subject objective functions, which sum to the total objective function) is to be evaluated. This internal SIGL is set to 10. As long as NSIG was set to a value less then or equal to 10/2 or 10/3, then the gradients would be evaluated to an appropriate precision to make the gradient search algorithm work efficiently. With many subjects, if SIGL=10 is the precision to which each individual objective function is evaluated, and they are all of the same sign, then the sum objective function could have a resulting precision of $\log_{10}(N)$+SIGL, where N is the number of subjects, for a maximum of 15, the limiting precision of double precision. Thus with 100 subjects, the actual precision that the total objective function is evaluated could be 12. One should not necessarily rely on this, so it is safest to suppose the more conservative precision of 10, for which a suitable NSIG would be 3.

For analytical problems, those which do not utilize $DES, one can usually expect a reasonably efficient convergence to the minimum of the objective function with NSIG=3. However, with differential equation problems (those used for ADVAN 6, 8, 9, 13, 14, or 15), the limiting precision that objective function values may be evaluated is not based on the internal SIGL of 10, but rather, on the TOL level set by the user (where TOL represents the relative significant digits precision to which differential equations are to be integrated, so the precision is 10^{-TOL}), which is used by PREDPP when differential equations are integrated. The relationship between the predicted value and the individual subject's maximized objective function is complex, but one can use the rule of thumb that the individual's objective function is evaluated to a precision of the smaller of TOL and the internal SIGL. Thus, when a user specifies a TOL=4, then it may well be that the sum objective function has no greater precision than 4. If the user then specifies

NSIG=3, then the main search algorithm evaluates finite gradients using step size h that varies theta at the 3rd significant digit. This results in 1 significant digit precision remaining in evaluating the finite difference gradients. The search algorithm is now attempting to maximize the objective function to 3 significant digits, when it is working with gradients that are accurate to only 1-2 significant digits. This results in inefficient advancement of the objective function, causing NONMEM to make repeated evaluations within an iteration, as well as iterations for which the objective function is barely moving. NONMEM can then spend many hours trying to obtain precision in its parameters which are impossible to obtain. Eventually it may stop because the maximum iterations were used up, or when it realizes that it could not reach the desired precision.

With this understanding of the search algorithm process, and recognizing the complex relationship between the step size needed for each parameter and the finite difference method used in each part of the algorithm, the optimization algorithm was changed to allow the user to specify SIGL, and for the algorithm to set up the appropriate step size for a given finite difference method, based on the user-supplied SIGL. While some trial and error may still be required by the user for a given problem, certain general rules may be considered.

1) Set SIGL, NSIG, and TOL such that:
   SIGL<=TOL
   NSIG<=SIGL/3

With these options, the algorithm sets up the following:
For forward finite difference, h is set to SIGL/2 precision
For central finite difference, h is set to SIGL/3 precision
For forward second order difference, h is set to SIGL/3 precision
The individual fits for evaluating optimal eta values will be maximized to a precision of the user-supplied SIGL value

Optimization of population parameters occurs until none of the parameters change by more than NSIG significant digits.

For the $COV step, the step size for evaluating the R matrix (central difference second derivative) is set to SIGL/4, which according to numerical analysis, yields the optimal precision of SIGL/2 for the second derivative terms. If only the S matrix is evaluated (central difference first derivative), then the step size for it is set to SIGL/3. (But see $COV: Additional Options and Behavior for a way to set SIGL and TOL for $COV, distinct from the option for the $EST command).

If the user sets NSIG>SIGL/3, and specifies SIGL, then the optimization algorithm will do the following, which is a less than optimal setup:

For forward finite difference, h is set to NSIG precision
For central finite difference, h is set to NSIG precision
For forward second order difference, h is set to NSIG precision

The individual fits for evaluating optimal eta values will be maximized to a precision of the user-supplied SIGL value
Optimization of population parameters occurs until none of the parameters change by more than NSIG significant digits.
For the $COV step, the step size for evaluating the R matrix (central difference second derivative) is set to SIGL/4, which according to numerical analysis, yields the optimal precision of SIGL/2 for the second derivative terms. If only the S matrix is evaluated (central difference first derivative), then the step size for it is set to SIGL/3.

If the user does not specify SIGL, or sets SIGL=100, then the optimization algorithm will perform the traditional NONMEM VI optimization, which as discussed above, may not be ideal:

For forward finite difference, h is set to NSIG precision
For central finite difference, h is set to NSIG precision
For forward second order difference, h is set to NSIG precision
The individual fits for evaluating optimal eta values will be maximized to a precision of SIGL=10
Optimization of population parameters occurs until none of the parameters change by more than NSIG significant digits.

For the $COV step, the step size for evaluating the R and S matrix is set to NSIG, as is done in NONMEM VI. This is far from optimal, particularly for analyses requiring numerical integration, and is often the cause of the inability to evaluate the R matrix.

Command syntax:
Example:

```
$EST METHOD=1 INTERACTION SIGL=9 NSIG=3
```

To see the advantage of properly setting NSIG, TOL , and SIGL, consider the following problem, which is example 6 at the end of this document. Data were simulated with 17 PK and 18 PD observations for each of 50 subjects receiving a bolus of drug, followed by short infusion a week later. The PK model has 2 compartments (Vc, k12, k21) with first-order (k10) and receptor-mediated clearance (Vmax, Kmc). The PD model is indirect response, with receptors generated by zero order process (k03), and removed by first order process (k30) or via drug-receptor complex (Vmax, Kmc). There are 46 population parameters, variances/covariances, and intra-subject error coefficients, and thee differential equations. In the table below are listed the estimation times (not including a $COV step) using various SIGL, NSIG, and TOL values. Note that when not setting SIGL (NM 6 method), the problem would take a very long time. When SIGL, NSIG, and TOL were set properly, estimation times were much less, with successful completions. Of course, as they say in the weight-loss commercials, individual results may vary, and such great differences in execution times will not occur for all problems.

Advan method	NSIG=3 TOL=6 SIGL=100 (NM6 style)	NSIG=2 TOL=6 SIGL=6	NSIG=1 TOL=4 SIGL=3
9	>30	22	10
6	>24	17	3
13 (new)	>20	8.5	2

I.27 The SIGLO level (NM72)

As of NONMEM 7.2.0, the user may obtain even greater control of the precision at which various parts of the estimation are performed by using the SIGLO option. If used, the SIGLO option is the precision to which the individual etas are estimated. The SIGL level set by the user continues to be the precision (or delta) setting for the finite difference algorithms in the higher level estimation process for THETAS, OMEGAS, and SIGMAS. By default, if SIGLO is not specified, then SIGLO is set to the same value as SIGL, and everything is evaluated in accordance with the previous paragraph. Should SIGLO be used, the recommended setting would be:

SIGLO<=TOL
SIGL<=SIGLO
NSIG<=SIGL/3

I.28 Alternative convergence criterion for FO/FOCE/Laplace (NM72)

Sometimes many iterations will occur with very little change in the objective function, even with SIGL/TOL adjustment. This may occur because a parameter may oscillate at the 2nd significant digit, for example, and NSIG was set to 3. The parameter may never settle down to a value that fluctuates at less than NSIG significant digits if its contribution to the objective function is very small. Thus, a minimum objective function is achieved, but NONMEM's traditional convergence test, based on all parameters changing by less then NSIG significant digits, is never satisfied. An alternative convergence test is to set CTYPE=4 in the $EST statement. NONMEM will then additionally test if the objective function has not changed by more then NSIG digits beyond the decimal point over 10 iterations. If this condition is satisfied, the estimation will terminate successfully.

I.29 Additional Control for $MSFI record (NM73)

NOMSFTEST (NM73)

Sometimes the MSFI error check is too strict, and prevents an MSF file from being utilized in a subsequent control stream file or problem. This occurs particularly when using classical

NONMEM methods. To turn off MSFI error checking, set NOMSFTEST (default is MSFTEST):

```
$MSFI myfilename NOMSFTEST
```

NEW (NM74)

When the problem that created the MSF file has successfully completed, calling for a resumed or new estimation is prevented when the method is FO/FOCE/Laplace. To allow analysis to continue, or to allow an analysis on a new data set, resuming from the final parameters of the MSF file, use the option NEW:

```
$MSFI myfilename NEW
```

Be careful in its use. Your data set (items) structure and model should be identical to the ones used for generating the MSF file.

VERSION (NM74)

You can now read MSF files generated by previous versions of NONMEM, using the VERSION option:

```
$MSFI myfile.msf VERSION=7.4.0
$MSFI myfile.msf VERSION=7.3.0
$MSFI myfile.msf VERSION=7.2.0
$MSFI myfile.msf VERSION=7.1.2
$MSFI myfile.msf VERSION=7.1.0
$MSFI myfile.msf VERSION=6.2
$MSFI myfile.msf VERSION=6.1
```

I.30 General New Options for $ESTIMATION Record (NM73).

OPTMAP=0 (default) (NM73)

0: Standard variable metric (Broyden, Fletcher, Goldfarb, and Shanno (BFGS)) optimization method used by NONMEM to find optimal eta values (aka EBE, CPE, MAP, or conditional mode estimates, referred to symbolically $\hat{\eta}$, or eta hat) for each subject at the mode of their posterior densities, using analytical derivatives of F with respect to etas, and analytical derivatives of H with respect to etas, that were supplied by NMTRAN or by the user.
1: Variable metric method, using numerical finite difference methods for first derivatives of F with respect to etas. Necessary when not all code used in evaluating F, G and H for observation event records is abbreviated code (some may be in verbatim code), and/or some portions of the computation of F, G and H are evaluated in a hidden subroutine specified by "$SUBROUTINES OTHER=" and the user-written code does not compute the eta derivatives. When OPTMAP=1 is present, values of G and H are ignored during eta optimization. This may be used to test user-coded derivatives, because two runs, one with OPTMAP=1 and one without it, should give very similar values for the OBJV, WRES, etc. if the user-coded derivatives are correct. That is, the analytic derivatives in G and H are ignored, and this option may be used when analytic derivatives are difficult to compute (e.g., user supplied code such as SDE).

2: Nelder Mead method, which uses a secant method, rather than relying on derivatives.

ETADER=0 (default) (NM73)

In evaluating the MAP objective function, the term log(Det(V)) must be evaluated to obtained the marginal or integrated posterior density, where V is the eta Variance matrix based on the subject's posterior density.
0: Expected value V, using analytical first derivatives
1: Expected value V, using forward finite difference numerical first derivatives. Needed if not all code evaluating F and Y derivatives with respect to eta are available for processing by NM-TRAN or in user supplied code.
2: Expected value V, using central finite difference numerical first derivatives. Needed if not all code evaluating F and Y derivatives with respect to eta are available for processing by NM-TRAN or in user supplied code. That is, the analytic derivatives in G and H are ignored, and this option may be used when analytic derivatives are difficult to compute (e.g., user supplied code such as SDE).

3: 2nd derivative method of evaluating V, using numerical second derivatives of -log(L) with respect to etas. This is equivalent to using the "Laplace NUMERICAL method, even though FOCE may be selected.

When relying on numerical derivatives by using OPTMAP>0 or ETADER>0, you may need to set the SLOW option for proper estimation of FOCE or Laplace (SLOW is not utilized by EM/BAYES methods). Note also that non Monte Carlo weighted residual diagnostics (such as NWRES, NWRESI, CWRES, CWRESI) use first derivatives of F with respect to eta, and the appropriate numerical derivatives will be used to assess them if ETADER>=1.

NUMDER=0 (default) (NM73)

The file root.fgh is produced if the user selects $EST NUMDER=1. The file lists the numerically evaluated derivatives of Y or F with respect to eta, where
G(I,1)=partial F with respect to eta(i))
G(I,J+1)=Second derivatives of F with respect to eta(i),eta(j)
H(I,1)=partial Y with respect to eps(i)
H(i,j+1)=partial Y with respect to eps(i),eta(j))

This option is useful for comparing with and checking analytic derivatives values.

The analytical derivatives values are stored in root.agh, if NUMDER=2 is selected. If you want both, set NUMDER=3.

MCETA=0 (default) (NM73)

0: Eta=0 is initial setting for MAP estimation (eta optimization) during FOCE/LAPLACE/ITS/IMPMAP, and sometimes IMP.
1: ETA=values of previous iteration is initial setting for MAP estimation, or ETA=0, whichever gives lower objective function.

>1: MCETA-1 Random samples of ETA, using normal random distribution with variance OMEGA, are tested. Plus previous ETA is tested, and ETA=0 is tested. The test is, whichever supplies the lowest objective function is the eta set used as initial parameters for the MAP optimization.

NONINFETA=0 (default) (NM73)

NONMEM has traditionally not assessed post-hoc eta hat (also known as empirical Bayes Estimates, EBE's, conditional mode etas, or conditional parametric etas (CPE)), if the derivative of the data likelihood with respect to that eta is zero for a given subject, and simply specified that eta as zero. This eta is called a non-influential eta. The true EBE is zero anyway, if this eta is not correlated by an off-diagonal omega element with an eta that is influential. If the non-influential eta is correlated with an influential eta, then the true EBE of the non-influential eta will in general not be 0. When NONINFETA=0, the default, then this traditional algorithm is in effect, so that all non-influential etas, even those correlated with influential etas, will be reported as 0 when outputted with $TABLE. However, if NONINFETA=1, then all etas are involved in the MAP estimation, regardless of their influence. This will result in non-influential etas reported as a non-zero value, if it is correlated with influential etas. From a pure statistical stand-point, this is the true EBE, although intuitively it may be puzzling for some users. Whether NONINFETA=1 or 0, the individual's objective function will change very little if at all, because NONMEM provides a corrective algorithm to assess the correct objective function. But for purposes of post-hoc evaluated etas, one may wish to set NONINFETA depending on the desired interpretation. The NONINFETA option applies only to FO/FOCE/Laplace. The Monte Carlo and EM methods have always used (even with earlier versions of NONMEM 7) the pure statistical option (NONINFETA=1).

FNLETA=1 (default) (NM72)

Set FNLETA to 0 if you do not want it to spend time performing the end FNLMOD (which evaluates final mixture proportions for each subject in mixture models) and FNLETA (which evaluates final etas) routines using the original algorithm after the estimation and covariance steps are completed. You may want to turn this off if each objective function call takes a long time, with very complex problems or large data sets. Be aware, that certain $TABLE outputs, such as the traditional WRES, RESI, and PRED, may or may not be properly evaluated if the FNLMOD and FNLETA steps are omitted.

Normally, when you do not set FNLETA, or when you set FNLETA to 1, regardless of the method that was used (classical or EM/Monte Carlo) to obtain the thetas, omegas and sigmas in the last $EST step, $TABLE parameters are estimated based on a "post-hoc" evaluation of the etas at the mode of the posterior density position (eta hat), and using the final theta/omegas/sigma estimates (clascical and EM), or at the posterior means of thetas/omegas/sigmas (BAYES) (so, population parameters used are those listed on the -1000000000 line of the .ext file). These eta hat values are identical to those evaluated during the estimation for ITS/FOCE/Laplace methods, but differ from the mean values estimated during an IMP, SAEM, BAYES analysis.

Setting FNLETA=0 prevents the post-hoc analysis, so that $TABLE parameters are evaluated based on the eta/phi values generated by the last iteration of the last $EST method implemented (those listed in the .phi file), which are mode of posterior values for ITS/FOCE/Laplace, and conditional means for IMP/SAEM.

Before nm75, the phis/etas after a BAYES analysis yields single sample position values of the very last iteration, and had limited use. With nm75, $TABLE parameters use final MCMC posterior means (those listed in the .phi file, collected during the stationary phase) for the phis, and posterior means of the thetas, omegas, sigmas (those listed on the -1000000000 line of the .ext file). For example, suppose you have a user defined variable in $ERROR or $PK that is a function of THETA(1) and ETA(1):

MU_1=THETA(1)
CL=EXP(MU_1+ETA(1)) ; phi(1)=theta(1)+eta(1)
W=CL+THETA(2)

Then W will be output to the table as

W=exp(MCMC mean(THETA(1)+ETA(1))+MCMC mean(THETA(2))

If you wish to have the MCMC mean of W:

W=MCMC mean (EXP(MU_1+ETA(1))+THETA(2))

you will need to collect individual samples of W with an additional WRITE statement, and then summarize the the results after NONMEM analysis, as shown in example 8 in I.108 Example 8: Sample History of Individual Values in MCMC Bayesian Analysis.

Regardless of the FNLETA setting, the .phi and .phm tables (see I.63 $EST: Additional Output Files Produced) always output the phi/eta values used for the particular method (mode of posterior, and approximate Fisher information based variances for ITS/FOCE/Laplace methods, Monte Carlo assessed conditional means and conditional variances for SAEM/IMP methods).

If you set FNLETA=2 (NM73), then the estimation step is not done, and whatever etas are stored in memory at the time are used in any subsequent $TABLE's. This has value if you loaded the individual etas from an MSF file, or from a $PHIS/$ETAS record, and you want to calculate $TABLE items based on those etas, rather than from a new estimation. For example:

```
$PROB
$INPUT C ID GRP AMT TIME DV1 DV CMTS EVID MDV
$DATA mydata.csv IGNORE=C
…
$MSFI=myresults.MSF
…
$EST METHOD=1 FNLETA=2
$TABLE ID TIME DV IPRED CMTS MDV EVID NOAPPEND NOPRINT FILE=mytable.tab
```

To summarize:

FNLETA=1: Diagnostics depending on EBE's such as CWRES, CIWRES, CIPRED, etc., will use EBE's based on the final estimation method (conditional mode for FO/FOCE/Laplace/ITS, conditional mean for IMP/SAEM, MCMC posterior means for BAYES), while user selected items will use EBE's from the FNLETA step (eta modes).

FNLETA=0: All table outputs (diagnostics and user selected items) will use EBE's from final estimation method (conditional modes for FO/FOCE/Laplace/ITS, conditional means for IMP/SAEM, MCMC posterior means for BAYES).

FNLETA=2: All table outputs will use a common set of EBE's from an imported source. When using an imported source, such as a *_ETAS.msf or *.phi file (from a $ETAS/PHIS record), ensure that their ID ordering mathes that of the $DATA data file.

FNLETA=3 (as of nm74): Like FNLETA=1, will call FNLETA, and all table outputs (diagnostics and user selected items) will use EBE's from the FNLETA step (eta modes).

KNUTHSUMOFF =0 (default) (NM74)

In NONMEM 7.4, the Knuth summing method is used to allow the most accurate summation of individual objective function values, even with large variations in values of the individual objective function. To turn this off, and allow a standard summation (not recommended except for comparison purposes from earlier versions), set KNUTHSUMOFF=1. With KNUTHSUM algorithm on by default, the SORT option is not necessary.

FPARAFILE (NM74)

As of NONMEM 7.4, computations of final etas (empirical Bayes estimates) after the last estimation record are evaluated are performed in a parallel computing setting, if parallel computing was requested, by the –parafile option on the command line, for example (see section I.73 Parallel Computing (NM72)). If you wish to turn parallel computing off for the final eta computation, then set FPARAFILE to off (the F stands for FNLETA=final eta step) on any $EST record in the problem:

```
$EST FPARAFILE=OFF …
```

Note that if FNLETA=0, then parallelization will not occur at anyway, since final etas are not re-evaluated after the last iteration of the last estimation process performed. Please note setting PARAFILE to OFF, rather than FPARAFILE will prevent parallelization for the estimation step itself.

NOSUB=0 (Default) (NM74)

Subscripts of thetas, etas, and epsilons may be replaced with meaningful aliases, such as THETA(CL) in place of THETA(1). See I.7 Expansions on Abbreviated and Verbatim Code under *$ABBR REPLACE feature for abbreviated code (NM73-NM75).* for how the $ABBR REPLACE command can be used to do this. These aliases will also be used as labels

for the final estimates in the NONMEM report file. If you like the convenience of using meaningful labels in your abbreviate code, but do not wish to see them expressed in the report file for a given estimation step, then set
```
$EST NOSUB=1
```
to turn this symbolic label substitution off.

To set the default NOSUB for the entire problem, specify the following $DEFAULT record:
$DEFAULT NOSUB=1

NOSUB also accepts a -1 to indicate revert to NONMEM default, which is to treat -1 as a 0. The NOSUB default setting is in effect throughout the control stream file, until another $DEFAULT record setting NOSUB is encountered.

BOOTDATA=0 (Default) (NM75)

By default (BOOTDATA=0), when data are selected based on $SIML BOOSTRAP, the randomly selected subjects are analyzed during the subsequent estimation method. If BOOTDATA=1, then the subjects not selected are analyzed. Usually, you may wish to simply perform an evaluation, not an estimation, on these non-selected data (MAXEVAL=0, or NITER=0), at the population parameter values based on estimation of the selected subjects. For example:

$SIML BOOTSTRAP=-1 SUBP=100
$EST METHOD=1 MAXEVAL=9999 BOOTDATA=0 ; Estimated using randomly selected subjects
$EST METHOD=1 MAXEVAL=0 BOOTDATA=1 ; Evaluate non-selected subjects at estimated parameters from selected subjects.

Or for EM methods:
$SIML BOOTSTRAP=-1 SUBP=100
$EST METHOD=ITS NITER=50 BOOTDATA=0 ; Estimated using randomly selected subjects
$EST METHOD=ITS NITER=0 BOOTDATA=1 ; Evaluate at estimation of parameters

I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)

BOOTSTRAP (NM73)

$SIML BOOTSTRAP=-1 SUBP=100
$EST METHOD=1 INTERACTION
The above example requests a bootstrap rearrangement (with replacement) of an existing data set, followed by analysis of that data set. The BOOTSTRAP number refers to how many subjects are to be randomly selected from the data set. Setting -1 or to a value larger than the number of subjects in the data set means to randomly select as many subjects as are in the data set. For example, if 400 subjects are in the simulation template data set, then 400 subjects are randomly selected (with replacement, so some are selected more than once, others not at all). In this case, NONMEM's simulator does not perform the usual activity of randomly creating DV values for a new data set, but rather selects a random set of subjects of an existing data set (which must

already have legitimate DV values), uniformly selected (using source 1) with replacement. This results in some subjects not being selected at all, and some subjects selected more than once.

NOREPLACE (NM73)

$SIML BOOTSTRAP=50 SUBP=100 NOREPLACE
$EST METHOD=1 INTERACTION

In the above example, 50 unique subjects are to be randomly selected from the simulation template data set. The NOREPLACE feature is reasonable if there are many more than 50 subjects to choose from template set (for example, 1000 subjects in the template, and for each sub-problem, 50 of them are randomly chosen without replacement, that is, without repeating a subject).

STRAT (NM73)

$SIML BOOTSTRAP=50 SUBP=100 NOREPLACE STRAT=CAT

A single stratification data item may be entered. In the above example, the data item CAT serves as the stratification. This splits the data set into distinct sub-sets, guaranteeing a specific number of subjects will be selected from each category. For example, if in the base data set CAT has values of 1 or 2, with 33 subjects in group 1 and 67 subjects in group 2 out of 100 total subjects, then exactly 33% of of subjects from group 1 will be randomly selected out of 50 total (16), and exactly 67% of subjects will be randomly selected from group 2 (34). This has value when desiring that a bootstrap analysis maintain the same proportion of subjects belonging to certain categories, such as gender, or age bracket. To stratify by both age bracket and gender, create a stratification data item that would be, for example, valued 1 for subjects who are male under 30, 2 for subjects that are female under 30, 3 for subjects who are male over 30, 4 for subjects who are female over 30. Any discrete numerical values will do, as long as the stratifier is not a continuous variable, and the subjects need not be sorted according to the stratification data item.

STRATF (NM73)

$SIML BOOTSTRAP=50 SUBP=100 NOREPLACE STRAT=CAT STRATF=FCAT

The option STRATF points to a data item that contains the fraction that should represent a category in the bootstrapped data set. Without STRATF, the number of subjects to be taken from a given category is proportional to the number of subjects in the base data set. If you want the category to be represented at a different proportion, then specify a STRATF data item, in this example, FCAT. Suppose FCAT=0.5 for CAT=1 and 0.5 for CAT=2 as well. Even though only 33% of subjects in the base data set belong to category 1, exactly 50% of subjects from group 1 will be randomly selected out of 50 total (25), and exactly 50% of subjects will be randomly selected from group 2 (25) in the formation of each bootstrap data set. This allows you to alter the proportions in each category from what is in the original data det.

BOOTSTRAPPING SINGLE SUBJECT DATA (NM74)

The BOOTSTRAP option in $SIML is most suitable for multi-subject data, in which there is an ID data column identifying the subjects. However, because NONMEM will parse data records

belonging to a single subject in single subject data as if they belonged to separate subjects, the BOOTSTRAP option can be used to create a series of data sets with random samples of all records in the data set, and this may have some utility in obtaining summary estimates statistics for an individual curve fit. This is shown in the following control3boot.ctl example:

```
$PROBLEM  THEOPHYLLINE   SINGLE SUBJECT DATA
$INPUT  DOSE=AMT TIME CP=DV CAT
$DATA  DATA3B
$SUBROUTINES  ADVAN2

$PK
CALLFL=1
KA=THETA(1)
K=THETA(2)
SC=THETA(3)

$ERROR
IPRED=F
W=1.0
; first observation after dose is part of "first subject".  So, put in dummy
record, CAT=3,
;  and  give  it  a  residual  variance  that  is  very  large,  so  it  does  not
influence the fit.
IF(CAT==3.0) W=1.0E+10
Y=F+W*ERR(1)

$THETA  (0,1.7)  (0,.102)  (0,29)
$OMEGA 0.2

$SIML (567666 NORMAL) (33012 UNIFORM) BOOTSTRAP=-1 STRAT=CAT SUBP=100

$ESTIMATION MAXEVAL=9999  PRINT=2
$COVR
$TABLE TIME CAT AMT CP IPRED W NOAPPEND NOPRINT file=control3boot.tab
```

The data file is as follows:

```
    320    .0     .    1
     .     .1    3.0   3
     .     .27   1.71  2
     .     .52   7.91  2
     .    1.     8.31  2
     .    1.92   8.33  2
     .    3.5    6.85  2
     .    5.02   6.08  2
     .    7.03   5.4   2
     .    9.     4.55  2
     .   12.     3.01  2
     .   24.3     .90  2
```

Note that to modify the original control3 problem, the CAT data item was added. Each dose record should have its own category number, so that it will be presented in every random dataset, rather than sometimes there and sometimes not, and sometimes duplicated. In this case, the single dose record is given category 1. The observed data are given category 2, from which a random set will be selected for each data set created. Furthermore, the dose record and the first

data record after it is considered "one individual", and therefore, this first data record (in this example the one for time=0.1) will always travel with the dose record. Since this means that the time=0.1 data record is not randomly selected, but always present in every random set, it was made to be a "dummy" record, one that was not originally in the control3 problem. To ensure that this dummy record does not contribute to the estimate, it is flagged to have a residual deviation of W=1.0E+10, so that it is given very little weight, and does not impact the estimate. One can readily give it a CAT=3, to distinguish it from the other data records during estimation, although its category number is not recognized as a separate category during the random data set creation, since the data record is associated with the dose record with category 1.

PARAFILE (NM74)

As of NONMEM 7.4, there is the option to perform simulations in a parallel computing setting, if parallel computing was requested, by the –parafile option on the command line, for example (see section I.73 Parallel Computing (NM72)). By default, simulations are not parallel computed because they are performed rapidly. There may be occasions, however, when the model is sufficiently complicated, and/or there are many subjects in the template data set, that simulation could benefit from parallel computing. If you wish to turn parallel computing on for the simulation step, then set PARAFILE to ON:
$SIM PARAFILE=ON …

Alternatively, set –simparon on the nmfe75 command line to turn on parallelization during simulation step. The –simparon switch takes precedence.

When modeling with super-ID nested ETA levels ($LEVEL record is present), parallelization will not occur, since these etas are shared across individuals, and there is no guarantee that all subjects sharing the same etas will be simulated by the same process.

RANMETHOD=[n|S|m|P] (NM73)

As of NM73, the RANMETHOD option is available for the $SIM record, to use alternative random numbers generators (default is NONMEM's traditional one, number 4):

$SIML RANMETHOD=[n|S|m|P]

Where n is the random number generator type, S is Sobol sequence, and m is the Sobol scrambler. See the description of RANMETHOD under I.36 Monte Carlo Importance Sampling EM.

As of NM74, RANMETHOD will also act on the P modifier, which will retain separate random number sequences for each subject, so that the random variable patterns are retained regardless of whether the simulation is done in single computing or parallel computing mode. So, when parallel computing, you may select
$SIM PARAFILE=ON RANMETHOD=P

Before NM74, a P descriptor could be given without an error message, but it was ignored because the parallel random number generation for each individual was not yet developed for $SIM.

NONMEM's default random number generator for the $SIM step is 4 (in contrast, default random number generator for $EST and $TABLE is 3). Number 4 is NONMEM's classic random number generator. Whatever random number source is selected, it affects all seed1 sources, and all source seed2 if not also selecting P. The P descriptor sets initial seed2 to 0 when creating starting seeds for each individual.

The Sobol method is used only to generate normally distributed random vectors of etas and epsilons, when the S descriptor is selected, and SEED1 of source 1 is used to set the seed. Among the Sobol sequence methods, the S2 method appears to provide the least biased random samples, that is nearly uniform distribution, with good mixing in multi-dimensional spaces.

REWIND(NM74)

The REWIND feature in $SIM allows the original data set to be used for all sub-problems. By default, if any data item is changed by a sub-problem, those data items remain changed for the start of the next sub-problem. If you want that each sub-problem start with using the values from the original data set, use the REWIND feature of $SIM. So, any changes to the data set made during simulation (when ICALL=4) of a sub-problem are used for that sub-problem only, and are not preserved for the next sub-problem. Keep in mind that any transgeneration you may have performed on the data set when using an $INFN when ICALL=1 will be considered original data set. For example:

```
$INFN
IF (ICALL==1) THEN
DOWHILE(DATA)
..modifying statements here
ENDDO
ENDIF
```

NOSUPRESET(NM74)

By default, (SUPRESET), with subsequent iterations of a super-problem, the simulation random number generator is re-initialized back to the seed that is listed in the $SIM record of the control stream file. It may be desirable that each iteration serves as a new random instance, so use NOSUPRESET.

Simulation Error Forgiveness (NM72)

As of NM72, if a simulation error occurs (such as "KA nonpositive"), or a user-implemented EXIT 1 is invoked during simulation, then another eta and eps sample set will be generated and tested. If ten such errors occur in the same subject, then it is supposed that the cause of the simulation error is not due to an occasional bad random sample, but is caused by a systematic error in the control stream file, and the simulation is terminated.

Extensions to Simulation Error Forgiveness (NM75)

You may wish to test a predictive value result, and if unacceptable, request a new set of etas and epsilons. For example,

```
$ERROR
…
IF(ICALL==4.and.IPRED<0.01 .and. TIME>20.0) EXIT 1 2300
```

If the first value following the EXIT statement is 1 (out of a possible range of 0-2), and the second value is between 1000 and 9999 (out of a possible range of 0-9999), then upon the condition being true, NONMEM will try another random eta and eps sample. Be careful that the condition does not occur too often (causing wasteful computation) or that it always occurs due to logic error in the control stream, or NONMEM can be caught in an infinite loop.

Note that if you wish to simply create a truncated normal distribution that only requires testing the eta value directly, this can be done with:

```
$PK
…
DO WHILE (ETA(1).GT.5)
CALL SIMETA(ETA)
ENDDO
```

as suggested in Guide VIII, in ABBREVIATED CODE section.

If it is desired that the simulation be immediately terminated, then generate an EXIT 2 code:

```
IF(ICALL==4.and.IPRED<0.1 .and. TIME>20.0) EXIT 2
```

This will also over-ride the simulation error forgiveness.

Providing Separate Random Number Sequences for Etas and Epsilons (NM75)

By default, etas and epsilons are generated from the same random source 1, with epsilons generated after etas are generated, for each subject, in sequence. You may wish to retain the sequence of randomly generated etas (for variables) using one source, and retain a separate sequence of randomly generated epsilons (for observed data) using another source. To do this, specify the random source index with SOURCE_EPS, and provide starting seed values for both sources, as follows

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA seedeps.csv IGNORE=C
…
$SIMULATION ONLYSIMULATION SUBPROBLEMS=5
SOURCE_EPS=2 ; specify that epsilons should be generated from source 2
(567811 NORMAL) ; starting seed for Source 1 for Etas
(2567811 NORMAL); stating seed for Source 2 for epsilons
(2933012 UNIFORM) ; starting seed for source 3
```

```
                    ; for something else (for calling RANDOM(R,3)).
```

This is useful if you generate data from two separate data file templates which differ only in the number of data points among subjects, but you have the same number of subjects, and you wish to retain the same etas between the two data sets generated across subproblems.

Simulating THETAS with t-Distribution (NM75)

By setting TTDF to a non-zero integer value, thetas can be simulated with TTDF degrees of freedom t-distribution, when setting also TRUE=PRIOR. For example:

```
$SIMULATION (567811 NORMAL) (2933012 UNIFORM) SUBPROBLEMS=10 ONLYSIM
TTDF=4 TRUE=PRIOR
```

Make sure to also have the priors appropriately set up, such as:

```
$PRIOR NWPRI PLEV=0.999
$THETAP  (1.68 FIXED)  (1.58 FIXED) (8.12E-01 FIXED) ( 2.37 FIXED)
$THETAPV BLOCK(4)
0.04 FIX
0.00 0.4
0.00  0.00 0.4
0.00  0.00 0.0 0.4
$OMEGAP BLOCK(4)
 1.65276E-01  FIXED
 4.62368E-03  1.33770E-01
 6.35274E-03  1.69155E-02  2.13881E-01
 -1.53098E-02  1.25855E-02  5.32524E-02  1.62771E-01
$OMEGAPD (50 FIX)
$SIGMAP BLOCK(1) 0.0544 FIXED
$SIGMAPD (300 FIXED)
```

See see *$TTDF 0 (default,NM75)* to specify degrees of freedom for each theta. Specifying $SIML STTDF over-rides the $TTDF settings.

CLOCKSEED=0 (default) (nm75)

As of nm75, starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if CLOCKSEED=1, for each source seed (SEED1) or source seed pair (SEED1, SEED2). This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication. If SEED2<=0, then the clock time will not be added to SEED2.

I.32 Probability Density Functions (NM742)

A series of built in probability density functions are now available. For a given probability density there is also a cumulative distribution function (densitycdf), and random number generating function (density_rng). The following provides an example of how the exponentially modified normal density (EXPMODNORMAL, which returns –log(density)) may be used (other examples are in ..\examples\densities):

ran_expmodnormal.ctl

```
$PROB TESTING RANDOM SAMPLERS
$ABBR FUNCTION EXPMODNORMAL(VQI,10)
$ABBR VECTOR VV(10)
$INPUT  AMT TVAL DV
$DATA  rsampler.csv

$PRED
QM=theta(1)
SIGV=THETA(2)
LAMBDA=THETA(3)
IF(ICALL==4) THEN
; Simulatiuon block.  Enter arguments into vector VV().
The first argument usually the random sample to be generated
VV(1)=1.0 ; set to 1 as place holder
VV(2)=QM
VV(3)=SIGV
VV(4)=LAMBDA
; Use normal source 3, and uniform source 2, defined with $SIML
" CALL EXPMODNORMAL_RNG(3,2,VV)
; The generated sample is stored in VV(1), so set DV to this value
Y=VV(1)
DV=Y
ELSE
; Estimation block.  The density function returns -log(pdf), so multiply by 2
to make it a -2LL.
VQI(1)=DV ; data value
VQI(2)=QM ; next are the parameters to the density.
VQI(3)=SIGV
VQI(4)=LAMBDA
; density function is called, using VQI vector as first argument.
; NMTRAN adds arguments to capture first and second derivatives.
WW=EXPMODNORMAL(VQI)
Y=2.0*WW
ENDIF

$THETA 30.0 (0.0,5.0) (0.0,0.4)

$SIMULATION (567811 NORMAL) (2933012 UNIFORM) (445678 NORMAL) SUBPROBLEMS=1
$EST METHOD=0 MAXEVAL=9999 PRINT=1 -2LL NOTHETABOUNDTEST
$COVR
$TABLE TVAL DV NOAPPEND NOPRINT FILE=ran_expmodnormal.tab
```

The density and densityCDF functions have arguments that are compatible with the FUNC system, in which function provides derivatives (XD), and second derivatives (XDD) (see I.76 Expanded Syntax and Capacity for User-Defined Functions (FUNCA) (NM74)). Thus, even random (eta associated) variables may serve as arguments to the parameters of the density functions. The source code of these densities are in ..\source\DISTRIB.f90, DISTRIBCDF.f90, and DISTRIBRNG.f90. Note that multi-variate densities do not have a corresponding CDF routine.

In the above example, the sources for the random number generator are defined with the $SIML record, and use the same generators as the RANDOM() subroutine. The RANDMTU random

number generating system (see *RANDMTU Function (NM75)* in section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75)) may also be used with these probability density function sample generators,. The RANDMTU() routine should be first directly called to initialize the source and seed, and subsequently the particular _RNG() subroutine may be called, making the source argument negative (or 0). Using the example of EXPMODNORMAL:

```
IF(NEWIND==0) THEN
RVAL=RANDMTU(1,2,5678,-1)
RVAL=RANDMTU(0,3,5678,-1)
ENDIF
  VV(1)=1.0
  VV(2)=0.0
  VV(3)=2.0
  VV(4)=3.5
" CALL EXPMODNORMAL_RNG(-2,-3,VV)
  RVAL=VV(1)
```

The EXPMODNORMAL_RNG function will call source 2 of RANDMTU(), and obtain the appropriate normal sample and uniform sample.

Below are the list of densities (these rotuines actually return –log(density)), which are modeled after the format from the Stan manual [20].

BERNOULLI

Given $\alpha \in [0,1]$, then for y={0,1},

$$\text{Bernoulli}(y \mid \theta) = \begin{Bmatrix} \theta & \text{if } y = 1 \\ 1-\theta & \text{if } y = 0 \end{Bmatrix}$$

BERNOULLILOGIT

Given $\alpha \in R$, then for y={0,1},

$$\text{BernouuliLogit}(y \mid \alpha) = \begin{Bmatrix} \log it^{-1}(\alpha) & \text{if } y = 1 \\ 1-\log it^{-1}(\alpha) & \text{if } y = 0 \end{Bmatrix}$$

$$\text{logit}^{-1}(\alpha) = \frac{\exp(\alpha)}{1+\exp(\alpha)}$$

BINOMIAL

Given integer N, $\theta \in [0,1]$, and n={o,,,,N}

$$\text{Binomial}(n \mid N, \theta) = \binom{N}{n} \theta^n (1-\theta)^{N-n}$$

BINOMIALLOGIT

Given integer N, $\alpha \in R$, and n={o,,,,N}

$$\text{Binomial logit}(n \mid N, \alpha) = \binom{N}{n} \text{logit}^{-1}(\alpha)^n (1 - \text{logit}^{-1}(\alpha))^{N-n}$$

BETABINOMIAL

Given integer N, $\alpha \in R^+$, $\beta \in R^+$, and n={0,,,,N}

$$\text{Beta Binomial}(n \mid N, \alpha\beta) = \binom{N}{n} \frac{\text{B}(n+\alpha, N-n+\beta)}{\text{B}(\alpha, \beta)}$$

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

$$\Gamma(x) = (x+1)!$$

HYPERGEOMETRIC

Given integer *N*, integer *a*, integer *b*, and n={max(0,*N*-*b*),,,,,min(*a*,*N*)),

$$\text{Hypergoemtric}(n \mid N, a, b) = \frac{\binom{a}{n}\binom{b}{N-n}}{\binom{a+b}{N}}$$

CATEGORICAL

For positive integer N, $\theta \in R^N$ are N non-negative set of values summing to one (N-simplex), then for y={1,,,,,N},

$$\text{Categorical}(y \mid \theta) = \theta_y$$

Values are entered as:
X(1)=y
X(2)=N
X(3)=th(1)
X(4)=th(2)

…
X(N+1)=th(N-1)

CATEGORICALLOGIT

For positive integer N, $\beta \in R^N$, then for y={1,,,,,N},

$$\theta_y = \frac{\exp(\beta_y)}{\sum_{k=1}^{N} \exp(\beta_k)}$$

$$\text{Categorical}(y \mid \theta) = \theta_y$$

Values are entered as:
X(1)=y
X(2)=N
X(3)=beta(1)
X(4)=beta(2)
…
X(N+1)=beta(N-1)

and
Beta(N) is assumed 0.

ORDEREDLOGISTIC

For integer K, such that $c_k<c_{k+1}$ for k={1,,,,,K-2}, and $\eta \in R$, then for k={1,,,,,K}

$$\text{OrderedLogistic}(k \mid \eta, c) = \begin{Bmatrix} 1-\text{logit}^{-1}(\eta - c_1) & \text{if } k = 1 \\ \text{logit}^{-1}(\eta - c_{k-1}) - \text{logit}^{-1}(\eta - c_k) & \text{if } 1 < k < K \\ \text{logit}^{-1}(\eta - c_{k-1}) & \text{if } k = K \end{Bmatrix}$$

NEGBINOMIAL

For $\alpha \in R^+$, and $\beta \in R^+$, then for positive integer y

$$\text{NegBinomial}(y \mid \alpha, \beta) = \binom{y+\alpha-1}{\alpha-1} \left(\frac{\beta}{\beta+1}\right)^{\alpha} \left(\frac{1}{\beta+1}\right)^{y}$$

NEGBINOMIAL2

For $\mu \in R^+$, and $\phi \in R^+$, then for positive integer y

$$\text{NegBinomial}(y \mid \mu, \phi) = \binom{y+\phi-1}{\phi-1} \left(\frac{\mu}{\mu+\phi}\right)^{y} \left(\frac{\phi}{\mu+\phi}\right)^{\phi}$$

NEGBINOMIAL2LOG

For $\eta = \log(\mu) \in R$, and $\phi \in R^+$, then for integer positive y

$$\text{NegBinomial2log}(y \mid \eta, \phi) = \text{NegBinomial2}(y \mid \exp(\exp(\eta), \phi)$$

POISSON

For $\lambda \in R^+$, then for integer n>0

$$\text{Poisson}(n \mid \lambda) = \frac{1}{n!}\lambda^n \exp(-\lambda)$$

POISSONLOG

For $\alpha = \log(\lambda) \in R$, then for integer n>0

$$\text{Poisson}(n \mid \alpha) = \frac{1}{n!}\exp(n\alpha - \exp(\alpha))$$

MULTINOMIAL

If K is positive integer, $\theta \in K - \text{simplex}$, then for integer y, where

$$\sum_{k=1}^{K} y_k = N$$

$$\text{Multinomial}(y \mid \theta) = \begin{pmatrix} N \\ y_{1,,,,,y_k} \end{pmatrix} \prod_{k=1}^{K} \theta_k^{y_k}$$

Enter values in X array as follows for the –log density multinomial :
X(1)=K
X(2)=y(1)
…
X(K+1)=y(K)
X(K+2)=TH(1)
..
X(2K)=TH(K-1)
For the random number generator,
X(1)=K
X(2)=N ; Enter total value
X(3)=1.0 ; place holder (multinomial_rng will fill in X(3) to X(K+1)
…
X(K+1)=1.0
X(K+2)=TH(1)
..
X(2K)=TH(K-1)

As a multivariate density, Multinomial does not have a CDF routine.

NORMAL

For $\mu \in R$, $\sigma \in R^+$, $y \in R$,

$$\text{NORMAL}(y \mid \mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right)$$

EXPMODNORMAL

For $\mu \in R$, $\sigma \in R^+$, $\lambda \in R^+$, $y \in R$,

$$\text{EXPMODNORMAL}(y \mid \mu,\sigma,\lambda) = \frac{\lambda}{2}\exp\left(\frac{\lambda}{2}(2\mu+\lambda\sigma^2-2y)\right)2\left(1-\Phi\left(\frac{\mu+\lambda\sigma^2-y}{\sigma}\right)\right)$$

SKEWNORMAL

For $\mu \in R$, $\sigma \in R^+$, $\alpha \in R$, $y \in R$,

$$\text{SKEWNORMAL}(y \mid \mu,\sigma,\alpha) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right)2\Phi\left(\alpha\frac{y-\mu}{\sigma}\right)$$

STUDENTT

For $\mu \in R, \nu \in R^+$ $\sigma \in R^+$, $\alpha \in R$,

$$\text{STUDENTT}(y \mid \nu,\mu,\sigma) = \frac{\Gamma(\nu+1/2)}{\Gamma(\nu/2)}\frac{1}{\sqrt{\nu\pi}\sigma}\left(1+\frac{1}{\nu}\left(\frac{y-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$$

Likelihood estimator STUDENTT2 in DISTRIB.f90 is an alternative coding, but otherwise equivalent to STUDENTT.
There are three types of STUDENTT random number generators available in DISTRIBRNG.f90:
STUDDENTT_RNG(K,X):
Given uniform Random generator K, and parameters X(2)..., return random variable X(1).
Uses the TDEV2 routine in GENERAL.f90, which uses efficient random number generators for NU=1,2,4,6, and 10. NU=X(2) must be integer valued.

STUDDENTT2_RNG(K,X):
Given Normal Random generator K, and parameters X(2)..., return random variable X(1).
NU=X(2) may be non-integer. Algorithm uses two normal random variables to generate one univariate t-variable.

STUDDENTT3_RNG(K,X):
Given normal generator uniform Random generator K1, uniform generator K2, and parameters X(2)..., return random variable X(1). NU=x(2) may by non-integer.
This routine uses the normal variable/square root of chi-square variable) algorithm.

There are also the STUDENTTB series routines in DISTRIB.f90 and DISTRIB_RNG.f90, which deal with pairs of random correlated t-distributed samples. Please read comments in the source code for more information.

DOUBLEEXPONENTIAL

For $\mu \in R$, $\sigma \in R^+$, $y \in R$,

$$\text{DOUBLEEXPONENTIAL}(y \mid \mu,\sigma) = \frac{1}{2\sigma}\exp\left(-\frac{|y-\mu|}{\sigma}\right)$$

LOGISTIC

For $\mu \in R$, $\sigma \in R^+$, $y \in R$,

$$\text{LOGISTIC}(y \mid \mu,\sigma) = \frac{1}{\sigma}\exp\left(-\frac{y-\mu}{\sigma}\right)\left(1+\exp\left(-\frac{y-\mu}{\sigma}\right)\right)^{-2}$$

GUMBEL

For $\mu \in R$, $\beta \in R^+$, $y \in R$,

$$\text{GUMBEL}(y \mid \mu,\beta) = \frac{1}{\beta}\exp\left(-\frac{y-\mu}{\beta}-\exp\left(-\frac{y-\mu}{\beta}\right)\right)$$

LOGNORMAL

For $\mu \in R$, $\sigma \in R^+$, $y \in R$,

$$\text{LOGNORMAL}(y \mid \mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma y}\exp\left(-\frac{1}{2}\left(\frac{\log(y)-\mu}{\sigma}\right)^2\right)$$

CHISQUARE

For $\nu \in R$, $y \in R^+$,

$$\text{CHISQUARE}(y \mid \nu) = \frac{2^{-\nu/2}}{\Gamma(\nu/2)}y^{\nu/2-1}\exp\left(-\frac{1}{2}y\right)$$

INVCHISQUARE

For $\nu \in R$, $y \in R^+$,

$$\text{INVCHISQUARE}(y \mid \nu) = \frac{2^{-\nu/2}}{\Gamma(\nu/2)}y^{-\nu/2+1}\exp\left(-\frac{1}{2y}\right)$$

SCALEDINVCHISQUARE

For $\nu \in R,\ y \in R^+$,

$$\text{SCALEDINVCHISQUARE}(y \mid \nu,\sigma) = \frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)}\sigma^{\nu} y^{-\nu/2-1}\exp\left(-\frac{1}{2}\nu\sigma^2\frac{1}{y}\right)$$

EXPONENTIAL

For $\beta \in R^+, y \in R$,

$$\text{EXPONENTIAL}(y \mid \beta) = \beta\exp(-\beta y)$$

GAMMA

For $\alpha \in R^+, \beta \in R^+,\ y \in R^+$,

$$\text{GAMMA}(y \mid \alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha-1}\exp(-\beta y)$$

INVGAMMA

For $\alpha \in R^+, \beta \in R^+,\ y \in R^+$,

$$\text{INVGAMMA}(y \mid \alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{-\alpha-1}\exp\left(-\beta\frac{1}{y}\right)$$

WEIBULL

For $\alpha \in R^+, \sigma \in R^+,\ y \in R^+$,

$$\text{WEIBULL}(y \mid \alpha,\sigma) = \frac{\alpha}{\sigma}\left(\frac{y}{\sigma}\right)^{\alpha-1}\exp\left(-\left(\frac{y}{\sigma}\right)^{\alpha}\right)$$

FRECHET

For $\alpha \in R^+, \sigma \in R^+,\ y \in R^+$,

$$\text{FRECHET}(y \mid \alpha,\sigma) = \frac{\alpha}{\sigma}\left(\frac{y}{\sigma}\right)^{-\alpha-1}\exp\left(-\left(\frac{y}{\sigma}\right)^{-\alpha}\right)$$

RAYLEIGH

For $\sigma \in R^+,\ y \in R^+$,

$$\text{RAYLEIGH}(y \mid \sigma) = \frac{y}{\sigma^2}\exp\left(-\frac{y^2}{2\sigma^2}\right)$$

PARETO

For real positive y_{min}, $\alpha \in R^+,\ y \geq y_{min}$,

$$\text{PARETO}(y \mid y_{\min}, \alpha) = \frac{\alpha y_{\min}^{\alpha}}{y^{\alpha+1}}$$

PARETO2

For $\mu \in R$, $\lambda \in R^+$, $\alpha \in R^+$, $y \geq \mu$,

$$\text{PARETO2}(y \mid \mu, \lambda, \alpha) = \frac{\alpha}{\lambda}\left(1 + \frac{y-\mu}{\lambda}\right)^{-(\alpha+1)}$$

BETA

For $\alpha \in R^+$, $\beta \in R^+$, $\theta \in (0,1)$

$$\text{BETA}(\theta \mid \alpha, \beta) = \frac{1}{\text{B}(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

DIRICHLET

For positive integer K, $\alpha \in (R^+)^K$, then for $\theta \in \text{K} - \text{simplex}$

$$\text{DIRICHLET}(\theta \mid \alpha) \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma\left(\alpha_k\right)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

Enter values as follows:
X(1)=K
X(2)=TH(1)
…
X(K)=TH(K-1)
X(K+1)=alpha(1)
…
X(2K)=alpha(K)
As a multivariate density, DIRICHLET does not have a CDF routine.

VON MISES

For positive real μ, real positive κ, and real $-\pi \leq y - \mu \leq \pi$,

$$\text{VonMises}(y \mid \mu, \kappa) = \frac{\exp(\kappa \cos(y-\mu))}{2\pi \text{I}_0(\kappa)}$$

Where $I_0(\kappa)$ is a modified Bessel function. There is no VonMisesCDF routine available. Two methods of random sample generation are available:
Vonmises_rng: Lucio Barabesi , Statistica Applicata, Vol 7 number 4, 1995, pp. 417-426, page 420,
Vonmises_rng2: Stan system [20], which in turn is based on D. J. Best and N. I. Fisher, Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28, No. 2 (1979), pp. 152-157

I.33 Some Improvements in Nonparametric Methods (NM73)

EXPAND (NM73)

$NONP EXPAND
After the parametric estimation is performed, the final eta MAP (or empirical Bayes estimates, EBE) estimates, based on the final SIGMAS, OMEGAS, and THETAS, are normally used as support points. If the natural distribution of etas among subjects is highly non-normal, with large tails, or there are several outlier subjects, the final Omega values may constrain the EBE's of these outliers so they do not fit these subjects well. When EXPAND is selected, an alternative set of EBE's are evaluated using the initial OMEGA values, but using the final THETAS and SIGMAS. It is recommended that the initial OMEGAs have inflated values relative to the final OMEGAS (which is usually the case), to allow the outlier subjects to be fitted with little constraint from the population distribution. For each subject, the EBE that provides the highest individual likelihood value (not the highest posterior density) , whether from the final fit EBE, or the expanded OMEGA EBE, is selected as a support point. This is the inflated variance recommendation from [8].

NPSUPP (NM73)

$NONP NPSUPP=50
Number of total support points to be used. If NPSUPP>number of subjects (NIND), then extra support points are randomly created from the final OMEGAS (even when EXPAND is selected for the base EBE support points). This is the extended Grid Method as described in [8].

NPSUPPE (NM73)

$NONP NPSUPPE=50
Number of total support points to be used. If NPSUPPE>number of subjects, then extra support points are randomly created from the initial, presumably inflated, OMEGAS (even when EXPAND is not selected for the base EBE support points).

BOOTSTRAP (NM73)

$NONP BOOTSTRAP
The original data set is fitted during the parametric estimation ($EST), and the eta support points from the original data set are used for the nonparametric version. However, a bootstrap sample, with subjects uniformly randomly selected with replacement from the original data set, is used for the nonparametric distribution analysis. This is the simplified bootstrap technique described in [9]. To provide a series of simplified bootstrap analyses, as an example,

```
$SIML (12345) SUBP=100
```

```
$EST METHOD=COND INTERACTION MAXEVAL=9999 NSIG=3 SIGL=10 PRINT=5 NOABORT
$NONP BOOTSTRAP EXPAND
```

In the above example, BOOTSTRAP option is given in $NONP, along with the $SIML statement, without a BOOTSTRAP option. On the first sub-problem NONMEM will pass the original data to the estimation step ($EST), to obtain final THETAS, OMEGAS, and SIGMAS, with EBE's adjusted for expansion (EXPAND), followed by a nonparametric density analysis on the original data set. On the second sub-problem, the estimation step is skipped, but the final THETAS, OMEGAS, SIGMAS, and EBE's from the first analysis are retained, and a nonparametric density analysis is performed on a bootstrap version of the original data set.

For a full bootstrap analysis method, as described in [9]:

```
$SIML (12345) SUBP=100 BOOTSTRAP=-1
$EST METHOD=COND INTERACTION MAXEVAL=9999 NSIG=2 PRINT=5 NOHABORT
$NONP EXPAND NSUPPE=50
```

In the above example, 100 bootstrap analyses are performed. The $SIML provides a bootstrap version of the original data set for estimation by $EST, this is followed by EBE assessment on the original data set, followed by nonparametric density assessment on the bootstrap data set.

STRAT,STRATF (NM73)

As with $SIML, options STRAT and STRATF are available for the $NONP BOOTSTRAP record to provide stratified selections (see *STRAT (NM73)* in I.31 Bootstrap, Selecting a Random Method, and Other Options for Simulation (NM73,NM74)).

NPESTIM=0 (default, NM75)

The default non-parametric estimation method for assessing support point probabilities is an (non-Monte Carlo) expectation-maximization (EM) method. You may also choose the non-negative least squares method (NNL, [29]), with NPESTIM=1. While it is touted to be faster than EM (NNL is quadratically convergent whereas EM is linearly convergent), several tests have not indicated that NNL is any strong speed advantage. The reason is, that the computation time increases by at least the square of the number of support points (MAX(NSPSUPP,NIND)) with the NNL method, (least squares methods require matrix inversion, which is at least an N^2 order process), whereas with EM the computation time increases in proporation to MAX(NSPSUPP,NIND). Thus the larger the number of support points, the greater the speed advantage of the EM method.

NPMAXITER=1000 (default, NM75)

The default maximum iterations for non-parametric estimation of assessing support point probabilities is 1000, which is usually more than enough.

Three files are produced providing nonparametric information:

root.npd

Each row contains information about a support point: The support point number, the ID from which the support point was obtained as an EBE of that subject (ID is -1 if this support point was randomly generated because NSUPP/NSUPPE was greater than number of subjects). The eta values of the support point are listed, followed by the cumulative probability (CUM) associated with each eta, followed by the joint density probability of that support point, if default or MARGINALS was selected. If ETAS was selected, then instead of cumulative probabilities, the support point eta vector that best fits that subject (ETM) is listed.

root.npe
The expected value etas and expected value eta covariances (ETC) are listed for each problem or sub-problem. Because only one line is written per problem or sub-problem, the column header is displayed (unless $EST NOLABEL=1) only once for the entire NONMEM run. However, each line contains information of table number, problem number, sub-problem number, super problem and iteration number.

root.npi
The individual probabilities are listed in this file. The header line (unless $EST NOLABEL=1) is written only once, at the beginning of the file, per NONMEM run. Each line contains information of table number, problem number, sub-problem number, super problem, iteration number, subject number, and ID. This is followed by the individual probabilities at each support point (of which there are NSUPP/NSUPPE or NIND of them, whichever is greater). The line with Subject number=0 contains the joint probability of each support point (the same as listed in root.npd under the column PROBABILITY). The total objective function is stored in the OBJ column, for subject 0, as well as in the report file tagged as #OBJN. For each support point K, the joint probability is equal to the sum of the individual probabilities over all subject numbers I. Thus row of subject number I, column of support K, contains the individual probability IPROB(I,K). The OBJ column in this file contains the objective function contribution of each subject *i*. The sum of the individual probabilities over all support points for any given line (subject), is equal to 1/NIND. The format of the file is fixed at (,1PE22.15), and cannot be changed. It is intended for use in further analysis by analytical software, and is designed to report the full double-precision information of each probability.

PARAFILE (NM74)

As of NONMEM 7.4, nonparametric analysis can be parallelized (see I.73 Parallel Computing (NM72)). If you wish to turn parallel computing off for nonparametric analysis, then set PARAFILE to off:

```
$NONP … PARAFILE=OFF …
```

I.34 Introduction to EM and Monte Carlo Methods

Expectation-maximization methods use a two step process to obtain parameters at the maximum of the likelihood. In the expectation step, the thetas, omegas, and sigmas are fixed, while for each individual, expected values (conditional means) of the eta's and their variances are evaluated. If necessary, expected values of gradients of the likelihood with respect to the thetas and sigmas are also evaluated, integrated over all possible values of the etas. From these

constructs, the thetas and sigmas are updated during the maximization step using these conditional means of the etas and/or the gradients. The omegas are updated as the sample variance of the individual conditional means of the etas, plus the average conditional variances of the etas. The maximization step is therefore typically a single iteration process, requiring very little computation time. The more accurately these constructs are evaluated during the expectation step, the more accurately the total likelihood will be maximized.

I.35 Iterative Two Stage (ITS) Method

Iterative two-stage evaluates the conditional mode (not the mean) and first order (expected) or second order (Laplace) approximation of the conditional variance of parameters of individuals by maximizing the posterior density. This integration step is the same as is used in FOCE or Laplace. Population parameters are updated from subjects' conditional mode parameters and their approximate variances by single iteration maximization steps that are very stable (usually converging in 50-100 iterations). Because of approximations used, population parameters almost, but not quite, converge towards the linearized objective function of FOCE. Iterative two stage method is about as fast as FOCE with simple one or two compartment models, and when set up with MU referencing (described below) can be several fold faster than FOCE with more complex problems, such as 3 compartment models, and differential equation problems.

The iterative two stage method is specified by

$EST METHOD=ITS INTERACTION NITER=50

where NITER (default 50) sets maximum number of iterations. For all new methods, it is essential to set INTERACTION if the residual error is heteroscedastic.

I.36 Monte Carlo Importance Sampling EM

Importance sampling evaluates the conditional (posterior) mean and variance of parameters of individuals (etas) by Monte Carlo sampling (integration, expectation step). It uses the posterior density which incorporates the likelihood of parameters relative to population means (thetas) and variances (etas) with the individual's observed data. By default, for the first iteration, the mode and first order approximation of the variance are estimated (called mode a posteriori, or MAP estimation) as is done in ITS or FOCE, and are used as the parameters to a normal distribution proposal (sampling) density. From this proposal density Monte Carlo samples are generated, then weighted according to the posterior density as a correction, since the posterior density itself is generally not truly normally distributed, and conditional means and their conditional variances are evaluated. For subsequent iterations, the normal density near the mean of the posterior (obtained from the previous iteration) is used as a proposal density. Population parameters (thetas, sigmas, and omegas) are then updated from subjects' conditional mean parameters, gradients, and their variances by single iteration maximization steps that are very stable, and improve the objective function. The population parameters converge towards the minimum of the objective function, which is an accurate marginal density based likelihood (exact likelihood). A series of options defined at the $EST command are available to the user to control the performance of the importance sampling, such as the number of Monte Carlo samples per

individual (ISAMPLE), and scaling of the proposal density relative to the posterior density (IACCEPT). Termination criteria (CITER, CALPHA, CTYPE, and CINTERVAL) may also be set, which are explained in detail in a later section. Typically, 300 Monte Carlo samples are needed, and 50-200 iterations are required for a randomly stationary objective function, that is, when the objective function does not vary in a directional manner beyond the Monte Carlo fluctuations.

The Importance sampling method is specified by

$EST METHOD=IMP INTERACTION

Followed by one or more of the following options:

NITER/NSAMPLE=50

Sets maximum number of iterations (default 50). Typically, 50-100 iterations are need to for a problem to have a randomly stationary objective function.

ISAMPLE=300

Sets number of random samples per subject used for expectation step (default 300). Usually 300 is sufficient, but may require 1000-3000 for very sparse data, and when desiring objective function evaluation with low Monte Carlo noise.

ISAMPEND=n, STDOBJ=d (NM73)

For importance sampling and direct sampling only, if ISAMPEND is specified as an integer value greater than ISAMPLE, and STDOBJ is set to a real value greater than 0, then NONMEM will vary the number of Monte Carlo samples under each subject between ISAMPLE and ISAMPEND, until the stochastic standard deviation of the objective function falls below STDOBJ.

IACCEPT=0.4

Expand proposal (sampling) density variance relative to conditional density so that on average conditional density/proposal density=IACCEPT (default 0.4). For very sparse data or highly non-linear posterior densities (such as with categorical data), you may want to decrease to 0.1 to 0.3.

IACCEPT=0.0 (NM7.3)

For importance sampling only, you may set IACCEPT=0.0, and NONMEM will determine the most appropriate IACCEPT level for each subject, and if necessary, will use a t-distribution (by altering the DF for each subject) as well. If IACCEPT=0, the individual IACCEPT values and DF values will be listed in root.imp, where root is the name of the control stream file.

ISCALE_MIN=0.1 (defaults for IMP, NM72)

ISCALE_MAX=10.0 (NM72)

In importance sampling, the scale factor used to vary the size of the variance of the proposal density in order to meet the IACCEPT condition, is in NM72 by default bounded by ISCALE_MIN of 0.1, and ISCALE_MAX=10.0. On very rare occasions, the importance sampling objective function varies widely, and the scale factor boundary may need to be reduced (perhaps ISCALE_MIN=0.3, ISAMPLE_MAX=3). After the importance sampling estimation, remember to revert these parameters to default operation on the next $EST step:
ISCALE_MIN=-100 ISCALE_MAX=-100.

Note: the values to ISCALE_MIN and ISCALE_MAX for the IMP method in NONMEM 7.1 and earlier were 0.01,100, respectively, and were not changeable by the user.

SEED=11456 (default)

The seed for the random number source used in Monte Carlo integration is initialized (default seed is 11456).

CLOCKSEED=0 (default) (nm75)

As of nm75, the actual starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if CLOCKSEED=1. This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication.

MAPITER=1 (default) (NM72)

By default, MAP estimation is performed only on the first iteration, to obtain initial conditional values (modes and approximate variances) to be used for the sampling density. Subsequently, the Monte Carlo assessed conditional means and variances from the previous iteration are used as parameters to the sampling density. However, the user can select the pattern by which MAP estimations are intermittently done, and their conditional statistics used for the sampling density. MAPITER=n means the first n iterations are to use MAP estimation to assess parameters for the sampling density. After these n iterations, the Monte Carlo conditional means and variances of the pervious iteration are used for the sampling density parameters of the present iteration. If MAPITER=0, then the first iteration will rely on conditional means and variances that are in memory. These may have come from an MSF file, or from a previous estimation step.

MAPINTER=0 (default) (NM72)

Every nth iteration, the MAP estimation should be used to provide parameters to the sampling density. Thus, if MAPITER=20 and MAPINTER=5, then for the first 20 iterations, MAP estimation is used, and thereafter, every 5^{th} iteration the MAP estimation is used. If MAPINTER=-1 (NM73), then mapinter will be turned on only if the objective function increases consistently over several iterations.

Setting an option to -100 will force NONMEM to select the default value for that parameter.

MAPCOV=1 (default) (NM74)

For iterations for which the MAP estimation is performed, by default (MAPCOV=1), the MAP estimated mode is used as the center (mean) for the sampling density, and the first order (or second order if Laplace option is used) approximate conditional variance is used as the variance of the sampling density. If MAPCOV=0, then only the mode is used for the sampling density's center, and the Monte Carlo assessed variance of the previous iteration is used as the sampling density's variance. If MAPCOV=2, then the Monte Carlo assessed conditional mean of the previous iteration is used, but the MAP first order (or second order if Laplace option) assessed variance is used for the sampling density. This option has been added for experimental purposes, and has no value for the user. It should be left at its default value of 1.

EONLY=1

Evaluate the objective function by performing only the expectation step, without advancing the population parameters (default is 0, population parameters are updated). When this method is used, NITER should equal 5 to 10, to allow proposal density to improve with each iteration, since mean and variance of parameters of normal or t distribution proposal density are obtained from the previous iteration. Also it is good to get several objective function values to assess the Monte Carlo noise in it.

As of NM74, if EONLY=2, then not only are the population parameters not updated with each iteration, neither are the individual conditional means/modes and conditional variances until the last iteration, if the variances of the population parameter estimates are estimated ($COV is requested). If EONLY=3, then the conditional modes and approximate variances from MAP estimation will be saved on the first iteration, and used for the sampling density of all subsequent iterations. This improves efficiency when selecting MAPITER=1, MAPINTER=0, so that the MAP estimation did not need to be repeatedly performed.

DF=4

The proposal density is to be t distribution with 4 degrees of freedom. Default DF=0 is normal density. The t distribution has larger tails, and is useful for situations where the posterior density has a highly non-normal distribution. For very sparse data or highly non-linear posterior densities (such as with categorical data), you may want to set DF to somewhere between 2 and 10.

RANMETHOD=[n|M|S|m|P] (default n=3) (NM72)

Where
n=0-4
m=0-3

By default, the random number generator used for all Monte Carlo EM and Bayesian methods use the Knuth method, ran3 of reference [5]. We feel this is the best random number generator for many purposes. However, you may choose alternative random number generators (n) as follows (n=0-4):
0: ran0 of reference [5], minimal standard generator
1: ran1 of reference [5], Bays and Durham.
2: ran2 of reference [5].
3: ran3 of reference [5], Knuth.
4: NONMEM's traditional random number generator used in $SIMULATION

For special purposes, a sobol [5] sequence method with or without scrambling [10] may be called upon, and only for the purpose of creating quasi-random samples of eta vectors. To select the sobol method without scrambling, add an S to RANMETHOD. For example,
RANMETHOD=2S
Selects random number generator ran2 for general purposes, and sobol sequence for the eta vector generation. The number m is reserved for the type of scrambing desired (m=0-3):
0: no scrambing (so S0 is the same as S)
1: Owen type scrambling
2: Faure-Tezuka type scrambling
3: Owen plus Faure-Tezuka type scrambling.

Other examples:
RANMETHOD=S1
Indicates sobol sequence with Owen scrambling for eta vector generation. Since there is no integer in the first position of RANMETHOD indicated, the general random number generator remains unchanged from the RANMETHOD specification previously specified, or ran method 3, if none was specified earlier.

RANMETHOD=1S2
Indicates ran1 type random number generator for general purposes, sobol sequence with Faure-Tezuka scrambling for eta vector generation.

The sobol sequence method of quasi-random number generation can reduce the Monte Carlo noise in the objective function evaluation during importance sampling under some circumstances. When the sampling density fits the posterior density well, such as with rich, continuous data, the sobol sequence method does not reduce the Monte Carlo noise by much. If you are fitting categorical data, or sparse data, and perhaps you are using the t distribution (DF>0) for the importance sampling density, then sobol sequence generation may be helpful in reducing Monte Carlo noise. The RANMETHOD specification propagates to subsequent $EST records in a given problem, but does not propagate to $CHAIN or $TABLE records.

In NM72, only DIRECT and IMP/IMPMAP methods could utilize the Sobol quasi-random method. As of NM73, Sobol may be used for BAYES and SAEM methods as well. From experience, The S0 and S1 methods produce considerable bias for SAEM and BAYES, whereas S2 and S3 perform better.

As of NM73, if you add a P descriptor to RANMETHOD, such as

RANMETHOD=P
RANMETHOD=3P
RANMETHOD=3S2P

then each subject will receive its own random number sequence, that will stay with that subject regardless of whether the job is run as a single process or parallel process. This assures that stochastically similar answers will be obtained for Monte Carlo estimation methods, regardless of the number of processes or different kinds of parallelization setups used to solve the problem. There is additional memory cost in using this option because the seed and seed status (additional internal variables of the random number generator that establish the random sequence) must be stored for each subject, and for SOBOL/QR sampling there may even be a reduction in speed because the random sampling algorithm has to be re-set for each subject. To reiterate, a single job run without the P descriptor will not be stochastically similar to a single job run with the P descriptor (although they will be statistically similar), or to any parallel job run. But, a single job run using the P descriptor will be stochastically similar to any parallel job run also using the P descriptor. If maintaining stochastic similarity regardless of how the job is run (single or any parallel profile) is important to you, then always set the P descriptor (so, RANMETHOD=P, at least).

Note on the t-Distribution Sampling Density (DF>0), and its Use With Sobol Method (RANMETHOD=S)

When using the t-distribution sampling density (DF>0), by default the algorithm creates a random vector from *n* independent univariate t-distributed samples. This is called the U algorithm, and the most efficient use of the U type t-distribution is when DF=1,2,4,5,8, or 10. These algorithms were designed to work well with the Sobol method's ability to reduce Monte Carlo noise.

As of NM74, another way of producing the vector is from a multi-variate t-distribution algorithm (suggested by Robert Leary), which can be selected by placing an M in the RANMETHOD descriptor, placed after the random number method number, for example:
RNAMETHOD=3MS2

(the default setting is U for composite univariate). The multivariate t-distribution algorithm (M) produces samples that have radially symmetric densities (that is, the density is a function of the sum of squares of vector elements) may provide a better fitting sampling density for some kinds of models, and hence, more efficient sampling. However, the individual random elements in the vector are not statistically independent, and when used with the Sobol method, does not result in as much reduced Monte Carlo noise as when the composite univariate t-distribution vector (U) is used. An alternative method is by a mixture of two normal densities, using IACCEPTL.

IACCEPTL =0 (default) (NM74)

If IACCEPTL is set to greater than 0 then NONMEM uses this value as a scale to a second multi-variate normal density, to cover long tails in the posterior density (hence L for long tails),

in combination with the ACCEPT value to cover the posterior density near the mode. For one half of ISAMPLE samples, IACCEPT is used to scale a multivariate-normal proposal density, and for the other half of ISAMPLE samples, IACCEPTL is used to scale another multivariate normal proposal density. Thus, a mixture of two normal densities, with two different variance scales, are used as the proposal density. This serves as a pseudo t-distribution, but assuring radial symmetry as well as statistically independent samples, that may be useful when using with the Sobol method. This method has been suggested by Robert Leary, recommending IACCEPT=1.0 and IACCEPTL=0.01.

If IACCEPT is set to 0, and IACCEPTL is set to a value greater than 0, then a search for the best IACCEPTL for each subject is made, starting the testing at the IACCEPTL value given by the user, while IACCEPT is fixed to 1. The root.imp file will contain the final values selected for each subject, listing the two IACCEPT scale values, the first one being 1 for near the mode, and the second for the long tails.

GRDQ=0 (default) (NM74)

The gradient quick option, called GRDQ, allows thetas that must be gradient assessed (such as those that are not mu-referenced) and SIGMAS to be more quickly evaluated by not evaluating the gradients for every one of the ISAMPLE random samples, but chooses a subset of the most important samples. This reduces the computational cost, since gradients of the objective function with respect to the thetas require more objective function calls than is usually required when evaluating mu-referenced thetas. If GRDQ>=1, then this is interpreted as the number of important samples to be used for theta gradient assessment per subject. If GRDQ<1, then GRDQ is interpreted as the fraction of ISAMPLEs to be used (GRDQ*ISAMPLE samples are used for theta gradient assessment). When GRDQ<0.0, then the number of samples used is ABS(GRDQ)*ISAMPLE/(Number of subjects with observations).
When GRDQ=0 (default), then all ISAMPLE samples are used to evaluate the theta gradients. Some experience suggests that if GRDQ is too low (<30 samples), the quality of standard error assessments may deteriorate, so some trial and error may be needed to determine to what extent the GRDQ can be reduced. Suggestion of the GRDQ algorithm courtesy of Robert Leary.

During estimation, the iteration output for importance sampling contains the objective function, average effective number of samples (eff.), actual samples (smpl., which is usually ISAMPLE), and average fitness (fit.) of proposal (sampling) density to the posterior density. The average effective number of samples per total samples is essentially the average ratio (or weights) of posterior to proposal density at all sample positions, and NONMEM tries to adjust the scaling of the variance to the proposal density so that on average, eff./smpl. approximates the IACCEPT value. The average fitness is the average absolute difference between the posterior density and proposal density among the samples, and indicates to what extent that the proposal density matches the posterior density. Typically if the IACCEPT is not too low, then the closer the shape of the posterior density is to a normal density, the higher the fitness, with perfect fitness being 1 if IACCEPT is set to 1.

I.37 Monte Carlo Importance Sampling EM Assisted by Mode a Posteriori (MAP) estimation

Sometimes for highly dimensioned PK/PD problems with very rich data the importance sampling method does not advance the objective function well or even diverges. For this the IMPMAP method may be used. At each iteration, conditional modes and conditional first order variances are evaluated as in the ITS or FOCE method, not just on the first iteration as is done with IMP method. These are then used as parameters to the multivariate normal proposal density for the Monte Carlo importance sampling step. This method is implemented by:

$EST METHOD=IMPMAP INTERACTION

This is equivalent to

$EST METHOD=IMP INTERACTION MAPITER=1 MAPINTER=1

I.38 Stochastic Approximation Expectation Maximization (SAEM) Method

As in importance sampling, random samples are generated from normal distribution proposal densities. However, instead of always centered at the mean or mode of the posterior density, the proposal density is centered at the previous sample position. New samples are accepted with a certain probability. The variance of the proposal density is adjusted to maintain a certain average acceptance rate (IACCEPT). This method requires more elaborate sampling strategy, but is useful for highly non-normally distributed posterior densities, such as in the case of very sparse data (few data points per subject), or when there is categorical data.

In the first phase, called the burn-in or stochastic mode, SAEM evaluates an unbiased but highly stochastic approximation of individual parameters (semi integration, usually 2 samples per individual). Population parameters are updated from individual parameters by single iteration maximization steps that are very stable, and improves the objective function (usually in 300-5000 iterations). In the second mode, called the accumulation mode, individual parameter samples from previous iterations are averaged together, converging towards the true conditional individual parameter means and variances. The algorithm leads to population parameters converging towards the maximum of the exact likelihood.

The SAEM method is specified by

$EST METHOD=SAEM INTERACTION

Followed by one or more of the following options:

NBURN=2000

Maximum number of iterations in which to perform the stochastic phase of the SAEM method (default 1000). During this time, the advance of the parameters may be monitored by observing the results in file specified by the FILE parameter (described later in the Format of Output Files section), and the advance of the objective function (SAEMOBJ) at the console may be monitored. When all parameters or the SAEMOBJ do not appear to drift in a specific direction,

but appear to bounce around in a stationary region, then it has sufficiently "burned" in. A termination test is available (described later), that will give a statistical assessment of the stationarity of objective function and parameters.

The objective function SAEMOBJ that is displayed during SAEM analysis is not valid for assessing minimization or for hypothesis testing. It is highly stochastic, and does not represent a marginal likelihood that is integrated over all possible eta, but rather, is the likelihood for a given set of etas.

NSAMPLE/NITER=1000

Sets maximum number of iterations in which to perform the non-stochastic/ accumulation phase (default 1000).

ISAMPLE=2 (defaults listed)

ISAMPLE_M1=2

ISAMPLE_M1B=2 (NM74)

ISAMPLE_M1A=0 (NM72)

ISAMPLE_M2=2

ISAMPLE_M3=2

IACCEPT=0.4

These are options for the MCMC Bayesian Metropolis-Hastings algorithm for individual parameters (ETAS) used by the SAEM and BAYES methods. For each ISAMPLE, SAEM performs *ISAMPLE_M1* mode (or kernel) 1 iterations using the population means and variances as proposal density, followed ISAMPLE_M1B mode 1B iterations using the individual conditional mean and individual conditional variance collected from previous iterations as proposal density, followed by ISAMPLE_M1A mode 1A iterations, testing model parameters from other subjects as possible values (by default this is not used, ISAMPLE_M1A=0), followed by *ISAMPLE_M2* mode 2 iterations, using the present parameter vector position as mean, and a scaled variance of OMEGA as variance [11]. Next, ISAMPLE_M3 mode 3 iterations are performed, in which samples are generated for each parameter separately. The scaling is adjusted so that samples are accepted IACCEPT fraction of the time. The final sample for a given chain is then kept. The average of the *isample* parameter vectors and their variances are used in updating the population means and variances. Usually, these options need not be changed.

The ISAMPLE_M1A method of sampling has limited use to assist certain subjects to find good parameter values by borrowing from their neighbors, in case the neighbors had obtained good values while the present subject has difficulty finding good samples. This mode should generally not be used, and can be inaccurate if not all subjects share the same $\boldsymbol{\mu}$ and $\boldsymbol{\Omega}$, such as in covariate modeling. Alternatively, use mode 1A sampling at the beginning of an SAEM

analysis for a few burn in iterations, then continue with a complete SAEM analysis with mode 1A sampling turned off, with more burn in and accumulated sampling iterations, for example:

$EST METHOD=SAEM INTERACTION NBURN=500 NITER=0 ISAMPLE_M1A=2
$EST METHOD=SAEM INTERACTION NBURN=500 NITER=1000 ISAMPLE_M1A=0

As of NM75, If the option MCETA is set to greater than ISAMPLE_M1, then for the first iteration of SAEM, ISAMPLE_M1, ISAMPLE_M2, and ISAMPLE_M3 will be set to MCETA, to facilitate a robust initial search for reasonable parameter values.

ISAMPEND=n (NM73)

For SAEM, if ISAMPEND is specified as an upper integer value (usually 10), then NONMEM will perform a ISAMPLE preprocess to determine the best ISAMPLE value. For the ISAMPLE preprocessing the used entered ISAMPLE value must be at least 2. It will perform 200 iterations during the ISAMPLE preprocess, and the last 50 iterations will be used to obtain average conditional variance/OMEGA (eta shrinkage) for each subject. The largest etashrinkage fraction*10 is the ISAMPLE for that subject. Thus,
ISAMPLE=2 ISAMPEND=10

Will assess a best ISAMPLE for each subject. The ISAMPLE will not be higher than 10 or lower than 2.

ISCALE_MIN=1.0E-06 (defaults for SAEM, BAYES, NM72)

ISCALE_MAX=1.0E+06 (NM72)

In MCMC sampling, the scale factor used to vary the size of the variance of the proposal density in order to meet the IACCEPT condition, is by default bounded by ISCALE_MIN of 1.0E-06, and ISCALE_MAX=1.0E+06. This should left alone for MCMC sampling, but on occasion there may be a reason to reduce the boundaries (perhaps to ISCALE_MIN=0.001, ISAMPLE_MAX=1000). After the SAEM estimation method, remember to revert these parameters back to default operation on the next $EST step:
ISCALE_MIN=-100 ISCALE_MAX=-100
The default operation is that NONMEM sets (ISCALE_MIN,ISCALE_MAX) to (0.1,10) for importance sampling (as described earlier), and to (1.0E-06,1.0E+06) for MCMC sampling.

NOCOV=[0,1] (NM73)

If covariance estimation is not desired for a particular estimation step, set NOCOV=1. It may be turned on again for the next estimation step with NOCOV=0. If NOCOV=1 is set for an FOCE/Laplace/FO method, this is equivalent to $COV NOFCOV setting. For ITS and IMP, covariance estimation can take some time for large problems, and you may wish to obtain only the objective function, such as in the case of $EST METHOD=IMP EONLY=1 after an SAEM estimation. NOCOV has no effect on BAYES analysis, as no extra time is required in assessing covariance for BAYES.

By default, standard error information for the classical methods (FO/FOCE/Laplace) will be given only if they are the last estimation method, even if NOCOV=0 for an intermediate estimation step. If NOCOV=1 for the FOCE/LAPLACE/FO method, and it is the last estimation step, then standard error assessment for it will be turned off.

DERCONT=[0,1] (NM73)

By the default value of the derivative continuity (DERCONT) is 0. When it equals 1, the partial derivative of the objective function with respect to thetas will perform an additional test to determine if a backward difference assessment is more accurate than a forward difference assessment. The forward difference assessment can differ greatly from the backward difference assessment in cases of extreme discontinuity when varying certain thetas by even just a small amount in the model results in a large change in objective function, (such as a viral model in which a very small change in the potency of an anti-viral agent results in widely varying time of return of viral load). This results in standard errors being poorly assessed for thetas that do not have inter-subject variances associated with them. Setting DERCONT to 1 slows the analysis, but can provide more accurate assessments of SE in such models. The DERCONT works only for the Monte Carlo EM algorithms such as IMP and SAEM.

CONSTRAIN=1 (NM72)

A built-in simulated annealing algorithm has been put in place for NONMMEM 7.2.0. Simulated annealing slows the rate of reduction of the elements of the OMEGA values during the burn-in phase of the SAEM method, allowing for a more global search of parameters. The subroutine CONSTRAINT performs this algorithm when the option CONSTRAIN is set to 1 or 5, where 1 is the default setting. This is by the constraint algorithm starting the Omegas at 1.5 times the initial values, and then controlling the rate at which the Omegas shrink during each iteration. CONTRAIN=2 or 6 performs simulated annealing on sigma parameters, CONSTRAIN=3 or 7 performs simulated annealing on both OMEGA and SIGMA parameters. CONSTRAIN=0 or 4, performs no simulated annealing on non-zero valued OMEGAS.

The user may modify the subroutine CONSTRAINT that performs the simulated annealing algorithm. The source code to the CONSTRAINT subroutine is available from the ..\source directory as constraint.f90 , and the user may copy this to their run directory, and as convenient, to rename it. Then, specify OTHER=name_of_source.f90 in the $SUBROUTINE record, as shown in example 9.

As of NM73, when CONSTRAIN>=4, simulated annealing is also performed on diagonal elements of OMEGAS that are fixed to 0 to facilitate estimation of any associated thetas. See I.55 $ANNEAL to facilitate EM search methods for this additional annealing technique. The subroutine CONSTRAINT may also be used to provide any kind of constraint pattern on any parameters.

As of NM72, the SAEM setting produces first order approximation standard errors, that is, MATRIX=S type, but not proper objective function for hypothesis testing.

PHITYPE=0 (NM74)

By default, after an estimation is performed, the phi(), conditional means of the individual parameters, and their variances, are reported in the *root*.phi file, where *root* is the root name of the control stream file. If you wish to have conditional mean etas reported, set PHITYPE=1. See *root.phi* in section I.63 $EST: Additional Output Files Produced for more information.

MAPITERS=0 (default) (NM75)

By default, no MAP estimation is performed with SAEM. To get good individual parameter values near the mode of the posterior density for the first iteration of SAEM, you can set MAPITERS=1. Alternatively, you can insert the record:
$EST METHOD=ITS NITER=0
Followed by
$EST METHOD=SAEM …

The mapping of parameters between Monolix and NONMEM SAEM is as follows:

Monolix	**NONMEM SAEM**
Number of Chains	ISAMPLE
K0	CONSTRAINT subroutine may be user modified to provide any constraining pattern on any population parameters
K1	NBURN
K2	NITER
Auto K1	CTYPE=1,2,3
Population Parameter settings menu:	
rho	IACCEPT
m1	ISAMPLE_M1
m2	ISAMPLE_M1A
m3	ISAMPLE_M2
m4	ISAMPLE_M3
No simulated annealing	CONSTRAIN=0
Simulated Annealing	CONSTRAIN=1,2,3 User may also define algorithm
SEED	SEED

Obtaining the Objective Function for Hypothesis Testing After an SAEM Analysis

After the analysis, suitable objective functions for hypothesis testing and second order standard errors can be obtained by importance sampling at the final population parameter values. Thus, one could issue this sequence of commands:
$EST METHOD=SAEM INTERACTION NBURN=2000 NITER=1000
$EST METHOD=IMP EONLY=1 ISAMPLE=1000 NITER=5

Here, after SAEM is performed, importance sampling, with MAP estimation done on its first iteration, is performed, but without updating the main population parameters. Sometimes the MAP estimation is problematic, and/or, the user wishes to use the SAEM's last conditional mean and variances as the parameters to the importance sampler's sampling density for the first iteration, so one may try:

$EST METHOD=SAEM INTERACTION NBURN=2000 NITER=1000
$EST METHOD=IMP EONLY=1 ISAMPLE=1000 NITER=5 MAPITER=0

For very large dimensioned problems (many Omegas), the IMP evaluated objective function can have a lot of stochastic variability (more than plus or minus 10 units), or continually increase with each iteration even though the population parameters are kept fixed. One way to reduce this volatility is to use IMPMAP instead of IMP, if the MAP estimation is not an issue:
$EST METHOD=IMPMAP EONLY=1 ISAMPLE=1000 NITER=5 MAPITER=0

Another way is to increase the ISAMPLE to 3000:
$EST METHOD=IMP EONLY=1 ISAMPLE=3000 NITER=5 MAPITER=0

and sometimes, using the combination of IMPMAP with ISAMPLE=3000 is needed. Using IMPMAP or increasing ISAMPLE do increase computation time, and it is a choice of which is more efficient.

As of NM74, another choice is to set EONLY=2, and the Monte Carlo variability of the objective function can be significantly reduced if, like the population parameters, the conditional means and variances from the SAEM estimation are also not updated after each IMP iteration:

$EST METHOD=SAEM INTERACTION NBURN=2000 NITER=1000
$EST METHOD=IMP EONLY=2 ISAMPLE=1000 NITER=5 MAPITER=0

Or, have the first iteration evaluate conditional modes and conditional MAP variances, and then use them for the subsequent iterations, by setting EONLY=3 and MAPITER=1:

$EST METHOD=SAEM INTERACTION NBURN=2000 NITER=1000
$EST METHOD=IMP EONLY=3 ISAMPLE=1000 NITER=5 MAPITER=1

Another set of commands for SAEM is the following, which begins with a short iterative two stage run to provide good initial eta estimates for each subject, followed by the SAEM analysis, which uses these initial eta estimates as a starting point for its Markov Chain Monte Carlo scan of each subject's conditional (posterior) density, followed by objective function evaluation:

$EST METHOD=ITS INTERACTION NITER=5
$EST METHOD=SAEM NBURN=1000 ISAMPLE=2 NITER=1000
$EST METHOD=IMP EONLY=2 ISAMPLE=1000 NITER=5 MAPITER=0

Values of NBURN, NITER, and ISAMPLE may be changed as needed.

If you want conditional mean values (values listed in root.phi) evaluated by MCMC sampling used in the SAEM method, but at a constant set of the final fixed parameters, then you could invoke EONLY=1 with the SAEM method as well:

$EST METHOD=ITS INTERACTION NITER=5
$EST METHOD=SAEM NBURN=1000 ISAMPLE=2 NITER=1000
$EST METHOD=SAEM EONLY=1 NBURN=200 ISAMPLE=2 NITER=1000
$EST METHOD=IMP EONLY=2 ISAMPLE=1000 NITER=5 MAPITER=0

ETASAMPLES=0 (default) (nm74)

As of NM74, one can obtain random samples of individual etas, while keeping population parameters fixed, and uses these for covariate and model diagnostics (see [21]). To do this, perform an SAEM analysis (or BAYES) after the primary estimation, keeping the population parameters fixed (ETYPE=1), only performing a BURN period so samples do not get accumulatively averaged, and set ISAMPLE to 10 or higher to collect suffient samples per subject. For example, note the $EST line in bold, after the usual SAEM analysis:

```
$SIZES ISAMPLEMAX=250
…
$EST METHOD=SAEM NBURN=500 NITER=1000 MASSRESET=1 ISAMPLE=2
     PRINT=20 NOPRIOR=1 RANMETHOD=P
$EST METHOD=SAEM NBURN=0 NITER=0 MASSRESET=0 ETASAMPLES=1 ISAMPLE=200 EONLY=1
$EST METHOD=IMP NITER=5 MASSRESET=1 ETASAMPLES=0 ISAMPLE=1000
     EONLY=1 PRINT=1 MAPITER=0
```

ETASAMPLES=1 causes individual ISAMPLE random eta samples per subject, to be written to *root*.ets, where *root* is the root name of the control stream file. MASSRESET is set to 1 to initialize the internal burn-in coefficients, and collect that information during the population parameter estimation. Then, have the next SAEM step accumulate eta samples (ETASAMPLES=1), and set MASSRESET=0 so the internal burn-in coefficeints are note reset, but use those obtained during the previous SAEM step. After this, you could request the IMP step (IMP needs to use its own burn-in or shaping coefficients, so re-initialize with MASSRESET=1).

Suppose your result was from a previous analysis, called precevample.ctl. You could do the following for the present problem:

```
; Obtain final result of previous example from its ext file
$EST METHOD=CHAIN NSAMPLE=0 TBLN=1 ISAMPLE=-1000000000 FILE=prevexample.ext
; Have a burn-in process with just 2 chains (ISAMPLE=2), to create appropriate
; burn-in coefficients
$EST METHOD=SAEM NBURN=0 NITER=0 MASSRESET=1 ISAMPLE=2 EONLY=1 RANMETHOD=P
; Use burned-in coefficients (MASSRESET=0) and collect many samples, and store
in .ets file.
$EST METHOD=SAEM NBURN=0 NITER=0 MASSRESET=0 ETASAMPLES=1 ISAMPLE=200 EONLY=1
```

The ETASAMPLES option of collection of individual parameter samples is comparable to the BIONLY option for BAYES method (*BIONLY=0 (default) (NM75)* in section I.39 Full Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method).

I.39 Full Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method

The goal of the MCMC Bayesian analysis [12,13] is not to obtain the most likely thetas, sigmas, and omegas, but to obtain a large sample set of probable population parameters, usually 10000-30000. The samples are not statistically independent, but when analysis is properly performed, they are uncorrelated overall. Various summary statistics of the population parameters may then be obtained, such as means, standard deviations, and even confidence (or credible) ranges. The mean population parameter estimates and their variances are evaluated with considerable stability. Maximum likelihood parameters are not obtained, but with problems of sufficient data, these sample mean parameters are similar to maximum likelihood values, and the standard deviations of the samples are similar to standard errors obtained with maximum likelihood methods. A maximum likelihood objective function is also not obtained, but, a distribution of joint probability densities is obtained, from which 95% confidence bounds (assuming a type I error of 0.05 is desired) can be constructed and tested for overlap with those of alternative models.

As with the SAEM method, there are two phases to the BAYES analysis. The first phase is the burn-in mode, during which population parameters and likelihood may change in a very directional manner with each iteration, and which should not be used for obtaining statistical summaries. The second phase is the stationary distribution phase, during which the likelihood and parameters tend to vary randomly with each iteration, without changing on average. It is these samples that are used to obtain summary statistics.

The Bayesian method is specified by

$EST METHOD=BAYES INTERACTION

Followed by one or more of the following parameter options:

NBURN=4000

Maximum number of iterations in which to perform the burn-in phase of the MCMC Bayesian method (default 4000). During this time, the advance of the parameters may be monitored by observing the results in file specified by the FILE parameter, and/or the objective function displayed at the console. The objective function progress is also written in OFV.TXT, and the report file. Full sets of population parameters and likelihood functions are also written in the file specified with the FILE= option. When all parameters and objective function do not appear to drift in a specific direction, but appear to bounce around in a stationary region, then it has sufficiently "burned" in. A termination test may be implemented to perform a statistical assessment of stationarity for the objective function and parameters. As mentioned earlier, the objective function (MCMCOBJ) that is displayed during BAYES analysis is not valid for assessing minimization or for hypothesis testing in the usual manner. It does not represent a

likelihood that is integrated over all possible eta (marginal density), but the likelihood at a given set of etas.

NSAMPLE/NITER=10000
Sets number of iterations in which to perform the stationary distribution for the BAYES analysis (default 10000).

ISAMPLE_M1=2 (defaults listed)

ISAMPLE_M1A=0 (NM72)

ISAMPLE_M1B=2 (NM74)

ISAMPLE_M2=2

ISAMPLE_M3=2

IACCEPT=0.4

These are options for the MCMC Bayesian Metropolis-Hastings algorithm for individual parameters (ETAS) used by the SAEM and BAYES methods. For Bayesian analysis, the MCMC algorithm performs *ISAMPLE_M1* mode 1 iterations using the population means and variances as proposal density, followed by ISAMPLE_M1B mode 1B iterations using the individual conditional mean and individual conditional variance collected from previous iterations as proposal density, followed by ISAMPLE_M1A mode 1A iterations, testing model parameters from other subjects as possible values (by default this is not used, ISAMPLE_M1A=0), followed by *ISAMPLE_M2* mode 2 iterations, using the present parameter vector position as mean, and a scaled variance of OMEGA as variance [11]. Next, ISAMPLE_M3 mode 3 iterations are performed, in which samples are generated for each parameter separately. The scaling is adjusted so that samples are accepted IACCEPT fraction of the time. The final sample for a given chain is then kept. Usually, these options need not be changed. There is only one chain of samples produced for a given NONMEM run (ISAMPLE is not used for MCMC, only for SAEM). If you would like additional chains, then create separate control stream files with different starting seed numbers.
As of NM75, if the option MCETA is set to greater than ISAMPLE_M1, then for the first iteration of SAEM, ISAMPLE_M1, ISAMPLE_M2, and ISAMPLE_M3 will be set to MCETA, to facilitate a robust initial search for reasonable parameter values.

ISCALE_MIN=1.0E-06 (defaults for SAEM, BAYES, NM72)

ISCALE_MAX=1.0E+06 (NM72)
In MCMC sampling, the scale factor used to vary the size of the variance of the proposal density in order to meet the IACCEPT condition, is by default bounded by ISCALE_MIN of 1.0E-06, and ISCALE_MAX=1.0E+06. This should left alone for MCMC sampling, but on occasion there may be a reason to reduce the boundaries (perhaps to ISCALE_MIN=0.001,

ISAMPLE_MAX=1000). After the SAEM estimation method, remember to revert these parameters back to default operation on the next $EST step:
ISCALE_MIN=-100 ISCALE_MAX=-100
The default operation is that NONMEM sets (ISCALE_MIN,ISCALE_MAX) to (0.01,100) for importance sampling (as described earlier), and to (1.0E-06,1.0E+06) for MCMC sampling.

IKAPPA=1(default)

The individual parameters are averaged using a weight $N^{-IKAPPA}$ for the Nth iteration (so a simple average with each iteration's value equally weighted would be IKAPPA=1), in obtaining the mean and variance-covariance for the ISAMPLE_M1B mode. A value of 0.75 can sometimes provide an improved decorrelation efficiency when performing standard Bayesian analysis.

PSAMPLE_M1=1 (defaults listed)

PSAMPLE_M2=-1

PSAMPLE_M3=1

PACCEPT=0.5

These are the options for the MCMC Metropolis-Hastings algorithm. These options only have meaning for population parameters (theta/sigma) that are not Gibbs sampled. Normally NONMEM determines whether THETA and SIGMA parameters are Gibbs sampled or not, based on the model setup (see MU_ Referencing section below). For each iteration, a vector of thetas/sigmas are generated using a multivariate normal proposal density that has mean/variances based on the previous samples, done *PSAMPLE_M1* times. Next, a vector of parameters are generated using a multivariate normal proposal density with mean at the present parameter position, and variance scaled to have samples accepted with PACCEPT frequency. This is done *PSAMPLE_M2* times (if *PSAMPLE_M2*<0, then program performs this as many times as there are M-H parameters). Finally, each parameter is individually sampled PSAMPLE_M3 times. The final accepted parameter vector is kept. Usually these options do not need to be changed from their default values, listed above.

PSCALE_MIN=0.01 (NM73)

PSCALE_MAX=1000 (NM73)

In MCMC sampling, the scale factor used to vary the size of the variance of the proposal density population parameters (theta/sigma) that are not Gibbs sampled, in order to meet the PACCEPT condition, is by default bounded by PSCALE_MIN of 0.01, and PSCALE_MAX=1000. This should left alone for MCMC sampling, but on occasion there may be a reason to expand the boundaries (perhaps to PSCALE_MIN=1.0e-06, PSCALS_MAX=1.0E+06).

OSAMPLE_M1=-1 (defaults listed)

OSAMPLE_M2=-1

OSAMPLE_M3=-1

OACCEPT=0.5

These are the options for the MCMC Metropolis-Hastings algorithm for OMEGA sampling. If OSAMPLE_M1<0 (default), then the OMEGA's are Gibbs sampled using the appropriate Wishart proposal density, and the other options (OSAMPLE_M2 and OACCEPT) are not relevant. Otherwise, for each iteration, a matrix of OMEGAs are generated using a Wishart proposal density that has variance based on the previous samples, done *OSAMPLE_M1* times. Next, a matrix of OMEGAS are generated using a Wishart proposal density at the present OMEGA values postion, and degrees of freedom (dispersion factor for variances) scaled to have samples accepted with OACCEPT frequency. This is done *OSAMPLE_M2* times (if *OSAMPLE_M2*<0, then program performs this as many times as there are non-fixed omega elements). Then, individual cholesky elements of OMEGA are varied, each OSAMPLE_M3 times (if OSAMPLE_M3<0, then program performs this as many times as there are non-fixed omega elements). The final OMEGA matrix is kept. Usually these options do not need to be changed from their default values, listed above.

NOPRIOR=[0,1]

If prior information was specified using the $PRIOR statement (available since NM 6, release 2.0, and described in the Help manual ..\guides\VIII.pdf: use only NWPRI option for the new $EST methods), then normally the analysis is set up for three stage hierarchical analysis. By default NOPRIOR=0, and this prior information will be used. However, if NOPRIOR=1, then for the particular estimation, the prior information is not included in the analysis. This is useful if you want to not use prior information during a maximization (METHOD=IMP, CONDITIONAL, IMPMAP, SAEM, or ITS), but then use it for the Bayesian analysis (METHOD=BAYES).

As of NM73, when NOPRIOR=1 is set, the estimation will not use TNPRI prior information (TNPRI should only be used with FO/FOCE/Laplace estimations). In previous versions of NONMEM, NOPRIOR=1 did not act on TNPRI priors.

MAPITERS=0 (default) (NM75)

As with SAEM, by default, no MAP estimation is performed with BAYES. To get good individual parameter values near the mode of the posterior density for the first iteration of BAYES, you can set MAPITERS=1. Alternatively, you can insert the record:
$EST METHOD=ITS NITER=0
Followerd by
$EST METHOD=BAYES …

THIN=1 (default,NM74)

As of nm74, the Bayesian records retained in the raw output file may be adjusted by every THINth iteration. So, if THIN=10, then every 10th iteration is recorded in the raw output file. The PRINT option controls only the iterations printed to the console and NONMEM report file.

BAYES_PHI_STORE=0 (default) (NM75)

As of nm75, samples of phi/eta are collected at each BAYES iteration, and summarized to provide mean phi and phc() in the root.phi table, as described above. By default, the individual phi values from each iteration are not stored. However, if you set

```
$EST … BAYES_PHI_STORE=1
```

then phi and eta values from each BAYES iteration will be stored in root.iph. For non-mixture problems, only records of SUBP=0 are recorded, as there are no sub-population divisions. For mixture problems, the SUBP=0 records contain the composite phis and etas (the average of these across all non-negative iterations are in the root.phi table), and the SUBP>0 records contain the phis and etas appropriate to each sub-population SUBP (the average of these across all non-negative iterations are in the root.phm table). The root.iph file can become quite large, so it should be used only on the final analysis. This root.iph file can be read by the utility program NEFFI (see I.98 NEFF and NEFFI Utility Programs (NM74)).

BIONLY=0 (default) (NM75)

BIONLY stands for Bayesian individual parameters only, and when set to 1, will create new samples of individual parameters only, but will keep the population parameters fixed. This is equivalent to EONLY=1 used for SAEM and IMP estimation. Of course, for this to be useful, either write statements capturing the individual parameters samples in the control stream must be entered (see example 8 near end of this manual), or the BAYES_PHI_STORE option must be set to 1, to collect individual samples in the .iph file.

Here is an example showing its use:

```
$EST METHOD=BAYES INTERACTION NBURN=200
     NITER=1000 PRINT=100 NOPRIOR=1 CINTERVAL=100 BIONLY=1
     BAYES_PHI_STORE=1
```

The BIONLY option of BAYES, with collection of individual samples in the .iph file, is comparable to the ETASAMPLES option of SAEM, with collection of individual samples in the .ets file (*ETASAMPLES=0 (default) (nm74)* in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method).

I.40 No U-Turn Sampling (NUTS) Markov Chain Monte Carlo (MCMC) Bayesian Analysis Method (NM74)

The No U-Turn sampling algorithm was developed by Hoffman, Gelman, and others of the STAN Development Team ([19,20]). The algorithm developed in NONMEM is a limited form of the No-U-Turn MCMC Bayesian environment that is available in STAN, focused on analyzing population PK/PD models, with normally distributed THETA priors (or t-distributed

theta priors, see TTDF), and Wishart/Gamma distributed OMEGA and SIGMA priors, or using the LKJ correlation prior (see OLKJDF and SLKJDF options) for OMEGAS and SIGMAS. The algorithm in NONMEM is based on references [19,20], and helpful suggestions by Bob Carpenter, Andrew Gelman, Matt Hoffman, Michael Betancourt, and Sebastian Weber.

Typical Bayesian algorithms search for individual parameters (phis or etas) and population parameters (thetas, omegas, and sigmas) in separate stages. While this provides for rapid generation of an MCMC sample, the samples can be heavily correlated, especially if there are high correlations in the OMEGA matrix. The NUTS sampler uses a directed search using partial derivatives and scaling techniques using posterior density knowledge from previous samples to reduce the correlation of the parameters from one iteration to the next. While each iteration takes longer to generate with NUTS, the samples may be 10-100 times decorrelated relative to a standard MCMC sampling. Thus, 10000 samples with NUTS may be worth 100000 samples with traditional MCMC algorithms. See the NEFF (Number of EFFective samples) utility for analyzing the quality of an MCMC run, I.98 NEFF and NEFFI Utility Programs (NM74).

Because the NUTS algorithm relies on derivatives, it is best if analytical derivaties are created for each of the estimated parameters. For OMEGAS and SIGMA's, these are done automatically, but for thetas, analytical derivatives are created only if they are MU referenced (see MU Reference section below). So MU reference all thetas that are to be estimated. It is okay to set their OMEGA to 0, the analytical derivative will still be utilized.

The easiest way to use the NUTS algorithm in NONMEM is to use the AUTO option (see section I.43 Some General Options and Notes Regarding EM and Monte Carlo Methods for more details on the AUTO feature, *AUTO=0 (default) (NM73)*). For example, stanrb42.ctl uses AUTO=1:

```
$EST METHOD=NUTS AUTO=1 PRINT=20
```

The example stanrb10.ctl shows another setup for using NUTS:

```
$EST METHOD=ITS NITER=0 file=stanrb10_its.ext
$EST METHOD=NUTS NBURN=1000 NITER=2000 PRINT=20 OLKJDF=3.0 file=stanrb10.ext
```

The OLKJDF option specifies the degrees of freedom to the LKJ decorrelation density for the OMEGA prior. The OLKJDF should be set to a value greater than 0 to use the LKJ decorrelation prior for Omegas, as recommended by the STAN group. Also see comment below about using LKJ decorrelation versus inverse Wishart prior. Experience has shown that a low non- or weakly-informative OLKJDF should be at least 2, and no greater than the number of the Omega diagonals. If OLKJDF is set to 1, then there may be considerable pull towards high correlations which reduces efficiency of sampling. A single iteration of ITS is helpful to center the initial etas at their modes, as a good facilitator for initiating the NUTS run that follows it (NITER=0, so population parameters are not advanced).

An alternative method is to use the traditional BAYES method to rapidly generate samples for an initial mass matrix, which can then be passed on to the NUTS algorithm (example stanrb9):

```
$EST METHOD=BAYES NBURN=2000 NITER=2000 PRINT=50 MASSRESET=1
```

```
file=stanrb9_bayes.ext KAPPA=0.75
$EST METHOD=NUTS NBURN=500 NITER=2000 PRINT=20 file=stanrb9.ext
     OLKJDF=3.0 MASSRESET=0 KAPPA=1.0 MADAPT=250
```

Notice that the MASSRESET is set to 1 to initialize the mass matrix accumulator at the BAYES step, and then set MASSRESET=0 at the NUTS step, so that the mass matrix is not re-initialized, but rather, carried over, from the BAYES step. Also, you may wish to set KAPPA=0.75 during BAYES so that the accumulated mass matrix favors values collected during the latter portion of the BAYES analysis (for the NUTS step itself, KAPPA should be set back to 1). This technique can sometimes make the burn-in (warm-up) period for NUTS execute faster, and/or improve the de-correlation. MADAPT is set to ½ of NBURN, during which the mass matrix is continuously updated every NUTS_BASE iterations (which in this case by default is 0.025*NBURN=25).

Other stanrb* examples in the examples directory show the various ways in which the problem may be analyzed.

Several versions of a differential equation problem, example6, are also in the examples directory, example6hmt*, to be compared with example6classic2. and example6classico3. Comparing example6hmto26.ctl using pure NUTS algorithm with example6hmto19 which uses a pre-warmup from a previous BAYES estimation, you can see that the pre-warmup can reduce time of computation.

The following is a list of options the user may play with.

METHOD=NUTS

This sets the Bayesian analysis for the NUTS method.

NUTS_MASS=B (default)

By default, NUTS_MASS=B, the NUTS method is to use a block diagonal mass matrix for scaling its search. The Thetas/sigmas/omegas and their correlations will be scaled with one block matrix, and parameter sets of each individual will have their own block matrix correlations. Correlations between thetas and individual parameters will not be accounted for. Most efficient.

F

Full mass matrix. A full correlation matrix between thetas, omegas, sigmas, and individual parameters among all individuals will be accounted for. Computationally very expensive. When using METHOD=BAYES as a preparation for NUTS, as shown in example stanrb_177.ctl, make sure you set the NUTS_MASS (if you will be using something different from the default B value) at the METHOD=BAYES record, as well as MASSRESET=1, as this is required to set the appropriate memory allocation, and store the posterior variance-covariance (mass matrix) information that the NUTS algorithm will then use.

D

Diagonal mass matrix. No correlations between parameters will be considered.

BD
Block mass matrix covering Thetas, Sigmas, Omegas, and diagonal mass matrix on individual parameters.

DB
Diagonal mass matrix covering Thetas, Sigmas, Omegas, and block mass matrix on individual parameters.

BBD
Thetas and Sigmas are blocked together, Omegas are in their own block, and individual parameters are diagonal.

BBB
Thetas and Sigmas are blocked together, Omegas are in their own block, and individual parameters are blocked within each subject.

The mass matrix is generated by accumulating previous samples of parameters and obtaining their variance-covariance, so that the NUTS algorithm performs an efficient search in the domain of the empirical posterior density. It is best to acquire this mass matrix by first perfoming a couple of thousand iterations using standard Gibbs Bayesian analysis, followed by a NUTS process.

MASSRESET=-1 (default)

By default mass matrix information accumulation is turned off. However, MASSRESET=1 should be set when performing a Gibbs/MH Baysian analysis to initialize accumulation of mass matrix information, followed by the NUTS algorithm, with MASSRESET=0 set, so the mass matrix accumulator information is not reset to 0, but adds to the information acquired during the previous Gibbs/MH process. IF you have a BAYES record setup previous to the NUTS record, make sure to set the NUTS_MASS value at the BAYES record first, so it accumulates the correct type of mass matrix information, and allocates the appropriate memory for its storage.

NUTS_MAXDEPTH=10 (default)

This sets the maximum number of total branchings to try in the NUTS algorithm in the search for the next decorrelated sample. If many messages are received of reaching the maximum buildtree level, increase NUTS_MAXDEPTH.

MADAPT=-1 (default)

If MADAPT/=-1 (also called PMADAPT), the mass matrix is updated throughout the NUTS analysis every NUTS_BASE iterations, for the first MADAPT iterations for the parameters, then changes no further after that. If MADAPT=-1, then the STAN method of warmup and mass matrix accumulation is used (according to the STAN manual [20], Optimization Algorithms section). When using MADAPT=-1, the tuning options, NUTS_INIT, NUTS_BASE, and NUTS_TERM are useful.

KAPPA=1(default)

The parameters are averaged using a weight N^{-KAPPA} (also called PKAPPA) for the Nth iteration (so a simple average with each iteration's value equally weighted would be KAPPA=1), in obtaining the mass matrix. A value of 0.75 gives the best results when preparing a mass matrix during the BAYES step, in anticipation of the NUTS step.

NUTS_GAMMA=0.05(default)

Gamma factor in the NUTS algorithm. Should not be changed, and NUTS experts (the Stan developers) recommend 0.05

NUTS_DELTA=0.8(default)

This is essentially the sample acceptance rate for the NUTS sampling process, equivalent to PACCEPT in standard MH sampling. NUTS experts recommend 0.8.

TTDF=0(default)

TTDF stands for Theta t-density degrees of freedom. When 0, the usual normal density prior is used as a prior density for thetas. When TTDF>0, then a t-distributed prior is used. TTDF may be set >0 when using METHOD=BAYES as well, but thetas will then be M-H sampled using the PSAMPLE_M1, PSAMPLE_M2, and PSAMPLE_M3 settings. TTDF may be a real number. See record $TTDF below to specify degrees of freedom for each theta.

TPU=0(default,NM75)

This is a $EST option that specifies, if >0, to use the THETA_PRIORU routine in ..\source\THETA_PRIORU.f90 as a template to provide the desired probability density function. To do this, you can make a copy of THETA_PRIORU.f90, call the file USERPRIORT.f90, for example, modify its THETA_PRIORU subroutine, then specify this file as an OTHER routine:
$SUBR … OTHER=USERPRIORT.f90

OLKJDF=0(default)

OLKJDF stands for Omega LKJ density degrees of freedom. When 0, the usual inverse Wishart prior is used for Omegas. When OLKJDF>0, then the LKJ density is used as the prior, with OLKJDF degrees of freedom for all omega blocks. In addition, only diagonal elements of the OMEGA prior are used, assuming a density dependent on the OVARF value. OLKJDF may be set >0 when using METHOD=BAYES as well, but Omegas will then be M-H sampled using the OSAMPLE_M1, OSAMPLE_M2, and OSAMPLE_M3 settings. See record $OLKJDF below to specify LKJ correlation degrees of freedom for each omega block.

OVARF=1(default)

OVARF is the weight factor to STD prior to the log sqrt OMEGA diagonal elements, the normal density of the log square root of OMEGA centered about log square root of Omega prior, and scaled with OVARF (see below). That is,
log(sqrt(Omega(i))) ~ Normal(log(sqrt(OmegaPrior(i))),1/OVARF).

If OVARF<0, then a half-t-distribution of degrees of ABS(OVARF) is used as the prior to the sqrt of OMEGA diagonal elements. Use OVARF=-1 for the half-Cauchy distribution.

SLKJDF=0(default)

SLKJDF stands for Sigma LKJ density degrees of freedom. When 0, the usual inverse Wishart prior is used for Sigmas. When SLKJDF>0, then the LKJ density is used as the prior, with SLKJDF degrees of freedom. In addition, only diagonal elements of the Sigma prior are used. SLKJDF may be set >0 when using METHOD=BAYES as well, but Sigmas (in cholesky format) will then be M-H sampled using the PSAMPLE_M1, PSAMPLE_M2, and PSAMPLE_M3 settings (choleskys of sigma elements are treated as extensions of the THETA parameters in M-H sampling methods). See record $SLKJDF below to specify LKJ correlation degrees of freedom for each sigma block.

SVARF=1(default)

SVARF is the weight factor to STD prior to the log sqrt Sigma diagonal elements, the normal density of the log square root of Sigma centered about log square root of Sigma prior, and scaled with SVARF (see below). That is,
log(sqrt(Sigma(i))) ~ Normal(log(sqrt(SigmaPrior(i))),1/SVARF).

If SVARF<0, then a half-t-distribution of degrees of ABS(SVARF) is used as the prior to the sqrt of SIGMA diagonal elements. Use SVARF=-1 for the half-Cauchy distribution.

$OLKJDF (0 default, NM75)

The $OLKJDF is a separate record that allows the user to specify LKJ decorrelation degrees of freedom for each OMEGA block. For example:

$OLKJDF 4.5 3.5 -2.0
Where 4.5 degrees of freedom are specified for the first omega block, 3.5 for the second, and -2.0 specifies 2 degrees of freedom for the third omega block, but a user-defined definition of the standard deviations of the diagonals for the third omega block. For any blocks without degrees of freedom defined, it is 0. Use the OMEGA_STD_PRIORU.f90 file in the ..\source directory as a template to modify the OMEGA_STD_PRIORU to provide the desired probability density function. Set $SUBR OTHER=OMEGA_STD_PRIORU.f90 to use the user modified template (you may rename the file for organizational purposes). $EST OLKJDF over-rides $OLKJDF.

$OVARF 0(default, NM75)

The OVARF is a separate record that allows the user to specify the weighting (inverse variance) to the standard deviations LKJ decorrelation degrees of freedom for each OMEGA block. For example:

$OVARF 2.0 5.0

Where 2.0 is specified for the first omega block, 5.0 for the second. If the corresponding $OLKJDF value is negative then this is argument STDSSP in user-defined OMEGA_STD_PRIORU.f90. $EST OVARF over-rides $OVARF.

$SLKJDF 0(default)

The SLKJDF is a separate record that allows the user to specify LKJ decorrelation degrees of freedom for each SIGMA block. For example:

$SLKJDF 4.5 3.5 -2.0

Where 4.5 degrees of freedom are specified for the first sigma block, 3.5 for the second, and -2.0 specifies 2 degrees of freedom for the third sigma block, but a user-defined definition of the standard deviations of the diagonals for the third sigma block. For any blocks without degrees of freedom defined, it is 0. Use the SIGMA_STD_PRIORU.f90 file in the ..\source directory as a template to modify the SIGMA_STD_PRIORU to provide the desired probability density function. Set $SUBR OTHER=SIGMA_STD_PRIORU.f90 to use the user modified template. $EST SLKJDF over-rides $SLKJDF.

$SVARF 0(default,NM75)

The SVARF is a separate record that allows the user to specify the weighting (inverse variance) to the standard deviations LKJ decorrelation degrees of freedom for each SIGMA block. For example:

$SVARF 2.0 5.0

Where 2.0 is specified for the first sigma block, 5.0 for the second. If the corresponding $SLKJDF value is negative then this is argument STDSSP in the user-defined SIGMA_STD_PRIORU.f90. $EST SVARF over-rides $SVARF.

$TTDF 0 (default,NM75)

The ttdf is a separate record that allows the user to specify the t-distribution degrees of freedom for each theta. So,

$TTDF (2.0)x2 (3.0)x2

sets t-distribution degrees of freedom for thetas 1 and 2 to value of 2, and DF for thetas 3 and 4 are set to 3.0. By default $TTDF value is 0, for any thetas whose degrees of freedom is not specified. These will be used in estimation (integer or real) and simulation (truncated integer only) unless $EST TTDF or the $SIM TTDF is set, respectively, which set the t degrees of freedom to a single TTDF value for all thetas.

NUTS_TRANSFORM=0(default)

When NUTS_TRANSFORM=0, model parameters are transformed using the mass matrix, for population parameters. If NUTS_TRANSFORM=1, the momentum parameters are transformed using the mass matrix. It is best to set NUTS_TRANSFORM to 1 if NUTS_TEST is set to 1, and NUTS_TRANSFORM should be set to 0 when NUTS_TEST is set to 0.

NUTS_EPARAM=0 (default)

When NUTS_EPARAM=0, parameters are entered into the NUTS algorithm parameterized as Thetas and phis, Cholesky. When NUTS_EPARAM=1, parameters are entered into the NUTS algorithm parameterized as Thetas and etas. When NUTS_EPARAM=2, parameters are entered into the NUTS algorithm parameterized as Thetas and Choleksy of Omega inverse*eta.

NUTS_OPARAM=1 (default)

When NUTS_OPARAM=1, Omega elements are parameterized in a correlation cholesky format that constrains correlations to be between -1 and 1. When NUTS_OPARAM=0, then the full Omega elements are parameterized in cholesky format.

NUTS_SPARAM=1 (default)

When NUTS_SPARAM=1, Sigma elements are parameterized in a correlation cholesky format that constrains correlations to be between -1 and 1. When NUTS_SPARAM=0, then the full Sigma elements are parameterized in cholesky format.

NUTS_REG=0.0 (default)

By default, the mass matrix is made slightly diagonal dominant by adding a fraction of the diagonal element. If NUTS_REG>0.0, then the mass matrix is made slightly diagonal dominant by adding the value of NUTS_REG. When OLKJDF>0, then NUTS_REG=1.0 may provide a more efficient sampling process.

NUTS_STEPITER=1(default)

An initial step size is calculated for the first NUTS_STEPITER iterations.

NUTS_STEPINTER=0(default)

An initial step size is calculated every NUTS_STEPINTER iterations.

NUTS_TEST=0(default)

The acceptance of a sample is tested using an algorithm as originally in algorithm 6 of Hoffman and Gelman [19] (default, NUTSTEST=0), or as performed in STAN (NUTSTEST=1)

NUTS_INIT=0.075 (default)

When using the STAN algorithm (MADAPT=-1) for mass matrix and step size development during the burn-in (warmup) stage, when NUTS_INIT<1 serves as the fraction of NBURN iterations for Stage I of the warmup period ([20]). When NUTS_INITS>1, then the explicit number of iterations is interpreted. Similarly, when MADAPT>0, this period is also used to accumulate NUTS_INIT*NBURN iterations before using the mass matrix.

NUTS_BASE=0.025 (default)

When using the STAN algorithm (MADAPT=-1) for mass matrix and step size development during the burnin-in (warmup) stage, NUTS_BASE (if NUTS_BASE>=1) or NUTS_BASE*NBURN (if NUTSBASE<1) serves as the number of iterations for the first Stage II of the warmup period ([20]), and doubles in iteration number with each subsequent segment of Stage II. The total number of stage II iterations is bounded by NBURN-NUTSINIT-NUTS_TERM. If NUTS_INIT=75, NUTS_BASE=25, NUTS_TERM=150, and NBURN=1000, then you have 5 segments of stage II, so that

NUTS_INIT+NUTS_TERM+NUTS_BASE+NUTS_BASE*2+NUTS_BASE*4=NBURN=
75 + 150 +25 +25*2+25*4+25*8+25*16=1000

When MADAPT>0, this period is also used to update the mass matrix every NUTSBASE iterations until MADAPT total iterations have been prformed.

If NUTS_BASE<=-1.0, then NUTS_BASE will be set to the largest block section of the mass matrix plus 10. This assures that a large enough base set of samples are collected before the mass matrix is used.

If NUTS_BASE<-1, then in addition, the number of stage II iterations is ABS(NUTS_BASE). The actual NBURN will be based on the above equation, but not to exceed the user specified NBURN, which serves as the max NBURN. With NUTS_BASE<-1.0, set NBURN to a large number (4000 or so). The AUTO feature set NUTS_BASE to -3.

NUTS_TERM=0.05 (default)

When using the STAN algorithm (MADAPT=-1) for mass matrix and step size development during the burn-in (warmup) stage, NUTS_TERM serves as the number of iterations for Stage III of the warmup period ([20]), to make final adjustments in step size.

MUFIRSTREC, OBJQUICK

For simple problems, the NO U-Turn process can have excessive overhead and run slowly. To have it run faster, you can do the following:

Set MUFIRSTREC=1 in $PRED or $PK. MUFIRSTREC=1 selects the covariate of the first record of the subject, rather than averaging among its records when using that covariate in a MU reference.

OBJQUICK=0.
Standard NONMEM processing of the model occurs.

OBJQUICK=1
Certain tests and initializations are skipped.

OBJQUICK=2.
A simplified modeling process occurs, but which cannot be used when $LEVEL or $MIX is used in the model. Also, parallelization is not performed.

Usage:

```
$PRED
include nonmem_reserved_general
MUFIRSTREC=1
OBJQUICK=1
```

The OBJQUICK and MUFIRSTREC can also speed up the other analysis methods, such as ordinary BAYES, FAST FOCE, ITS, and the EM methods.

Note on Combinations of Option Settings NUTS_MASS, NUTS_EPARAM, NUTS_OPARAM, NUTS_SPARAM

Certain combinations of option settings work well, others do not.

Default:
NUTS_EPARAM=0 NUTS_MASS=B
These settings are the most efficient for many of the problems tested so far. They offer the greatest speed efficiency and sampling (Neff/Nsample) efficiency. On occasion, one or two thetas will have low efficiencies relative to the rest. The AUTO=1 option allows an easy setup of this configuration (see section I.43 Some General Options and Notes Regarding EM and Monte Carlo Methods for more details on the AUTO feature, *AUTO=0 (default) (NM73)*). Example stanrb42.ctl uses the AUTO=1 feature.

NUTS_EPARAM=1 NUTS_MASS=D
When the problem is submitted to the NUTS algorithms with etas rather than phis (NUTS_REPARAM=1), the NUTS_MASS=B does not yield good efficiencies on thetas. Therefore, NUTS_MASS=D needs to be used. However, this reduces speed efficiencies by about 5 fold, but evens out the theta efficiencies so the lowest Neff/Nsample efficiency is about 3x higher than the lowest sampling efficiency of the NUTS_EPARAM=0 setting. The AUTO=3 option allows an easy setup of this configuration.

NUTS_EPARAM=2 NUTS_MASS=BD
With NUTS_EPEARM=2, this is called the “Matt trick” in the Stan community, and offers very high sampling efficiencies of 3 to 5 fold than that of the default settings. However, the speed is

about 4-8 times slower from the default settings when performing NUTS in NONMEM, so there may or may not be greater overall efficiency, in terms of number of independent samples per unit time. Furthermore, this method uses a conditional likelihood equation that differs from the standard that the population analysis community is used to: It drops the NIND*LOG(DET(OMEGA)) term (where NIND=number of subjects), resulting in a conditional likelihood that has very different distribution properties and does not fully represent the contribution of the individual etas to the likelihood. An example of using NUTS_EPARAM=2 is ..\examples\stanrb19. The AUTO=2 option allows an easy setup of this configuration, and example ..\examples\stanbrb39 uses this option.

The Stan community also supports the notion of using LKJ correlation priors for supporting Omegas, rather than inverse Wishart priors. This is reasonable when using LKJ decorrelation as an uninformative prior, and there is no previous knowledge of the scale of the variances. For example, one could use Identity matrix for the Omega priors (diagonal OMEGA values=1, and very low degrees of freedom, and the LKJ decorrelation prior will not introduce much bias, whereas the inverse Wishart prior would introduce considerable bias. However, the uninformative inverse Wishart prior is reasonable to use if the diagonal Omega values are in a reasonable range of where the data are. In PK/PD modeling, we have the benefit of obtaining reasonable variances from first performing a maximum likelihood analysis, (using FOCE, ITS, IMP, ITS, or SAEM), which often do not require any priors, and then supplying these results as priors for the NUTS analysis, as long as the degrees of freedom is set to <=D, the dimension of the block matrix. In such cases, the Inverse Wishart as an uninformative prior (DF<=D), but with the variances obtained from an earlier maximum likelihood analysis on the same data, is equivalent to LKJ correlation prior in terms of quality and lack of bias.

When using Omega information from a previous study to supply as an informative prior to a present study, the Inverse Wishart format of the prior information conjugates well with the information in the cross products of etas provided by the present study, and the informative prior information from the previous study offers a natural statistical support, as if the data of the previous study were added to the present study. Furthermore, the inverse Wishart prior supplies the information of the entire block, off-diagonals and diagonals, whereas the LKJ correlation prior method (OLKJDF>0) only uses the diagonal elements, and some general notion of correlation in the OLKJDF value. Such a natural interpretation for the inverse Wishart is evident in the mathematical structure of the total likelihood, when using NUTS_EPARAM=0 or NUTS_EPARAM=1, and Omega priors with inverse wishart distribution.

Prepare a Single Burn-In for Multiple Stationary Chains

Because the burn-in of NUTS is computationally expensive, you can use the MSF file to perform a one-time burn-in in one control stream file, and then have subsequent control stream files use the burn-in information (including the mass density information) and evaluate their stationary phases in parallel (at different starting seeds). For example, start with the following burn-in control stream, where NITER=0:

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX
       V1X QX V2X SDIX SDSX
```

```
$DATA example1.csv IGNORE=C
$SUBROUTINES ADVAN3 TRANS4
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
Y = F + F*EPS(1)
$THETA (2.0)X4
$OMEGA BLOCK(4) VALUES(0.15,0.01)
$SIGMA  (0.6 )   ;[P]
$PRIOR NWPRI
$THETAP (2.0 FIX)X4
$THETAPV BLOCK(4) FIX VALUES(10000,0.0)
$OMEGAP BLOCK(4) FIX VALUES(0.2,0.0)
$OMEGAPD (4 FIX)
$SIGMAP 0.06 FIX
$SIGMAPD (1 FIX)
$EST METHOD=ITS NITER=0 PRINT=10 file=nuts_resume.ext
$EST METHOD=NUTS INTERACTION  AUTO=1 NITER=0 PRINT=10 file=nuts_resume.txt
     MSFO=nuts_resume.msf
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

You can use the MSF file to resume an analysis, in this case with NITER non-zero (the stationary phase), and specify a seed:

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX
       V1X QX V2X SDIX SDSX
$DATA example1.csv IGNORE=C
$SUBROUTINES ADVAN3 TRANS4
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
Y = F + F*EPS(1)
$PRIOR NWPRI NTHETA=4, NETA=4, NTHP=4, NETP=4, NEPS=1, NEPP=1
$MSFI nuts_resume.msf
$EST METHOD=NUTS AUTO=1 NITER=1000 PRINT=10 file=nuts_resume2.txt
     SEED=2234556
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

You can run additional control streams as the one above, in parallel, each with a different starting seed. These will serve as separate chains.

I.41 A Note on Setting up Prior Information

Prior information is important for MCMC Bayesian analysis, but not necessary for maximization methods. Of greatest importance are priors to the Omegas. As a general rule, if your data set consists of fewer subjects than 100 times the dimension of the Omega matrix to be estimated, then you should have at least uninformative OMEGA prior information. Priors to THETAS are assumed multivariate normal, and priors to OMEGAS and SIGMAS are assumed inverse Wishart distributed. Alternatively, a residual variance in the form of its square root, may be modeled via THETA (a sigma-like Theta parameters is set up in example 2). For a thorough reference to the options in the $PRIOR record, see ..\guides\VIII.pdf. The following describes the setup for most Bayesian analysis purposes.

To set up the $PRIOR NWPRI statement, keep in mind the following:
NTHETA=number of Thetas to be estimated
NETA=number of Etas (Omegas) to be estimated (and is to be described by an NETAxNETA OMEGA matrix)
NEPS=number of epsilons (Sigmas) to be estimated (and is to be described by an NEPSxNEPS SIGMA matrix)
NTHP=number of thetas which have a prior
NETP=number of Omegas with prior
NEPP=Number of Sigmas with prior (NM73). Before NM73, the NEPP option was ignored, as supplying priors for Sigma's was not activated.

For example:

```
$PRIOR NWPRI NTHETA=4, NETA=4, NEPS=1 NTHP=4, NETP=4, NEPP=1
```

Then the $THETA records list the parameters, in order, the following:
NTHETA of initial thetas
NTHP of Priors to THETAS
Degrees of freedom to each OMEGA block Prior
Degrees of freedom to each SIGMA block Prior

The $OMEGA records list the variances, in order, the following:
NETAxNETA of initial OMEGAS
NTHPxNTHP of variances of Priors to THETAS
NETPxNETP of priors to OMEGAS, matching the block pattern of the initial OMEGAS

The $SIGMA records list the variances, in order, the following:
NEPSxNEPS of initial SIGMAS
NEPPxNEPP of priors to SIGMAS, matching the block pattern of the initial SIGMAS (NM73).

So we may have the following example control stream file portion:

```
$THETA 2.0 2.0 4.0 4.0 ; Initial Thetas
$OMEGA BLOCK(4) ; Inital Parameters for OMEGA
0.4
0.01 0.4
0.01 0.01 0.4
0.01 0.01 0.01 0.4
$SIGMA 0.1

$PRIOR NWPRI NTHETA=4, NETA=4, NEPS=1, NTHP=4, NETP=4, NEPP=1

; Prior information of THETAS (NTHP=4 of them)
$THETA (2.0 FIX) (2.0 FIX) (2.0 FIX) (2.0 FIX)

; Variance to prior information of THETAS (NTHPxNTHP=4x4 of them).
; Because variances are very large, this means that the prior
; information to the THETAS is highly uninformative.  Note that the
; order of $THETA values among the THETA records, and the order
; of $OMEGA values among the OMEGA records, is very important,
; But $THETAs and $OMEGAs can be interspersed.
$OMEGA BLOCK(4)
10000 FIX
0.00 10000
0.00  0.00 10000
0.00  0.00 0.0 10000

; Prior to OMEGA (NETPxNETP=4x4 if them)
$OMEGA BLOCK(4)
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2

; Set degrees of freedom of OMEGA Prior (one value per OMEGA block)
; Uninformative Omega prior is designated by having a DF that is equal to
; the dimension size of the Omega block.
$THETA (4 FIX)

; Prior to SIGMA (NEPPxNEPP=1x1 if them)
$SIGMA 0.05 FIX

; Set degrees of freedom of SIGMA Prior (one value per SIGMA block)
; Uninformative SIGMA prior is designated by having a DF that is equal to
; the dimension size of the Sigma block.
$THETA (1 FIX)
```

By default, the number of prior experiments is 1. However, perhaps you have more than one previous study, and you wish to average their contribution, forming a composite average set of prior parameters to influence the present analysis. In this case, add NEXP=n to the $NWPRI record above, where n is the number of experiments. Then, add the prior information of each additional study with additional $THETA, $OMEGA, and $SIGMA statements. The order is then:

$THETA records list the parameters, in order, the following:
NTHETA of initial thetas

Exp 1:
NTHP of Priors to THETAS
Degrees of freedom to each OMEGA block Prior
Degrees of freedom to each SIGMA block Prior
Exp 2:
NTHP of Priors to THETAS
Degrees of freedom to each OMEGA block Prior
Degrees of freedom to each SIGMA block Prior
…

The $OMEGA records list the variances, in order, the following:
NETAxNETA of initial OMEGAS
Exp 1:
NTHPxNTHP of variances of Priors to THETAS
NETPxNETP of priors to OMEGAS, matching the block pattern of the initial OMEGAS
Exp 2:
NTHPxNTHP of variances of Priors to THETAS
NETPxNETP of priors to OMEGAS, matching the block pattern of the initial OMEGAS
…

The $SIGMA records list the variances, in order, the following:
NEPSxNEPS of initial SIGMAS
Exp 1:
NEPPxNEPP of priors to SIGMAS, matching the block pattern of the initial SIGMAS
Exp 2:
NEPPxNEPP of priors to SIGMAS, matching the block pattern of the initial SIGMAS

Additional examples of setting up prior information for various problems are shown in the example problems listed at the end of this document.

As of NM73, you can use more informative names as follows:
$THETAP for theta priors
$THETAPV for variance to theta priors
$OMEGAP for omega priors
$OMEGAPD for degrees of freedom (or dispersion factor) for omega priors
$SIGMAP for SIGMA priors
$SIGMAPD for degrees of freedom (or dispersion factor) for SIGMA priors

This allows you to intersperse these records at will in the control stream files, but it also gives NMTRAN an alternative source for values to NTHETA, NETA, NTHT, NETP, NEPS, and NEPP that is typically given in the $PRIOR NWPRIOR record. However, if these values are also listed in $PRIOR NWPRI, then these values are chosen over what is surmised from the informatively labeled theta/omega/sigma records. Thus, the above control stream file could be structured as follows, with the various records in any order, and a shortened $PRIOR record (in the following example uninformative priors are used):

```
$PRIOR NWPRI
```

```
; Prior information of THETAS (NTHP=4 of them)
$THETAP (2.0 FIX) (2.0 FIX) (2.0 FIX) (2.0 FIX)

$THETA 2.0 2.0 4.0 4.0 ; Initial Thetas
$OMEGA BLOCK(4) ; Inital Parameters for OMEGA
0.4
0.01 0.4
0.01 0.01 0.4
0.01 0.01 0.01 0.4

; Set degrees of freedom of SIGMA Prior (one value per SIGMA block)
$SIGMAPD (1 FIX)

;intial parameters to sigma
$SIGMA 0.1



; Set degrees of freedom of OMEGA Prior (one value per OMEGA block)
$OMEGAPD (4 FIX)

; Prior to OMEGA (NETPxNETP=4x4 if them)
$OMEGAP BLOCK(4)
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2


; Variance to prior information of THETAS (NTHPxNTHP=4x4 of them).
$THETAPV BLOCK(4)
10000 FIX
0.00 10000
0.00  0.00 10000
0.00  0.00 0.0 10000

; Prior to SIGMA (NEPPxNEPP=1x1 if them)
$SIGMAP 0.05 FIX
```

Informative prior information may come from a previous study. Typically, they are used as follows:

The theta priors for the present analysis are obtained from the estimates of thetas from the previous study. For example, in the report file of the previous study:

```
FINAL PARAMETER ESTIMATE
 THETA - VECTOR OF FIXED EFFECTS PARAMETERS
         TH 1      TH 2      TH 3      TH 4
         1.64E+00  1.57E+00  7.58E-01  2.35E+00
```

would be placed in the present study control stream file as:

```
$THETAP   (1.64 FIXED) (1.57 FIXED) (0.758 FIXED)
          (2.35 FIXED)
```

The variance-covariance to theta priors of the present analysis are obtained from the variance-covariance submatrix pertaining to the theta estimates from the previous study. For example, the information in the report file of the previous study:

```
COVARIANCE MATRIX OF ESTIMATE
            TH 1      TH 2      TH 3      TH 4
 TH 1
+        2.33E-03
 TH 2
+        4.76E-04  2.86E-03
 TH 3
+        7.87E-04  1.27E-04  5.35E-03
 TH 4
+        7.80E-05  2.36E-04  1.76E-03  2.98E-03
```

would be placed in the control stream file of the present study as:

```
$THETAPV BLOCK(4)
        2.33E-03 FIXED
        4.76E-04  2.86E-03
        7.87E-04  1.27E-04  5.35E-03
        7.80E-05  2.36E-04  1.76E-03  2.98E-03
```

The omega priors of the present analysis are obtained from the estimates of omegas from the previous study. For example, from the report file of the previous study:

```
OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS
            ETA1      ETA2      ETA3      ETA4
 ETA1
+        1.75E-01
 ETA2
+        8.33E-03  1.51E-01
 ETA3
+        2.98E-02  1.74E-02  2.41E-01
  ETA4
+       -8.05E-03  1.84E-02  5.14E-02  1.62E-01
```

you transpose as follows to the control stream of the present study:

```
$OMEGAP BLOCK(4)
    1.75E-01 FIXED
    8.33E-03  1.51E-01
    2.98E-02  1.74E-02  2.41E-01
   -8.05E-03  1.84E-02  5.14E-02  1.62E-01
```

Similarly for Sigma priors, the results of the previous study:

```
SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ***
            EPS1
 EPS1
+         5.28E-02
```

Is transposed to the present study control stream as:

```
$SIGMAP (5.28E-02 FIXED)
```

The degrees of freedom to the omega priors of the present analysis are at most the total number of subjects in the previous study. Dr. Mats Karlsson has proposed the following formula for selecting degrees of freedom:

DF=2*[(Omega estimate of previous analysis)/(SE of omega of previous analysis)]^2

Or

DF=2*[(Omega estimate of previous analysis)/(SE of omega of previous analysis)]^2+1

to adjust for degrees of freedom loss in the estimate of Omega of the previous study.

For an OMEGA block, use the smallest DF calculated among the OMEGA diagonal estimates in that block.

A similar formula would apply for SIGMA priors, with the proviso that the DF be no larger than the total number of data points that apply for that sigma in the previous study (for example, if there are two sigmas, one for PK data, and another for PD data, then the sigma for PK data gets no more than total number of PK data points in the previous study).

For convenient transfer of information from a previous analysis to prior information for a subsequent analysis, see section I.96 priorget: Transfer Results of an Analysis to NMTRAN Prior Information (NM75).

As of nm74, the degrees of freedom to the inverse wishart algorithms used for OMEGAS and SIGMAS may be any real number greater than 0. Thus, the inverse wishart matrix distribution can substitute for inverse gamma matrix distribution as follows. The parameter beta is the rate parameter (with inverse units of the variable), and alpha is the shape parameter, to a gamma distribution. This gamma distribution to the inverse residual variance can be expressed with an equivalent Wishart distribution to the inverse residual variance. Set 2alpha for the $SIGMAPD, and beta/alpha for $SIGMA. The gamma distribution to the inverse residual variance is equivalent to the inverse gamma distribution of the residual variance.

I.42 Monte Carlo Direct Sampling (NM72)

On rare occasions, direct Monte Carlo sampling may desired. This method is the purest method for performing expectation maximization, in that it creates completely independent samples (unlike MCMC), and there is no chance of causing bias if the sampling density is not similar enough to the conditional density (unlike IMP). However, it is very inefficient, requiring ISAMPLE values of 10000 to 300000 to properly estimate the problem. The method can be implemented by issuing a command such as

$EST METHOD=DIRECT INTERACTION ISAMPLE=10000 NITER=50

On occasion it can have some use in jump starting an importance sampling method, especially if the first iteration of importance sampling fails because it relies on MAP estimation, and the problem is too unstable for it. Thus, one could perform the following, where just a few iterations of direct sampling begin the estimation process:

$EST METHOD=DIRECT INTERACTION ISAMPLE=10000 NITER=3
$EST METHOD=IMP INTERACTION ISAMPLE=1000 NITER=50 MAPITER=0

Notice that since MAPITER=0, the first iteration of IMP method relies on starting parameters for its sampling density that came from the DIRECT sampling method.

I.43 Some General Options and Notes Regarding EM and Monte Carlo Methods

AUTO=0 (default) (NM73)

If option AUTO=1 is selected, then several options will be set by NONMEM that will allow best settings to be determined. The user may still override those options set by AUTO, by specifying them on the same $EST record. For example,

```
$EST METHOD=ITS AUTO=1 PRINT=10
$EST METHOD=SAEM AUTO=1 PRINT=50
$EST METHOD=IMP PRINT=1 EONLY=1 NITER=5 ISAMPLE=1000
$EST METHOD=BAYES AUTO=1 NITER=1000 FILE=auto.txt PRINT=100
```

The settings of AUTO=1 for each method are as follows:

```
METHOD=DIRECT INTERACTION ISAMPLE=1000 CTYPE=3 NITER=500 STDOBJ=10
                     ISAMPEND=10000 NOPRIOR=1 CITER=10 CINTERVAL=0 CALPHA=0.05
                     EONLY=0

METHOD=BAYES INTERACTION CTYPE=3 NITER=10000 NBURN=4000
                     NOPRIOR=0 CITER=10 CINTERVAL=0 CALPHA=0.05
                     IACCEPT=0.4 ISCALE_MIN=1.0E-06 ISCALE_MAX=1.0E+06
                     PACCEPT=0.5 PSCALE_MIN=0.01 PSCALE_MAX=1000
                     PSAMPLE_M1=1 PSAMPLE_M2=-1 PSAMPLE_M3=1 OSAMPLE_M1=-1
                     OSAMPLE_M2=-1 OACCEPT=0.5 ISAMPLE_M1=2 ISAMPLE_M1A=0
                     ISAMPLE_M2=2 ISAMPLE_M3=3 ISAMPLE_M1B=2 MCETA=0

METHOD=SAEM INTERACTION CTYPE=3 NITER=1000 NBURN=4000
                     ISAMPEND=10 NOPRIOR=1 CITER=10 CINTERVAL=0 CALPHA=0.05
                     IACCEPT=0.4 ISCALE_MIN=1.0E-06 ISCALE_MAX=1.0E+06
```

```
                    ISAMPLE_M1=2 ISAMPLE_M1A=0 ISAMPLE_M1B=2 ISAMPLE_M2=2
                    ISAMPLE_M3=2 CONSTRAIN=1 EONLY=0 ISAMPLE=2 MCETA=0

METHOD=ITS INTERACTION CTYPE=3 NITER=500
                    NOPRIOR=1 CITER=10 CINTERVAL=1 CALPHA=0.05

METHOD=IMP INTERACTION CTYPE=3 NITER=500 ISAMPLE=300 STDOBJ=10
                    ISAMPEND=10000 NOPRIOR=1 CITER=10 CINTERVAL=1 CALPHA=0.05
                    IACCEPT=0.0 ISCALE_MIN=0.1 ISCALE_MAX=10 DF=0 MCETA=3
                    EONLY=0 MAPITER=1 MAPINTER=-1

METHOD=IMPMAP INTERACTION CTYPE=3 NITER=500 ISAMPLE=300 STDOBJ=10
                    ISAMPEND=10000 NOPRIOR=1 CITER=10 CINTERVAL=1 CALPHA=0.05
                    IACCEPT=0.0 ISCALE_MIN=0.1 ISCALE_MAX=10 DF=0 MCETA=3
                    EONLY=0
```

As of nm74, for IMP estimation a second auto value, AUTO=2 is available. Same settings as AUTO=1, with additional:

```
GRDQ=-1.0 DERCONT=1 RANMETHOD=3S2P
```

As of nm75, for SAEM estimation, a second auto value, AUTO=2 is available. Same settings as AUTO=1, with additional:

```
MAPITERS=1
MCETA=100
```

to turn MAP assessment assist on the first iteration.

As of nm75, for BAYES estimation, a second auto value, AUTO=2 is available. Same settings as AUTO=1, with additional:

```
MAPITERS=1
MCETA=100
```

to turn MAP assessment assist on the first iteration.

As of nm74, an AUTO=1 feature is available for NUTS algorithm

```
METHOD=NUTS INTERACTION CTYPE=0 NITER=2000 NBURN=10000
        NOPRIOR=0 NUTS_STEPITER=1 NUTS_STEPINTER=0 NUTS_TEST=0
        NUTS_INIT=75 NUTS_BASE=-3 NUTS_TERM=50 NUTS_GAMMA=0.05
        NUTS_DELTA=0.8 KAPPA=1.0 IKAPPA=1.0
        NUTS_REG=0.0 MADAPT=-1
        NUTS_EPARAM=0 NUTS_OPARAM=1 NUTS_SPARAM=1 NUTS_MASS=B
        NUTS_TRANSFORM=0 NUTS_MAXDEPTH=10
```

A second auto value, AUTO=2, may be used with NUTS estimation to setup the alternative sampling strategy, “Matt trick” (options that differ from AUTO=1 are shown in bold):

```
METHOD=NUTS INTERACTION CTYPE=0 NITER=2000 NBURN=10000
        NOPRIOR=0 NUTS_STEPITER=1 NUTS_STEPINTER=0 NUTS_TEST=0
        NUTS_INIT=75 NUTS_BASE=-3 NUTS_TERM=50 NUTS_GAMMA=0.05
        NUTS_DELTA=0.8 KAPPA=1.0 IKAPPA=1.0
        NUTS_REG=0.0 MADAPT=-1
        NUTS_EPARAM=2 NUTS_OPARAM=1 NUTS_SPARAM=1 NUTS_MASS=BD
        NUTS_TRANSFORM=0 NUTS_MAXDEPTH=10
```

A third auto value, AUTO=3, may be user with NUTS estimation to setup the alternative sampling strategy of eta sampling (options that differ from AUTO=1 are shown in bold):

```
METHOD=NUTS INTERACTION CTYPE=0 NITER=2000 NBURN=10000
        NOPRIOR=0 NUTS_STEPITER=1 NUTS_STEPINTER=0 NUTS_TEST=0
        NUTS_INIT=75 NUTS_BASE=-3 NUTS_TERM=50 NUTS_GAMMA=0.05
        NUTS_DELTA=0.8 KAPPA=1.0 IKAPPA=1.0
        NUTS_REG=0.0 MADAPT=-1
        NUTS_EPARAM=1 NUTS_OPARAM=1 NUTS_SPARAM=1 NUTS_MASS=D
        NUTS_TRANSFORM=0 NUTS_MAXDEPTH=10
```

The AUTO option is ignored by the FO/FOCE/Laplace methods. The AUTO setting itself transfers to the next $EST within the same $PROB, just like any other option settings explicitly set by the user in the control stream file, so AUTO remains on or off until then next AUTO option specified. For example, in the following example:

```
$EST METHOD=ITS AUTO=1 PRINT=10
$EST METHOD=SAEM AUTO=1 PRINT=50
$EST METHOD=IMP PRINT=1 EONLY=1 NITER=5 ISAMPLE=1000
$EST METHOD=BAYES AUTO=1 FILE=auto.txt PRINT=100 NITER=1000
```

the IMP statement also has AUTO=1. However, for the following example:

```
$EST METHOD=ITS AUTO=1 PRINT=10
$EST METHOD=SAEM AUTO=1 PRINT=50
$EST METHOD=IMP PRINT=1 EONLY=1 NITER=5 ISAMPLE=1000 AUTO=0
$EST METHOD=BAYES AUTO=1 FILE=auto.txt PRINT=100 NITER=1000
```

the AUTO setting is turned off for IMP, and turned back on for BAYES. Any option settings implicitly set by the AUTO feature does not transfer to the next $EST statement. Also, when using AUTO=1, the transfer of any options settings explicitly set by the user from previous $EST statements may or may not occur for those options set by the AUTO option, depending on the situation.

The mapping of parameters between S-ADAPT and NONMEM is as follows

S-ADAPT	**NONMEM**
Pmethod=4	IMPMAP
Pmethod=8	IMP
Pmethod=1	ITS
Pmethod=6	DIRECT
Npopiter	NITER
Npopc	ISAMPLE
Npop	MCETA
optmethod	OPTMAP
covest	ETADER
Gefficiency	IACCEPT
Gamma_min	ISCALE_MIN
Gamma_max	ISACLE_MAX

S-ADAPT	NONMEM
DFRAN	DF
Popconv_test	CTYPE
Popconv_rows	CITER
Popconv_alpha	CALPHA
Ndelpar	MAPINTER
Poperr_type=3	$COV MATRIX=S
Poperr_type=8	$COV MATRIX=R
Poperr_type=9	$COV
POPFINAL subroutine	CONSTRAINT subroutine may be user modified to provide any constraining pattern on any population parameters
RANMETHOD	RANMETHOD
SEED	SEED

I.44 MU Referencing

The new methods in NONMEM are most efficiently implemented if the user supplies information on how the THETA parameters are associated arithmetically with the etas and individual parameters, wherever such a relationship holds. Calling the individual parameters phi, the relationship should be

```
phi_i=mu_i(theta)+eta(i)
```

For each parameter i that has an eta associated with it, and mu_i is a function of THETA.
The association of one or more THETA's with ETA(1) must be identified by a variable called MU_1. Similarly, the association with ETA(2) is MU_2, that of ETA(5) is MU_5, etcetera. Providing this information is as straight-forward as introducing the MU_ variables into the $PRED or $PK code by expansion of the code.

For a very simple example, the original code may have the lines

```
CL=THETA(4)+ETA(2)
```

This may be rephrased as:

```
MU_2=THETA(4)
CL=MU_2+ETA(2)
```

Another example would be:

```
CL=(THETA(1)*AGE**THETA(2))*EXP(ETA(5))
V=THETA(3)*EXP(ETA(3))
```

which would now be broken down into two additional lines, inserting the definition of a MU as follows:

```
MU_5= LOG(THETA(1))+THETA(2)*LOG(AGE)
MU_3=LOG(THETA(3))
CL=EXP(MU_5+ETA(5))
V=EXP(MU_3+ETA(3))
```

Note the arithmetic relationship identified by the last two lines, where MU_5+ETA(5) and MU_3+ETA(3) are expressed. This action does not change the model in any way.

It is better to have a linear relationship between all thetas and MU's (as we shall see below)

```
MU_5= THETA(1)+THETA(2)*LOG(AGE)
MU_3=THETA(3)
CL=EXP(MU_5+ETA(5))
V=EXP(MU_3+ETA(3))
```

The above parameterization would also entail log transforming initial values of THETA(1) and THETA(3).

If the model is formulated by the traditional typical value (TV, mean), followed by individual value, then it is straight-forward to add the MU_ references as follows:

```
TVCL= THETA(1)*AGE**THETA(2)
CL=TVCL*EXP(ETA(5))
TVV=THETA(3)
V=TVV*EXP(ETA(3)
MU_3=LOG(TVV)
MU_5=LOG(TVCL)
```

This also will work because only the MU_x= equations are required in order to take advantage of EM efficiency. It is not required to use the MU_ variables in the expression EXP(MU_5+ETA(5)), since the following are equivalent:

```
CL=TVCL*EXP(ETA(5))=EXP(LOG(TVCL)+ETA(5))=EXP(MU_5+ETA(5))
```

but it helps as an exercise to determine that the MU_ reference was properly transformed (in this case log transformed) so that it represents an arithmetic association with the eta.

Again, it is preferable to re-parameterize so that the MU's are linear functions of all thetas:

```
LTVCL= THETA(1)+THETA(2)*LOG(AGE)
CL=EXP(LTVCL+ETA(5))
LTVV=THETA(3)
V=EXP(LTVV+ETA(3)
MU_3=LTVV
MU_5=LTVCL
```

An incorrect usage of MU modeling would be:

```
MU_1=LOG(THETA(1))
MU_2=LOG(THETA(2))
MU_3=LOG(THETA(3))
CL=EXP(MU_1+ETA(2))
V=EXP(MU_2+MU_3+ETA(1))
```

In the above example, MU_1 is used as an arithmetic mean to ETA(2), and a composite MU_2 and MU_3 are the arithmetic means to ETA(1), which would not be correct. The association of MU_x+ETA(x) must be strictly adhered to.

Once one or more thetas are modeled to a MU, the theta may not show up in any subsequent lines of code. That is, the only usage of that theta may be in its connection with MU. For example, if

```
CL=EXP(THETA(5)+ETA(2))
```

So that it can be rephrased as

```
MU_2=THETA(5)
CL=EXP(MU_2+ETA(2))
```

But later, suppose THETA(5) is used without its association with ETA(2):

```
...
CLZ=THETA(5)*2
```

Then THETA(5) cannot be MU modeled, because it shows up as associated with ETA(2) in one context, but as a fixed effect without association with ETA(2) elsewhere. However, if

```
MU_2=THETA(5)
CL=EXP(MU_2+ETA(2))
…
CLZ=CL*2
```

Then this is legitimate, as the individual parameter CL retains the association of THETA(5) with ETA(2), when used to define CLZ. That is, THETA(5) and ETA(2) may not be used separately in any other part of the model, except indirectly through CL, in which their association is retained.

Suppose you have:

```
CL=THETA(5)+THETA(5)*ETA(2)
```

One should see this as:

```
CL=THETA(5)*(1+ETA(2))
```

So the way to MU model this is:

```
MU_2=1.0
CL=THETA(5)*(MU_2+ETA(2))
```

Which would mean that in the end, THETA(5) is not actually MU modeled, since MU_2 does not depend on THETA(5). One would be tempted to model as follows:

```
MU_2=THETA(5)
CL=MU_2+MU_2*ETA(2)
```

But this would be incorrect, as MU_2 and ETA(2) may not show up together in the code except as MU_2+ETA(2) or its equivalent. Thus, THETA(5) cannot be MU modeled. In such cases, remodel to the following similar format:

```
CL=THETA(5)*EXP(ETA(2))
```

So that THETA(5) may be MU modeled as:

```
MU_2=LOG(THETA(5))
CL=EXP(MU_2+ETA(2))
```

Again, for EM methods, better to re-parameterize as:

```
MU_2=THETA(5)
CL=EXP(MU_2+ETA(2))
```

And log transform the initial value of THETA(5).

Sometimes, a particular parameter has a fixed effect with no random effect, such as:

```
Km=THETA(5)
```

with the intention that Km is unknown but constant across all subjects. In such cases, the THETA(5) and Km cannot be Mu referenced, and the EM efficiency will not be available in moving this Theta. However, one could assign an ETA to THETA(5), and then fix its OMEGA to a small value, such as 0.0225 =0.15^2 to represent 15% CV, if OMEGA represents proportional error. This often will allow the EM algorithms to efficiently move this parameter, while retaining the original intent that all subjects have similar, although not identical, Km's. Very often, inter-subject variances to parameters were removed because the FOCE had difficulty estimating a large parametered problem, and so it was an artificial constraint to begin with. EM methods are much more robust, and are adept at handling large, full block OMEGA's, so you may want to incorporate as many etas as possible when using the EM methods.

You should Mu reference as many of the THETA's as possible, except those pertaining to residual variance (which should be modeled through SIGMA whenever possible). If you can afford to slightly change the theta/eta relationship a little to make it MU referenced without unduly influencing the model specification or the physiological meaning, then it should be done.

When the arithmetic mean of an ETA is associated with one or more THETA's in this way, EM methods can more efficiently analyze the problem, by requiring in certain calculations only the evaluation of the MU's to determine new estimates of THETAs for the next iteration, without having to re-evaluate the predicted value for each observation, which can be computationally expensive, particularly when differential equations are used in the model. For those THETA's that do not have a relationship with any ETA's, computationally expensive gradient evaluations must be made to provide new estimates of them for the next iteration.

If you provide a MU reference to THETA's associated with ETAS whose OMEGA value is fixed to 0 (and if you do not turn these MU-reference off with a MUM=N designation, see below), these thetas will also be evaluated by gradient evaluations. However, as of NONMEM 7.4, these gradients will be evaluated using analytical derivatives, which are usually faster and more accurate (see I.48 The FAST Option for use with FOCE/ITS and Differential Equation ($DES) Models (NM74) to read how this is done). This will automatically evaluate its gradient using the analytical eta derivatives, and may increase the speed of analysis, although trial and error is recommended for each case. Set MUM to N for those thetas you want not mu referenced (for example MUM=N(4,5) means to not use mu-reference for thetas 4 and 5), and their derivatives will then be evaluated by finite difference.

There is additional increased efficiency in the evaluation of the problem if the MU models are linear functions with respect to THETA. As mentioned in the previous examples above, we could re-parameterize such that

```
MU_5=THETA(1)+THETA(2)*LOG(AGE)
CL=EXP(MU_5+ETA(5))
MU_3=THETA(3)
```

```
V=EXP(MU_3+ETA(3))
```

This changes the values of THETA(1) and THETA(3) such that the re-parameterized THETA(1) and THETA(3) are the logarithm of the original parameterization of THETA(1) and THETA(3). The models are identical, however, in that the same maximum likelihood value will be achieved. The only inconvenience is having to anti-log these THETA's during post-processing.

The added efficiency obtained by maintaining linear relationships between the MU's and THETA's is greatest when using the SAEM method and the MCMC Bayesian method. In the Bayesian method, THETA's that are linearly modeled with the MU variables have linear relationships with respect to the inter-subject variability, and this allows the Gibbs sampling method to be used, which is much more efficient than the Metropolis-Hastings (M-H) method. By default, NONMEM tests MU-THETA linearity by determining if the second derivative of MU with respect to THETA is nearly or equal to 0. Those THETA parameters with 0 valued second derivatives are Gibbs sampled, while all other THETAS are M-H sampled. In the Gibbs sampling method, THETA values are sampled from a multi-variate normal conditional density given the latest PHI=MU+ETA values for each subject, and the samples are always accepted. In M-H sampling, the sampling density used is only an approximation, so the sampled THETA values must be tested by evaluating the likelihood to determine if they are statistically probable, requiring much more computation time.

As much as possible, define the MU's in the first few lines of $PK or $PRED. Do not define MU_ values in $ERROR. Have all the MU's particularly defined before any additional verbatim code, such as write statements. NMTRAN produces a MUMODEL2 subroutine based on the PRED or PK subroutine in FSUBS, and this MUMODEL2 subroutine is frequently called with the ICALL=2 settings, more often than PRED or PK. The fewer code lines that MUMODEL2 has to go through to evaluate all the MU_s' the more efficient.

MU parameters should be completely defined on every call to PK. While conditional assignments to MU are permitted, make sure to define them for all possibilities, such as:

```
IF(GROUP==1) THEN
MU_2=THETA(3)
ELSE
MU_2=THETA(4)
ENDIF
```

Time dependent covariates, or covariates changing with each record within an individual, cannot be part of the MU_ equation. For example
MU_3=THETA(1)*TIME+THETA(2)
should not be done. Or, consider
MU_3=THETA(2)*WT
Where WT is not constant within an individual, but varies with observation record (time). This would also not be suitable. However, we could phrase as
MU_3=THETA(2)
CL=WT*(MU_3+ETA(3))

where MU_3 represents a population mean clearance per unit weight, which is constant with time (observation record), and is more universal among subjects. The MU variables may vary with inter-occasion, but not with time.

Suppose we have a situation where WT has an unknown power term associated with it modeled as THETA(3) in this example:
CL=THETA(2)*WT**THETA(3)*EXP(ETA(1))
Normally, we could efficiently linear model this as follows:
MU_1=THETA(2)+THETA(3)*LOG(WT)
CL=EXP(MU_1+ETA(1))

with THETA(2) transformed into the log of clearance domain. However, if WT changes record by record within the individual, then LOG(WT) may not be in the Mu modeling. We would then remove the THETA(3)*LOG(WT) term from MU_1:
MU_1=LOG(THETA(2))
CL=WT**THETA(3)*EXP(MU_1+ETA(1))
And THETA(3) itself would not be MU modeled.

For NONMEM 7.2.0, NMTRAN is programmed to detect some MU modeling errors. Nonetheless, the user should verify that these rules are followed.

Examples at the end of the document show examples of MU modeling for various problem types. Study these examples carefully. When transposing your own code, begin with simple problems and work your way to more complex problems.

At this point one may wonder why bother inserting MU references in your code. MU referencing only needs to be done if you are using one of the new EM or Gibbs sampling methods to improve their efficiency. The EM methods may be performed without MU references, but it will be several fold slower than the FOCE method, and the problem may not even optimize successfully. If you choose one of the new methods, and you do not incorporate MU referencing into your model, you are likely to be disappointed in its performance. For simple two compartment models, the new EM methods are slower than FOCE even with the MU references. But, for 3 compartment models, or numerical integration problems, the improvement in speed by the EM methods, properly MU modeled, can be 5-10 fold faster than with FOCE. Example 6 described at the end of the SIGL section is one example where importance sampling solves this problem in 30 minutes, with R matrix standard error, versus FOCE which takes 2-10 hours or longer, and without even requesting the $COV step. So, for complex PK/PD problems that take a very long time in FOCE, it is well worth putting in MU references and using one of the EM methods, even if you may need to rephrase some of the fixed/random (theta/eta) effects relationships. In addition, FOCE is a linearized optimization method, and is less accurate than the EM and Bayesian methods when data are sparse or when the posterior density for each individual is highly non-normal.

It cannot be stressed too much that MU referencing and using the new EM methods will take some time to learn how to use properly. It is best to begin with fairly simple problems, to understand how a particular method behaves, and determine the best option settings. When

setting up a problem for the new EM methods, you should start out with some trial runs, and a limited number of iterations, and observe its behavior. Here are some starting points for the various methods:

$EST METHOD=ITS NITER=100
$EST METHOD=SAEM NBURN=500 NITER=500
$EST METHOD=IMP NITER=100 ISAMPLE=300

The convergence tests should not be used during trial runs. The convergence tests for the EM methods can be fooled into running excessively long, or ending the problem prematurely. For example, the iterations of SAEM are Markov chain dependent, and therefore, certain parameters may meander slowly. The convergence tester, if CITER and CINTERVAL are not properly set to span these meanderings, may never detect stationarity for all the parameters, and therefore may never conclude the analysis. For IMP, the parameters between iterations are less statistically correlated, and the convergence tester is a little more reliable for it.

NMTRAN does some checking of MU statements. If you wish to turn this off (checking mu statements can take a long time for very large control stream files), then include the NOCHECKMU option on the $ABBR record:
$ABBR NOCHECKMU

MUM=MMNNMD

These options allow the MU reference equations for each theta to be optionally used or not used. By default, if a theta parameter is MU referenced, it will be used to facilitate theta parameter estimation. However, the user may "turn off" specific parameters so their Mu referencing is not used. M indicates that the parameter should be Mu modeled (assuming there is an association of a Mu for that theta, which the program will verify), and N indicates it should not be Mu modeled. In the above example, thetas 1,2,5,6 are MU modeled, and 3,4 are not to be Mu modeled. D (for default) indicates you want the program to decide whether to MU model, useful for specifying back to a default option in a future $EST statement, if the present setting is N.

The MUM parameter can also be used to specify which THETAS are used in a mixture problem by marking the position with an X. For example:

MUM=DDDDX

Where THETA(5) is involved in mixture modeling (in a $MIX statement). This is only necessary for covariate dependent mixture models, such as:

```
$MIX
IF(KNOWGENDER==1) THEN
IF(GENDER==1) THEN
P(1)=1.0
P(2)=0.0
ELSE
P(1)=0.0
P(2)=1.0
```

```
ENDIF
ELSE
P(1)=THETA(5)
P(2)=1-THETA(5)
ENDIF
```

and it guarantees that the new estimation methods are aware of the proper parameters.

An alternative method for specifying MU modeled parameters is by using the following syntax:
MUM=v_1(n_1):v_2(n_2):v_3(n_3)...

Where v refers to a letter (N,M,D, or X), and n refers to a number list. For example, to specify thetas 3,5 through 8 to not be MU modeled, theta 2 is a population mixture parameter, and thetas 6,12 are to be MU modeled,

MUM=N(3,5-8):X(2):M(6,12)

Thetas not specified are given a default D designation.

GRD=GNGNNND

By default, if a theta parameter has a Mu associated with it, and its relationship to its Mu is sufficiently linear (the program tests this by evaluating the partial second derivative of MU with respect to theta), then the program will use Gibbs sampling for that parameter. However for Mu modeled parameters, the user can override these decisions made by the program, and force a given parameter to be Gibbs sampled (G), or Metropolis-Hastings sampled (N). In the above example, thetas 1 and 3 are to be Gibbs sampled, and the other thetas are M-H sampled. If the parameter is not Mu modeled, or its Mu modeling is turned off by an MUM option setting, the program performs an M-H sampling. D (for default) specifies you want the program to decide whether to use Gibbs sampling.

For SIGMA parameters, if a particular SIGMA is associated with only one data point type, and conversely, the data point type has only that one SIGMA parameter defining its residual error, and that data point type is not linked by an L2 item with any other data point types, then that SIGMA will by default be Gibbs sampled with a chi-square distribution. Otherwise, that SIGMA parameter will be sampled by Metropolis-Hastings. You can force Meroplis-Hastings by specifying an N. The first m letters of GRD refer to the m THETA's. Then, the m+1th letter refers to SIGMA(1,1), m+2 refers to SIGMA(2,2), etc (going along the diagonal of SIGMA). Not all thetas and sigmas need to be designated. If just the Thetas are designated, for example then the designations for SIGMA are assumed to be D.

For example, for
Y=IPRED + (CMT-1)*IPRED**GAMMA*EPS(1) +(2-CMT)*IPRED*EPS(2)
And with no correlation set between SIGMA(1,1) and SIGMA(2,2), then both SIGMA(1,1) and SIGMA(2,2) will be Gibbs sampled.

Mixed homoscedastic/heretoscedastic residual errors are not Gibbs sampled:

Y=IPRED + IPRED*EPS(1) +EPS(2)

GRD=DDDDDDSSN

The S and D specification are used only for Monte Carlo EM methods. The S specification is optional, and can improve the speed of IMP, IMPMAP, and SAEM methods. Sometimes, users model parameters that could have been a Sigma parameter, but model them as Theta parameters instead, such as:

Y=IPRED+THETA(7)*IPRED*EPS(1)+THETA(8)*EPS(2)

These theta parameters are therefore "Sigma-like", and are typically not MU referenced. To have the S designation, these thetas are not allowed to be involved in evaluating the predicted function F, or compartment values A(x). Specifying theta parameters 7 and 8 as "sigma-like" in this example (note 7^{th} and 8^{th} position of S in the GRD option setting) indicates to the program that when it evaluates forward difference partial derivatives to these thetas (which it must when etas are not associated with theta parameters), it does not have to re-evaluate the predicted function, which can be computationally expensive, especially if one of the differential equation solver ADVAN's are used.

Another example in which the theta can be designated S, is something like the following:

```
$PK
EMAX=THETA(4)
EC50=THETA(5)
…
$ERROR
IPRED=EMAX*F/(EC50+F)
Y=IPRED+IPRED*EPS(1)
…
$EST … GRD=TS(4-5)
```

Note that EMAX and EC50 are not MU modeled, and they are not involved in the evaluation of F or A(x), so they would benefit from an S designation in terms of efficient evaluation in a Monte Carlo EM estimation, especially for differential equation problems.

An alternative method for specifying GRD modeled parameters is by using the following syntax:
GRD=$t_1v_1(n_1)$: $t_2v_2(n_2)$: $t_3v_3(n_3)$...

Where t refers to a parameter type (T for theta, S for SIGMA), v refers to a letter (S,D, or N), and n refers to a number list. For example, to specify thetas 3,5 through 8 to be Gibbs samples, theta 4 is sigma-like, and sigmas 1-3 are to be Metropolis-Hastings processed,

GRD=TG(3,5-8):TS(4):SN(1-3)

Thetas and sigmas not specified are given a default D designation. The SN() designation is also used by EM methods to not determine the derivatives of the objective function with respect to the Sigmas analytically (which is faster), but numerically.

I.45 Degrees of Freedom when Assessing Omegas

Previous to nm751, for EM methods and MCMC Bayesian methods, the default correction for degrees of freedom was
M/(M-LDF)
where M is the number of subjects, and LDF is the loss of degrees of freedom. The LDF for EM/BAYES methods was 1 by default. Such a correction was not used in FOCE/Laplace analysis. Normally, this correction is very small, as M should be at least 10 for a proper population analysis. However, a difference in FOCE versus EM analysis may be noticed if the number of subjects is less than 10.

Because the correction is very small, for most problems, the LDF need not be adjusted. However, on the occasions where there are very few subjects, it would be most accurate to have LDF(I) be the number of to-be-estimated thetas involved in MU_I. For example, you could enter the following (note the LDF must be accessed with verbatim code):

```
" LDF(3)=2.0
MU_3=THETA(4)+LOG(BWT)*THETA(5)
CL=EXP(MU_3+ETA(3))
```

As of nm751, the default LDF for EM/BAYES is 0, and you need to purposely add in degrees of freedom with LDF, and it will act on all analysis methods.

Or, you can ask NONMEM to evaluate the most appropriate LDF for you by setting the LDF to a negative value, for example:

```
" LDF(1:4)=-1.0
```

This will specify LDF according to the number of non-fixed thetas associated with a particular MU reference. Therefore, when setting LDF=-1, it is necessary to have proper mu referencing, even for FOCE/Laplace. If you are curious about what the NONMEM assessed LDF values are, you can access them with:

```
$PK
Include nonmem_reserved_general
…
" write(*,*) TLDF(1),TLDF(2)
```

The LDF=-1 will correct for all analysis methods (including FOCE/Laplace, and EM) except Bayes, for which this correction is not necessary. If you set LDF>0, this will use that degrees of freedom for all methods, including Bayes. You can of course filter as follows:

```
Include nonmem_reserved_general
LDF(3)=2.0
If(IBMETHOD==EST_BAYES) LDF(3)=0.0
```

In summary, for nm751 and higher, if you wish to correct for degrees of freedom loss, let NONMEM determine the best values and for which methods, and make sure thetas are mu referenced. So, for example:

```
$PK
LDF=-1
MU_1=THETA(1)+THETA(2)*LOG(AGE)
MU_2=THETA(3)
```

Etc.
will determine the correct degrees of freedom loss, and impose this correction for EM/FOCE/Laplace, but not for BAYES.

If there is a reason that you need your problem to handle LDF's in the manner of previous NONMEM versions, insert the following after $PROBLEM:
$LDFOLD

and this will revert to the old default handling of LDF's. Only the new methods (ITS/IMP/SAEM/BAYES) are affected by a change in default behavior regarding LDF's of NONMEM version 7.5.0 to 7.5.1.

I.46 Termination testing

A termination test is available for importance sampling, iterative two stage, burn-in phase of SAEM, and the burn-in phase of MCMC Bayesian. It is during burn-in that one wishes to know when the sampling has reached the stationary distribution for SAEM and BAYES. The second, sampling stage in SAEM and BAYES still is determined by how many samples (NITER or NSAMPLE) are desired to contribute to the final answer, so "convergence" does not apply there. There are four parameters set in the $EST statement to specify the termination options:

CTYPE

CTYPE=0 no termination test (default). Process goes through the full set of NBURN (SAEM or BAYES) or NITER (IMP, IMPMAP or ITS) iterations
CTYPE=1. Test for termination on objective function, thetas, and sigmas, but not on omegas.
CTYPE=2. Test for termination on objective function, thetas, sigmas, and diagonals of omegas.
CTYPE=3. Test for termination on objective function, thetas, sigmas, and all omega elements.
CTYPE=4: As of NONMEM 7.2.0, there is an alternative test for FO/FOCE/Laplace. NONMEM will test if the objective function has not changed by more then NSIG digits beyond the decimal point over 10 iterations. If this condition is satisfied, the estimation will terminate successfully. The traditional criterion for successful termination of a classical NONMEM method is that if all of the parameters change by no more than NSIG significant digits, then successful termination results.

CINTERVAL

Every CINTERVAL iterations is submitted to the convergence test system. If CINTERVAL is not specified, then the PRINT option is used as CINTERVAL. If neither PRINT nor CINTERVAL are specified, then default CINTERVAL is listed as 9999, which is interpreted as CINTERVAL=1. If CINTERVAL=0 (NM73), then a best CINTERVAL will be found, then used.

CITER or CNSAMP

Number of latest PRINT or CINTERVAL iterations on which to perform a linear regression test (where independent variable is iteration number, dependent variable is parameter value). If CITER=10, then 10 of the most recent PRINTed or CINTERVAL iterations, are used for the linear regression test. CITER=10 is the default.

CALPHA

CALPHA=0.01-0.05. Alpha error rate to use on linear regression test to assess statistical significance. The default value is 0.05.

At each iteration, the program performs a linear regression on each parameter (which parameters depends on the CTYPE option: if CTYPE=3, then all parameters). If the slope of the linear regression is not statistically different from 0 for all parameters tested, then convergence is achieved, and the program stops the estimation. If you complete NBURN (for SAEM or BAYES methods) or NITER (for IMP, IMPMAP, or ITS methods) iterations and convergence has not occurred, the optimization stops (or goes to the next mode) anyway. So if you want the termination test to properly take effect, give a rather high value to NBURN (1000-10000 for SAEM/BAYES) or NITER (200-1000 for ITS/MAP/IMPMAP) so you don't run out of iterations.

Typically, consecutive importance sampling iterations tend to be nearly statistically uncorrelated, and so it is reasonable to have CITER=10 consecutive iterations (CINTERVAL=1) tested at the alpha=0.05 level. For MCMC methods SAEM and BAYES, consecutive iterations can be highly correlated, so to properly detect a lack of change in parameters, you may want to test every 10^{th} to 100^{th} iteration (CINTERVAL =10 to 100), so that the linear regression on parameter change is spread out over a larger segment of iterations.

An alternative method to convergence testing is to set NBURN to a very high number (10000), monitor the change in MCMCOBJ or SAEMOBJ, and enter ctrl-K (see section I.14 Interactive Control of a NONMEM batch Program) when you feel that the variations are stationary, which will end the burn-in mode and continue on to the statistical/accumulation mode. It is better to provide a large NBURN number, and end it at will with ctrl-K, or allow the convergence tester to end it, rather than to have a small NBURN number and have the burn-in phase end prematurely.

The termination test for the Monte Carlo methods can often be very conservative, and may result in very long run times, even when the objective or likelihood function as well as the parameters appear randomly stationary by eye. To make the termination test more liberal, use one of the lower level CTYPE's (CTYPE=1 or CTYPE=2) to test the more important parameters, or reduce CALPHA to 0.01 or 0.001. Once the objective function is randomly stationary, then often the analysis has converged statistically, so CTYPE=1 is often enough. Remaining parameters that appear to continue to change in a directional manner may often not have much impact on the fit. This can be particularly true of covariances of OMEGAs.

I.47 Use of SIGL and NSIG with the new methods

For the new analysis methods, SIGL is also used to set up forward-difference or central difference gradients as needed. Such finite difference gradients need to be set up for sigma parameters and thetas not MU modeled to etas, or where OMEGA values of etas to which the thetas are MU associated are set to 0.

NSIG is used only with the iterative two stage method, among the new methods. The iterative two stage is not Monte Carlo, and has a more deterministic, smooth trajectory for its parameter movements with each iteration. In this case, NSIG is used as follows: The average of the last CITER/2 parameters are evaluated and compared with the average of the next to last CITER/2 parameters. If CITER is odd valued, (CITER+1)/2 will be used. For example, for CITER=5, at iteration 102, iterations 97-99 are compared with iterations 100-102. If they differ by no more than NSIG significant digits, then this parameter is considered to have converged. When this is true for all parameters tested, optimization is completed.

I.48 The FAST Option for use with FOCE/ITS and Differential Equation ($DES) Models (NM74)

As of nm74, the FAST option is available for FOCE/ITS methods. The FAST method allows use of analytical theta derivatives to facilitate FOCE analysis, especially when using differential equaiton modesl ($DES) (although analytical models are also helped). The method by which this works is based on Almquist et al. [7]. All thetas should be MU-referenced in the manner described in I.44 MU Referencing. For thetas that should not have inter-subject variability associated with them, or should not be MU referenced, Mu reference it anyway by adding addional etas and directly assigning them to these thetas through MU referencing, but set the associated omega values to 0.0 FIXED. Next, set FAST option on the $EST record. For example:

```
$PK
MU_1=THETA(1)+THETA(2)*LOG(AGE)
MU_2=THETA(3)
MU_3=THETA(4) ; for THETA's associated with 0 FIXED omegas, must have strict
; assignment.  Functional transformation, such as MU_3=LOG(THETA(4)), should not be done.
KA=EXP(MU_1+ETA(1))
CL=EXP(MU_2+ETA(2))
V=EXP(MU_3+ETA(3))
…
$OMEGA BLOCK(2)
0.2
0.01 0.2

$OMEGA (0.0 FIXED)
…
$EST METHOD=1 INTERACTION FAST …
```

Notice that theta(4) is associated with eta(3), but as the parameter modeled by theta(4) should not have inter-subject variability, the omega(3,3) is fixed to 0. Also, before nm751, a strict assignment must be made for MU's that have 0 valued OMEGAS, without functional transformation (MU_X=THETA(Y) format). If there is a function transformation between MU and THETA, and the associated OMEGA is 0, then NONMEM will not use the analytical derivative evaluation for the associated theta. As of nm751, functional (but still one-to-one) transformation is allowed in the mu referencing for the FAST algorithm to take effect on that theta. The reason for requiring MU-mapping for all thetas, is that NMTRAN provides analytical eta derivatives for all etas. With MU-mapping (or referencing), that particular eta derivative is interpreted as the corresponding theta derivative by NONMEM, and utilized in FAST algorithms, similar to what has been classically done for OMEGA derivatives. In the above example, analytical derivatives of f with respect to eta(1) are used to generate derivatives of F with respect to theta(1) and theta(2), and similar for theta3 and theta4:
df/d(th1)=df/d(eta1)
df/d(th2)=df/d(eta1)*LOG(AGE)
df/d(th3)=df/d(eta2)
df/d(th4)=df/d(eta3)

For thetas associated with non-zero omegas, MU-reference equations need not be linear with respect to the theta's, but often linear mu-referencing stabilizes the problem, even for FOCE.

For Monte Carlo EM algorithms, even if a theta should not normally be mu-referenced, you can MU-reference it anyway, but set its associated OMEGA diagonal to 0, as described in I.44 MU Referencing. Then, if you set MUM=M(x) for that theta (x), this will evaluate its gradient using the analytical eta derivatives as shown above. The FAST option need not be set for Monte Carlo EM/BAYES algorithms for this to occur. Setting MUM=M(x) can increase speed for analytical models, but may in fact slow down analysis for differential equation $DES problems, because all of the first derivative assessments are turned on, many of which are in fact not needed by IMP/SAEM. Trial and error is recommended in turning on the analytical derivatives feature for thetas that have 0 valued Omegas.

Switches OBJQUICK and MUFIRSTREC can speed up the analysis even further (section *MUFIRSTREC, OBJQUICK*).

I.49 Options to Include Various Constants to the Objective Function (NM74)

LNTWOPI

As of NONMEM 7.4, you may select to have the objective function reported including the N*LOG(2pi) constant term, where N is the total number of normally distributed data values in the data set. Specify option LNTWOPI on the $EST record.

OLNTWOPI

As of NONMEM 7.4, you may select to have the objective function reported including the NETA*NIND*LOG(2pi) constant term for SAEM and BAYES, where NETA is the number of etas, and NIND is number of individuals. Specify option OLNTWOPI on the $EST record.

PRIORC

As of NONMEM 7.4, you may select to have the objective function reported include the constant term to the prior, if the prior is utilized in the objective function evaluation. Specify option PRIORC on the $EST record.

I.50 List of $EST Options and Their Relevance to Various Methods

Option	Classical	ITS	DIRECT	IMP	IMPMAP	SAEM	BAYES	NUTS
-2LL	X	X	X	X	X	X	X	X
ATOL (ADVAN9/13/14/15/16/17)	X	X	X	X	X	X	X	X
AUTO		X	X	X	X	X	X	X
BAYES_PHI_STORE							X	X
BIONLY							X	X
BOOTDATA	X	X	X	X	X	X		
CALPHA		X	X	X	X	X	X	X
CENTERING	X							
CINTERVAL		X	X	X	X	X	X	X
CITER/CNSAMP		X	X	X	X	X	X	X
CLOCKSEED			X	X	X	X	X	X
CONDITIONAL	X	X	X	X	X	X	X	X
CONSTRAIN		X	X	X	X	X	X	X
CTYPE	(CTYPE 4)	X	X	X	X	X	X	X
DERCONT			X	X	X	X		
DF				X	X			
DFS (CHAIN only)								
EONLY			X	X	X	X		
ETABARCHECK	X							
ETADER	X	X		X	X			
ETASAMPLES						X	X	X
ETASTYPE	X	X	X	X	X	X		
FAST	X	X						
FILE	X	X	X	X	X	X	X	X
FNLETA	X	X	X	X	X	X	X	X
FORMAT/DELIM	X	X	X	X	X	X	X	X
FPARAFILE	X	X	X	X	X	X	X	X
GRD		X	X	X	X	X	X	X
GRDQ				X	X			
GRID	X (Stieltjes)							
HYBRID	X							
IACCEPT				X	X	X	X	X
IACCEPTL				X	X			
INTERACTION	X	X	X	X	X	X	X	X
IKAPPA								X
ISAMPEND			X	X	X	X		
ISAMPLE						X	X	X

Option	Classical	ITS	DIRECT	IMP	IMPMAP	SAEM	BAYES	NUTS
ISAMPLE_M1						X	X	X
ISAMPLE_M1A						X	X	X
ISAMPLE_M1B						X	X	X
ISAMPLE_M2						X	X	X
ISAMPLE_M3						X	X	X
ISCALE_MAX				X	X	X	X	X
ISCALE_MIN				X	X	X	X	X
KAPPA								X
KNUTHSUMOFF	X	X	X	X	X	X	X	X
LAPLACE	X	X	*	*	X	*	*	*
LEVCENTER(for $LEVEL)	X	X	X	X	X	X	X	X
LEVWT (for $LEVEL)	X	X	X	X	X	X	X	X
LIKE	X	X	X	X	X	X	X	X
LNTWOPI	X	X	X	X	X	X	X	X
MADAPT								X
MAPCOV				X	X			
MAPINTER				X	X			
MAPITER				X	X			
MAPITERS						X	X	
MASSRESET							X (in prep. For NUTS)	X
MAXEVAL	X							
MCETA	X	X		X	X			
MSFO	X	X	X	X	X	X	X	X
MUM		X	X	X	X	X	X	X
NBURN						X	X	X
NITER/NSAMPLE		X	X	X	X	X	X	X
NOABORT	X	X	X	X	X	X	X	X
NOCOV	(when last estimation step)	X	X	X	X	X	X	X
NOHABORT	X	X	X	X	X	X	X	X
NOLABEL	X	X	X	X	X	X	X	X
NONINFETA	X							
NOOMEGABOUNDTEST	X							
NOPRIOR	X	X	X	X	X	X	X	X
NOSIGMABOUNDTEST	X							
NOSLOW	X	X						
NOSUB	X	X	X	X	X	X	X	X
NOTHETABOUNDTEST	X							
NOTITLE	X	X	X	X	X	X	X	X
NSIG/SIGDIGITS	X	X						
NUMDER	X	X	X	X	X	X	X	X
NUMERICAL	X	X	*	*	X	*	*	*
NUTS_BASE								X
NUTS_DELTA								X
NUTS_EPARAM							X (in prep for NUTS)	X

Option	Classical	ITS	DIRECT	IMP	IMPMAP	SAEM	BAYES	NUTS
NUTS_GAMMA								X
NUTS_INIT								X
NUTS_MASS							X (in prep for NUTS)	X
NUTS_MAXDEPTH								X
NUTS_OPARAM							X (in prep for NUTS)	X
NUTS_REG							X (in prep for NUTS)	X
NUTS_SPARAM							X (in prep for NUTS)	X
NUTS_STEPINTER								X
NUTS_STEPITER								X
NUTS_TERM								X
NUTS_TEST								X
NUTS_TRANSFORM								X
OACCEPT							X	X
OLKJDF								X
OLNTWOPI						X	X	X
OMEGABOUNDTEST	X							
OMITTED	X	X	X	X	X	X	X	X
OPTMAP	X	X		X	X			
ORDER	X	X	X	X	X	X	X	X
OSAMPLE_M1							X	X
OSAMPLE_M2							X	X
OSAMPLE_M3							X	X
OVARF								X
PACCEPT							X	X
PARAFILE	X	X	X	X	X	X	X	X
PARAFPRINT	X	X	X	X	X	X	X	X
PHITYPE		X	X	X	X	X	X	X
POSTHOC	X	X	X	X	X	X	X	X
PREDICTION	X	X	X	X	X	X	X	X
PRINT	X	X	X	X	X	X	X	X
PRIORC	X	X	X	X	X	X	X	X
PSAMPLE_M1							X	X
PSAMPLE_M2							X	X
PSAMPLE_M3							X	X
PSCALE_MAX							X	X
PSCALE_MIN							X	X
RANMETHOD=nSmP			X	X	X	X	X	X
REPEAT	X							
REPEAT1	X							
REPEAT2	X							

Option	Classical	ITS	DIRECT	IMP	IMPMAP	SAEM	BAYES	NUTS
SADDLE_HESS	X							
SADDLE_RESET	X							
SEED			X	X	X	X	X	X
SIGL	X	X	X	X	X	X		
SIGLO	X	X		X	X			
SIGMABOUNDTEST	X							
SLKJDF								X
SLOW	X	X						
SORT	X							
STDOBJ				X	X			
STIELTJES	X							
SVARF								X
TBLN (CHAIN only)								
THETABOUNDTEST	X							
THIN							X	X
TPU								X
TTDF								X
ZERO	X							

*May be needed to suppress error messages from NMTRAN or NONMEM.

I.51 When to use each method

While there is some overlap in usage of the various EM methods, some basic guidelines may be noted. MC Importance Sampling EM (IMP) is most useful for sparse (few data points per subject, that is, fewer data points than there are etas to be estimated for a given subject) or rich data, and complex PK/PD problems with many parameters. The SAEM method is most useful for very sparse, sparse, or rich data, and for data with non-normal likelihood, such as categorical data. The iterative two stage (ITS) method is best for rich data, and rapid exploratory methods, to obtain good initial parameters for the other methods. The FOCE method is useful for rich data, and in cases where there are several or more thetas that do not have ETA's associated with them.

I.52 Composite methods

Composite methods may be performed by giving a series of $EST commands. The results of the estimation method are passed on as initial parameters to the next $EST method. Also, any settings of options of the present method are passed on by default to the next $EST method.

One suggestion is to perform in the following order (although trial and error is very important):
1) Iterative two stage for rapid movement of parameters towards reasonable values (10-30 iterations)
2) SAEM if model is complex, or data are very sparse, with 300-3000 iterations, depending on model complexity. Obtain maximum likelihood parameters
3) Importance Sampling if model is complex with 300-3000 samples, 50-100 iterations, depending on model complexity. Obtain maximum likelihood parameters
4) Evaluate at final position by importance sampling. Obtain maximum likelihood value and standard errors

5) Perform MCMC Bayesian analysis on your favorite model, 200-1000 burn in samples (having started at maximum, no more is necessary), 10000-30000 stationary samples. Obtain complete distribution of parameters, to obtain mean, standard error, confidence bounds

An example control stream file follows.

Iterative two stage with 50 iterations

```
$EST METHOD=ITS INTERACTION NITER=50 SIGL=7 NSIG=2
```

SAEM with 200 iterations for stochastic mode, 500 iterations for accumulated averaging mode

```
$EST METHOD=SAEM INTERACTION NBURN=200 NITER=500
```

Importance sampling for 10 iterations, expectation step only (this evaluates OBJF without moving population parameters). Note that SIGL=7 that was set for the previous $EST command is assumed for this $EST command as well

```
$EST METHOD=IMP INTERACTION ISAMPLE=1000 NITER=10 EONLY=1
```

MCMC Bayesian Analysis, with 200 burn in samples, and 10000 stationary samples:

```
$EST METHOD=BAYES INTERACTION NBURN=200 NSAMPLE=10000
```

Here is the full control stream file:

```
$PROBLEM  Setup of Data for Bayesian Analysis
$INPUT   SET ID JID TIME   DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X
SDIX SDSX
$DATA     samp5.csv

$SUBROUTINES ADVAN3 TRANS4
; At least An uninformative Prior on OMEGAS is
; recommended for MCMC Bayesian
$PRIOR NWPRI NTHETA=4, NETA=4, NTHP=0, NETP=4, NPEXP=1
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
Y = F + F*EPS(1)

$THETA 2.0 2.0 4.0 4.0 ; Initial Thetas
$OMEGA BLOCK(4) ; Inital Parameters for OMEGA
0.4
0.01 0.4
0.01 0.01 0.4
```

```
0.01 0.01 0.01 0.4
$SIGMA 0.1

; Set the Priors.  Good Idea if Doing MCMC Bayesian
$OMEGA BLOCK(4) ; Prior to OMEGA
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2
$THETA (4 FIX) ; Set degrees of freedom of OMEGA PRior

;ITS. Store results in sampl5_extra.txt
$EST METHOD=ITS INTERACTION FILE=samnp5l_extra.TXT
     NITER=30 PRINT=5 NOABORT MSFO=.msf
     SIGL=6
; Next to SAEM.  Option settings carry over from
; previous $EST by default. So results are added to
; same file
$EST METHOD=SAEM NBURN=200 NITER=500 PRINT=100
; Calculate OBJF by importance sampling
$EST METHOD=IMP EONLY=1 NITER=5 ISAMPLE=3000 PRINT=1
; Store results of Bayesian in its own file
$EST METHOD=BAYES FILE=.TXT NBURN=200 NITER=3000
     PRINT=100
; Do an FOCE just for comparison
$EST  METHOD=COND INTERACTION MAXEVAL=9999 NSIG=2
      SIGL=6 PRINT=5
$COV MATRIX=R
```

More examples of composite analysis are given at the end of this document.

I.53 $THETAI ($THI) AND $THETAR ($THR) Records for Transforming Initial Thetas and Reporting Thetas (NM73)

Initial thetas in the $THETA record may be functionally transformed with the $THETAI (or $THI) record, and final thetas may then be reverse transformed for report purposes using $THETAR (or $THR). This has particular value when it is desired that the thetas by estimated within NONMEM in the log domain, but you want the convenience of inputting and outputting them in the natural domain, such as when performing linear MU referencing. For example,

```
$THETAI
THETA(1:NTHETA)=LOG(THETAI(1:NTHETA))
THETA(NTHETA+1:NTHETA+NTHP)=LOG(THETAI(NTHETA+1:NTHETA+NTHP))
```

Or

```
$THETAI
THETA(1:NTHETA)=LOG(THETAI(1:NTHETA))
THETAP(1:NTHP)=LOG(THETAPI(1:NTHP))
```

Where ntheta=number of to be estimated thetas, and nthp=number of theta priors. Or, leave it to NONMEM to supply the range (which is by default NTHETA+NTHP).

```
$THETAI
THETA=LOG(THETAI)
```

This record will convert any initial thetas in a $THETA record, or thetas obtained from a chain file, but will not convert thetas from an MSF file. Furthermore, the variance to the theta priors will be appropriately converted, when using $PRIOR NWPRI ($PRIOR TNPRI receives variance-covariance information from MSF files, and this information is in the model theta domain).

For reporting thetas, the inverse function should be supplied:

```
$THETAR
THETAR=EXP(THETA)
```

Or

```
$THETAR
THETAR(1:NTHETA)=EXP(THETA(1:NTHETA))
THETAPR(1:NTHP)=EXP(THETAP(1:NTHP))
```

The code in $THETAI and $THETAR is transferred to the FORTRAN compiler without interpretation.

An example is shown with thetair.ctl:

```
$PROB RUN# From Example 1
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$THI
THETA(1:NTHETA)=DLOG(THETAI(1:NTHETA))
THETAP(1:NTHP)=DLOG(THETAPI(1:NTHP))

$THR
THETAR(1:NTHETA)=DEXP(THETA(1:NTHETA))
THETAPR(1:NTHP)=DEXP(THETAP(1:NTHP))

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
Y = F + F*EPS(1)
```

```
; Initial values of THETA
$THETA (7.389056099)X4
;INITIAL values of OMEGA
$OMEGA BLOCK(4) VALUES(0.2,0.001)
;Initial value of SIGMA
$SIGMA
(0.6 )   ;[P]

$PRIOR NWPRI
;prior information on thetas
$THETAP (7.389056099 FIX)X4
;variance to theta priors
$THETAPV BLOCK(4) FIX VALUES(545981.5003,0.0)

; Prior information to the OMEGAS.
$OMEGAP BLOCK(4)
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2
$OMEGAPD (4 FIX)

$EST METHOD=ITS INTERACTION NOABORT CTYPE=3 PRINT=5 NOPRIOR=1
$EST METHOD=BAYES INTERACTION NOABORT NBURN=200 NITER=500 CTYPE=3
PRINT=50 NOPRIOR=0
$EST METHOD=1 INTERACTION NSIG=3 SIGL=10 PRINT=1 NOABORT
MAXEVAL=9999 NOPRIOR=1
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

Note the use of informative names for the prior information (see I.41 A Note on Setting up Prior Information).

When using $THETAI, be aware that the any bounds described in the $THETA record are also submitted to the $THETAI transformations. Therefore, bounds that are likely to cause a domain error in the transformation process should not be included. For example, When Theta is log transformed, the lower bound of 0 should not be included. The transformation itself will prevent reported thetas from being less than 0.

```
$THI
THETA(1)=LOG(THETAI(1))

$THETA
(0,5.0) ; not good

…
$THETA
(0.1,5.0) ; okay.

…
$THETA
5.0 ; okay.
```

I.54 A note on Analyzing BLQ Data (NM73)

Since NONMEM VI, SIGMA(x,x) has been allowed to be used on the right hand side of equations in the control stream file. This has offered a means to obtaining the residual variance in code, for example:

```
IPRED = F
SD=SQRT(SIGMA(1,1))*IPRED
Y=IPRED+IPRED*EPS(1)
...
$SIGMA 0.01
```

Whereas previously, to obtain SD, a theta needed to be used as the residual coefficient in place of SIGMA:

```
$ERROR
IPRED = F
SD=THETA(1)*IPRED
...
Y=IPRED + SD*EPS(1)
...
$THETA 0.1
$SIGMA (1.0 FIXED)
```

Furthermore, if some data are below level of quantitation (BLQ), and it is desired to use an integral of the normal density to represent that the value can be anywhere below BLQ, this can be modeled using THETA as follows, requiring the Laplace method:

```
$ERROR
IPRED = F
SD = THETA(3)*IPRED
LOQ=0.1
DUM = (LOQ - IPRED) /SD
CUMD = PHI(DUM)+1.0E-30
IF (DV.GT.LOQ) THEN
      F_FLAG = 0
      Y = IPRED + SD*ERR(1)
ELSE
      F_FLAG = 1
      Y = CUMD
      MDVRES=1
ENDIF
$SIGMA (1.0 FIXED)
$THETA
-2.3 4.2 0.3
```

When performing an EM analysis, such as importance sampling, remember to designate the THETA that serves as the residual coefficient as a sigma-like parameter, by setting GRD appropriately:

```
$EST METHOD=IMP LAPLACE INTERACTION CTYPE=3 NOHABORT GRD=TS(3) PRINT=1
```

If you are using SIGMA instead, then code as follows:

```
$ERROR
IPRED = F
SD=SQRT(SIGMA(1,1))*IPRED
LOQ=0.1
DUM = (LOQ - IPRED) / SD
CUMD = PHI(DUM)+1.0E-30
IF (DV>LOQ) THEN
      F_FLAG = 0
      Y = IPRED + IPRED*EPS(1)
ELSE
      F_FLAG = 1
      Y = CUMD
      MDVRES=1
ENDIF
$THETA
-2.3 4.2
$SIGMA 0.1
```

In this case, the SIGMA is not being used purely as a scale parameter in a normal density variance matrix, but is also being used as a parameter in another distribution (the integrated normal density). When using an EM or Bayes method, it is best to indicate that this SIGMA should not be estimated using the usual analytical method for calculating SIGMA derivatives, but using numerical derivatives, by designating the GRD appropriately:

```
$EST METHOD=IMP LAPLACE INTERACTION CTYPE=3 NOHABORT GRD=SN(1) PRINT=1
```

I.55 $ANNEAL to facilitate EM search methods (NM73)

Syntax:
$ANNEAL number-list1:value1 number-list2:value2
etc. for as many lists that are needed.

Example:
$ANNEAL 1-3,5:0.3 6,7:1.0
Sets starting diagonal Omega values for purposes of simulated annealing. Thus, initial values of OMEGA(1,1), OMEGA(2,2), OMEGA(3,3), and OMEGA(5,5) are set to 0.3, while initial OMEGA(6,6) and OMEGA(7,7) are set to 1.0. When $EST CONSTRAIN>=4, an algorithm in constraint.f90 will initially set the omegas to these values, and then shrink these OMEGA values more and more with each iteration, and eventually shrinks the OMEGA's to 0, the intended target value for that Omega. This is a technique that may be used especially with SAEM, to provide an annealing method for moving thetas that have 0 omega values associated with them. The default is the use of gradient methods, which are good for problems starting near the solution, whereas the annealing method is more suitable for problems starting far from the solution.

An example is anneal.ctl, an EMAX model in which the Hill coefficient does not have inter-subject variance (that is, its omega variance is set to 0):

```
$PROB Emax model with hill=3
$INPUT ID DOSE DV
$DATA anneal.dat IGNORE=@
$PRED

 MU_1 = THETA(1)
```

```
 EMAX = EXP(MU_1+ETA(1))
 MU_2 = THETA(2)
 ED50 = EXP(MU_2+ETA(2))
 MU_3 = THETA(4)
 E0   = EXP(MU_3+ETA(3))

 MU_4=THETA(3)
 HILL = EXP(MU_4+ETA(4))

 IPRED = E0+EMAX*DOSE**HILL/(ED50**HILL+DOSE**HILL)
 Y     = IPRED + EPS(1)

$THETA  4.1 ; 1. Emax
$THETA  6.9 ; 2. ED50
$THETA  0.001 ; 3. Hill
$THETA  2.3 ; 4. E0

$OMEGA BLOCK(2) 0.1
                0.01 0.1
$OMEGA 0.1
$OMEGA 0.0 FIXED

$ANNEAL 4:0.3

$SIGMA 1
$ESTIMATION METH=SAEM INTER NBURN=1000 NITER=500 ISAMPLE=5 IACCEPT=0.3 CINTERVAL=25 CTYPE=0
NOABORT PRINT=50 CONSTRAIN=5 SIGL=8
$ESTIMATION METH=IMP INTER PRINT=1 NITER=0 ISAMPLE=10000 EONLY=1 CONSTRAIN=0 MAPITER=0 DF=4
$COV MATRIX=R UNCONDITIONAL
```

The user may modify the subroutine CONSTRAINT that performs the simulated annealing algorithm. The source code to the CONSTRAINT subroutine is available from the ..\source directory as constraint.f90 , and the user may copy this to their run directory, and as convenient, to rename it. Then, specify OTHER=name_of_source.f90 in the $SUBROUTINE record, as shown in example 9. The subroutine CONSTRAINT may also be used to provide any kind of constraint pattern on any parameters.

Another technique is to use an initial Monte Carlo search method using $EST METHOD=CHAIN ISAMPEND, and then use the standard gradient method for SAEM, as follows:

```
$PROB Emax model with hill=3
$INPUT ID DOSE DV
$DATA anneal.dat IGNORE=@
$PRED

 MU_1 = THETA(1)
 EMAX = EXP(MU_1+ETA(1))
 MU_2 = THETA(2)
 ED50 = EXP(MU_2+ETA(2))
 MU_3 = THETA(4)
 E0   = EXP(MU_3+ETA(3))

 MU_4=THETA(3)
 HILL = EXP(MU_4+ETA(4))

 IPRED = E0+EMAX*DOSE**HILL/(ED50**HILL+DOSE**HILL)
 Y     = IPRED + EPS(1)

$THETA  4.1 ; 1. Emax
$THETA  6.9 ; 2. ED50
$THETA  (-3.0,0.001,3.0) ; 3. Hill
$THETA  2.3 ; 4. E0

$OMEGA BLOCK(2) 0.1
                0.01 0.1
$OMEGA 0.1
```

```
$OMEGA 0.0 FIXED

$SIGMA 1
$EST METHOD=CHAIN ISAMPLE=1 ISAMPEND=30 NSAMPLE=30 FILE=anneal2.chn
$ESTIMATION METH=SAEM INTER NBURN=4000 NITER=200 ISAMPLE=5 IACCEPT=0.3 CINTERVAL=25 CTYPE=3
NOABORT PRINT=100
$ESTIMATION METH=IMP INTER PRINT=1 NITER=0 ISAMPLE=10000 EONLY=1 MAPITER=0
$COV MATRIX=R UNCONDITIONAL
```

Notice that the range of Monte Carlo search for the Hill coefficient is from -3 to 3, the specified lower and upper bound values (note that theta(3) is actually the log of the Hill coefficient). See I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records.

For importance sampling, the CONSTRAINT subroutine is set up so that when CONSTRAIN=4, it uses NBURN as the first NBURN iterations of the total NITER to be for the annealing process, and the remaining NITER-NBURN iterations will continue the problem with the annealing OMEGA being set to 0. See ..\examples\anneal4.ctl, and notice that NBURN=40, NITER=60, and CONSTRAIN=4.

I.56 $COV: Additional Options and Behavior

Example syntax:

```
$COV UNCONDITIONAL TOL=10 SIGL=10 SIGLO=11 NOFCOV ATOL=6 RESUME
```

If $COV is specified, then for IMP, IMPMAP, and ITS methods, standard error information will be supplied for every $EST statement.

Standard error information for the classical methods (METHOD=0, METHOD=1) will be given only if they are the last estimation method, and only if NOFCOV is not specified.

If UNCONDITIONAL is specified, then for the IMP and IMPMAP EM methods, if the R information matrix is not positive definite, the program will modify the matrix to be positive definite, will report that it has done so, and provide the standard errors. The user should use the standard error results with caution should a non-positive definite flag occur.

The ITS and SAEM methods can only evaluate the S matrix, and will do so even if MATRIX=R is requested. The banner information will show what type of variance was evaluated.

The BAYES method always supplies standard errors, correlation matrix, and covariance matrix, even when $COV step is not requested, as these results are a direct result of summarizing the accumulated NITER samples. Furthermore, the matrices are always positive definite, and therefore always successful.

To obtain the eigenvalues to the correlation matrix, even for the BAYES method, a $COV step must be issued with the PRINT=E feature.

TOL, SIGL, SIGLO (NM72)

The TOL (used by PREDPP when differential equations are integrated) and SIGL and SIGLO may be set specifically for the $COV step, distinct from those used during $EST. This special option for $COV is not so important for the new EM or BAYES methods, which are able to obtain suitable standard errors using SIGL, SIGLO, and TOL that are also used for estimation, but classical NONMEM methods in particular can require a different significant digits level of evaluation (usually more stringent) during the $COV step than during $EST. Keep in mind that when evaluating the R matrix, SIGL and TOL should be at least 4 times that of what one would normally set NSIG. If evaluating only the S matrix, then SIGL,SIGLO, TOL should be at least 3 times that of what one normally sets NSIG. For example, during $EST, NSIG=2, SIGL=6, TOL=6 may be sufficient, but during $COV, you may need SIGL=12 TOL=12 to avoid positive definiteness issues. The MATRIX, TOL, and SIGL have no relevance to the variance results for a BAYES method, which are derived from samples generated during the estimation step. If TOL is set in the $COV record, but SIGL and/or SIGLO are not, then the TOL is not changed. Also, if TOL is set for the $COV record, then this TOL is used for all compartments.

FAST (NM74)

This is equivalent to FAST for the $EST record (see I.48 The FAST Option for use with FOCE/ITS and Differential Equation ($DES) Models (NM74)), but for the $COV record. If $EST FAST is set, then $COV will be set to FAST, unless you specify SLOW or NOSLOW at the $COV record.

ATOL (NM72)

The absolute tolerance option pertains to using ADVAN13, and as of NM73, to ADVAN9 as well, where ATOL is the accuracy for derivatives evaluated near zero. The same ATOL value is set for all compartments. The ATOL by default is 12. Usually the problem runs quickly when using ADVAN13 with this setting. On occasion, however, you may want to reduce ATOL (usually equal to that of TOL), and improve speeds of up to 3 to 4 fold. ATOL may be set at the $EST or $COV command. Keep in mind that ATOL is changed for the $COV step only if SIGL and/or SIGLO are also specified at the $COV record.

KNUTHSUMOFF (NM74)

In NONMEM 7.4, the Knuth summing method is used to allow the most accurate summation of individual objective function values, even with large variations in values of the individual objective function. To turn this off, and allow a standard summation (not recommended except for comparison purposes from earlier versions), set KNUTHSUMOFF=1. If KNUTHSUMOFF was set in the $EST step, but not in the $COV step, the KNUTHSUMOFF value of the last $EST record will be used.

POSDEF =0 (default) (NM75)

If POSDEF=1, then if the R matrix is not positive definite, then during its inversion, it will be forced positive definite, by method of shifting the eigenvalues by an amount slightly greater than the most negative eigenvalue. If POSDEF=2, then R matrix is made positive definite by making

the negative valued eigenvalues approximately 1/100 of the least positive eigenvalue (courtesy of Larry Schaeffer, U. Guelph, Ontario, Ca, see ..\guides\PDforce.pdf). If POSDEF=3, then R matrix is made positive definite by using the absolute values of the eigenvalues. POSDEF is 0 (no correction) by default for classical covariance step, and 3 for EM methods. The preconditioning method, described below, is a more sophisticated, but time-consuming method of making the R matrix positive definite.

CHOLROFF =0 (default) (NM74)

If CHOLROFF is set to 1, then one part of the R matrix evaluation (before inversion) will be evaluated in the manner of earlier versions of NONMEM. This is strictly for comparison with earlier versions for diagnostic purposes. If CHOLROFF=0, then a part of the R matrix evaluation will undergo cholesky decomposition, and if not positive definite, only then will it undergo evaluation according to earlier versions of NONMEM. If CHOLROFF=2, then if cholesky decomposition fails, it will be slightly adjusted to be positive definite. The CHOLROFF=0 is the safest option, in that the Cholesky decomposition (if positive definite) provides more precision in evaluating the R matrix. The CHOLROFF=2 setting is useful if you wish to increase the rate of success in obtaining an R matrix (followed by its inverse) that could be suitable for SIR sampling. If you use CHOLROFF=2, then positive definiteness has been corrected before the inverse occurs, so POSDEF will have no additional effect. If you use POSDEF, then CHOLROFF should be 0 or 1.

NOFCOV (NM72)

No $COV step for any classical estimation steps. This would be useful if you wanted EM estimation analyses with variance-covariance assessment performed, and a final FOCE analysis performed, but did not want the program to spend time on standard error assessments for FOCE, which can take a long time relative to the other methods.

RESUME (NM73)

If an MSFO=msffile specification was made in the $EST step, and analysis was interrupted during the $COV step for the FO/FOCE/Laplace method, then the $COV step may be resumed where it was interrupted by executing another control stream file that uses the $MSFI record specifying the MSFO file of the interrupted analysis, and the RESUME option is entered at the $COV record:

```
…
$MSFI=msffile
…
$COV RESUME
```

In addition to the main msf file and _ETAS file, files _RMAT, and _SMAT files will generated by the MSFO option of the previous control stream or problem, which are needed by the RESUME option of the new control stream or problem.

PARAFPRINT=1 (default, NM74)

The print iteration intervals to the parallelization log file can be controlled by this option during parallelization of the $COV step.

PARAFILE=OFF (NM75)

Specifically turn off parallelization during the $COV step.

THBND =1 (default) (NM74)

For covariance assessment for FOCE/FO/Laplace, if THBND=1, any theta boundaries specified in the $THETA record causes NONMEM to impose a non-linear transformation of the theta parameters so that the transformed parameters may vary from –infinity to infinity. It does this with logistic transformations. This is suitable during the estimation step, but may be desirable to have this off (THBND=0), and assess partial derivatives of the objective function with respect to the thetas themselves, or some linear transformation of these thetas. By default THBND=1, in keeping with the behavior of earlier NONMEM versions, which effectively has THBND=1. Usually boundaries that are fairly wide will not impact how the variance-covariance is assessed, such as when a lower bound of 0 is given, but if you have very narrow boundaries set, then this can impact the assessment of the variance-covariance of the estimates considerably, and you may wish to set THBND=0. If no lower or upper bounds are given to thetas in $THETA record, this option has no impact.

The degree to which boundary settings can impact the variance of a parameter is linearly approximated by the relation:

$$\mathrm{I}_c = \mathrm{I}_u + \left(\frac{(H-L)}{(H-x)(x-L)} \right)^2$$

Where I refers to information (inverse variance), c=boundary constrained assessment (THBND=1), u=non-boundary constrained assessnt (THBND=0), x is estimate of theta, H is high (upper) boundary, L is lower boundary. For the standard error to be decreased by no more than 10%, the second term should not be larger than 20% (as this is a 1/variance term), so no larger than 0.2. Assuming boundaries equidistant to the estimate, this comes to sqrt(4/0.2)=4.47, that is, the distance between estimate and nearest boundary should be no smaller than 4.47 times the unconstrained standard error.

IMP/SAEM/ITS do not evaluate the variance/covariance taking into account boundary settings, and act always effectively as THBND=0.

Importance Sampling of the Variance-Covariance of the Parameter Estimates (NM74)

Based on the Perl-Speaks-Nonmem Sampling-Importance-Resampling algorithm (SIR), after the variance-covariance matrix is assessed by the $COV step by the usual deterministic finite-difference evaluation of the Fisher information matrix for FO/FOCE/Laplace, it can be used as the variance-covariance to a proposal density to obtain Monte Carlo importance samples of population parameters, in the neighborhood of the minimum of the objective function. The

samples are generated with normal distribtution sampling in the unconstrained parameter domain (see Appendix K of NONMEM7_technical_guide.pdf describing unconstrained parameter domain), and then transformed back to the original domain. The Monte Carlo samples will be listed in the raw output file (.ext), along with their importance sampling weights. The weighted average parameter values are listed on the line labeled iteration -1000000000, and the weighted sample variance-covariances are listed on the line labeled -1000000001, in accordance with raw output file format. The other statistics labeled -100000000x, are also reported, and should be similarly interpreted as if they were results reported for a BAYES analysis. Iteration 0 contains the objective function minimum position, along with its importance sampling weight, which is 1. Re-sampling is not performed, however this can be done by a post-processing utility software (such as R), keeping in mind that each sample should be weighted according to its relative WEIGHT listing. For example, creating a uniform random variable r, when the sum of the normalized weights of the jth sample in the list first is equal to or exceeds r, that is, when:

$$r \le \frac{\sum_{k=1}^{j} W_k}{(\text{sirsample}+1)W_{-1000000000}}$$

select that sample j (sirsample+1 is the number of total samples, including the 0^{th} iteration one). In this way, samples will be selected in proportion to their weight listings. Similarly, to determine a quantile position q for a particular parameter, sort the samples for that parameter, and adding their weights in order of the sorted samples, when the normalized summed weight first exceeds q:

$$q \le \frac{\sum_{k=1}^{j} W_k}{(\text{sirsample}+1)W_{-1000000000}}$$

select that parameter as the qth quantile. Read about the utilities that perform frequency and quantile sorting (table_quant), or resampling (table_resample) in I.79 table_quant, and table_resample Utility Programs for Analyzing $COV Sampling-Importance-Resampling (SIR) data (NM74).

The following options may be set for the $COV record to control the SIR sampling process. Note: the RESUME option only resumes an interrupted generation of finite difference of the covariance estimation. If your analysis is interrupted during a SIR sampling process, RESUME will start at the beginning of a new set of sample generation.

SIRSAMPLE=0 (default) (NM74)

By default SIRSAMPLE=0, so SIR process does not occur. Setting SIRSAMPLE to a value greater than 0 will produce SIRSAMPLE importance samples. These will be placed in the raw output file as a table
with heading "Importance Sampling of Variance-Covariance (SIR):
Goal Function=AVERAGE FITNESS OF PROPOSAL DENSITY".

As of nm75, SIRSAMPLE may contain a list of integers, one for each of the SRNITER iterations (use double quotes around the list):

SIRSAMPLE=”1000,200,500”

If there are fewer than SIRNITER values, the last value in the list will be used as the SIRSAMPLE value for the remaining iterations.

SIRNITER=1 (default) (NM74)

The number of times you want SIR sampling to be performed. While the proposal density for the first iteration comes from the main $COV step, subsequent iterations borrow the proposal density variance from the SIR sample variance of the previous iteration, centered around the SIR average of the parameters.

SIRCENTER=0 (default) (NM74)

Where the sampling (proposal) density is to be centered. On the first iteration, the mean of the sampling density is at the estimate. On subsequent iterations, the mean of the sampling density is at the estimate (SIRCENTER=0) or at the mean of the (transformed) samples of the previous iteration (SIRSAMPLE=1).

SIR_CAPCORR=1.0 (default) (NM75)

Limit the correlation of omegas and sigmas generated form the proposal density to have |correlation| of SIR_CAPCORR or less. This is similar to PSN’s cap_correlation option.

SIRMINWT=0.001, SIRMAXWT=1000.0 (default) (NM75)

Limit the weight range that a sample may have relative to the proposal density, to avoid extreme values.

IACCEPT=1 (default)(NM74)

The acceptance ratio acts similarly to importance sampling in EM analysis. For objective function profiles that are particularly heavy tailed (this could happen when there are relatively few subjects, and or data), then IACCEPT=0.4 may be more suitable, so the proposal density variance is made to be greater than the original variance-covariance, and more sampling is done farther away from the center (objective function minimum).

IACCEPTL=0 (default)(NM74)

The IACCEPTL option performs the same as listed for *IACCEPTL =0 (default) (NM74)* in section I.36 Monte Carlo Importance Sampling EM. It adds a normal density sampling scaled according to the IACCEPTL, alternating with normal density sampling scaled according to IACCEPT.

SIRDF=n (NM74)

The proposal density is to be a t distribution with n degrees of freedom. Default is 0, a normal density. You may wisth to utilize a t-distribution with SIRDF degrees of freedom, to provide heavy tail sampling. This has a similar purpose as setting IACCEPT less than 1.

SIRSEED= 11456 (default)(NM75)

Starting seed for SIR analysis.

SIRCLOCKSEED=0 (default) (nm75)

As of nm75, the actual starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if SIRCLOCKSEED=1. This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication.

RANMETHOD=blank (default)(NM74)

The RANMETHOD settings for the SIR analysis perform the same actions as described for RANMETHOD in the Importance Sampling EM section. See *RANMETHOD=[n|M|S|m|P] (default n=3)* in I.36 Monte Carlo Importance Sampling EM. If RANMETHOD with a 'P' attribute is set, then parallelization occurs by the usual parsing of subjects among the nodes, and retaining the seed pattern for each subject, so that the sample pattern is re-produced regardless of how the job is split up. As of NM75, if RANMETHOD does not have the 'P' attribute, then parallelization occurs by parsing the SIRSAMPLE objective function evaluations evenly among the nodes. In such a case, the sampling pattern cannot be made to be reproducible. There are efficiency advantages to splitting the job up among the samples rather than among subjects within each objective function (OFV) evaluation, if the OFV is rapidly calculated, or there are fewer subjects for parsing than there are nodes available.

SIRPARAFPRINT=1 (default, NM75)

The print iteration intervals to the parallelization log file can be controlled by this option during parallelization of the SIR analysis.

SIRPARAFILE=OFF (NM75)

Specifically turn off parallelization during the SIR analysis.

SIRPRINT=0 (default)(NM74)

Set the console print iterations interval. This does not impact the iterations listed in the raw output file.

FILE=blank (default)(NM74)

By default, the raw output file is whatever was listed in the $EST step, or root.ext, where root is the root name of the control stream file. You can re-direct SIR sample listings to an alternative file with this option.

FORMAT=s1PE12.5 (default)(NM75)

By default, the raw output file format is whatever was listed in the $EST step, or s1PE12.5. You can change its format with the above option. The first character defines the delimiter, which may be s for space, t for tab, comma for a comma delimited file with aligned fields (so, padded with spaces), c for comma delimited file with no spaces, or q for comma delimited file with no spaces and double quotes around column names that have commas (such as "OMEGA(2,1)").

If you specify FORMAT=QCSV then this is equivalent to FORMAT=q1PG23.16
If you specify FORMAT=CSV then this is equivalent to FORMAT=c1PG23.16

SIRTHBND=THBND (default) (NM74)

As with the deterministic covariance assessment step, when SIRTHBND=1, the transformed parameters are sampled, so that no sample is below the $THETA lower bound specification, and no higher than the $THETA upper bound specification. To allow a boundariless search in the original theta domain, SIRTHBND should be 0. By default, SIRTHBND is the value of THBND, which in turn is 1 by default.

Using $RCOV with SIR sampling (NM75)

You can use $RCOV (or $RCOVI) (see I.67 $RCOV and $RCOVI Record For Inputting Variance-Covariance information from another problem (NM75))) to introduce a covariance of estimates matrix from a *.cov file of any previous analysis, or generated by any other means. The following shows how one could generate the covariance matrix by the usual $COV step from one control stream problem, say rcov.ctl:

```
…
$EST METHOD=1 INTERACTION NSIG=3 PRINT=10 NOABORT MAXEVAL=9999
     FORMAT=,1PE23.15
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

Which generates rcov.cov, rcov.ext, and rcov.phi. Thes files in turn can be used in a subsequent analysis (Suppressing estimation with FNLETA=2):

```
…
$CHAIN FILE=rcov.ext NSAMPLE=0 ISAMPLE=-1000000000 FORMAT=,1PE23.15
$RCOV FILE=rcov.cov TBLN=1 DELIM=,
$ETAS FILE=rcov.phi TBLN=1 DELIM=,
$EST METHOD=1 INTERACTION NSIG=3 PRINT=10 NOABORT MAXEVAL=9999 FNLETA=2
     FORMAT=,1PE23.15
$COV MATRIX=R UNCONDITIONAL SIRSAMPLE=1000 NITER=3
```

Thus the $RCOV record serves the same purpose as the option -covmap_input in Perl-Speaks-NONMEM.

Preconditioning the R Matrix to Improve Precision and Success Rate of $COV Step (NM74)

Preconditioning of the R matrix, as described by Aoki et al [??], and implemented in Perl Speaks NONMEM, can improve the precision, as well as increase the chance of success of the $COV step.

PRECOND =0 (default) (NM74)

By default, PRECOND is 0, and no preconditioning of the R matrix is performed. When PRECOND=n, then up to n preconditioning cycles are performed. This is used in combination with the PFCOND setting.

PRECONDS =TOS (default) (NM74)

By default, if preconditioning is performed, it is done on Thetas (T), Omegas (O), and Sigmas(S). Specify PRECONDS=T to do only thetas, PRECONDS=TO to do only thetas and omegas, etc.

PFCOND =0 (default) (NM74)

PFCOND means "forced" preconditioning. Preconditioning occurs exactly PFCOND times, without testing if the R matrix is positive definite or not on each preconditioning cycle. On the remaining PRECOND-PFCOND cycles, the R matrix is tested for positive definiteness, and upon success, will terminate the preconditioning cycles.

PRETYPE =0 (default) (NM74)

By default (PRETYPE=0), the R matrix corrector is V*square_root(eigenvalue), as described in Aoki. If you set PRETYPE=1, then corrector is V*square_root(eigenvalue)*Vtranspose. If you set PRETYPE=2, then corrector is the correlation version of PRETYPE=1.

FPOSDEF =0 (default) (NM74)

If FPOSDEF>0, then if the R matrix is not positive definite, it will be forced positive definite. If PRECOND>0, this will occur after the PRECONDth try. The FPOSDEF operates on preconditioned R matrix, while POSDEF described earlier operates on any non-preconditioned R matrix, that is, when MATRIX when PRECOND=0.

The positive definite algorithm used is as follows. If FPOSDEF=1, then if the R matrix is not positive definite, it will be forced positive definite, by method of shifting the eigenvalues by an amount slightly greater than the most negative eigenvalue. If FPOSDEF=2, then R matrix is made positive definite by making the negative valued eigenvalues approximately 1/100 of the

least positive eigenvalue. If FPOSDEF=3, then R matrix is made positive definite by using the absolute values of the eigenvalues. Setting FPOSDEF=3 appears to be the best method here.

I.57 A Note on Covariance Diagnostics

There are several conditions that can occur in assessing the variance-covariance matrix of the estimates, which are best defined according to eigenvalues that it detects in them.

1) Positive definite means there are only positive eigenvalues. NONMEM outputs proper variance-variance matrices.
2) Non-positive definite means there is at least one eigenvalue that is less than or equal to zero.
3) Positive-semidefinite means there are no negative eigenvalues, but at least one zero valued eigenvalue (singular).
4) Non-positive-semidefinite means there is at least one negative eigenvalue.
5) Non-positive-semidefinite and singular means there is at least one negative eigenvalue, and at least one zero valued eigenvalue. Non-inverted matrices may be outputted by NONMEM.
6) Non-positive-semidefinite and non-singular means there is at least one negative eigenvalue, and no zero valued eigenvalue. Alternative diagnostic matrices may be outputted by NONMEM.
7) Negative-definite means there are only negative eigenvalues.
8) Non-negative-definite means there is at least one eigenvalue that is greater than or equal to zero.

NONMEM tests for conditions 1), 5), and 6), and outputs appropriate result matrices, or diagnostic matrices, as it is able.

Alternative expressions would be unsuitable to describe the condition of the matrices. For example, non-positive-definite (2) does not mean the same as positive-semi-definite (3). Similarly, non-positive-definite (2) is not exactly the same as non-positive-semidefinite (4). The set of non-negative-definite matrices (8) includes matrices that are positive-definite (1), positive-semi-definite (3), and a subset of non-positive-semidefinite (4) not including those with all negative eigenvalues.

I.58 Adding Nested Random Levels Above Subject ID (NM75)

Note: The following section provides an improved method of modeling random levels above subject ID that differs from NONMEM 7.4 and earlier. This method will give similar results, but will provide standard errors to thetas that reflect the variability in the highest random level, rather than at the lowest, subject level. It does this by pointing the Mu referencing to the eta of the highest random level, rather than at the individual level, and preventing level etas summing to 0 (centering), by setting $EST LEVCENTER=0.

Suppose you wish to model inter-site variability, or inter-trial variability, so that several subjects belong to a trial. An easy, method would be to use the $LEVEL feature. Consider the following control stream fragment, which in addition to inter-subject variability eta(1) for clearance (CL), there is inter-site variability eta(5) :

```
$PK
```

```
MU_5=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_5+ETA(5)+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

...
$LEVEL
SID=(5[1])

…
$EST METHOD=ITS INTERACTION PRINT=1 NSIG=2 NITER=500 SIGL=8 FNLETA=0 NOABORT
CTYPE=3 MCETA=0
LEVCENTER=0 ; Do not force level etas to sum to 0
$EST METHOD=1 INTERACTION PRINT=5 NSIG=2 NBURN=1000 NITER=500 SIGL=10
FNLETA=0 NOHABORT SLOW NONINFETA=1 MCETA=20
$COV MATRIX=R UNCONDITIONAL SIGL=10
```

Note that LEVCENTER=0 so as not to force level etas to sum to 0. Notice also that FNLETA=0 is set, so that the etas reflect what were used in the estimation. If FNELTA=0 is not set, super ID eta values outputted using $TABLE will incorrectly differ with each subject, rather than averaged for each LEVEL item value.

Let us suppose that the data item named SID is the site ID. NONMEM needs to know that SID is to be associated with eta(5), and in turn eta(1) is nested within eta(5). The data file need not be sorted for super ID values. The $LEVEL record gives this information:

```
$LEVEL
SID=(5[1])
```

such that SID is a super ID data item associated with eta(5) (inter-site eta), and eta(1) nests within eta(5) (5[1]). NONMEM will then perform appropriate summary statistics for eta(5), and make the appropriate constraints on eta(5), so eta(5) changes by site, that is, by every SID value change, and not by every ID value change. You may have additional parameters having site variability etas and their suitable nesting etas, such as for V1, Q, and V2:

```
$PK
MU_5=THETA(1)
MU_6=THETA(2)
MU_7=THETA(3)
MU_8=THETA(4)
CL=DEXP(MU_5+ETA(5)+ETA(1))
V1=DEXP(MU_6+ETA(6)+ETA(2))
Q=DEXP(MU_7+ETA(7)+ETA(3))
V2=DEXP(MU_8+ETA(8)+ETA(4))
S1=V1

...
$LEVEL
SID=(5[1],6[2],7[3],8[4])
```

Perhaps in addition to SID, you have country ID, let's call that data item CID. Perhaps there are several sites belonging to one country, some other sites belonging to another country, etc. This would provide a nesting level of 2 above that of ID, and is expressed as follows, for example (..\examples\superid2_*.ctl):

```
$PK
MU_9=THETA(1)
MU_10=THETA(2)
```

```
MU_11=THETA(3)
MU_12=THETA(4)
CL=DEXP(MU_9+ETA(9)+ETA(5)+ETA(1))
V1=DEXP(MU_10+ETA(10)+ETA(6)+ETA(2))
Q=DEXP(MU_11+ETA(11)+ETA(7)+ETA(3))
V2=DEXP(MU_12+ETA(12)+ETA(8)+ETA(4))
S1=V1

...
$LEVEL
SID=(5[1],6[2],7[3],8[4])
CID=(9[5],10[6],11[7],12[8])
```

Thus, for clearance, eta(9) is the country variability that has nested in it the site variability eta(5), which in turn has nested in it the subject variability (the standard ID data) eta(1). When performing FOCE with $LEVEL, you must use the SLOW option in $EST, and MATRIX=R for the covariance step $COV should be selected. Notice in these examples the thetas are mu referenced to the highest random level.

Sometimes you may wish to have inter-site variability, but not inter-subject variability. In the following example, residual SD is to change only with each site, and the inter-subject variance is set to 0:

```
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_4=THETA(3)
CL=DEXP(MU_1+ETA(1))
V=DEXP(MU_2+ETA(2))
W=DEXP(MU_4+ETA(4)+ETA(3))
S1=V

$ERROR
IPRED=F
Y = F + F*W*EPS(1)

; Initial values of THETA
$THETA 0.5  1.0 -.5
;INITIAL values of OMEGA
$OMEGA BLOCK(2)
0.02
0.001 0.02

$OMEGA
(0.0 FIXED)

$OMEGA
0.2

$SIGMA
1.0 FIXED

$LEVEL
SID=(4[3])
```

As of nm74, instead of making a nesting connection to an eta of 0 variance, you can reference eta 0, which tells NONMEM that there is no variability for that level underneath. Thus the above model can be re-written to:

```
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
CL=DEXP(MU_1+ETA(1))
V=DEXP(MU_2+ETA(2))
W=DEXP(MU_3+ETA(3))
S1=V

$ERROR
IPRED=F
Y = F + F*W*EPS(1)

; Initial values of THETA
$THETA 0.5  1.0 -0.8
;INITIAL values of OMEGA

$OMEGA BLOCK(2)
0.02
0.001 0.02

$OMEGA
0.2

$SIGMA
1.0 FIXED

$LEVEL
SID=(3[0])
```

with no need to have an extra ETA as a dummy variance. Thus, NONMEM is informed that eta(3) is a random effect to W, associated with changes in SID, rather than the usual ID, and no other eta shares the variability for that parameter.

As of NM74, the syntax to giving the level mapping may use the TO(:)/BY pattern method. For example,

```
SID=(5[1],6[2],7[3],8[4])
CID=(9[5],10[6],11[7],12[8])
```

may be expressed as:

```
SID=(5:8[1])
CID=(9:12[5])
```

So, 5:8[1] means etas 5 to 8 nest into etas beginning at 1. The “to” value in the brackets is not necessary. However, you may wish to express a by step:

```
SID=(5:11by3[12])
```

means etas 5 to 11, by steps of 3, nest into etas beginning at 12, also by steps of 3. So, etas 5,8,11 map into etas 12,15,18, respectively. If the by step value must differ between the two lists, then the by value must be given in both lists:

```
SID=(5:11by3[12by4])
```

So, etas 5,8,11 nest into 12,16,20, respectively. Negative steps may also be used:

```
SID=(11:5by-3[20])
```

means etas 11, 8, 5 map into etas starting at 20, also by steps of -3 (20, 17, 14).

Commas should only be used to separate x[y] structures, but commas may not appear within x or within y. Also, x may have a to(:)/BY pattern and y need only have a value indicating a starting number of the nested etas, as given in the examples above, but the converse may not be true. That is, the y may imply its to/by pattern from x, but x may not imply its to/by pattern from y.

Nesting below the subject ID as for previous versions of NONMEM, as shown for inter-occasion variability, example 7.

This method using $LEVEL is approximate to modeling data such that the SID data item is renamed as ID, and the the original ID data item is renamed as OCC (for occasion). Then, the 16 SID levels become 16 “super “ subjects, each super subject having 50 occasions to represent the original 50 subjects within each level. This of course places considerable computational demand, and executes much more slowly than the $LEVEL method. The following example (superid3_20.ctl) shows the layout of how this would be done. Notice the $OMEGA … SAME(49) line, to allow for 49 additional occasions (formerly subjects) within each subject (formerly SID) sharing the repeated 49 omega blocks to represent “inter-occasion” (formerly inter-subject) variability.

```
$SIZES LVR=160
$PROB RUN#
$ABBR DERIV2=NO
$INPUT C OCC TIME DV AMT RATE EVID MDV CMT ROWNUM ID
$DATA superid3_occ.csv IGNORE=C

$SUBROUTINES ADVAN2 TRANS2
$ABBR REPLACE ETA(OCC_KA)=ETA(,4 TO 151 BY 3)
$ABBR REPLACE ETA(OCC_CL)=ETA(,5 TO 152 BY 3)
$ABBR REPLACE ETA(OCC_V)=ETA(,6 TO 153 BY 3)

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)

KA=DEXP(MU_1+ETA(1)+ETA(OCC_KA))
CL=DEXP(MU_2+ETA(2)+ETA(OCC_CL))
V=DEXP(MU_3+ETA(3)+ETA(OCC_V))
S2=V

$ERROR
IPRE=F
```

```
Y = IPRE + IPRE*EPS(1)

$THETA  1.79067E-01 -5.31302E+00 -3.08220E+00

$OMEGA BLOCK(3)
 3.14841E-02
-5.55179E-03  2.64664E-02
 2.73335E-02 -6.08426E-03  5.23413E-02

$OMEGA BLOCK(3)
 1.00860E-02
 1.48793E-04  9.80457E-03
 5.38548E-04  6.57755E-04  9.72772E-03

$OMEGA BLOCK(3) SAME(49)

$SIGMA
 2.99892E-03      ;[P]

$EST METHOD=ITS INTERACTION PRINT=1 NSIG=2 NITER=0 SIGL=8 FNLETA=0
     NOABORT CTYPE=3 MCETA=0
$COV MATRIX=R UNCONDITIONAL SIGL=10
```

The equivalent model using $LEVEL would be as follows (superid3_21.ctl):

```
$PROB RUN#
$INPUT C ID TIME DV AMT RATE EVID MDV CMT ROWNUM SID
$DATA superid3.csv IGNORE=C

$SUBROUTINES ADVAN2 TRANS2

$PK
MU_4=THETA(1)
MU_5=THETA(2)
MU_6=THETA(3)
KA=DEXP(MU_4+ETA(4)+ETA(1))
CL=DEXP(MU_5+ETA(5)+ETA(2))
V=DEXP(MU_6+ETA(6)+ETA(3))
S2=V

$ERROR
IPRE=F
Y = IPRE + IPRE*EPS(1)

$THETA 0.2 -4 -2
$OMEGA BLOCK(3)
0.1
0.001 0.1
0.001 0.001 0.1

$OMEGA BLOCK(3)
0.3
0.001 0.3
0.001 0.001 0.3

;Initial value of SIGMA
```

```
$SIGMA
0.1     ;[P]

$LEVEL
SID=(4[1],5[2],6[3])

$EST METHOD=ITS INTERACTION PRINT=1 NSIG=2 NITER=500 SIGL=8 FNLETA=0 NOABORT
CTYPE=3 MCETA=0
$EST METHOD=1 INTERACTION PRINT=5 NSIG=2 NBURN=1000 NITER=500 SIGL=10
FNLETA=0 NOHABORT SLOW NONINFETA=1 MCETA=20
$COV MATRIX=R UNCONDITIONAL SIGL=10
```

LEVWT=0 (default) (NM74)

By default, LEVWT=0, and weights each level value equally, regardless of number of subjects per level value. If you wish to weight according to number of subjects for that value, set LEVWT=1 on the $EST record.

LEVCENTER=[0|1] (no default) (NM75)

If LEVCENTER=1, this ensures the etas of super ID random levels sum to 0. In earlier versions of NONMEM, the default (and only) action was to ensure the etas of super ID random levels sum to 0. To obtain similar results as earlier versions of NONMEM, set LEVCENTER=1 on the $EST record.

LEVOBJTYPE=(0 default) (NM75)

If LEVOBJTYPE=0, then for IMP/ITS/FOCE/Laplace, the reported ojective function (and what is used for estimation) is the average of two cycles of optimization. The first optimization floats all etas, and the second optimization fixes the super-ID etas to the appropriate average of etas per super-id group from the first optimization, and then floats the ID (and below) level etas. If LEVOBJTYPE=1, then the OBJ reported (and used for estimation) is only that of the second optimization cycle. The LEVOBJTYPE=0 provides the most consistent results between FOCE/Laplace and the EM and Bayes methods, particularly with super-ID levels higher than 1 (so, site ID and country ID, for example). The LEVOBJTYPE=1 provides an OBJ more consistent with theory.

By default, NONMEM sets the estimation process to SLOW when a $LEVEL option is used. As of nm751, you may set NOSLOW or FAST. While there is an increase in speed, but there may not be as good optimization progress as with the SLOW option. It may be best to use SLOW for the $EST step, and use NOSLOW or FAST for the $COV step.

I.59 Adding Nested Random Levels Above Subject ID in Versions NM73 and NM74

In previous versions of NONMEM provding the $LEVEL feature, the paradigm used to model nested random levels above subject ID differed from the method described in the previous section in in two ways. First, the MU referencing pointed to the individual (ID) level rather than

the top random level. Secondly, the etas to random levels above ID were constrained to sum to 0, with the OMEGA estimation corrected for the loss of 1 degree of freedom. This older method applies some additional constraints on the model, and offers greater stability, especially for EM/BAYES methods, if the number of distinct items in a level are few (<5).

This original method of $LEVEL modeling was desgined to replicate the following paradigm. Consider defining separate thetas for each value pertaining to a super ID data item, so that theta is shared only by the subjects with the particular SID value. This is suitable if there are not too many distinct values of the super ID data item, otherwise, the number of thetas can become very large, and the analysis may take a considerable amount of time. This analysis method could be performed in earlier versions of NONMEM, but the many thetas that needed to be mapped with the different levels could make the NMTRAN code quite large and tedious to write. Fortunately since NM73, a series of substitution variable techniques and short-hand entries for initial values can be used, and this method is now easier to program in NMTRAN.

Here is an example to code using separate thetas pertaining to each value of the SID data item (example superid3_6):

```
$SIZES LTH=60
$PROB RUN#
$INPUT C ID TIME DV AMT RATE EVID MDV CMT ROWNUM SID TYPE L2
$DATA superid3_6.csv IGNORE=C

$SUBROUTINES ADVAN2 TRANS2
$ABBR REPLACE THETA(SID_KA)=THETA(,4 to 19)
$ABBR REPLACE THETA(SID_CL)=THETA(,20 to 35)
$ABBR REPLACE THETA(SID_V)=THETA(,36 to 51)
$ABBR DECLARE DOWHILE I
$ABBR DECLARE INTEGER NSID

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
NSID=16
THSUM_KA=0.0
THSUM_CL=0.0
THSUM_V=0.0
I=1
DO WHILE (I<=NSID)
THSUM_KA=THSUM_KA-THETA(I+3)
THSUM_CL=THSUM_CL-THETA(I+19)
THSUM_V=THSUM_V-THETA(I+35)
I=I+1
ENDDO

IF(SID<NSID) THEN
KA=DEXP(MU_1+ETA(1)+THETA(SID_KA))
CL=DEXP(MU_2+ETA(2)+THETA(SID_CL))
V=DEXP(MU_3+ETA(3)+THETA(SID_V))
ELSE
; for the last SID level, NSID, use the negative sum of the thetas of the other SID levels,
; so that the sum of all thetas is 0, that is, the super-nested average theta is 0.
KA=DEXP(MU_1+ETA(1)+THSUM_KA)
CL=DEXP(MU_2+ETA(2)+THSUM_CL)
V=DEXP(MU_3+ETA(3)+THSUM_V)
ENDIF

S2=V

$ERROR
IPRE=F
```

```
IF(TYPE==0) Y = IPRE + IPRE*EPS(1)
IF(TYPE==1.AND.SID<NSID) Y=THETA(SID_KA)+EPS(2) ; The fitting of the pseudo-data (TYPE>0)
IF(TYPE==1.AND.SID==NSID) Y=THSUM_KA+EPS(2)     ; constrains the SID level thetas to be
IF(TYPE==2.AND.SID<NSID) Y=THETA(SID_CL)+EPS(3) ; constrained, and modeled using extra
IF(TYPE==2.AND.SID==NSID) Y=THSUM_CL+EPS(3)     ; Sigma variances 2-4.
IF(TYPE==3.AND.SID<NSID) Y=THETA(SID_V)+EPS(4)
IF(TYPE==3.AND.SID==NSID) Y=THSUM_V+EPS(4)

$THETA 0.2 -4 -2
       (0.1)x15 (0.0 FIXED)
       (0.1)x15 (0.0 FIXED)
       (0.1)x15 (0.0 FIXED)

$OMEGA BLOCK(3) VALUES(0.1,0.001)

$SIGMA
0.1     ;[P]

$SIGMA BLOCK(3) VALUES(0.3,0.001) ; This is the inter-SID variance.

$EST METHOD=1 INTERACTION PRINT=1 NSIG=2 SIGL=10 FNLETA=0 NOHABORT NONINFETA=1 MCETA=20
$COV MATRIX=R UNCONDITIONAL SIGL=10
```

Notice the use of variable replacement mapping ($ABBR REPLACE), short-hand entries for initial thetas, omegas, and sigmas, and that the sum of the thetas to the SID data item are fixed to 0 by constraining the theta pertaining to the highest SID value (NSID) to be the negative sum of the thetas to the other SID values (1 through NSID-1) using a DOWHILE loop.

For this method, some pseudo-data must be added to the data file:

Original data portion (TYPE=0):

```
C       ,       ID,     TIME,       DV,     AMT,    RATE,     EVID,     MDV,     CMT,  ROWNUM,     SID,_    TYPE,  L2
0.00E+00,1.00E+00,0.00E+00,0.00E+00,1.00E+00,0.00E+00,1.00E+00,1.00E+00,1.00E+00,1.00E+00,1.00E+00,0.00E+00,1.00E+00
0.00E+00,1.00E+00,1.00E-01,2.44E+00,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,2.00E+00,1.00E+00,0.00E+00,2.00E+00
0.00E+00,1.00E+00,2.00E-01,4.45E+00,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,3.00E+00,1.00E+00,0.00E+00,3.00E+00
0.00E+00,1.00E+00,5.00E-01,9.93E+00,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,4.00E+00,1.00E+00,0.00E+00,4.00E+00
0.00E+00,1.00E+00,1.00E+00,1.65E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,5.00E+00,1.00E+00,0.00E+00,5.00E+00
0.00E+00,1.00E+00,2.00E+00,2.05E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,6.00E+00,1.00E+00,0.00E+00,6.00E+00
0.00E+00,1.00E+00,5.00E+00,1.82E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,7.00E+00,1.00E+00,0.00E+00,7.00E+00
0.00E+00,1.00E+00,1.00E+01,7.20E+00,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,8.00E+00,1.00E+00,0.00E+00,8.00E+00
0.00E+00,1.00E+00,2.00E+01,1.29E+00,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,9.00E+00,1.00E+00,0.00E+00,9.00E+00
0.00E+00,1.00E+00,5.00E+01,6.80E-03,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+01,1.00E+00,0.00E+00,1.00E+01
0.00E+00,1.00E+00,1.00E+02,1.42E-06,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.10E+01,1.00E+00,0.00E+00,1.10E+01
0.00E+00,2.00E+00,0.00E+00,0.00E+00,1.00E+00,0.00E+00,1.00E+00,1.00E+00,2.00E+00,1.20E+01,1.00E+00,0.00E+00,1.00E+00
0.00E+00,2.00E+00,1.00E-01,2.73E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.30E+01,1.00E+00,0.00E+00,2.00E+00
0.00E+00,2.00E+00,2.00E-01,2.79E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.40E+01,1.00E+00,0.00E+00,3.00E+00
0.00E+00,2.00E+00,5.00E-01,2.68E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.50E+01,1.00E+00,0.00E+00,4.00E+00
0.00E+00,2.00E+00,1.00E+00,2.32E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.60E+01,1.00E+00,0.00E+00,5.00E+00
0.00E+00,2.00E+00,2.00E+00,1.74E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.70E+01,1.00E+00,0.00E+00,6.00E+00
0.00E+00,2.00E+00,5.00E+00,1.30E+01,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.80E+01,1.00E+00,0.00E+00,7.00E+00
```

…

Added data portion (TYPE=1,2,3), to provide variance constrained among the SID values, and bind it to the inter-SID $SIGMA variance :

```
C       ,       ID,     TIME,       DV,     AMT,    RATE,     EVID,     MDV,     CMT,  ROWNUM,     SID,    TYPE,  L2
0.00E+00,8.01E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+00,1.00E+00,1.00E+00
0.00E+00,8.01E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+00,2.00E+00,1.00E+00
0.00E+00,8.01E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+00,3.00E+00,1.00E+00
0.00E+00,8.02E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,2.00E+00,1.00E+00,1.00E+00
0.00E+00,8.02E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,2.00E+00,2.00E+00,1.00E+00
0.00E+00,8.02E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,2.00E+00,3.00E+00,1.00E+00
0.00E+00,8.03E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,3.00E+00,1.00E+00,1.00E+00
0.00E+00,8.03E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,3.00E+00,2.00E+00,1.00E+00
0.00E+00,8.03E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,3.00E+00,3.00E+00,1.00E+00
0.00E+00,8.04E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,4.00E+00,1.00E+00,1.00E+00
0.00E+00,8.04E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,4.00E+00,2.00E+00,1.00E+00
0.00E+00,8.04E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,4.00E+00,3.00E+00,1.00E+00
0.00E+00,8.05E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,5.00E+00,1.00E+00,1.00E+00
0.00E+00,8.05E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,5.00E+00,2.00E+00,1.00E+00
0.00E+00,8.05E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,5.00E+00,3.00E+00,1.00E+00
0.00E+00,8.06E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,6.00E+00,1.00E+00,1.00E+00
0.00E+00,8.06E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,6.00E+00,2.00E+00,1.00E+00
0.00E+00,8.06E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,6.00E+00,3.00E+00,1.00E+00
0.00E+00,8.07E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,7.00E+00,1.00E+00,1.00E+00
0.00E+00,8.07E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,7.00E+00,2.00E+00,1.00E+00
0.00E+00,8.07E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,7.00E+00,3.00E+00,1.00E+00
```

```
0.00E+00,8.08E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,8.00E+00,1.00E+00,1.00E+00
0.00E+00,8.08E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,8.00E+00,2.00E+00,1.00E+00
0.00E+00,8.08E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,8.00E+00,3.00E+00,1.00E+00
0.00E+00,8.09E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,9.00E+00,1.00E+00,1.00E+00
0.00E+00,8.09E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,9.00E+00,2.00E+00,1.00E+00
0.00E+00,8.09E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,9.00E+00,3.00E+00,1.00E+00
0.00E+00,8.10E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+01,1.00E+00,1.00E+00
0.00E+00,8.10E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+01,2.00E+00,1.00E+00
0.00E+00,8.10E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.00E+01,3.00E+00,1.00E+00
0.00E+00,8.11E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.10E+01,1.00E+00,1.00E+00
0.00E+00,8.11E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.10E+01,2.00E+00,1.00E+00
0.00E+00,8.11E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.10E+01,3.00E+00,1.00E+00
0.00E+00,8.12E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.20E+01,1.00E+00,1.00E+00
0.00E+00,8.12E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.20E+01,2.00E+00,1.00E+00
0.00E+00,8.12E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.20E+01,3.00E+00,1.00E+00
0.00E+00,8.13E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.30E+01,1.00E+00,1.00E+00
0.00E+00,8.13E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.30E+01,2.00E+00,1.00E+00
0.00E+00,8.13E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.30E+01,3.00E+00,1.00E+00
0.00E+00,8.14E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.40E+01,1.00E+00,1.00E+00
0.00E+00,8.14E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.40E+01,2.00E+00,1.00E+00
0.00E+00,8.14E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.40E+01,3.00E+00,1.00E+00
0.00E+00,8.15E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.50E+01,1.00E+00,1.00E+00
0.00E+00,8.15E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.50E+01,2.00E+00,1.00E+00
0.00E+00,8.15E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.50E+01,3.00E+00,1.00E+00
0.00E+00,8.16E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.60E+01,1.00E+00,1.00E+00
0.00E+00,8.16E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.60E+01,2.00E+00,1.00E+00
0.00E+00,8.16E+02,0.00E+00,1.00E-12,0.00E+00,0.00E+00,0.00E+00,0.00E+00,2.00E+00,1.00E+00,1.60E+01,3.00E+00,1.00E+00
```

The idea in doing this is to cause the following term to be added to the objective function:

$$\sum_{i=1}^{N_{SID}} [\boldsymbol{\theta}_i' \boldsymbol{\Sigma}_\theta^{-1} \boldsymbol{\theta}_i + \log(\boldsymbol{\Sigma}_\theta)]$$

Where $\boldsymbol{\theta}_i$ is the vector of SID thetas, and $\boldsymbol{\Sigma}_\theta$ is the variance among the SID thetas. For the above example, $\boldsymbol{\theta}_i$ is a 3x1 vector, one element each for KA (TYPE=1), CL (TYPE=2), and V (TYPE=3), for i=1 to NSID, where NSID is the number of possible values of SID, which in this example NSID=16. The $\boldsymbol{\Sigma}_\theta$ matrix is the 3x3 block matrix to Epsilons 2,3, and 4. NONMEM is fooled into constructing the above term by use of the additional data records for which DV_{ij}=0 (or nearly so), for which are modeled $IPRED_{ij}$=theta(3+(TYPE-1)*j+i), for i=1 to 16 SID values, and j=1 to 3 TYPE values. NONMEM thus adds, for each TYPE>0 data record, objective function value terms $(DV_i - IPRED_i)\Sigma^{-1}(DV_i - IPRED_i)$ that evaluates to $\boldsymbol{\theta}_i' \boldsymbol{\Sigma}_\theta^{-1} \boldsymbol{\theta}_i$, and the control stream file places a dependency of the last $\boldsymbol{\theta}_i$ of each element (that is, each of the three TYPE's) such that $\sum_{i=1}^{N_{SID}} \boldsymbol{\theta}_i = \mathbf{0}$. The L2 data item allows NONMEM to assess correlation (hence off-diagonal elements to the SIGMA block) between the three TYPEs, within a given SID. Thus for the added data portion, NONMEM sees 16 "subjects", one for each of the SID values, each of which have 3 "data points", one for each PK parameter (TYPE).

The above problem can alternatively be coded more easily using the $LEVELS mapping of etas as follows (example superid3_1), without needing to add pseudo data to the data file:

```
$PROB RUN#
$INPUT C ID TIME DV AMT RATE EVID MDV CMT ROWNUM SID
$DATA superid3.csv IGNORE=C

$SUBROUTINES ADVAN2 TRANS2

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
KA=DEXP(MU_1+ETA(1)+ETA(4))
CL=DEXP(MU_2+ETA(2)+ETA(5))
V=DEXP(MU_3+ETA(3)+ETA(6))
S2=V
```

```
$ERROR
IPRE=F
Y = IPRE + IPRE*EPS(1)

; Initial values of THETA
$THETA 0.2 -4 -2
;INITIAL values of OMEGA
$OMEGA BLOCK(3)
0.1
0.001 0.1
0.001 0.001 0.1

$OMEGA BLOCK(3) ; Inter-SID variance
0.3
0.001 0.3
0.001 0.001 0.3

;Initial value of SIGMA
$SIGMA
0.1     ;[P]

$LEVEL
SID=(4[1],5[2],6[3])

$EST METHOD=ITS INTERACTION PRINT=1 NSIG=2 NITER=500 SIGL=8 FNLETA=0 NOABORT CTYPE=3 MCETA=0
     LEVCENTER=1
$EST METHOD=IMP INTERACTION PRINT=1 NSIG=2 NITER=500 SIGL=8 FNLETA=0 NOABORT CTYPE=3 MCETA=0
             ISAMPLE=300 MAPITER=0
$EST METHOD=SAEM INTERACTION PRINT=10 NSIG=2 NITER=100 SIGL=8 FNLETA=0 NOABORT CTYPE=3 MCETA=0
             ISAMPLE=2 CONSTRAIN=0
$EST METHOD=IMP EONLY=1 INTERACTION PRINT=1 NSIG=2 NITER=5 SIGL=8 FNLETA=0 NOABORT CTYPE=3
             MCETA=0 ISAMPLE=300 MAPITER=0
$EST METHOD=BAYES INTERACTION PRINT=10 NSIG=2 NBURN=1000 NITER=500 SIGL=8 FNLETA=0
             NOABORT CTYPE=3
$EST METHOD=1 INTERACTION PRINT=5 NSIG=2 NBURN=1000 NITER=500 SIGL=10 FNLETA=0 NOHABORT
             SLOW NONINFETA=1 MCETA=20
$COV MATRIX=R UNCONDITIONAL SIGL=10
```

Notice that the MU referencing points to the individual level (MU 1,2,3), rather at the SID level as was done in the previous section (superid_21.ctl, MU 4,5,6). Also, LEVCENTER=1 is set in the present example (superid3_1.ctl), to center (constrain) the super ID etas to 0 (this is done by default in earlier versions of NONMEM, but in nm75, you must set LEVCENTER=1 to obtain similar results as in earlier versions).

Both methods (LEVCENTER=0 or LEVCENTER=1) provde similar estimates of population parameters. However, the standard errors of the thetas in superid3_1 are smaller than from superid3_21. The standard errors in superid3_1 represent what one would get from information at the 800 subjects level, which is equiavalent to the error of THETA(1), THETA(2), and THETA(3) obtained in superid3_6, with the “difference of SIDS from overall mean” thetas 4 to 51 (or the OMEGA(4,4), OMEGA(5,5), OMEGA(6,6)) absorbing the larger standard error based on the fewer (only 16 of them) SID levels. In contrast, the standard errors of the thetas in superid3_21 incorporate the uncertainty due to variability between sites, and that there are only 16 of these sites. To get this standard error associated with site uncertainty from superid3_1, you can obtain this by dividing the site OMEGA by number of sites, and square-rooting it. For example, SQRT(OMEGA(4,4)/16)=site standard error of theta(1).

I.60 Model parameters as log t-Distributed in the Population (NM73)

Sometimes one may suspect that PK/PD model parameters are actually log t-distributed among the population, with degrees of freedom NU, instead of the usual log normal distributed. To simulate such data for a two compartment model as an example, consider the following control stream file, ..\examples\tdist6_sim.ctl:

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT SID
$DATA tdist_sim.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
NU=4.0
CLA=ETA(1)/SQRT(OMEGA(1,1))
V1A=ETA(2)/SQRT(OMEGA(2,2))
QQA=ETA(3)/SQRT(OMEGA(3,3))
V2A=ETA(4)/SQRT(OMEGA(4,4))
CLB=ETA(5)
V1B=ETA(6)
QQB=ETA(7)
V2B=ETA(8)
CLR=(CLA*CLA+CLB*CLB)/NU
V1R=(V1A*V1A+V1B*V1B)/NU
QQR=(QQA*QQA+QQB*QQB)/NU
V2R=(V2A*V2A+V2B*V2B)/NU
CL=EXP(MU_1+ETA(1)*SQRT((EXP(CLR)-1.0)/CLR))
V1=EXP(MU_2+ETA(2)*SQRT((EXP(V1R)-1.0)/V1R))
Q= EXP(MU_3+ETA(3)*SQRT((EXP(QQR)-1.0)/QQR))
V2=EXP(MU_4+ETA(4)*SQRT((EXP(V2R)-1.0)/V2R))
S1=V1

$ERROR
Y = F + F*EPS(1)

; Initial values of THETA
$THETA 1.68338E+00  1.58811E+00  8.12694E-01  2.37435E+00
;INITIAL values of OMEGA
$OMEGA BLOCK(4)
0.03
0.01  0.03
-0.006 0.01  0.03
0.01 -0.006  0.01 0.03

$OMEGA (1.0 FIXED) (1.0 FIXED) (1.0 FIXED) (1.0 FIXED)

$SIGMA
0.01

$SIMULATION (567811 NORMAL) (2933012 UNIFORM) ONLYSIMULATION SUBPROBLEMS=1
$TABLE  ID TIME CONC DOSE RATE EVID MDV CMT ETA1 ETA2 ETA3 ETA4 CL V1 Q V2
NOAPPEND ONEHEADER FILE=tdist6.csv  NOPRINT
```

The data file produced, tdist6.csv, will have CL, V1, Q, and V2 t-distributed among the 100 subjects, with NU degrees of freedom.

Now, to analyze the data, we may first analyze it by assuming a normal distribution, as in this control stream file, ..\examples\tdist6.ctl:

```
$PROB RUN# Example 1 (from samp5l)
```

```
$INPUT ID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA tdist6.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
NU=4.0
CL=EXP(MU_1+ETA(1))
V1=EXP(MU_2+ETA(2))
Q=EXP(MU_3+ETA(3))
V2=EXP(MU_4+ETA(4))
S1=V1

$ERROR
Y = F + F*EPS(1)

;$THETA 1.68338E+00  1.58811E+00  8.12694E-01  2.37435E+00
$THETA 2 2 2 2
$OMEGA BLOCK(4)
0.3
0.001 0.3
0.001 0.001 0.3
0.001 0.001 0.001 0.3

$SIGMA
0.3

$EST METHOD=ITS LAPLACE INTERACTION MAXEVAL=9999 PRINT=5 NOHABORT SIGL=8 CTYPE=3 NITER=200
$EST METHOD=IMP INTERACTION MAXEVAL=9999 PRINT=1 NOABORT ISAMPLE=3000 NITER=200 SIGL=8 DF=1
$EST METHOD=1 LAPLACE INTERACTION MAXEVAL=9999 PRINT=1 NOHABORT
$COV MATRIX=R UNCONDITIONAL
```

Note that Laplace is used for conditional estimation, since the posterior density will by quite a bit not normally distributed. For importance sampling a t-distribution proposal density is used, to approximately match the posterior density shape. The result will be thetas and sigmas that approximate the simulation values used, whereas the OMEGAS will be increased by a factor of about NU/(NU-2) (see [12], bottom of page 341).

When estimating in the manner in which it was simulated, the thetas, sigmas, and omegas will more closely match the simulated values (..\examples\tdist7.ctl):

```
$PROB RUN# Example 1 (from samp5l)
$INPUT ID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA tdist6.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
NU=4.0
CLA=ETA(1)/SQRT(OMEGA(1,1))
V1A=ETA(2)/SQRT(OMEGA(2,2))
QQA=ETA(3)/SQRT(OMEGA(3,3))
V2A=ETA(4)/SQRT(OMEGA(4,4))
;CLA=ETA(1)/0.173
;V1A=ETA(2)/0.173
;QQA=ETA(3)/0.173
;V2A=ETA(4)/0.173
CLB=ETA(5)
```

```
V1B=ETA(6)
QQB=ETA(7)
V2B=ETA(8)
CLR=(CLA*CLA+CLB*CLB)/NU
V1R=(V1A*V1A+V1B*V1B)/NU
QQR=(QQA*QQA+QQB*QQB)/NU
V2R=(V2A*V2A+V2B*V2B)/NU
DEL=1.0E-08
IF (CLR.GT.40.0) CLR=40.0
IF (V1R.GT.40.0) V1R=40.0
IF (QQR.GT.40.0) QQR=40.0
IF (V2R.GT.40.0) V2R=40.0
CLRQ=1.0
V1RQ=1.0
QQRQ=1.0
V2RQ=1.0
IF(CLR.GT.DEL) CLRQ=SQRT((EXP(CLR)-1.0)/CLR)
IF(V1R.GT.DEL) V1RQ=SQRT((EXP(V1R)-1.0)/V1R)
IF(QQR.GT.DEL) QQRQ=SQRT((EXP(QQR)-1.0)/QQR)
IF(V2R.GT.DEL) V2RQ=SQRT((EXP(V2R)-1.0)/V2R)
CL=EXP(MU_1+ETA(1)*CLRQ)
V1=EXP(MU_2+ETA(2)*V1RQ)
Q= EXP(MU_3+ETA(3)*QQRQ)
V2=EXP(MU_4+ETA(4)*V2RQ)
S1=V1

$ERROR
Y = F + F*EPS(1)

;$THETA 1.68338E+00  1.58811E+00  8.12694E-01  2.37435E+00
$THETA 2 2 2 2
$OMEGA BLOCK(4)
0.1
0.01 0.1
0.01 0.01 0.1
0.01 0.01 0.01 0.1

$OMEGA (1.0 FIXED) (1.0 FIXED) (1.0 FIXED) (1.0 FIXED)

$SIGMA
0.1

$EST METHOD=ITS INTERACTION MAXEVAL=9999 PRINT=5 NOHABORT SIGL=9 CTYPE=3 NITER=200
            NONINFETA=1 MCETA=10
$EST METHOD=IMP INTERACTION MAXEVAL=9999 PRINT=1 NOHABORT ISAMPLE=3000 NITER=200
     SIGL=9 DF=2 RANMETHOD=3S1P CTYPE=3 MCETA=10
$EST METHOD=1 INTERACTION MAXEVAL=9999 PRINT=1 NOHABORT NSIG=3 SIGL=9 NONINFETA=1 SLOW MCETA=30
$COV MATRIX=R UNCONDITIONAL
```

Note that constructions such as

```
CL=EXP(MU_1+ETA(1)*SQRT((EXP(CLR)-1.0)/CLR))
```

violate the strict MU_x+ETA(x) rule recommended for EM analysis, because the term

```
 SQRT((EXP(CLR)-1.0)/CLR)
```

is multiplied by ETA(1). Nonetheless for this example, the importance sampling works quite well. Note also that

```
SQRT((EXP(CLR)-1.0)/CLR)
```

approaches 1 as NU approaches infinity, and therefore the random effect of CL approaches normality.

The above method generates independent univariate (U) t-distributed samples, one for each of the PK parameters (CL, V1, Q, V2), but retaining the correlation imposed by the first OMEGA block on ETAS 1-4. An atlternative method is to generate multivariate (M) t-distributed sample vectors (see *Note on the t-Distribution Sampling Density (DF>0), and its Use With Sobol*

Method (RANMETHOD=S) in section I.36 Monte Carlo Importance Sampling EM for U versus M type t-samples), as follows (..\examples\tdist11_sim.ctl):

```
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
NU=4.0
CHISQ=SQRT(
(ETA(5)*ETA(5)+ETA(6)*ETA(6)+ETA(7)*ETA(7)+ETA(8)*ETA(8))/NU )
CL=EXP(MU_1+ETA(1)/CHISQ)
V1=EXP(MU_2+ETA(2)/CHISQ)
Q= EXP(MU_3+ETA(3)/CHISQ)
V2=EXP(MU_4+ETA(4)/CHISQ)
S1=V1
```

However, when it comes time to estimate, the above algorithm is unstable (tdist11.ctl). The U method algorithm of tdist7.ctl is more stable for extimation, and is able to accurately analyze even data generated with the M method of tdist11 (see ..\examples\tdist12.ctl), even though the two algorithms are not identical.

I.61 Format of NONMEM Report File

The format of the NONMEM report file has been slightly modified, with improvements to allow third party software to more easily identify portions of the result file. As described above, the user has now the ability to request a series of classical or new estimation methods within the same problem if he so chooses. Each of the new methods produces slightly different banner text and termination status text in the report file. For example, an iterative two stage analysis may be requested, followed by an MCMC Bayesian method, followed by an FOCEI method. The theta, sigma, and omega results of the iterative two stage method will be passed on as initial values for the MCMC Bayesian method, to facilitate the MCMC Bayesian analysis, which in turn can supply initial values for the FOCEI method. Each of these intermediate analyses will provide output to the NONMEM report file, and will be identified by unique text for that method. To allow a program to consistently find the appropriate positions in the file without having to search for specific words in the text, the report file is augmented with special tag labels that remain constant, regardless of the method used.

The tags always begin with #, followed by four letters to indicate the tag type, followed by a colon (:). The following tags are presently defined:

#PARA: (NM72)

This tag identifies the parallelization file and number of nodes used, if parallel estimation is performed.

#TBLN: (NM72)

This tag specifies that following it, on the same line, will be found an integer that refers to the number of this estimation method. This number is also the table number listed in the title to tables in the various output files (raw output file, .cov, .cor, etc). The table number is incremented for each $EST statement, across all problems in the control stream file.

#METH:

This tag specifies that following it, on the same line, will be found a text that describes the method, for example *First Order Conditional Estimation Method with Interaction*.

#TERM:

This tag indicates that beginning on the next lines, text describes the termination status of the analysis. Included in the results are average of the individual etas (ETABAR), its standard error (SE), P-value on the null hypothesis that ETABAR is not statistically different from 0, and eta and epsilon shrinkage. Shrinkage is not reported after an FO analysis. As of nm75, shrinkage information is reported for the BAYES method. See below for more information on shrinkage.

The individual etas used to assess ETABAR/SE/p-value/Shrinkage are modes of the posterior density for ITS/FOCE/Laplace for each individual, or conditional mean etas for IMP/SAEM for each individual, as of the last iteration.

ETABAR, SE, P-Value, and Shrinkage are not always accurately calculated after an SAEM analysis, as these are averaged over the entire set of iterations of the reduced stochastic mode (assuming NITER>0), during which the estimates of thetas, omegas, and sigmas are also averaged. After an SAEM analysis, run a $EST METHOD=IMP EONLY=1 to obtain good post-analysis estimates of shrinkage, standard errors, and objective function, as described earlier.

#TERE:

This tag indicates the end of the lines describing the termination status of the analysis. Thus, a software program may transfer all lines between #TERM: and #TERE: to a summary file.

#OBJT:

Indicates that following it, on the same line, is the text describing the objective function, such as *Minimal Value Of Objective Function*.

#OBJV:

Indicates that following it, on the same line, is the objective function value. However, a more efficient way of extracting numerical results from the analysis is from the raw output file (see below).

#OBJS:

Indicates that following it, on the same line, is the objective function standard deviation (MCMC Bayesian analysis only). However, a more efficient way of extracting numerical results from the analysis is from the raw output file (see below).

#OBJN: (nm73)
Indicates that following it, on the same line, is the nonparametric objective function value.

#CPUT: (nm73)
Total cpu time in seconds. This is an accurate assessment of CPU usage of the entire problem, whether done in single or parallel mode.

Shrinkage and ETASTYPE (NM74)

Inter-subject shrinkage for each eta is calculated as:
ETASHRINKSD=100%*[1-SD(eta(j))/sqrt(omega(j,j))]

where eta(j) is the individual's empirical bayes estimate (FOCE/LAPLACE/ITS) or sampled conditional mean (IMP/SAEM) for the *j*th eta. The above definition of eta shrinkage is designated ETASHRINK in versions earlier than nm74. The variance version is also calculated as of nm74:
ETASHRINKVR=100%*[1-VAR(eta(i))/omega(i,i)]

The reason for the two types of calculations is that both types appear in the literature.

As of nm73, additional shrinkage information, called EBVSHRINK (nm73) or EBVSHRINKSD (nm74), is the ETA shrinkage based on the average empirical Bayes variance (EBV), the etc(j,j), or phc(j,j) listed in the .phi or .phm table, standard deviation version:

$$\text{EBVSHRINKSD} = 100\%\left(1 - \sqrt{1\text{-etc}_{\text{ave}}(j,j) / \text{Omega}(j,j)}\right)$$

$$\text{EBVSHRINKSD} = 100\%\left(1 - \sqrt{1\text{-phc}_{\text{ave}}(j,j) / \text{Omega}(j,j)}\right)$$

Where $\text{etc}_{\text{ave}}(j,j)$ is average etc(j,j) among included subjects, and $\text{phc}_{\text{ave}}(j,j)$ is average phc(j,j) among included subjects, for eta(j) or phi(j). Again, etc/phc is evaluated as first order approximation of the posterior variance around the mode (FOCE/ITS), second order approximation around the mode (LAPLACE), or Monte Carlo assessed posterior variance around the conditional mean (IMP/SAEM).

As of nm74, the variance version is also available, called EBVSHRINKVR:

$$\text{EBVSHRINKVR} = 100\%(\text{etc}_{\text{ave}}(j,j) / \text{Omega}(j,j))$$

$$\text{EBVSHRINKVR} = 100\%(\text{phc}_{\text{ave}}(j,j) / \text{Omega}(j,j))$$

As of nm75, the relative information is available, evaluated as

$$\text{RELATIVEINF} = 100\% / \mathit{diag}[(I - \Omega^{-1}\mathbf{V}_{\text{ave}})^{-1}]$$

where V_{ave} is the matrix containining the elements of the $\text{phc}_{\text{ave}}(i,j)$ matrix. Typically, RELATIVEINF is approximately equal to 100-EBVSHRINKVR.

As of nm73, if the eta shrinkage is less than 0, it will reported as a value of 1.0E-10. Less than 0 shrinkage can occur due to limited precision evaluations, and/or sometimes with classical NONMEM methods.

Eta shrinkage is averaged for all individuals if ETASTYPE=0. As of nm73, should you wish to correct for some individuals not contributing at all to one or more etas (this may or may not be desirable, depending on your needs), the shrinkage can be calculated by NONMEM to not include these etas by setting ETASTYPE=1 in the $EST record. This will average shrinkage information only among individuals that provided a non-zero derivative of their data likelihood with respect to that eta, and will not include subjects with a non-influential eta, that is in which the derivative of the data likelihood is zero. Furthermore, you may specify eta *i* of particular subjects to be excluded, by setting a reserved variable ETASXI(i) to 1 in $PK or $PRED, or specify eta *i* of certain subjects to be included, by setting ETASXI(i)=2 (ETASXI stands for eta shrinkage exclude/include):

```
IF(ID==3)  ETASXI(1)=1
IF(ID==23) ETASXI(3)=2
```

The results outputted in the NONMEM report file refer to average eta shrinkage. See the section I.63 $EST: Additional Output Files Produced on root.phi, for additional information one can obtain about eta shrinkage for each subject.

Residual error shrinkage standard deviation version for each residual error is evaluated for simple problems as
EPSSHRINKSD=100%*[1-SD(IWRES)]
(see [14]).

In nm73 and earlier, the above definition was called EPSSHRINK. As of nm74, the variance version is also now available:
EPSSHRINKVR=100%*[1-VAR(IWRES)]

For more complicated problems, the data and individual predicted values that contribute to assessing the shrinkage for each epsilon is not as straight-forward. For example, if EPS(1) is proportional error to PK data, and EPS(2) is proportional error to PD, and they are not connected by an off-diagonal sigma, then EPS1 shrinkage pertains to PK data residuals, and EPS2 shrinkage pertains to PD data residuals. If they are related by an off-diagonal SIGMA, then their shrinkage is related, and they will have similar or identical shrinkage values.

If two epsilons pertain to the same data, such as proportional EPS and additive EPS for PK data:

Y=F+F*EPS(1)+EPS(2)

Then the same epsilon shrinkage is associated with EPS(1) and EPS(2). However, if F=0 for some data, then such values contribute to EPS(2) shrinkage assessment, but not to EPS(1) shrinkage assessment. In such cases, shrinkage to EPS(1) and EPS(2) may differ slightly, where EPS(1) shrinkage incorporates only residuals to data with predicted values that are non-zero, and EPS(2) shrinkage incorporates residuals to all PK data.

See also *ETASAMPLES=0 (default) (nm74)*in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method for other methods of coariate model diagnostics.

I.62 $EST: Format of Raw Output File

A raw output file will be produced that provide numerical results in a columnar format. The raw output file name is by default *root*.ext, where *root* is the root name of the control stream file. Or, the name is provided by the user using a new FILE= parameter added to the $EST record. A raw output file has the following format:

A header line that begins with the word Table, such as:

```
TABLE NO.  4: MCMC Bayesian Analysis: Goal Function=AVERAGE VALUE OF LIKELIHOOD FUNCTION
```

This header line provides the analysis text (same as given on the #METH: line in the main report file), followed by the goal function text (same as given on the #OBJT: line in the report file).

The next line contains the column headers to the table, such as (this is actually all on one line in the file):

```
ITERATION     THETA1        THETA2        THETA3        THETA4        SIGMA(1,1)    OMEGA(1,1)
OMEGA(2,1)    OMEGA(2,2)    OMEGA(3,1)    OMEGA(3,2)    OMEGA(3,3)    OMEGA(4,1)    OMEGA(4,2)
OMEGA(4,3)    OMEGA(4,4)    OBJ
```

This is followed by a series of lines containing the intermediate results from each printed iteration (six significant digits), based on the PRINT= option setting:

```
           10  1.73786E+00  1.57046E+00  7.02200E-01  2.35533E+00  6.18150E-02  1.82955E-01
 -3.18352E-03  1.46727E-01 -4.38860E-02  2.58155E-02  1.45753E-01 -4.58791E-02  6.28773E-03
  5.06262E-02  1.50017E-01 -2301.19773603667
```

For the above example, each of the values, up to the next to last one, occupies 13 characters, including the delimiter (in this example the delimiter is a space). The last value is the objective function, which occupies 30 characters, to allow for the largest range of objective function values, and the greatest expression of precision.

The iteration number, which is the first value in every line, is typically positive, but also may be negative under the following conditions:

1) The burn-in iterations of the MCMC Bayesian analysis are given negative values, starting at –NBURN, the number of burn-in iterations requested by the user. These are followed by positive iterations of the stationary phase.
2) The stochastic iterations of the SAEM analysis are given negative values. These are followed by positive iterations of the accumulation phase.
3) Iteration -1000000000 (negative one billion) indicates that this line contains the final result (thetas, omegas, and sigmas, and objective function) of the particular analysis. For BAYES analysis, this is the mean of the non-negative iterations (stationary samples) listed before it.

4) Iteration -1000000001 indicates that this line contains the standard errors of the final population parameters. For BAYES, it is the sample standard deviation of the stationary samples.
5) Iteration -1000000002 indicates that this line contains the eigenvalues of the correlation matrix of the variances of the final parameters.
6) Iteration -1000000003 indicates that this line contains the condition number , lowest, highest, Eigen values of the correlation matrix of the variances of the final parameters.
7) Iteration -1000000004 indicates this line contains the OMEGA and SIGMA elements in standard deviation/correlation format
8) Iteration -1000000005 indicates this line contains the standard errors to the OMEGA and SIGMA elements in standard deviation/correlation format
9) Iteration -1000000006 indicates 1 if parameter was fixed in estimation, 0 otherwise.
10) Iteration -1000000007 lists termination status (first item) followed by termination codes. See I.63 $EST: Additional Output Files Produced under *root.xml (NM72)* for interpreting the codes.
11) Iteration -1000000008 lists the partial derivative of the log likelihood (-1/2 OFV) with respect to each estimated parameter. This may be useful for using tests like the Lagrange multiplier test.
12) Additional special iteration number lines may be added in future versions of NONMEM.

The raw output file is provided automatically, independent of the formatted files that may be requested by the user using the $TABLE command.

For the output files generated during the $EST step, the following parameters may be specified:

FILE=my_example.ext

Parameters/objective function printed to this raw output file every PRINT iterations. Default is *control*.ext, where *control* is name of control stream file.

DELIM=s or FORMAT=t or FORMAT=,

Delimiter to be used in raw output file FILE. The character defines the delimiter, which may be s for space, t for tab, comma for a comma delimited file with aligned fields (so, padded with spaces), c for comma delimited file with no spaces, or q for comma delimited file with no spaces and double quotes around column names that have commas (such as “OMEGA(2,1)”).

DELIM=s1PE15.8 or FORMAT=s1PG15.8 or FORMAT=tF8.3

In addition to the delimiter, a format (FORTRAN style) may be defined for the presentation of numbers in the raw OUTPUT file.

Default format for additional output files produced by the $EST step is s1PE12.5

The variables DELIM and FORMAT are interchangeable.

If you specify FORMAT=QCSV then this is equivalent to FORMAT=q1PG23.16

If you specify FORMAT=CSV then this is equivalent to FORMAT=c1PG23.16

The lines produced in the ext file may be very long. You may optionally provide a line length, followed by a continuation marker to be tagged at the end of each line (e), and/or a continuation marker to be tagged at the beginning of the continuing line.

FORMAT=s1PE15.8:160&

will print lines of at most 160 characters, followed by a & for each line that needs to be continued (if using an ampersand, and it is at the end of the line in the control stream file, place a ; after it so it is not interpreted as a continuation indicator by the NMTRAN control stream file reader).

FORMAT=s1PE15.8:160&c

Will print lines of at most 160 characters, with & tagged at the end of the line to be continued, and a c at the beginning of the continued line.

FORMAT=s1PE15.8:160sc

Will print lines of at most 160 characters, with no character at the end of each line to be continued, and a c at the beginning of the continued line. S represents "space", and a space may not serve as a continuation marker because of its ambiguity, so it serves here as a place holder in the FORMAT definition. These line continuation formats are ignored in $TABLE records, but are used in the $EST record for all additional file formats, and can are used in $EST CHAIN=METHOD and $CHAIN records.

NOTITLE=[0,1]

If NOTITLE=1 (default=0), then the Table header line will not be written to the raw output file specified by FILE=.

NOLABEL=[0,1]

If NOLABEL=1 (default=0), then the column label line will not be written to the raw output file specified by FILE=.

ORDER (NM72)

The order in which the thetas, omegas, and sigmas are listed in the output file is by default as follows: Thetas (T), SIGMAS(S), OMEGAS(O). The SIGMA and OMEGA matrices are listed in lower triangular order, row-wise:

1
2 3
4 5 6
7 8 9 10

You may change the order in which these are displayed, by specifying the ORDER option. The THETAS are referenced with a T, SIGMAS with S, OMEGAS with O, lower triangular with L, upper triangular with U. The first three letters given in the ORDER option refer to which parameters are listed in order (T, S, O), and the fourth letter is U or L to indicate matrix element order for sigmas and omegas. Thus,

ORDER=TSOL

Is the default ordering. This is different from the ordering that is given in the report file for displaying the variance matrix, which is TOSU. In TOSU ordering, Thetas are listed first in the raw output file, followed by omegas, followed by sigmas, and the omegas and sigma elements are listed in row-wise upper-triangular order (or column-wise, lower triangular order):

```
1  2  3  4
   5  6  7
      8  9
        10
```

I.63 $EST: Additional Output Files Produced

In addition to the raw output file described in the previous section, the following files are created automatically, with root name based on the root name of the control stream file. The delimiter and numerical format is whatever is specified with $EST … FORMAT=, for the raw output file.

root.cov

Full variance-covariance error matrix to thetas, sigmas, and omegas

root.clt (NM74)

Lower triangular form of Variance-covariance error matrix to thetas, sigmas, and omegas, in TOSL order, for easy cut-and-paste transfer to place into control stream files as prior information.

root.cor

Full correlation matrix to thetas, sigmas, and omegas

root.coi

Full inverse covariance matrix (Fischer information matrix) to thetas, sigmas, and omegas

root.phi

Individual phi parameters (phi(i)=mu(i)+eta(i), for ith parameter), and their variances phc(,). For parameters not MU referenced phi(i)=eta(i). When a classical method is performed (FOCE, Laplace), then mode of posterior eta(i) are printed out, along with their Fisher information (first order expected value for FOCE, second order for Laplace) assessed variances etc(,).
For ITS, these parameters are the modes of the posterior density, with first-order approximated expected variances (or second order variances if $EST METHOD=ITS LAPLCE is used).

For IMP, IMPMAP, SAEM methods, they are the Monte Carlo evaluated mean parameters and variances of the parameters under the posterior density.
For MCMC Bayesian, before nm75, they are random single samples of phi(), as of the last position. Their variances are zero. As of nm75, the .phi table contains the average of phi() values collected throughout the stationary distribution (positive iterations) phase, so they also contain variances.

Individual objective function values (obji) are also produced. In the case of MCMC, before nm75, the individual likelihood of the last sample is reported. As of nm75, average the MCMC individual likelihood among during the stationary phase is reported. Note that the variances of the phis (phc(,)) include variability contributed by thetas for MCMC analysis, unlike the other methods where the variances of the phis are evaluated with thetas fixed.

As of nm75, variances phc/etc for phis/etas of levels above the subject are now available in the phi table.

As of nm74, if $EST PHITYPE=1 is specified, then conditional mean etas are reported in the phi table. regardless of the analysis method. See *PHITYPE=0 (NM74)*in section I.38 Stochastic Approximation Expectation Maximization (SAEM) Method.

root.phm (NM72)

These vaues have similar meaning as the results in root.phi, but individual phi/eta/obji parameters per sub-population are recorded. This file is only produced in $MIXTURE problems.

The PHITYPE option also acts on this file.

As of nm75, for MCMC Baysian analysis, the items listed in this table consist of the average values among all the iterations during the stationary distribution phase. Before nm75, the values consisted of values from the last MCMC iteration.

The conditional variances in the root.phi and root.phm files can represent the information content provided by a subject for a given eta or phi. For example, if data supplied by the subject is rich, then the variance tends to be smaller. If little data is supplied by the subject for that eta, then the conditional variance will approach its omega. In fact, a subject's shrinkage can be evaluated as follows:

In accordance with the SD formula:

$$\mathrm{ETAshrinkage}_i\% = 100\%(1-\sqrt{1-\mathrm{phc}_i(j,j)/\mathrm{Omega}(j,j)})$$

Or by the variance formula:

$$\mathrm{ETAshrinkage}_i\% = 100\%\,\mathrm{phc}_i(j,j)/\mathrm{Omega}(j,j)$$

For subject *i*, eta or phi *j*.

root.iph (NM75)

As of nm75, samples of phi/eta are collected at each BAYES iteration, and summarized to provide mean phi and phc() in the root.phi table, as described above. By default, the individual phi values from each iteration are not stored. However, if you set

```
$EST … BAYES_PHI_STORE=1
```

then phi and eta values from each BAYES iteration will be stored in root.iph. For non-mixture problems, only records of SUBP=0 are recorded, as there are no sub-population divisions. For mixture problems, the SUBP=0 records contain the composite phis and etas (the average of these across all non-negative iterations are in the root.phi table), and the SUBP>0 records contain the phis and etas appropriate to each sub-population SUBP (the average of these across all non-negative iterations are in the root.phm table). The root.iph file can become quite large, so it should be used only on the final analysis.

root.shk (NM72)

This file presents composite eta shrinkage and epsilon shrinkage information, the same as given in the report file between the #TERM: and #TERE: tags, but in rows/column format, and with adjustable formatting.
Type 1=etabar
Type 2=Etabar SE
Type 3=P val
Type 4=%Eta shrinkage SD version
Type 5=%EPS shrinkage SD version
Type 6=%Eta shrinkage based on empirical Bayes Variance (SD version)
Type 7=number of subjects used.
Type 8=%Eta shrinkage variance version
Type 9=%Eta shrinkage based on empirical Bayes Variance (variance version)
Type 10=%EPS shrinkage variance version
Type 11=%Relative information

root.shm (NM73)

As of NM73, the .shm table (which stands for shrinkage map) will contain information which etas were excluded in the eta shrinkage assessment. The syntax is as follows:

For each subject, sub-population, the value listed in column ets(j) contains the information about whether and how that eta was included in the etabar/shrinkage calculations. It is a binary value of the format x.abcdef, where each of the letters may be 0 or 1. If the eta is excluded from the etabar/eta shrinkage summary that is recorded in the main NONMEM report file or the .shk file, then x=1, otherwise it is 0. The remaining binary digits after the decimal point describes conditions about this eta that were involved in deciding whether to exclude this eta:
a: set to 1 if NONMEM assessed this eta as non-influential (the derivative of the data likelihood with respect to that eta is 0). This exclusion criterion is only acted on (that is, actually excludes this eta, indicated by x=1), if etastype=1.

b: set to 1 if NONMEM excluded this eta for this sub-model (sub-population), for this subject, because this was not the best fitting sub-model for this subject. Thus all etas of that subject for all sub-models that are not the optimally fitting will have this bit set, and only the optimal sub-model will have B cleared (0) for all its etas.
c: set to 1 if NONMEM determined that this eta had no influence for this sub-model. This bit is not set to 1 if bit B is 1. This bit is not set to 1 for non-population-mixture models. Also, this exclusion criterion is set and acted upon when FOCE/Laplace are used, but is not set or acted on for the Em methods. IF NONINFETA is set to 1, then FOCE/Laplace behave similarly to EM methods, and will not set this bit even if the eta has no influence.
d: set if the eta is excluded based on selecting the hybrid option in $EST.
e: Set if the user requested an exclusion based on ETASXI(i)=1 setting in $PK or $PRED for eta i.
f: Set if the user requested an inclusion based on ETASXI(i)=2 setting in $PK or $PRED for eta i. Be careful about using this, as it overrides all other exclusion criteria except bit B. The F bit is the only one that indicates inclusion when set, rather than exclusion.

root.grd (NM72)

This file contains gradient values for classical NONMEM methods.

The format of these files are subject to FORMAT, ORDER, NOLABEL, and NOTITLE options in the $EST command, the same as for the raw output file.

root.xml (NM72)

An XML markup version of the contents of the NONMEM report file is produced automatically. The rules (schema, document type definition) by which it is constructed are given in output.xsd and output.dtd, in the NONMEM ..\util or ..\run directory.

In NM73, termination_textmsgs catalogs termination text messages by number, which can be mapped to ..\source\textmsgs.f90.

In nm73, termination status catalogs the error status:
For traditional analyses, an error number is listed. If negative, the analysis was user-interrupted

For EM/Bayes analysis, error numbers map as follows (changed for nm75):
0,8: optimization was completed
1,9: optimization not completed (ran out of iterations)
2,10: optimization was not tested for convergence
3,11: optimization was not tested for convergence and was user interrupted
4,12: optimization was not completed and was user interrupted
16,24: objective function is infinite or all individual objective fuctions are zero. problem ended
32,40: All individual objective fuctions are zero. problem ended
8,9,10,11,12,24,40: reduced stochastic/sationary portion was not completed prior to user interrupt

The expectation only step may also be considered an “optimization” in the context of these codes.

These codes are also listed on line -1000000007 in the .ext file (NM74).

root.cnv (NM72)

This file contains convergence information for the Monte Carlo/EM methods, if CTYPE>0:
-2000000000=mean of last CITER values.
-2000000001=standard deviation of last CITER values (for objective function, STD of second to last CITER values)
-2000000002=linear regression p-value of last CITER values against iteration number.
-2000000003=Alpha used to assess statistical significance (p-value<alpha)

Please note the following:
The Sigma values are in their Cholesky format, as this is the form in which convergence of these values are tested.
The Alpha are those based on ones actually used for convergence test of that parameter, or which would have been used on that parameter if CTYPE were of proper type. The alpha may be bonferoni corrected because of multiple comparisons, depending on number of parameters that were tested or would have been tested. Objective function alphas are not bonferoni corrected.

For importance sampling and iterative two stage, the average objective function listed in root.cnv could be used as an alternative to the final objective function for likelihood ratio tests.

root.smt (NM72)

S matrix, if $COV step failed.

root.rmt (NM72)

R matrix, if $COV step failed.

root.imp (NM73)

The root.imp file is produced if the user selects importance sampling with option IACCEPT=0.0. In such cases, this file lists the final IACCEPT and DF values that NONMEM selected for each subject.

Three files are produced providing nonparametric information:

root.npd (NM73)

Each row contains information about a support point: The support point number, the ID from which the support point was obtained as an EBE of that subject (ID is -1 if this support point was randomly generated because NSUPP/NSUPPE was greater than number of subjects). The eta values of the support point are listed, followed by the cumulative probability (CUM) associated with each eta, followed by the joint density probability of that support point, if default or

MARGINALS was selected. If ETAS was selected, then instead of cumulative probabilities, the support point eta vector that best fits that subject (ETM) is listed.

root.npe (NM73)

The expected value etas and expected value eta covariances (ETC) are listed for each problem or sub-problem. Because only one line is written per problem or sub-problem, the column header is displayed (unless NOLABEL=1) only once for the entire NONMEM run. However, each line contains information of table number, problem number, sub-problem number, super problem and iteration number.

root.npi (NM73)

The individual probabilities are listed in this file. The header line (unless NOLABEL=1) is written only once, at the beginning of the file, per NONMEM run. Each line contains information of table number, problem number, sub-problem number, super problem, iteration number, subject number, and ID. This is followed by the individual probabilities at each support point (of which there are NSUPP/NSUPPE or NIND of them, whichever is greater). The line with Subject number=0 contains the sum of the probabilities of all the subjects, and is similar or exactly equatl to the joint probability of each support point listed in root.npd under the column PROBABILITY. That they are not equal is due to the convergence limit of the non-parametric analysis. Row of subject number I, column of support K, contains the individual probability IPROB(I,K). The sum of the individual probabilities over all support points for any given line (subject), is equal to 1/NIND. Thus, the sum of all items across rows and columns (not including subject 0) sums to 1. The format of the file is fixed at (,1PE22.15), and cannot be changed. It is intended for use in further analysis by analytical software, and is designed to report the full double-precision information of each probability.

root.npl (NM74)

The individual data likelihoods (not including the parameter density) are listed in this file. The header line (unless NOLABEL=1) is written only once, at the beginning of the file, per NONMEM run. Each line contains information of table number, problem number, sub-problem number, super problem, iteration number, subject number, and ID. This is followed by the individual likelihoods at each support point (of which there are NSUPP/NSUPPE or NIND of them, whichever is greater). Unlike the .npi file, there is no line with Subject number=0. The row of subject number I, column of support K, contains the individual likelihood LIK(I,K). The format of the file is fixed at (,1PE22.15), and cannot be changed. It is intended for use in further analysis by analytical software, and is designed to report the full double-precision information of each probability.

The LIK(I,K) of the .npl file and the IPROB(I,K) are related to each other as follows:

IPROB(I,K)=PI(K)*LIK(I,K)/exp(-1/2*OBJ(I))/NIND

where NIND is number of subjects, PI(K) is the probability of a support point (found as item PROBABILITY in the .npd file), and OBJ((I) is the objective function contribution of subject I (found under OBJ column of the .npl or .npi file).

root.fgh (NM73)

This file is produced if the user selects $EST NUMDER=1 or 3. The file lists the numerically evaluated derivatives of Y with respect to eta, where
G(I,1)=partial Y with respect to eta(i))
G(I,J+1)=Second derivatives of Y with respect to eta(i),eta(j)
H(I,1)=partial Y with respect to eps(i)
H(i,j+1)=partial Y with respect to eps(i),eta(j))

root.agh (NM73)

This file is produced if the user selects $EST NUMDER=2 or 3. The file lists the analytically evaluated derivatives of Y with respect to eta, from the PK(), ERROR(), and/or PRED() routines in FSUBS, where
G(I,1)=partial Y with respect to eta(i))
G(I,J+1)=Second derivatives of Y with respect to eta(i),eta(j) (not always evaluated by FSUBS)
H(I,1)=partial Y with respect to eps(i)
H(i,j+1)=partial Y with respect to eps(i),eta(j))

root.cpu (NM73)

The cpu time in seconds is reported in this file. It is an accurate representation of the computer usage, whether single or parallel process. The same problem when run singly or in parallel will report a similar cpu time. This is in contrast with elapsed time, which is improved with parallelization.

root.vpd (NM74)

Variance-covariance among user-defined parameters and PREDPP parameters are placed in this file, associated with conditional individual variances of the .phi file. The COMACT column value is usually 2, or 3 if $NONPARAMETRIC ETAS was implemented. The subsequent columns contain the values of user-defined table items, followed by variance-covariance of these items (labled VAR1_VAR2, wher VAR1 and VAR2 are the particular table items).

root.vpt (NM75)

Variance-covariance among user-defined parameters and PREDPP parameters are placed in this file, associated with conditional individual variances of the .phi file, as well as the variance-covariance of the fixed effects parameters (THETAS, SIGMAS, OMEGAS) of the .cov file. The COMACT column value is 1 to indicate the record contains the values during a COMACT=1 pass, that is with ETAs set to 0. Thus, an individual predicted value (IPRED) on this record would in fact by the population predicted value (PRED). The subsequent columns

contain the values of user-defined table items, followed by variance-covariance of these items (labled VAR1_VAR2, wher VAR1 and VAR2 are the particular table items). The variance IPRED_IPRED for a record with COMACT=1 would then have essentially the statistical error associated with the variance-covariance of the appropriate thetas. When COMACT=2, table variables are evaluated having the etas set to the empirical bayes etimates (depending on the last estimation method used, and what the FNLETA value is set to), variance-covariance of these itmes are evaluated based on conditional individual variances of the .phi table, and variance-covariance of the appropriate thetas, sigmas, and omegas, as applicable. If COMACT=3, then non-parametric estimated etas are used.

root.ets (NM74)

Random samples of individual etas when $EST METHOD=SAEM ETASAMPLES=1 or $EST METHOD=BAYES ETASAMPLE=1. ETASAMPLES=1 causes individual ISAMPLE random eta samples per subject, to be written to *root*.ets, where *root* is the root name of the control stream file.

root.bfm (NM75)

When OFVTYPE=8 during optimal design ($DESIGN OFVTYPE=8) the progress of average individual conditional variances are shown in the root.bfm file (where root is name of control stream file), and the final one on the -1000000000 line, during an optimization. The RAW file (by default named root.ext) which shows only the starting, and final values of the standard errors of population parameters, has extra information.

msfroot_ETAS

When MSF or MSFO option is used to specify an MSFO file in the $EST record:
$EST … MSFO=*msfroot.msf*

then in addition to the main MSF file msfroot.msf, file msfroot_ETAS.msf containing individual etas and phis generated during estimation (except FO, which does not generate non-zero etas during estimation), will also be produced, and provide additional information when a $MSFI record is used in a subsequent problem or control stream. In addition, this files stores information useful for resuming an interrupted EM/BAYES estmation. The _ETAS file does not contain POSTHOC evaluated etas, only estimated (during $EST) evaluated etas. Also, ensure that the ordering of subjects in the present problem matches that of the problem when the msf file was generated (for example, do not change the data file structure or IGNORE/ACCEPT filtering between the two problems).

msfroot_RMAT, msfroot_SMAT (nm73)

When MSF or MSFO option is used to specify an MSFO file in the $EST record:
$EST … MSFO=*msfroot.msf*

then in addition to the main MSF file msfroot.msf and msfroot_ETAS.msf, files msfroot_RMAT.msf and msfroot_SMAT.msf containing intermediate information on the R matrix and S matrix will also be produced if a $COV record was implemented. These files provide information when a $MSFI record along with a $COV … RESUME record is used in a subsequent problem or control stream. See *RESUME (NM73)* in section I.56 $COV: Additional Options and Behavior.

I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records

The METHOD=CHAIN option of the $EST command allows the user to create a series of random initial values of THETAS and OMEGAS, or for reading in initial population parameters from a file of rectangular (rows/column) format.

Consider the following example.

```
$EST METHOD=CHAIN FILE=example1.chn DELIM=,
     NSAMPLE=5 CTYPE=0 ISAMPLE=3 DF=100
     SEED=122234 RANMETHOD=2 IACCEPT=0.5
```

In this example, NSAMPLE random samples of THETAS and OMEGAS will be generated and written to a file specified by FILE, using “comma” as a delimiter. SEED sets the starting seed for the random number sequence.

As of nm75, the actual starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if CLOCKSEED=1. This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication.

By default (CTYPE=0), random values of theta are generated from a uniform distribution spanning from lower bound theta to upper bound theta specified in the $THETA statement. If a boundary for a theta is not specified, then (1-IACCEPT)*THETA is used for a lower bound, and (1+IACCEPT)*THETA is used for an upper bound. For the SIGMA values their Cholesky-decomposed values are uniformly varied between (1-IACCEPT)*SIGMA and (1+IACCEPT)*SIGMA (but see below for the option DFS as of NM73). If CTYPE=1, then regardless of lower and upper bound designations on the $THETA statements, all thetas are uniformly varied using the IACCEPT factor. If CTYPE=2, then, the random values of theta are created based on a normal distribution, with the initial $THETA in the control stream file as the mean, and the second set of $OMEGAs as the variance, if there is a $PRIOR command with NTHP non-zero. This is the best way and most complete way to define the sampling density for the THETAs. Otherwise, if NTHP=0, the variance for THETA is obtained from the first set of $OMEGA, and requires that the THETA’s be MU modeled, and those THETAs not MU modeled will be varied by the uniform distribution method as described for CTYPE=0.

The omega values are sampled using a Wishart density of variance listed in the $OMEGA command, and DF is the degrees of freedom for randomly creating the OMEGAS. If DF=0, then

the dimensionality of the entire OMEGA matrix is used as the degrees of freedom. As of NM73, if DF>one million, then OMEGA elements are fixed at their initial values.

The format of the chain file that is created is exactly the same as the raw output files, including iteration numbers. In the above example, after the 5 random samples are made, ISAMPLE=3 (the third randomly created sample) is selected, and brought in as the initial values. If ISAMPLE=0, then the initial values are not set to any of the randomly generated samples, but will just be what was listed in $THETA and $OMEGA of the control stream file.

If NSAMPLE=0, but ISAMPLE=some number, then it is expected that FILE already exists, and its iteration number specified by ISAMPLE is to be read in for setting initial values:

```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=3
```

One could create a control stream file that first creates a random set of population parameters, and then sequentially uses them as initial values for several trial estimation steps:

```
$PROBLEM #1
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA     wexample11.csv IGNORE=@
$SUBROUTINES ADVAN3 TRANS4
$PK
...
$ERROR
...
$THETA 2.0 2.0 4.0 4.0 ; Initial Thetas
$OMEGA BLOCK(4) ; Initial Parameters for OMEGA
2
0.01 2
0.01 0.01 2
0.01 0.01 0.01 2
$SIGMA 0.5

; First problem, creates NSAMPLE=5 random sets of initial parameters, stores
; them in example11.chn.  Then, selects the first sample ISAMPLE=1
; for estimation
$EST METHOD=CHAIN FILE=wexample11.chn NSAMPLE=5 CTYPE=2 ISAMPLE=1 DF=4
     SEED=122234 IACCEPT=0.8
$EST  METHOD=COND INTERACTION MAXEVAL=9999 NSIG=2 SIGL=10 PRINT=5 NOABORT
      FILE=wexample11_1.ext

$PROBLEM #2
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA     wexample11.csv IGNORE=@ REWIND

$THETA 2.0 2.0 4.0 4.0 ; Initial Thetas
$OMEGA BLOCK(4) ; Inital Parameters for OMEGA
0.4
0.01 0.4
0.01 0.01 0.4
0.01 0.01 0.01 0.4
$SIGMA 0.1
```

```
; Second problem, selects sample ISAMPLE=2 for initial settings, from file
wexample11.chn. Won't recreate the file, as NSAMPLE=0
$EST METHOD=CHAIN FILE=wexample11.chn NSAMPLE=0 ISAMPLE=2
$EST  METHOD=COND INTERACTION MAXEVAL=9999 NSIG=2 SIGL=10 PRINT=5 NOABORT

; etcetera, for samples 3, 4, and 5, executed as problems 3, 4, and 5.
```

In the above example, the five estimations are performed in sequence. To perform these in parallel in a multi-processor or multi-computer environment, a pre-processing program could set up and execute a control stream file which would have as one of the commands

```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=5 ISAMPLE=0 DF=20
```

A copy of this control-stream file could be made, and the pre-processing program could make five new "child" control stream files, with the NSAMPLE this time set to 0 (so that it does not create a new chain file, but uses the already existing one), and ISAMPLE= entries modified in the following five ways, each differing by only the ISAMPLE number:

First control stream file:
```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=1 DF=20
```
second control stream file:
```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=2 DF=20
```
third control stream file:
```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=3 DF=20
```
fourth control stream file:
```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=4 DF=20
```
fifth control stream file:
```
$EST METHOD=CHAIN FILE=example1.chn NSAMPLE=0 ISAMPLE=5 DF=20
```

Each control stream file points to a different ISAMPLE position in the .chn file, so each would use these as the respective initial positions. Each of these "child" control stream files could be loaded on to a job queue, as separate processes. If the user is running a multi-core computer, this would be quite straight forward.

An existing chain file could actually be a raw output file from a previous analysis, with a list of iterations. In the following example:

```
$EST METHOD=CHAIN FILE=example1_previous.txt NSAMPLE=0
     ISAMPLE=-1000000000 TBLN=4
```

could pick up the final result of the previous analysis, since ISAMPLE points to the iteration number, and -1000000000 is the iteration number for the final estimate. As of nm75, you may also select the table number TBLN, as raw output files can contain multiples tables of results, with similar sample numbers. If TBLN is not specified, the first matching iteration (isample) number is selected among all the tables in the file. Thus, the CHAIN method in this usage is really just an input command to bring in values from a raw output-type file format. Of course, users may have the chain file created by any program, not just NONMEM, so long as it has the raw output file format, with delimiter specified by DELIM/FORMAT (which is space by default).

(NM73) If the option ISAMPEND is set to a value greater than ISAMPLE, then NONMEM will evaluate the objective function (using FOCEI method) for each sample between numbers ISAMPLE and ISAMPEND in the file, and then select the one with the smallest objective function. For example,

```
$EST METHOD=CHAIN FILE=random.txt NSAMPLE=20 ISAMPLE=1 ISAMPEND=20
```

randomly creates 20 sets of initial parameters, and selects the one with the lowest objective function.

If METHOD=CHAIN is used, it must be the first $EST command in the particular $PROB. Furthermore, because the settings it uses for FILE, NSAMPLE, ISAMPLE, IACCEPT, CTYPE, and DF are functionally different from the way the other $EST methods use them, these settings from METHOD=CHAIN are not passed on to the next $EST command, which must be an estimation method. However, other parameters such as DELIM, FORMAT, SEED, AND RANMETHOD will be passed on as default delimiter/format to the next $EST command. However, the RANMETHOD does not propagate to the $CHAIN record.

DFS=-1 (default)(NM73)

As of NM73, the SIGMA matrix may be randomly created with an inverse Wishart distribution centered about the initial SIGMA values, with degrees of freedom DFS for dispersion. If DFS=-1 which is the default, then the method of earlier versions of NONMEM will be used, with the cholesky elements uniformly varied over the interval (1-iaccept)*initial value and (1+iaccept)*initial value. If DFS>one million, then SIGMA is fixed at the initial values. If DFS=0, then the dimensionality of the entire SIGMA matrix is used as degrees of freedom.

$CHAIN Record

Any initial settings of THETA, OMEGA, and SIGMA that are read in by $EST METHOD=CHAIN are applied only for the estimation step. The $SIML command will not be affected, and will still use the initial settings given in $THETA, $OMEGA, and $SIGMA statements, or from an $MSFI file. To introduce initial THETAs omegas and sigmas that will cover the entire scope of a given problem, use the $CHAIN record:

```
$CHAIN FILE=example1_previous.txt NSAMPLE=0
     ISAMPLE=-1000000000
```

The following options are available for $CHAIN, and have the same actions as for $EST METHOD=CHAIN: FILE, NSAMPLE, ISAMPLE, SEED, CLOCKSEED, RANMETHOD, FORMAT, ORDER, CTYPE, DF, DFS, IACCEPT, NOLABEL, NOTITLE. Setting SEED or RANMETHOD in a $CHAIN record does not propagate to $EST METHOD=CHAIN or any other $EST record.

ISAMPEND (NM73) has a different action with $CHAIN than with $EST METHOD=CHAIN.

If the option ISAMPEND is set to a value greater than ISAMPLE, then NONMEM uniformly randomly selects one of these samples between ISAMPLE and ISAMPEND. This is particularly useful in combination with the SIML record:

```
$CHAIN FILE=test2.chn ISAMPLE=3 ISAMPEND=10 NSAMPLE=10 SEED=6234
$SIML (112345) (334567 NORMAL) SUBP=4
$EST METHOD=IMP INTERACTION NITER=40 PRINT=1 NOABORT SIGL=4
CTYPE=3 CITER=10
```

In the above example, for the first subproblem, a file called test2.chn is created and stores NSAMPLE (10) randomly created sets of thetas, omegas, and sigmas, numbered 1 to NSAMPLE. Then, a sample of parameters is selected from this file uniformly randomly between ISAMPLE (3) and ISAMPEND (10), and these parameters are used to create a data set for the first sub-problem, and an estimation is performed. For the second sub-problem, a new file of parameters does not need to be created, but another sample is selected randomly uniformly between samples 3 and 10, from which a new data set is created and estimation analysis performed.

The parameter file may already exist, perhaps as a raw output file from a previous MCMC Bayesian analysis, and it is desired to randomly selected sets of parameters:

```
$CHAIN FILE=example1.chn ISAMPLE=0 ISAMPEND=10000 NSAMPLE=0 SEED=6234
$SIML (112345) (334567 NORMAL) SUBP=100
```

In the above example, NSAMPLE=0, so this means the file example1.chn already exists, which is in fact the raw output file example1.txt from the MCMC Bayesian analysis of example1. Samples from 0 to 10000 (the stationary distribution range) are selected randomly. Even though samples in physically close proximity in the file may have some correlation, selecting randomly among the entire set assures de-correlation, while assuring the samples taken represent the empirical distribution of uncertainty of the parameters. In general sampling is performed between the larger of ISAMPLE and the lowest iteration (sample) number of a raw output file, and the smaller of ISAMPEND and the largest iteration number in the file. So, it is safe to make ISAMPEND=1000000 for example, to cover most Bayesian sample set sizes. If ISAMPEND is specified in the $CHAIN record, then $SIML's TRUE=PRIOR will be ignored.

SELECT=0 (default)(NM73)

When SELECT=0, and ISAMPEND>=ISAMPLE, then the default action for selecting between ISAMPLE and ISAMPEND is taken, which for $EST METHOD=CHAIN is to find the one giving the best OBJ at the initial values, and for $CHAIN is to randomly select a sample, with replacement, as described above. Alternative actions may be obtained, which apply to both record types:

SELECT=1, the sample is selected sequentially from ISAMPLE to ISAMPEND with each new use of $CHAIN/$SIML with multiple sub-problems for the given problem, and with each new $EST METHOD=CHAIN with multiple sub-problems and across problems. When ISAMPEND is reached, the sample selection begins at ISAMPLE again.

SELECT=2, uniform random selection of sample, without replacement. Should the sample selection become exhausted, which would occur if CHAIN or $CHAIN records are utilized for more than ISAMPEND-ISAMPLE+1 times, subsequent sample selection then occurs with replacement.

SELECT=3, uniform random selection of sample, with replacement (this is equivalent to SELECT=0 for $CHAIN).

I.65 $ETAS and $PHIS Record For Inputting Specific Eta or Phi values (NM73)

Sometimes it is desired to bring in specific eta or phi values and using them as initial values, just as is done for thetas using the $THETA record. The simplest syntax is to enter a single set of etas:

$ETAS 0.4 3.0 3.0 5.0

from the control stream file. All of the subjects in the data set will be given these set of initial values of etas. Alternatively, enter them as phi values, convenient for EM methods:
$PHIS 0.4 3.0 3.0 5.0

The eta values will then be evaluated as eta(i)=phi(i)-mu(i) for each eta, where mu(i)=mu_i is evaluated according to their definitions in the $PK section.

Alternatively, enter initial etas and/or phis for an entire set of subjects from a .phi or .phm (in the case of mixture problems) of a previous analysis:

$ETAS FILE=myprevious.phi FORMAT=s1pE15.8 TBLN=3

Where FORMAT should at least have the delimiter appropriate to read the file, and TBLN is the table number in the file. If TBLN is not specified, then the first set of etas/phis are brought in. In matching the etas/phis to the data set given in $DATA of the control stream file, the attempt will be to match ID numbers rather than subject numbers, if an ID column in the file exists, which it will, if you are using a .phi or .phm file generated from a previous nonmem analysis. The phc/etc variances will also be brought in. Make sure that subject numbers and/or ID numbers match up uniquely and unambiguously to the subjects of the data set given in $DATA, so there is no possibility of confusion (which is not an issue if the .phi file produced by the control stream used the same data structure as the present control stream).

The etas inputted by $ETAS/$PHIS can be used in several ways. In BAYES, SAEM, and IMP MAPITER=0 they are used as the starting etas (in the first iteration). In MAP estimation matters, such as METHOD=1, or ITS, or IMP MAPITER>0, or IMPMAP, and if MCETA>0, then these etas are one of the initial eta vector positions tested (during the first iteration), and the one giving the lowest OBJ is then selected. In cases where FNLETA=2, the estimation step is skipped, and etas inputted from $ETAS are passed directly to the Final processing steps. That is, these etas are treated as if they were the final result of an estimation. The final processing steps

use routines such as FNLETA, FNLMOD, PRRES, NP4F, that contribute to generating $TABLE, $SCATTER outputs, including the various WRES diagnostics, where applicable. When METHOD=0, these initial etas are not used, as this method does not require initial etas.

One purpose to bringing initial eta/phi and etc/phc values is you can readily resume an analysis, if an MSF file was not set up in the previous analysis (the MSF file system is still the most complete information transfer for resuming an analysis):

```
$PROB RUN# example3 (from ad1tr1m2s)
$INPUT C SET ID JID TIME CONC=DV DOSE=AMT RATE EVID MDV CMT VC1 K101 VC2 K102 SIGZ PROB
$DATA example3.csv IGNORE=C

$SUBROUTINES ADVAN1 TRANS1

$MIX
P(1)=THETA(5)
P(2)=1.0-THETA(5)
NSPOP=2

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
VCM=DEXP(MU_1+ETA(1))
K10M=DEXP(MU_2+ETA(2))
VCF=DEXP(MU_3+ETA(3))
K10F=DEXP(MU_4+ETA(4))
Q=1
IF(MIXNUM.EQ.2) Q=0
V=Q*VCM+(1.0-Q)*VCF
K=Q*K10M+(1.0-Q)*K10F
S1=V

$ERROR
Y = F + F*EPS(1)

$THETA 4.3 -2.9 4.3 -0.67 0.7
$OMEGA BLOCK(2)
 .04
 .01 .027

$OMEGA BLOCK(2)
 .05
 .01 .06

$SIGMA
0.01

$PHIS FILE=etafile3_phi.phm FORMAT=S1PE15.7  TBLN=3
$EST METHOD=CHAIN FILE=etafile3.chn ISAMPLE=5 NSAMPLE=0
$EST METHOD=IMP MAPITER=0 CTYPE=3 INTERACTION NSIG=3 PRINT=1 NITER=3
```

Or, use FNLETA=2 to use the etas that were brought in to evaluate predicted values, without performing a new population estimation:

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X SDIX SDSX
$DATA etafile.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$PK
MU_1=THETA(1)
MU_2=THETA(2)
```

```
MU_3=THETA(3)
MU_4=THETA(4)
LCL=MU_1+ETA(1)
CL=DEXP(LCL)
LV1=MU_2+ETA(2)
V1=DEXP(LV1)
LQ=MU_3+ETA(3)
Q=DEXP(LQ)
LV2=MU_4+ETA(4)
V2=DEXP(LV2)
S1=V1

$ERROR
IPRED=F
Y = F + F*EPS(1)

; Initial values of THETA
$THETA 1.68693E+00  1.61129E+00  8.19604E-01  2.39161E+00

;INITIAL values of OMEGA
$OMEGA BLOCK(4)
 1.65062E-01 -7.41489E-04  1.31429E-01  1.24115E-02  1.59565E-02  1.87547E-01 -1.27356E-02
1.39056E-02  3.32699E-02  1.49906E-01
;Initial value of SIGMA
$SIGMA
5.71632E-02    ;[P]

$ETAS FILE=etafile_phi.phi FORMAT=S1PE15.7  TBLN=6

$EST METHOD=1 INTERACTION NSIG=3 PRINT=1 FNLETA=2
$TABLE ID CL V1 Q V2 FIRSTONLY NOAPPEND NOPRINT FILE=etafile.par FORMAT=,1PE13.6
$TABLE ID ETA1 ETA2 ETA3 ETA4 LCL LV1 LQ LV2 FIRSTONLY NOAPPEND NOPRINT FILE=etafile.eta
$TABLE ID TIME IPRED DV CPRED CWRES NOAPPEND ONEHEADER FILE=etafile.tab NOPRINT
```

I.66 Improved Flexibility for using PRIOR information in TNPRI Problems (nm75)

A typical setup for using prior information from a first run (control stream), and using $PRIOR TNPRI for a second run, is to specify the MSFO file in the first run (call it pre_tnpri). In the second run, the MSF file is picked up with an MSFI record, and the first $PROB imports the prior information:

while the second $PROB evaluates the problem with a TNPRI prior:

```
$PROB Read in MSFI file.
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C
$MSFI pre_tnpri.msf ONLYREAD
$SUBROUTINES ADVAN=3 TRANS4
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
```

```
Y = F + F*EPS(1)

$PRIOR TNPRI (PROBLEM 2) IFND=0 MODE=0 PLEV=0 IVAR=0
$EST OMIT
```

Then, the information is used in the next $PROB (problem 2 in the same control stream) to be used as prior information:

```
$PROB Evaluation
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C REWIND
$THETA (0.001, 2.0)x4
$OMEGA BLOCK(4) VALUES(0.15,0.01)
$SIGMA
(0.6 )   ;[P]


$EST METHOD=1 INTERACTION NSIG=3 PRINT=10 NOABORT MAXEVAL=9999
     FORMAT=S1PE23.15
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

Keep in mind that the MSF file will contain the last covariance analysis performed in pre_tnpri. This includes covariance analysis for IMP, SAEM, BAYES, or SIR.

IVAR=0: Uses Rinv variance-covariance of a former problem (as of NMVI), if former problem used METHOD=1. If former problem used EM, BAYES, or SIR method, will retrieve the variance-covariance matrix that was specified with $COV MATRIX in the former problem.
IVAR=1: Uses Rinv*S*Rinv of a former problem (as of NMVI)
IVAR=2: Uses S of a former problem (as of nm75)

As of nm75, to use S-matrix assessed variance-covariance in a TNPRI problem, set IVAR=2:

```
$PRIOR TNPRI (PROBLEM 2) IFND=0 MODE=0 PLEV=0 IVAR=2
```

Make sure that when requesting the appropriate variance type, this was computed for the previous problem. For example, if METHOD=1, and $COV MATRIX=R was set for the previous problem, then S matrix is not available for the present problem and IVAR=2 should not be used. For IMP, SAEM, BAYES, or SIR evaluated covariances from the previous problem, use IVAR=0 for the present problem, regardless of whether $COV MATRIX=R, S ,or RSR (no MATRIX specification) was used.

To have more flexibilty in inputting desired prior information from text files, see the section below on $RCOV.

I.67 $RCOV and $RCOVI Record For Inputting Variance-Covariance information from another problem (NM75)

The $RCOV record can be used to bring in variance-covariance matrix of estimates results from a previous problem, and use it for subsequent use in assessing total standard errors of table items without having to re-calculate the variance with a $COV step. Consider the control stream

snippet below that brings in the covariance-variance of estimates from a previous control stream, located in rcov.cov, to be used in

```
$EST METHOD=1 FNLETA=2 FORMAT=,1PE23.15
$RCOV FILE=rcov.cov TBLN=1 DELIM=,
$TABLE ID TIME IPRED CL V1 Q V2 LTVCL VARCALC=3 FILE=rcov3.tab NOPRINT
NOAPPEND
```

Or, the record $RCOVI may be used to bring in the variance-covariance information from the inverse-covariance file:

```
$RCOVI FILE=rcov.coi TBLN=1 DELIM=,
```

So, $RCOV reading in variance-covariance from rcov.cov of a previous run can be used in place of executing the $COV step, and then used in evaluating the total standard error for table items (VARCALC=3). Thus, you may bring in a variance-covariance evaluated by any possible method (IMP, SAEM, BAYES, SIR) from the previous run, and use it for evaluating total standard errors of table items.

As another example, final parameter results using the $CHAIN record, and final empirical Bayes estimates of etas (and their covariances) using the $ETAS record, may also be read in from the previous run:

```
$CHAIN FILE=rcov.ext NSAMPLE=0 ISAMPLE=-1000000000 FORMAT=,1PE23.15
$ETAS FILE=rcov.phi TBLN=1 DELIM=,
$RCOV FILE=rcov.cov TBLN=1 DELIM=,
```

The final thetas, sigmas, omegas, etas, covariances of etas, and covariances of thetas, omegas and sigmas from the previous run can then be used without any re-estimation by setting FNLETA=2, and without any re-evaluation of the $COV step.

```
$EST METHOD=1 FNLETA=2 FORMAT=,1PE23.15
$TABLE ID TIME IPRED CL V1 Q V2 LTVCL VARCALC=3 FILE=rcov3.tab NOPRINT
       NOAPPEND
```

$RCOV can also be used to import a prior in a $PRIOR TNPRI problem, replacing the variance-covariance that was last written to an MSF file with alternative values. This may have value if you wish to use variance-covariance estimated by an alternative method in a previous run, (IMP, SAEM, BAYES, SIR), and use it as the prior in a TNPRI problem. A proper MSFI file was generated from a previous run (call it pre_tnpri), and it is desired to input variance-covariance from another source, perhaps yet another previous run (call it sirsampling, because the SIR assessed covariance will be used), so that the contents of the MSFI file, along with the $RCOV record, are read in on the first problem:

```
$PROB Read in MSFI file.
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C
;Comment out ONLYREAD in next line if msf file does not contain covariance
;step. If covariance step failed, it should still be okay to use ONLYREAD
$MSFI pre_tnpri.msf ONLYREAD
$SUBROUTINES ADVAN=3 TRANS4
```

```
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
Y = F + F*EPS(1)

; Bring in prior means
$CHAIN FILE=sirsampling.ext ISAMPLE=-1000000000 NSAMPLE=0
; Bring in prior variance
$RCOV FILE=sirsampling.cov TBLN=3 DELIM=S
$PRIOR TNPRI (PROBLEM 2) IFND=0 MODE=0 PLEV=0 IVAR=0
$EST OMIT
```

Then, the information is used in the next $PROB (same control stream) to be used as prior information:

```
$PROB Evaluation
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C REWIND
$THETA (0.001, 2.0)x4
$OMEGA BLOCK(4) VALUES(0.15,0.01)
$SIGMA
(0.6 )   ;[P]

; Optionally retrieve initial thetas, omegas, sigmas from a $CHAIN record
$CHAIN FILE=sirsampling.ext ISAMPLE=37 NSAMPLE=0
$EST METHOD=1 INTERACTION NSIG=3 PRINT=10 NOABORT MAXEVAL=9999
     FORMAT=S1PE23.15
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

Final parameter estimates from a previous run may also be entered in using the $CHAIN record, as shown above, to serve as the centrality parameters of the prior.

The lower and upper bound settings in the $THETA record on the second problem normally must match those of the problem that generated the prior. To get around this, you can set the NOMSFTEST option on the $MSFI record in the first problem. You will still get warnings if the bounds do not match.

Alternatively, you can omit an MSFI input altogether as follows:

```
$PROB Read in MSFI file.
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C
;$MSFI pre_tnpri.msf ONLYREAD
$SUBROUTINES ADVAN=3 TRANS4
$PK
MU_1=THETA(1)
```

```
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
Y = F + F*EPS(1)

; If no MSFI file, then enter $THETA, $OMEGA, $SIGMA records with some
; type of reasonable starting values. Only boundary values will be used,
; however, for internal parameterization.
$THETA (0.001, 2.0)x4
$OMEGA BLOCK(4) VALUES(0.15,0.01)
$SIGMA
(0.6 )   ;[P]

; Bring in prior means
$CHAIN FILE=sirsampling.ext ISAMPLE=-1000000000 NSAMPLE=0
; Bring in prior variance
$RCOV FILE=sirsampling.cov TBLN=3 DELIM=S
$PRIOR TNPRI (PROBLEM 2) IFND=0 MODE=0 PLEV=0 IVAR=0
$EST OMIT
```

Then, the information is used in the next $PROB (same control stream) to be used as prior information:

```
$PROB Evaluation
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
$DATA example1b.csv IGNORE=C REWIND
$THETA (0.001, 2.0)x4
$OMEGA BLOCK(4) VALUES(0.15,0.01)
$SIGMA
(0.6 )   ;[P]

; Optionally retrieve initial thetas, omegas, sigmas from a $CHAIN record
$CHAIN FILE=sirsampling.ext ISAMPLE=37 NSAMPLE=0
$EST METHOD=1 INTERACTION NSIG=3 PRINT=10 NOABORT MAXEVAL=9999
     FORMAT=S1PE23.15
$COV MATRIX=R PRINT=E UNCONDITIONAL
```

Final parameter estimates from a previous run may also be entered in using the $CHAIN record, as shown above, to serve as the centrality parameters of the prior.

I.68 Obtaining individual predicted values and individual parameters during MCMC Bayesian Analysis

Usually it is enough to obtain the population parameters thetas, omegas, and sigmas for each accepted sample, which is listed in the raw output file specified by FILE= of the $EST command. Occasionally one wishes to obtain a distribution of individual parameters, or even predicted values. This is done be incorporating additional verbatim code. This is best shown by

example 8. The BAYES_EXTRA_REQUEST is set to 1, informing NONMEM that PRED/PK/ERROR are to be called after an example has been accepted. The sample is indicated as accepted when NONMEM sets BAYES_EXTRA to 1. An IF block can be written by the user to, for example, write the individual parameters in a separate file (as shown in example 8), or the user may simply desire to obtain the minimum, maximum values obtained.

I.69 Imposing Thetas, Omegas, and Sigmas by Algebraic Relationships: Simulated Annealing Example

Additional algorithmic constraints may be imposed upon the model parameters, by use of the subroutine CONSTRAINT. This feature is available only for the EM and Bayesian algorithms. One use would be to slow the rate of reduction of the diagonal elements of the OMEGA values during the burn-in phase of the SAEM method. This is shown in example 9, where a user supplied annealing algorithm is used to replace the built-in one described earlier. By specifying OTHER=ANEAL.f90, where ANEAL.f90 was originally derived from a template of CONSTRAINT.f90 in the ..\source directory, the user supplied CONSTRAINT subroutine can be incorporated into the model. In example 9, whenever iteration number (ITER_NO) changes, a new OMEGA is evaluated that is larger than what was determined by the SAEM update. Typically, this expansion algorithm should be such that its impact decreases with each iteration.

I.70 Stable Model Development for Monte Carlo Methods

The Monte Carlo EM and Bayesian methods create samples of etas from multi-variate normal or t distributions. Because of this, some extreme eta values may be randomly selected and sent to the user-developed model specified in $PK, $PRED, $DES, and/or $ERROR. Usually these extreme eta positions are rejected by the Monte Carlo algorithm because of the poor resulting objective function. But occasionally, floating point overflows, divide by zero, or domain errors may occur, which can result in failure of the analysis. This may occur especially when beginning an analysis at poor initial parameter values. In NM72 NONMEM can recover from many of these errors, but there may be still occasion where such domain errors can terminate the analysis. Here are some suggestions to provide a more robust user model that protects against domain errors or floating point overflows, or allows NONMEM to reject these positions of eta that cause them and continue the analysis.

As of nm74, there are automated means of having your code protected against numerical errors. See the next section for that. The remainder of the present section is for historical purposes and for a greater understanding of protected code procedures.

If it is impossible to calculate the prediction due to the values of parameters (thetas or etas) from NONMEM, then the EXIT statement should be used to tell NONMEM that the parameters are inappropriate. The EXIT statement allows NONMEM to reject the present set of etas by setting an error condition index, which is in turn detected by classical NONMEM algorithms as well as the Monte Carlo algorithms. With the NOABORT switch of the $EST statement set, NONMEM may then recover and continue the analysis.

For example, if you have an expression that uses

```
LOG(X)
```
You may wish to flag all non-positive values and let NONMEM know when the present eta values are unacceptable by inserting:
```
IF(X<=0.0) EXIT
LOG(X)
```

On some occasions, you may need to have the calculations complete, then this expression could be transformed to:
```
LOG(ABS(X)+1.0E-300)
```
to avoid arguments to LOG that are non-positive.

If you have an expression which is ultimately exponentiated, then there is a potential for floating point overflow. An expression such as
```
EXP(X)
```
Which is likely to cause a floating point overflow could be filtered with
```
IF(X>100.0) EXIT
EXP(X)
```

Again, if the calculation must complete, such as when evaluating a user-defined likelihood, then you can place a limiting value, taking care that it causes little first derivative discontinuity:

```
EXPP=THETA(4)+F*THETA(5)
;Put a limit on EXPP, as it will be exponentiated, to avoid floating overflow
IF(EXPP.GT.40.0) EXPP=40.0
F_FLAG=1 ; Categorical data
; IF EXPP>40, then A>1.0d+17, A/B approaches 1, 1/B approaches 0 and Y is
; approximately DV
A=DEXP(EXPP)
B=1+A
Y=DV*A/B+(1-DV)/B      ; a likelihood
```

If your code uses SQRT() phrases, the expression within parentheses should be always positive. Sometimes expressions are calculated to near zero but slightly negative values, such as -1.1234444555E-16. Such values may legitimately be 0, but square rooting a negative number could result in failure of analysis. In such cases, the difficulty is due to the finite precision of the computer (e.g.,rounding error causing a value to be negative that would be non-negative on a machine with infinite precision) then the code should be written so as to produce the correct result. To protect against this,
```
SQRT(X)
```
could be converted to
```
SQRT(ABS(X))
```
Or
```
SQRT(SQRT(X*X))
```

The EXIT statement should not be used in such near-zero cases. It could lead to a failure in NONMEM with a message containing text such as

```
DUE TO PROXIMITY OF NEXT ITERATION EST. TO A VALUE AT WHICH THE OBJ.
FUNC. IS INFINITE
```

An EXIT may still be issued for values of X that are clearly negative because of erroneous inputs, and you may wish to flag this calculation, so that the estimation algorithm rejects this position:

```
IF(X<=-1.0E-06) EXIT
SQRT(ABS(X))
```

Such protection codes described above need not be inserted for every LOG(), EXP, or SQRT, but only if your analysis fails frequently or tends to be sensitive to initial values.

I.71 Stable Routines for Estimation Methods and Automated Protection Against Floating Point Exceptions (nm74)

As of nm74, a series of routines are available that protect against domain violations, divide by zero, and floating point overflows, so you won't need to insert the extra protective code lines described in the previous section. Each of these routines start with the letter P, followed by the name of the mathematical operation they are to perform. For example, PLOG is the protective code routine that performs the LOG operation. In addition, there are first derivative (such as PLOGD1), and second derivative (such as PLOGD2) companion routines available which NONMEM uses for analytical derivatives. The source code of these routines are available in ..\source\PROTECT.f90. If you wish to modify their behavior, then copy PROTECT.f90 to your run directory, rename and modify it, such as PROTECTB.f90, then refer to this modified code with

$SUBROUTINES OTHER=PROTECTB.f90

Make sure you modify the companion *D1 and *D2 routines appropriately as well.

The following protective code routines are available:

For all routines, if X=not a number, X is converted to machine precision value, which is about 1.0E-15, before performing an operation on it. If X>INFNTY (where INFNTY is approximately 1.0E+154), then X is converted to INFNTY before an operation is performed on it.

PLOG(x): returns LOG of x. If x<SMALLZ, where SMALLZ is approximately 2.8E-103, then LOG(SMALLZ) is returned.

PLOG10(x): returns LOG10 of x. If x<SMALLZ, where SMALLZ is approximately 2.8E-103, then LOG10(SMALLZ) is returned.

PSQRT(x): returns SQRT of x. If x<=0.0d+00, then 0 is returned.

PEXP(x): returns EXP of x. If x>100.0, then PEXP(100.0) is returned (avoids overflow).

PDZ(x): returns 1/x . Protects against divide by zero. If abs(x)<SMALLZ, then 1/SMALLZ is returned.

PZR(x): returns x . protects against zero. If abs(x)<SMALLZ, then SMALLZ is returned.

PNP(x): returns x. Protects against non-positive. If X<SMALLZ, then SMALLZ is returned.

PHE(x): returns x. Protects against high exponent. If X>100, then 100 is returned. Thus PEXP(x)=EXP(PHE(x)).

PNG(x): returns x. Protects against negative. If X<0.0, then 0.0 is returned.

PTAN(x): returns tan(x). Protects against returning infinity on inputs near pi/2.

PATAN(x): returns atan(x). Protects against large intputs.

PACOS, PASIN, returns acos(x), asin(x), respectively. If |X| is between 1.0 and 1+10**(-08), then x is submitted as 1 or -1. So, "dirty ones" are cleaned up, but values clearly beyond 1 are allowed to trip up the function, so the user is aware of the logical error in the code, and fix the issue.

Instead of replacing various operations with protected code operations by hand, you can ask NMTRAN to automatically convert your code to protected code with the following statement:

$ABBR PROTECT

NMTRAN will automatically replace all LOG (or DLOG) with PLOG, EXP (or DEXP) with PEXP, SQRT (or DSQRT) with PSQRT, / operations with *PDZ(), and B**E operations with PEXP(E*PLOG(B)).

When you use $ABBR PROTECT, you will find a considerable improvement in estimation stability, regardless of estimation method used.

I.72 Clinical Trial Design Evaluation and Optimization (NM75)

The design process can help you in designing or evaluating a clinical trial. It may be desired to evaluate specified time points, or find the optimal time points, dose levels, and number of time points appropriate for a particular sub-design, etc. The design algorithms have been modeled after POPED by Hooker et al., and PFIM by Mentre et al. (see references [22-27]). The following features are available.

$DESIGN

followed by a set of options.

APPROX=FO (default)

The nlme approximation method is specified here. First order (no interaction) is the default, and the appropriate type of covariance matrix is evaluated or used in optimization of the design. Other options are:

FOI
First order interaction with interaction
FOCE
first order conditional estimation
FOCEI
first order conditional estimation with interaction
LAPLACE
Laplace conditional estimation
LAPLACEI
Laplace conditional estimation with interaction

OFVTYPE=1 (default)

The objective function types are comparable to those of PopED:

0,1,3,4,5: design type: D-optimality, -log(det(FIM)), where FIM=Fisher Information Matrix (inverse of variance-covariance).
2: design type: A-optimality, -1/tr(1/FIM)
6: design type: DS-optimality, -log(det(FIM))+log(det(FIMuninteresting))
To identify parameters as uninteresting, place UNINT at the parameter in the same manner you would place FIX. For example:

```
$THETA 0.1   (0.5 UNINT) 0.4

$OMEGA BLOCK(2)
0.1 UNINT ; Sets entire block to uninteresting
0.01 0.1
```

7: design type: R-optimality (relative standard error), -1/tr(sqrt(1/FIM)/Parameter))

8: optimal design type: Individual Bayesian FIM, -log(det(Bayes FIM)), as described in the PFIM 4.0 manual [26]. In this case, the ETC() are optimized, as listed in the .phi table. The ETC() are the conditional variance-covariances of the individual's parameters. If just one subject (or subject type) is in the data file, then the criterion is that one subject's Bayesian FIM. If more than one subject (or subject type), then the Bayesian FIM of all the subjects, averaged together, is the criterion for design.

The change in variances with each printed iteration are listed in the additional raw file (.ext extension by default). The shrinkage information is reported in the main report file, and the individual empirical Bayes variances (ETC()) are reported in the .phi file. For shrinkage information, only the following have meaning:

EBVSHRINKSD(%)
EBVSHRINKVR(%)
EPSSHRINKSD(%)

EPSSHRINKVR(%)

When OFVTYPE=8, the change of average individual conditional variances (average empirical Bayes conditional variance) with each iteration are shown in the root.bfm file (where root is name of control stream file), and the final one on the -1000000000 line, during an optimization. The RAW file (by default named root.ext) shows only the starting, and final values of the standard errors of population parameters, as extra information.

9: Same as option 2, and using the UNINT filter.
10: Same as option 7, and using the UNINT filter.

GROUPSIZE=1 (default)

The GROUPSIZE is comparable to that of POPED, in which the FIM is multiplied by this number to provide the subject number size of the dataset template. For a template of one subject, GROUPSIZE would then offer the variance-covariance expected from GROUPSIZE number of subjects.

FIMTYPE or FIMDIAG=0

FIMTYPE or FIMDIAG may be set to the following, and corresponds to fimcalc.type in POPED:

0 (default): Full information matrix, using finite difference assessment for Theta, Omega, and Sigma variances and covariances.

No asssumptions are made about whether the approximated variance (C=h*Sigma*h+g*OMEGA*g, as described in Guide I, section E.2) used in FO is dependent on any Thetas via h and g.

1: Create a block diagonal information matrix of the estimates that assumes there is neglible correlation between Thetas and (Omegas,Sigmas). Also, independence of variance C with respect to thetas is assumed (that is, h and g's dependence on theta is ignored) . This is similar to the fim.calc.type=1 option in POPED or the diagonal option in PFIM, and performs finite difference at the y=E(f) position on the thetaxtheta portion of the information matrix (for thetas that are mu referenced, an analytical evaluation is made, and increases speed considerably), analytical evaluation of (SIGMAS OMEGAS)x(SIGMAS OMEGAS) portion of information matrix, and Thetas x (SIGMAS OMEGAS) cross variances are set to 0.

2: Create a block information matrix, by performing finite difference on the thetaxtheta portion of the information matrix, independence of variance C with respect to thetas is not assumed (that is, h and g's dependence on theta is not ignored), performs analytical evaluation of (SIGMAS OMEGAS)x(SIGMAS OMEGAS) portion of information matrix, and Thetas x (SIGMAS OMEGAS) are cross variances set to 0.

3: Create a full information matrix by performing finite difference on the Theta x Theta portion of the information matrix, and on the Thetas x (SIGMAS OMEGAS) portion of the information matrix, and performs analytical evaluation of (SIGMAS OMEGAS)x(SIGMAS OMEGAS) portion of information matrix.

For the film.calc.type=4 equivalent, see VARCROSS below.

With FIMTYPE=1, it is worth while to mu reference all non-fixed thetas. This will allow analytical evalutation of the thetaXtheta portion of the information matrix, and can provide considerable speed improvement. Even derivatives of mu-referenced thetas with associated OMEGA() fixed to 0 will be analytically evaluated.

VARCROSS=0 (default)

By default (VARCROSS=0), NONMEM interprets residual variance such as the following:

```
$PK
…
IPRED=F
Y=IPRED+EPS(1)+IPRED*EPS(2)
…
$SIGMA 0.5 0.15
```

as residual variance
var=SIGMA(1,1)+IPRED*IPRED*SIGMA(2,2)

To model variance as residual standard deviation squared:

```
var=(sqrt(SIGMA(1,1))+IPRED*sqrt(SIGMA(2,2))**2=
SIGMA(1,1)+2*sqrt(SIGMA(1,1)*SIGMA(2,2))*IPRED+IPRED*IPRED*SIGMA(2,2)
```

as in the manner of the PFIM 4.0 software, one would normally model the residual variance using thetas, such as:

```
$PK
…
IPRED=F
W=SQRT(THETA(4))+SQRT(THETA(5))*IPRED
Y=IPRED+W*EPS(1)
…
$SIGMA (1.0 FIXED)
```

However, when FIMTYPE=1 is used, the theta(4) and theta(5) are not identified as Sigma type parameters, and the appropriate contribution by theta(4) and theta(5) is not estimated. To provide an interpretation of residual variance incorporating the the cross-term
2*sqrt(SIGMA(1,1)*SIGMA(2,2))*IPRED
model using the SIGMAS as shown in the first example, and set VARCROSS to 1:

```
$PK
…
IPRED=F
Y=IPRED+EPS(1)+IPRED*EPS(2)
…
$SIGMA 0.5 0.15
…
```

```
$DESIGN … VARCROSS=1 FIMTYPE=1
```

FIMTYPE=1 VARCROSS=1 is equivalent to fim.calc.type=4 in POPED, and diagonal option in PFIM. Note that the variances are still modeled, rather than the standard deviation form. See ..\examples\optdesign\warfarin_pfim.ctl for an example of its use.

EOPTD=1

For each iteration, this creates a random sample of thetas, omegas, or sigmas, using the prior information. $PRIOR NWPRI prior information is required, and PLEV=0.999 must be specified. Best used with STGR. See example optdesign16.

SEED=223345

Set the starting seed for any random samples to be created, whether for EOPTD=1, or for FOCE type FIM in creating random etas (see below).

CLOCKSEED=0 (default)

As of nm75, the actual starting seed will be 10000*(seconds after midnight)+SEED (SEED may be set to 0 for this purpose), if CLOCKSEED=1. This allows a control stream to produce different stochastic results for automated replications, without the need to modify the seed value in the control stream file in each replication.

MODE=0 (default), 1, or 2

Used for specifying data and prediction value type when specifying APPROX=FOCEI.

MODE=0 means FOCEI with data at F(ETAsim), and predicted function evaluated at f(ETAsim), is to be used. This method works well.
MODE=2 means linearized FOCEI, with data at F(ETAsim)-G*ETAsim, and predicted function at F(ETAsim). Works well.
MODE=1 means FOCEI with data at F(ETAsim), and predicted function evaluated at the mode, f(ETAhat), is to be used (see below). The results are not satisfactory.

DATASIM=0 (default)

Normally, y-expectation evaluation of the FIM is performed. To actually simulate data, set DATASIM=1, and along with APPROX=FOCEI, this will produce simulated etas as well. So,

```
$DESIGN APPROX=FOCEI MODE=1 NELDER FIMDIAG=0 DATASIM=1 GROUPSIZE=32 OFVTYPE=0
```

will produce the most empirical, “clinical trial simulation” (CTS) style covariances, complete with simulated etas and eps, and standard FIM is assessed. If FIMDIAG>0, then a y-expectation covariance will be evaluated, but mode will be evaluated with the simulated data.

Additional Control Options for $DESIGN

The following options may be set within the $DESIGN record, and they operate exactly as their equivalents in the $EST record. In fact, these options specified in $DESIGN are transferred to the ESTM record in the FCON file, and NONMEM treats these in the same manner as if they came from the $EST record, but with the interpretation that is appropriate for the optimal design or evaluation setting.

SIGL
SIGLO
ABORT/NOABORT/NOHABORT
MAXEVAL (=0 indicates design evaluation, which is default, whereas MAXEVAL>0 indicates design optimization)
PRINT (control iteration printing during optimal design)
NUMERICAL/NONUMERICAL
-2LL/LIKELIHOOD/LLIKELIHOOD
SLOW/NOSLOW/FAST
POSTHOC
NOPRIOR
FORMAT
FILE
FNLETA

Thus, the following are equivalent:

```
$DESIGN FIMDIAG=1 OFVTYPE=6 APPROX=FO
        MAXEVAL=9999 NOHABORT PRINT=20 SIGL=10 POSTHOC

$DESIGN FIMDIAG=1 OFVTYPE=6 APPROX=FO
$EST    MAXEVAL=9999 NOHABORT PRINT=20 SIGL=10 POSTHOC
```

In addition, $DESIGN sets up the covariance step as $COV MATRIX=R UNCONDITIONAL without the user requiring this record entered in the control stream. If you wish to specify additional control for the covariance, you can add these in a $COV record, such as:

```
$DESIGN FIMDIAG=1 OFVTYPE=6 APPROX=FO
        MAXEVAL=9999 NOHABORT PRINT=20 SIGL=10 POSTHOC
$COV PRINT=E
```

An Example of a Design Evaluation

The options described above can be used to set up the design criteria, and can be used for finding the best design parameters (optimal design), or for evaluating design parameters proposed by the user (design evaluation). We will consider a simple example of design evaluation, before we consider additional options needed for optimal design. Using the example of ..\examples\optdesign\warfarin_pfim.ctl, note the following record:

```
$DESIGN GROUPSIZE=200 FIMTYPE=1 VARCROSS=1
```

The csv file contains only one subject (that is one subject type), but the subject number size may be specified with GROUPSIZE. Here, you request the variance-covariance one should expect from 200 subjects, all of the one particular subject type (equivalent time values, doses, covariates) defined in the csv file. You can have more than one subject in the csv, each with their own design (different doses, sample times, etc.). Also, by use of STRAT and STRATF (see below), one can desire a specific weighting of each subject type to contribute to the FIM.

FIMTYPE=1 means to evaluate the variance-covariance with the block-diagonal modality. VARCROSS=1 means to treat the residual variance model in the manner of PFIM 4.0, as described in the manual.

The intended result for this problem is essentially the FIM information and its derived components. In warfarin_pfim.coi, is the inverse covariance matrix (the FIM itself), for a sample size of 200 subjects. In the .res file, the -log(det(PFIM)) is given. Also, standard errors are given in report form here. The .ext file has essentially the same information is the .res file, but is in rows-column format. Much of these outputs (.res, .ext., .coi) are already a part of standard NONMEM outputs, they are now just recruited for optimal design and design evaluation.

Note the usual variance-covariance intended for an actual data set is created, but rather the asymptotic, y-expectation evaluation version, since the $DESIGN record was specified. Much of the control stream setup is as is if one were performing an FO estimation analysis on a real data set, while the addition of the $DESIGN record then re-purposes the analysis for design evaluation (or optimization if MAXEVAL>0).

View the standard error and variance-covariance outputs in the NONMEM report file, .ext file, .cov, .cor, and .coi files. For optimal design, you can view the final optimal design in the final data set, which can be outputted with the $TABLE step (see Examples for Optimal Design, below).

Options for Setting up Types of Optimal Design

The additional options for $DESIGN listed below are for optimizing parts of the design components. For example, the DESEL, DESELSTRAT, DESELMIN, DESELMAX can be specified for all the various design elements that you want optimized. It might be TIME for time samples, AMT for dose, or some type of covariate specific for the problem. Certainly, any combination of covariates can be requested to be optimized.

NELDER

Use Nelder method to search for optimal continuous parameters

FEDOROV

Use to find ideal set of discrete time points from a larger set of possible time points.

RS
Random Search method to find optimal continuous parameters

STGR
Stochastic Gradient method to find optimal continuous parameters

DISCRETE
Find optimal number of time points for each sub-design (subject template), and use NELDER method to find optimal continuous parameters.

DISCRETE_RS
Find optimal number of time points for each sub-design (subject template), and use RS method to find optimal continuous parameters.

DISCRETE_SG
Find optimal number of time points for each sub-design (subject template), and use STGR method to find optimal continuous parameters.

Next, data items must be specified to be optimized. This is done using the following options:

DESEL=data item
The data item (column name) that contains the design element (DESEL) values that are to be modified and optimized. For example, TIME column would indicate that you want times to be estimated.

DESELSTRAT=data item indicating stratification
The DESELSTRAT data item should contain integer indices to indicate what values are to be shared, and estimated together. If a record contains a value of 0 in the DESELSTRAT column, then this record is not included in the estimation process, and its value (say its time value in DESEL=TIME column) will not be changed. If the record contains a value >0 in DESELSTRAT, let us suppose a 1, then all records with the value of 1 in DESELSTRAT will share the same time value (or whatever DESEL selected), extimated together. Those records with value 2 will be another set of records which will share a time value, etc. Thus, within a subject, there may be a PK record and a PD record which should share the same time value. Also, a group of subjects may share the same time values. Within a subject, times will be automatically sorted as they are changed, so that NONMEM's time ordering policy is not violated.

DESELMIN=data item containing the minimal value

DESELMAX=data item containing the maximal value

You must impose boundaries on the values that are being optimized. That is done with these two data items. Only those records with a stratification value >0 in DESELSTRAT column will require a min and max value, and only those records that define that stratification value for the first time.

DESEL, DESELMIN, DESELMAX, and DESELSTRAT may be repeated for as many design elements are to be optimized. For example for times and amounts:

```
DESEL=TIME DESELMMIN=TMIN DESELMAX=TMAX DESELSTRAT=TSTRAT
DESEL=AMT DESELMIN=AMTMIN DESELMAX=AMTMAX DESELSTRAT=AMTSTRAT
```

NMIN=data item containing minimal number of time points to the subject

NMAX=data item containing maximal number of time points to the subject

If NMIN<0 or NMAX<0, then most previous non-negative value is used.
The NMIN and NMAX column are only used for the DISCRETE* analyses, to bound the number of time points that may be permitted for a given subject.
With DISCRETE*, the total N of time points among all subjects is determined by the total number of time points whose MDV=0 in the starting data set.

STRAT=data item containing grouping or stratification number pertaining to that subject

STRATF=data item containing starting fraction representation for the stratification value in data item STRAT

If STRAT and STRATF are specified, and there is at least one STRAT value >0, then the STRATF values are optimized, and represent the weight to the contribution of that subject to the Information matrix. For STRAT<=0, then their STRATF values are not optimized, and remain fixed at their initial values, but are still used as weights to the information matrix. It is up to the user to ensure that the sum of STRATF values among unique STRAT values sum to 1. If value of STRATF<0.0, then that subject is not included in the assessment.

Examples of Optimal Design

Here are portions of some examples of scripts located in examples\optdesign. Note that results are reported in the outputted tables. Study the data sets as well, as these establish part of the setup.

The following use FO by default.

Optdesign.ctl; FEDOROV search for ideal time points (fixed number), from a possible set of discrete time points given in the data set. The data set contains many time points, only 5 of which have MDV=0 (see optdesignw.csv). The FEDOROV algorithms tries different combinations of 5 time points among those given in the data set, and finds the best ones. Final (optimal) data set is listed in optdesign.tab.

```
$DESIGN FEDOROV FIMTYPE=1 MAXEVAL=9999 SIGL=12
$TABLE ID TIME EVID MDV DV NOPRINT NOAPPEND EXCLUDE_BY XCLD
       FILE=optdesign.tab
```

Optdesign2.ctl; Nelder search for ideal time points (fixed number). The data set has initial times, and these times are varied continuously, until an optimal set are found. The algorithm assures that the new times are sorted appropriately so as not to violate NONMEM rules. Notice in this example that the Omega values are UNINTed, and only the thetas and the residual proportional error are of interest.

```
$THETA
 1.68338E+00  1.58812E+00  8.12710E-01  2.37436E+00
;$OMEGA BLOCK(4) VALUES(0.0225,0.001)
$OMEGA (0.0225 UNINT)X4
$SIGMA
( 0.0225) ( 0.0001 UNINT)

$DESIGN NELDER FIMDIAG=1 OFVTYPE=6
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
        APPROX=FO MAXEVAL=9999 NOHABORT PRINT=20 SIGL=10 POSTHOC
$TABLE ID TIME EVID MDV DV IPRED NOPRINT NOAPPEND FILE=optdesign2.tab
       FORMAT=S1PE23.16
```

Optdesign5.ctl; Search of best times, using stochastic gradient method of search

```
$DESIGN FIMDIAG=0 STGR
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX APPROX=FO
        MAXEVAL=1000 SIGL=10 SIGLO=12 nohabort PRINT=50
$TABLE ID TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign5.tab
FORMAT=S1PE23.16
```

Optdesign6.ctl; Search of best times, using Random Search method of search

```
$DESIGN RS DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
           MAXEVAL=400 SIGL=12 nohabort PRINT=100
$TABLE ID TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign6.tab
FORMAT=S1PE23.16
```

Optdesign11.ctl; Nelder search of best times, as well as optimizing for number of time points in each subject. Note that the MDV are set to 0 or 1, to add or remove data points to a subject. To select for STGR, use DISCRETE_SG (optdesign12.ctl), and for RS, use DISCRETE_RS (optdesign13.ctl).

```
$DESIGN DISCRETE FIMTYPE=1
        NMIN=NMIN NMAX=NMAX DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN
DESELMAX=TMAX
        MAXEVAL=400 SIGL=10 nohabort PRINT=100
$TABLE ID TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign11.tab
FORMAT=S1PE23.16
```

Optdesign14.ctl; Nelder search of best times, as well as optimizing the weight contributions to the FIM for each subject.

```
$DESIGN NELDER FIMTYPE=1 STRAT=STRAT STRATF=STRATF DESEL=TIME
DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
           MAXEVAL=400 SIGL=12 nohabort PRINT=10
$COV PRINT=E
$TABLE ID STRAT STRATF TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign14.tab
FORMAT=S1PE23.16
```

Optdesign15.ctl; Nelder search of best times, DISCRETE optimization of number of data points for a subject, as well as optimizing the weight contributions to the FIM for each subject.

```
$DESIGN DISCRETE FIMTYPE=1 NMIN=NMIN NMAX=NMAX STRAT=STRAT
        STRATF=STRATF DESEL=TIME DESELSTRAT=TSTRAT
        DESELMIN=TMIN DESELMAX=TMAX
        MAXEVAL=400 SIGL=10 nohabort PRINT=10
$TABLE ID STRAT STRATF TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign15.tab
FORMAT=S1PE23.16
```

Optdesign16.ctl; STGR search of best times, and random samples of THETA are generated at each iteration, for an “expected value” of FIM, since EOPTD=1. Note the need for $PRIOR information.

```
$PRIOR NWPRI PLEV=0.9999
$THETAP  (1.68338E+00 FIXED) (1.58812E+00 FIXED) (8.12710E-01 FIXED)
(2.37436E+00 FIXED)
$THETAPV BLOCK(4) FIXED VALUES(0.03,0.001)

$DESIGN STGR EOPTD=1 FIMTYPE=1 SEED=1222345 DESEL=TIME DESELSTRAT=TSTRAT
DESELMIN=TMIN DESELMAX=TMAX
           MAXEVAL=9999 SIGL=12 nohabort PRINT=10 NOPRIOR=1
$TABLE ID CLN V1N QN V2N TIME EVID MDV DV NOPRINT NOAPPEND
FILE=optdesign16.tab  FORMAT=S1PE23.16
```

Optdesign17.ctl; The following evaluates FO with interaction for the FIM.

```
$DESIGN NELDER FIMTYPE=1
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
        APPROX=FOI MAXEVAL=1000 SIGL=10 nohabort PRINT=100
$TABLE ID TIME EVID MDV DV NOPRINT NOAPPEND FILE=optdesign17.tab
FORMAT=S1PE23.16
```

Optdesign21.ctl; The following evaluates FOCE with interaction, non-MODE, for the FIM. FOCE is best executed with a number of replicate subjects, perhaps 100, each with their own random set of etas. You may use $SIM record to introduce seed values. Because of the new REPL option in $DATA, you can have a template data file that contains just one example subject for each design type desired, and ask for these to be replicated REPL times. For example, in optdesign21.csv, there is only one subject type, but one can want 100 replicates of these:

```
$DATA optdesign21.csv IGNORE=C REPL=100
```

```
…
$DESIGN NELDER MODE=2 FIMTYPE=1 GROUPSIZE=0.01 DESEL=TIME DESELSTRAT=TSTRAT
DESELMIN=TMIN DESELMAX=TMAX APPROX=FOCEI
           MAXEVAL=400 SIGL=10 nohabort FNLETA=0
$SIM (1222345 NORMAL) (2933012 UNIFORM)
$TABLE ID ETA1 ETA2 ETA3 ETA4 TIME EVID MDV DV NOPRINT NOAPPEND
FILE=optdesign21.tab  FORMAT=S1PE23.16
```

Also, notice that GROUPSIZE=0.01. In this way, 100 subjects are in the data set, but you may want the FIM to represent the FIM per subject, not the FIM of 100 subjects. Thus GROUPSIZE is set to 1/100.

Optdesign22.ctl; The following evaluates FOCE with interaction (modal) for the FIM. In this case, a SEED for source 1 may be entered at the $DESIGN record. If SEED is given at the $DESIGN record, as well as on a $SIM record, the SEED given at $DESIGN SEED option will have precedence for seed source 1.

```
$DATA optdesign21.csv IGNORE=C REPL=100
…
$DESIGN NELDER MODE=1 FIMTYPE=1 GROUPSIZE=0.01 SEED=1222345 APPROX=FOCEI
           DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
           MAXEVAL=400 SIGL=10 nohabort FNLETA=0 PRINT=10
$TABLE ID ETA1 ETA2 ETA3 ETA4 TIME EVID MDV DV NOPRINT NOAPPEND
FILE=optdesign22.tab  FORMAT=S1PE23.16
```

Optdesign25.ctl; The following evaluates FO for the FIM, for best times, and best doses.

```
$DESIGN NELDER FIMTYPE=1 DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN
DESELMAX=TMAX
           DESEL=AMT DESELSTRAT=AMTSTRAT DESELMIN=AMTMIN DESELMAX=AMTMAX
           MAXEVAL=400 SIGL=10 nohabort
$TABLE ID TIME DOSE EVID MDV DV NOPRINT NOAPPEND FILE=optdesign25.tab
FORMAT=S1PE23.16
```

See example optex6d17.ctl for an A-optimality Nelder Mead search of best time points for a TMDD problem.

See example optex6d17_8.ctl for an example of Bayes FIM (OFVTYPE=8). The table optex6d17_8.tab reports the optimal data set, and optex6d17_8.tab records the intermediate results of conditional variances during optimization. Final conditional variances are reported in optex6d17_8.phi.

You can also run several problems in one control stream, to use one optimizer followed by another. To do this across two problems note the following setup in warfarin_optimize.ctl:

```
$PROB 1
…
$DESIGN RS FIMDIAG=1 GROUPSIZE=32 OFVTYPE=0 DESEL=TIME DESELSTRAT=TSTRAT
           DESELMIN=TMIN DESELMAX=TMAX
           MAXEVAL=4000 SIGL=10 nohabort PRINT=100
$TABLE ID TIME EVID MDV DV IPRED NOPRINT NOAPPEND FILE=warfarin_optimize.tab
FORMAT=S1PE23.16
$TABLE ID TIME AMT RATE EVID MDV DV TSTRAT TMIN TMAX NOPRINT NOHEADER
NOAPPEND FILE=warfarin_1.CSV  FORMAT=S1PE23.16
```

```
…
$PROB 2
$INPUT ID TIME AMT RATE EVID MDV DV TSTRAT TMIN TMAX
$DATA warfarin_1.csv NOOPEN
(1X,10(1X1PE23.16))

…
$DESIGN NELDER FIMDIAG=1 GROUPSIZE=32 OFVTYPE=0
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
        MAXEVAL=4000 SIGL=10 nohabort PRINT=100
$TABLE ID TIME EVID MDV DV IPRED NOPRINT NOAPPEND FILE=warfarin_optimize2.tab
FORMAT=S1PE23.16
$TABLE ID TIME AMT RATE EVID MDV DV TSTRAT TMIN TMAX NOPRINT NOAPPEND
FILE=warfarin_nelder.CSV  FORMAT=S1PE23.16
```

Notice the way in which $DATA of the second problem is used to pick up the file created in the first problem, to resume where the first problem left off regarding the modified time points.

You can also have one problem stack two or more $DESIGN records, as shown in warfarin_optimize3.ctl:

```
$DESIGN RS FIMDIAG=1 GROUPSIZE=32 OFVTYPE=0
           APPROX=FOI SIGL=10 NOHABORT PRINT=100 MAXEVAL=400
           DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
$DESIGN STGR DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
$DESIGN NELDER DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
```

The only disadvantage to this method is that while the .ext and .phi files will record the intermittent change and results of each optimal design step, the main report file (.res or .lst) will only report the results of the final analysis, and tables will be outputted only after the last analysis, so you won’t be able to observe intermediate observation times.

The $DESIGN record will use the values of the options of the previous $DESIGN record for most options, except those of DESEL, DESELSTRAT, DESELMIN, DESELMAX. These options need to be specified for each $OPT record.

You can use the $SIML TRUE=PRIOR with prior information to randomly generate thetas (and omegas and sigmas, if desired), and optimal time points for each of these theta sets will be obtained (priortrue.ctl):

```
…
$PRIOR NWPRI PLEV=0.99
$THETAP (0.15 FIXED) (0.80 FIXED) (1.0 FIXED)
$THETAPV BLOCK(3) ; 30% CV
0.0025 FIX
0.0   0.07
0.0   0.0   0.11

$SIM (4442223) SUBPROB=100 TRUE=PRIOR
$DESIGN NELDER FIMDIAG=1 GROUPSIZE=32 OFVTYPE=0
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX
```

```
        MAXEVAL=4000 SIGL=10 nohabort PRINT=100 NOPRIOR=1
$TABLE ID TVCL TVV TVKA TIME EVID MDV DV IPRED NOPRINT NOAPPEND
FILE=priortrue.tab
```

In this example, 100 sub-problems are generated, each with their own theta sets. The variance on the thetas is about 30%CV, and by observing the optimal times listed in priortrue.tab, you will get a sense of what kind of variability of optimal time points will be obtained.

As of nm751, you can add a prior from a previous analysis, or design, to a problem. Problem warfarin_desprior_opt4.ctl is a typical design evaluation, generating a .coi file. Then, warfarin_desprior_opt5.ctl considers another design, and adds the prior information in warfarin_desprior_opt4.coi, to the FIM:

```
…
$RCOVI FILE=warfarin_desprior_opt4.coi


$DESIGN GROUPSIZE=50 FIMTYPE=1 VARCROSS=1 APPROX=FO
        SIGL=12 MAXEVAL=9999 PRINT=100
        DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX


$TABLE ID TIME TSTRAT TMIN TMAX EVID MDV IPRED NOAPPEND NOPRINT
FILE=warfarin_desprior_opt5.tab
…
```

Such an example is given in reference [27].

The same problem is done without $RCOVI in warfarin_desprior_opt.ctl. Note that the first subject represents the design of warfarin_desprior_opt5, and the other 4 subjects represents the elementary designs in warfarin_desprior_opt4, with their TSTRAT 0 so that they do not change with the optimization, but remain as unchanged, prior information.

You can also add prior information via $PRIOR NWPRI. Only the dispersion information is used, as centrality is assumed at the set population parameters. For example, $THETAPV, $OMEGAPD, and $SIGMAPD information is used, along with $SIGMA and $OMEGA, to construct the prior information. Example warfarin_pfim2p.ctl shows this:

```
$THETA
0.15 ;[CL]
8.0  ;[V]
1.0  ;[KA]

$OMEGA (0.07) (0.02) (0.6)
$SIGMA 0.01

$PRIOR NWPRI
$THETAP
0.15 FIXED ;[CL]
```

```
8.0   FIXED ;[V]
1.0   FIXED ;[KA]

$THETAPV BLOCK(3)
0.00005 FIXED
0.0   0.06
0.0   0.0   0.015
$OMEGAP (0.07 FIXED) (0.02 FIXED) (0.6 FIXED)
$OMEGAPD (32 FIXED) (40 FIXED) (56 FIXED)
$SIGMAP 0.01 FIXED
$SIGMAPD (96 FIXED)

$DESIGN GROUPSIZE=32 FIMTYPE=1 VARCROSS=1 APPROX=FO NOPRIOR=0
```

Note that NOPRIOR can be used to turn on or off the prior contribution.

As of nm751, to include a theta that is only involved in the residual variance, and when using FIMDIAG=1, declare that variable “Sigma-like” with GRD=TS() option, as shown in examples\warfarin_pfim23.ctl for the gamma parameter:

```
$PROBLEM WARFARIN
$INPUT ID TIME AMT RATE EVID MDV DV TSTRAT TMIN TMAX
$DATA warfarin_pfim2.csv ignore=C


$SUBROUTINES ADVAN2 TRANS2
$PK
MU_1=LOG(THETA(1))
MU_2=LOG(THETA(2))
MU_3=LOG(THETA(3))
;MU_4=LOG(THETA(4))
CL=EXP(MU_1+ETA(1))
V=EXP(MU_2+ETA(2))
KA=EXP(MU_3+ETA(3))
gamma=THETA(4)
S2=V
F1=1.0

$ERROR
IPRED=A(2)/V
Y=IPRED + (IPRED**gamma)*EPS(1)

$THETA
0.15 ;[CL]
8.0   ;[V]
1.0   ;[KA]
1.0 ;[gamma]

$OMEGA (0.07) (0.02) (0.6)
$SIGMA 0.01
$DESIGN GROUPSIZE=32 FIMTYPE=1 VARCROSS=1 APPROX=FO
$EST GRD=TS(4) SIGL=10
```

The standard errors to gamma and sigma, however are somewhat inaccurate (by about 25%, when compared with the equivalent clinical trial simulations warfarin_pfim23_cts.ctl (FOCEI)

and warfarin_pfim23_ctsi.ctl (IMP)). Therefore, avoid incorporating a theta parameter in the residual variance assessment when performing optimal designs and evaluations.

Serial correlation models may also be implemented. According to Nyberg et . al [31], incorporating serial correlation can account for possible model mis-specification, and allow a wider distribution of times among the samples to be found, in order to make the sampling more robust. The following example, warfarin_fobc.ctl shows how this may be implemented (as of nm751, FIMDIAG=1 may be used with a serial correlation problem).

```
$PROBLEM WARFARIN
$INPUT ID TIME AMT RATE EVID MDV DV TSTRAT   TMIN   TMAX   REPZ
$DATA warfarinz.csv ignore=@
$SUBROUTINES ADVAN2 TRANS2
$ABBR DECLARE T(NO), INTEGER I, DOWHILE J
$PK
MU_1=LOG(THETA(1))
MU_2=LOG(THETA(2))
MU_3=LOG(THETA(3))
CL=THETA(1)*EXP(ETA(1))
V=THETA(2)*EXP(ETA(2))
KA=THETA(3)*EXP(ETA(3))
S2=V
F1=1.0
$ERROR
 IF(NEWIND.NE.2) I=0
 IF (NEWL2 ==1 .AND. MDV==0) THEN
    I=I+1
    T(I)=TIME
    J=1
    DO WHILE (J<=I)
    CORRL2(J,1)=EXP(-THETA(4)*(TIME-T(J)))
    J=J+1
    ENDDO
 ENDIF
IPRED=A(2)/V
Y=IPRED*(1.0+EPS(1)) + EPS(2)
$THETA
0.15 ;[CL]
8.0  ;[V]
1.0  ;[KA]
(0.1) ; [COR]
$OMEGA (0.07 ) (0.02 ) (0.6 )
$SIGMA (0.01 ) (0.001 FIX )
$DESIGN GROUPSIZE=100 FIMDIAG=1 APPROX=FO
         MAXEVAL=9999 SIGL=12 DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN
         DESELMAX=TMAX
$EST GRD=TS(4)
$TABLE ID TIME AMT EVID MDV DV IPRED NOPRINT NOAPPEND FILE=warfarin_fobc.tab
```

This results in 7-8 distinct time points (warfarin_fobc.tab). The analysis of real data itself may not require the CORRL2 algorithm, it is mainly designed to create a greater spread of distinct time samples for robustness, as mentioned earlier.

This above example requires some decision about the de-correlation rate THETA(4), which could be arbitrary. Also, it might be more desirable to consider that the farther apart that data are in terms of value (rather than in time), the less correlated they are, to provide a separation of samples based on the degree of change in concentration. In other words, in regions where the concentration changes rapidly, the samples may be more closely spaced, and in regions where concentration changes less rapidly, samples would be spaced farther apart. Furthermore, relative change in concentration could be most appropriate. For this purpose, the following control stream (..\examples\optdesign\warfarin_fobc5.ctl) would be suitable:

```
$PROBLEM WARFARIN
$INPUT ID TIME AMT RATE EVID MDV DV TSTRAT   TMIN   TMAX   REPZ
$DATA warfarinz2.csv ignore=@
$SUBROUTINES ADVAN2 TRANS2
$ABBR DECLARE T(NO), INTEGER I, DOWHILE J
$PK
MU_1=LOG(THETA(1))
MU_2=LOG(THETA(2))
MU_3=LOG(THETA(3))
CL=THETA(1)*EXP(ETA(1))
V=THETA(2)*EXP(ETA(2))
KA=THETA(3)*EXP(ETA(3))
S2=V
F1=1.0
$ERROR
 IPRED=A(2)/V
 MEASURE=DV
; MEASURE=LOG(DV) ; Alternative measure
 IF(NEWIND.NE.2) I=0
 IF (NEWL2 ==1 .AND. MDV==0) THEN
    I=I+1
    T(I)=MEASURE
    CORRL2(I,1)=1.0
    J=1
    DO WHILE (J<I)
    DVAL1=MEASURE
    DVAL2=T(J)
; 0.999 needed to make sure correlation between two different data
; points is less than 1, even if they are of identical value
    CORRL2(J,1)=0.999*EXP(-THETA(4)* &
SQRT( (DVAL1-DVAL2)*(DVAL1-DVAL2)/(DVAL1*DVAL1+DVAL2*DVAL2)   ) )
; Alternative correlation algorithm
;    CORRL2(J,1)=0.999*EXP(-THETA(4)*ABS((DVAL1-
DVAL2))/(ABS(DVAL1)+ABS(DVAL2)) )
    J=J+1
    ENDDO
 ENDIF
Y=IPRED*(1.0+EPS(1)) + EPS(2)
$THETA
0.15 ;[CL]
8.0  ;[V]
1.0  ;[KA]
(1.0 FIXED) ; [COR]
$OMEGA (0.07 ) (0.02 ) (0.6 )
$SIGMA (0.01 ) (0.001 FIXED)
$DESIGN GROUPSIZE=100 FIMDIAG=1 APPROX=FOI
```

```
         MAXEVAL=99999 PRINT=200 SIGL=9 DESEL=TIME DESELSTRAT=TSTRAT
DESELMIN=TMIN DESELMAX=TMAX
$EST GRD=TS(4)
$TABLE ID TIME AMT EVID MDV DV IPRED NOPRINT NOAPPEND FILE=warfarin_fobc5.tab
```

The measure now used is not TIME, but DV. The IPRED should not be used as a measure, as NONMEM cannot handle random variables (those with eta associated with them) within the CORRL2 structure. Note that the THETA(4) is fixed to 1, which is very reasonable for any problem, and we have removed the arbitrariness of having to select its value.

Table warfarin_fobc5.tab shows final times of samples that provide a reasonable distribution to cover incremental changes in the predictive values of the PK curve. What is also of value is that the standard errors approximate what one would get with a clinical trial simulation, not using the CORRL2 algorithm (cts_fobc5).

I.73 Parallel Computing (NM72)

General Concepts of Parallel Computing

If you have a run that takes a long time to estimate, you may submit it for parallel computing. This is the process of splitting the objective function evaluations of individual subjects among a set of computers or CPUs, to speed up analysis of a particular run. Only estimations ($EST) and covariance assessments ($COV) are parallel processed.

From our tests, we have found that the optimal number of processes needed depends on the problem. On one extreme, if the problem contains many subjects, and each subject takes a long time to evaluate because of a large number of differential equations, and/or a large number of dose events, so that one subject takes a minute to evaluate on each function evaluation, then as many cores as there are subjects would still be efficient. Our parallelization algorithm does not split up the problem beyond one subject per process. On the other hand, if the problem takes just 0.01 second to evaluate all subjects for a function evaluation, then it may not be worth using parallel processing. For each function call, the manager process packages a subset of subjects and sends the data to a worker process, then the worker process returns its results to the manager, and the manager summarizes the information from all of the workers. For the next function call, the procedure begins again.

The length of time to perform one subject's evaluation in a function call varies with the estimation method as well. In importance sampling, there is one function call per iteration, and if you have high ISAMPLE, then it can take some time to evaluate each subject. Such a problem is very efficiently parallelized. On the other hand, BAYES analysis performs only one sample per subject per function call, so it may perform a function evaluation very quickly on a single process, and parallelization may not improve computation time.

NONMEM can parallelize across computers as well as to individual cores on those computers. However, depending on your intranet connection between computers, the process will be a little slower across computers than among cores on the manager computer alone. Eight to 16 cores

per computer with about 2 GB RAM per core should be sufficient for almost any problem in NONMEM. Alternatively, 0.4 GB per core is more than enough for many NONMEM problems. If there is insufficient RAM, many operating systems utilize virtual memory (usually mapped to hard drives), but this may slow down execution.

The manager process is the user's process that runs the nmfe75 script, reads the control stream file, executes NMTRAN, and runs the main NONMEM process. The worker process is NONMEM in worker mode, not taking any input from the user, only from the manager NONMEM process.

If the manager process is on one computer and the worker process is on a second computer, then a network communication must be possible between these computers, and the manager computer must be able to have access to a network drive and directory that is mapped to a drive and directory that is locally accessible by the worker directory. It is possible for this directory to also be accessible from the worker computer as a network drive, but this can slow down the data transfer. If the manager process and the worker process are on the same computer, but are simply running on different cores, then they can communicate on an agreed upon directory on a local drive. Both manager and worker must have read and write privileges.

To obtain the greatest efficiency in parallel computing, make sure the LIM values to buffers 1, 3, 4, 13, and 15 are set to the largest needed for ensuring the buffers can be loaded all into memory, and no file reading and writing is required. See the section I.10 Changing the Size of NONMEM Buffers on how to do this.

File Passing Interface (FPI) Method

Two information passing methods between manager and worker processes are available, file passing interface (FPI), and message passing interface (MPI). The FPI method requires no additional software installation other than what is normally required to run a single process NONMEM run (that is, it needs only NONMEM plus compiler). All transfer of information between a manager NONMEM process and its worker processes is done by writing files to a directory throughout the analysis.

Message Passing Interface (MPI) method

The message passing interface (MPI) allows exchange of data much more rapidly than the FPI. MPI requires installation of free but ubiquitous use third party software, and we recommend you set this up for your cluster. Fortunately, MPI is free and available for most platforms and Fortran compilers. The MPI's speed is particularly notable over FPI when FOCE, Laplace, SAEM and BAYES are done. For ITS and IMP/IMPMAP, the speed difference is less noticeable. There is some initial file copying required between manager and worker directories (or computers), but after the initial loading of the NONMEM processes, all information transfer is via the message passing interface without requiring file transfer.

The PARAFILE

Parallel computing with NONMEM 7.2.0 uses a "parallel file" (or parafile) that controls the parallelization process implemented by NONMEM, and is written by the user. The NONMEM

installed ..\run directory has sample pnm files that can be used as a template. The name of the parallel file may be given at the command line as:

Nmfe75 myexample.ctl myexample.res –parafile=myparallel.pnm

(quotes of some kind may be needed for Windows, otherwise the parameters are improperly parsed). This parallel file will remain in effect throughout the control-stream file, to be used in all $EST methods.

If no –parafile switch was given, then the default name parallel.pnm is assumed. The reserved default name of parallel.pnm should not be used, as it is only for the worker process. Make sure no file called parallel.pnm exists in your manager's run directory.

The PARAFILE option may be alternatively set to the keywords ON or OFF. If a PARAFILE parameter is set to OFF in a $EST command, then parallelization does not occur for that $EST command. If a subsequent PARAFILE is set to ON, the parallelization occurs using the most recent PARAFILE file specification. If –parafile=off is given at the command line, then no parallelization is done for the entire control stream, regardless of PARAFILE options within the control stream file.

The format of the parallel file is best shown by this example, which is heavily commented to describe the meanings of the records and options available. This parafile example is set up for FPI method on Windows:

```
$GENERAL
NODES=2 PARSE_TYPE=3 PARSE_NUM=200 TIMEOUTI=60 TIMEOUT=10 PARAPRINT=0
TRANSFER_TYPE=0
; NODES=number of nodes (that is process, whether cores or computers)
; SINGLE node: NODES=1
; MULTI node (node means process, whether cores or computers): NODES>1
; WORKER node: NODES=0
;
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading time.
; This setting of parse_type will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU mode.
;
; timeouti=seconds to wait for node to start.  if not started in time,
; deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. As of nm75, only a warning
; is given if node takes longer than timeout minutes to return
; paraprint=1  print to console the parallel computing process.  Can be
; modified at run-time with ctrl-B toggle.
; Regardless of paraprint setting, <control_stream>.log always records
; parallelization progress.
;
; transfer_type=0 for file transfer, unloading and reloading workers with
```

```
; each estimation
; transfer_type=1 for mpi
; transfer type=2 for file transfer, maintaining a single loaded process
; throughout the run.

;THE EXCLUDE/INCLUDE may be used to selectively use certain nodes,
; out of a large list.
; $EXCLUDE 5-7 ; exclude nodes 5-7
; or
;$EXCLUDE ALL
;$INCLUDE 1,4-6

$NAMES ; Give a label to each node for convenience
1:MANAGER
2:WORKER1
3:WORKER2
4:WORKER3

$COMMANDS ;each node gets a command line, used to launch the node session.
; Command lines must be on one line for each process.  The following commands
; are for FPI method on Windows.

; First node is manager, so it does not get a command line when using FPI
1:NONE
;
; load on a core of the same computer as manager:
; For psexec, notice that the worker directories are named
; as the worker sees them, not as the manager sees them.  Very important
; distinction for remote worker computers.
; -w refers to working directory for particular process
2:psexec -d -w worker1\ cmd.exe /C nonmem.exe
; load on a core of the same computer as manager:
3:psexec -d -w worker2\ cmd.exe /C nonmem.exe
; load on a core of a different computer than manager:
4:psexec \\any_computer -d -w c:\share\worker3 cmd.exe /C nonmem.exe

$DIRECTORIES ; Names of directories as a manager sees them.
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY. Make it NONE if no
; common directory is to be used. This is the best option.
2:worker1\ ; NEXT SET ARE THE WORKER directories.
3:worker2\
4:w:\share\worker3\ ; This directory is on a different computer from manager

$IDRANGES ; USED IF PARSE_TYPE=3
1:1,50
2:51,100
```

You may load the problem as follows:
nmfe75 mycontrol.ctl mycontrol.res -parafile=fpiwini8.pnm

Strictly speaking, drive letter mapping on the manager side is not necessary. One could refer to the network drive as \\any_computer\share\worker3\ instead of w:\share\worker3 in the pnm file.

The most versatile PARSE_TYPE selections are 2 and 4. If you select PARSE_TYPE=0, make sure that PARSE_NUM>=(no. of subjects)/(no. of nodes), otherwise the problem may not run

properly. If you select PARES_TYPE=3, make sure all subjects are accounted for in the $IDRANGES listings.

By default, at the beginning of each step (estimation, covariance, wres diagnostics, etc.), the individuals are evenly distributed among the nodes. As of nm742, an option called PARSE_PRESERVE may be set to a non-zero value to have the most edfficient ID range settings that were determined from the previous step be used for the first iteration after the next step. For example, if the previous step was estimation, then the ID distributions among the nodes of the last iteration of estimation will be used as the start of covariance assessment. The supposition is that the load distribution that was assessed in estimation should be suitable for covariance assessment. Set the bit for the appropriate previous step:
If Previous estimation was estimation, use its last ID load distribution: set bit 0
If Previous estimation was chain method, use its last ID load distribution: set bit 1
If Previous estimation was covariance, use its last ID load distribution: set bit 2
If Previous estimation was wres assessment, use its last ID load distribution: set bit 3
If Previous estimation was final eta assessment, use its last ID load distribution: set bit 4
If Previous estimation was simulation, use its last ID load distribution: set bit 5
If Previous estimation was nonparametric, use its last ID load distribution: set bit 6

The most sensible is to set bits 0, 2, 3, and 4, as these algorithms tend to provide the most accurate load distribution assessments, suitable for launching the next step with the same initial load distribution (PARSE_PRESERVE=29). After several iterations, each step empirically adjusts its load distribution anyway, regardless of PARSE_PRESERVE setting.

The $NAMES record is optional. If left out, or if a name is not defined for a process, the default name is MANAGER for position 1, WORKER1 for position 2, WORKER2 for position 3, etc.

The structure of the COMMANDS lines for launching the worker nodes is completely dependent on your computing and parallel distribution environment, and the syntax requirements of the launching program. The psexec.exe program (located in the ..\run directory of the NONMEM folder) is available for Windows to launch a program on the same computer (as with the first 2 worker nodes), or on a remote computer (last worker node). An alternative launching program may be used. The –w option in psexec specifies the working directory (as the worker identifies it) from which the NONMEM programs is to be launched.

The index numbers that begin an item in a list (1:, 2:, etc), are optional. If present, it refers to node 1 (manager), node 2, node 3, etc. If not present, the item number is determined by the order in which the item was listed. It is best to use them for greater clarity.

In $DIRECTORIES, the directory names must follow syntax rules of the particular operating system. The $DIRECTORIES record is optional. If left out, or if a directory name is not given for a process. Then the default values are NONE for common directory (position 1), worker1 for the first worker (position 2), worker2 for the second worker (position 3), etc. These are interpreted as sub-directories to the present run directory.

There is no need to create the worker directories ahead of time (although its parent directory, whether local or network, must exist), or be concerned with populating them with the appropriate files, including the nonmem executable. NONMEM will take care of this automatically. For example, while w:\share needs to exist before the run, as it was the share directory that needed to be set up, w:\share\worker3 did not have to exist before the NONMEM run. Make sure that the managers and workers have appropriate read/write access to these directories, and proper privileges to load on remote computers.

The $COV statement also allows a PARAFILE setting, to turn on or off parallel computing for the $COV step for classical NONMEM methods, or changing the parallelization profile.
The $TABLE statement also allows a PARAFILE setting, to turn on or off parallel computing for calculation of weighted residuals, or changing the parallelization profile (nm74).

The $SIML and $NONP also have PARAFILE settings to turn on or off parallel computing for their calculations.

Sometimes the parallelization log files can become very large during the $EST and $COV steps. Each of these records have parafprint options to control the print intervals. For example.

$EST METHOD= BAYES parafprint=100

will cause only every 100th iteration to be printed.

Or, you can control the print iterations globally with the –parafprint option at the command line:
nmfe75 mycontrol.ctl myconotrl.res –parafile=mpiwini8.pnm –parafprint=100

Examples of PARAFILE files are given in NONMEM's ..\run directory as a list of *.pnm files. Examples are shown in the next sections as well. The files fpiwini8.pnm, fpilinux8.pnm, mpilinux8.pnm, and fpilinux8.pnm are particularly versatile, in that they are useful for multiple cores on a single computer, and are designed to be used in any run directory.

Substitution Variables in the parafile

Substitution variables provide flexibility in the use of the parafile. Certain substitution variables are reserved words as follows, which can be passed as arguments to the worker nonmem executable (although typically this is not necessary to do so). That is, they are placed at the end of a $COMMANDS process command line, coming after nonmem.exe, as arguments to nonmem.exe, as needed:

<control_stream>: substitute the control stream file name given at the command line of the nmfe75 script.

<licfile>: substitute the entire –licfile option, including its value, provided by the nmfe75 script. For example, -licfile=c:\mynonmem\license\nmlicense.lic is substituted into <licfile>.

<background>: substitute –background switch, if given by user on the nmfe75 command line.

<parafile>: substitute –parafile option, such as –parafile=myparallel.pnm, given at nmfe75 command line. Never use the <parafile> switch on a worker process.

Substitution variables need not be used just as arguments to the nonmem executables that are loaded. In some cases, they are needed in other parts of the command line of the process launch, or in the directory listing of $DIRECTORIES. In such cases, it is not desired to substitute the entire
–option=value
string, but just the value portion. Where the value of the option itself is to be substituted, use <<option>>. For example, suppose the nmexec option is used to specify an alternative nonmem executable name. In such cases, you would specify <<nmexec>> in place of the usual nonmem.exe:

```
3:psexec -d -w worker2\ cmd.exe /C <<nmexec>> <control_stream>
```

This principle of using <> versus <<>> applies to the other substitution parameters as well.

You may also define your own substitution parameters to be used in the pnm file, as long as the substitution variable begins with a [or <. For example, you may enter at the command line of nmfe75 the following variable [wd] for a worker directory definition:

Nmfe75 mycontrol.ctl mycontrol.res -parafile=mypara.pnm [wd]=c:\myworker

and your pnm file may contain the following loading $COMMANDS:

```
2:psexec -d -w [wd]\q1 cmd.exe /C nonmem.exe
3:psexec -d -w [wd]\q2 cmd.exe /C nonmem.exe
```

and $DIRECTORIES

```
2:[wd]\q1
3:[wd]\q2
```

For user defined variables, the value of the variable is substituted into the placeholder, rather than the entire [var]=value. Then c:\myworker will be substituted in place of [wd], in the $COMMANDS and $DIRECTORIES entries. Add as many substitution variables as you need to create a generalized pnm file.

To make the user substitution process even more flexible, default values for these variables may be defined, in case the user does not specify a value for it on the command line. For example, in ..\run\fpiwini8.pnm, There is a section called $DEFAULTS (or $DEFAULT), where a default value for [nodes] is given:

```
$DEFAULTS
[nodes]=8
```

, and in $GENERAL, [nodes] is used as the number of nodes:

```
$GENERAL
; [nodes] is a User defined variable
NODES=[nodes] PARSE_TYPE=2 PARSE_NUM=50 TIMEOUTI=500 TIMEOUT=2000 PARAPRINT=0
TRANSFER_TYPE=0
```

Make sure that $DEFAULTS is placed at the head of the file, so the default variable substitution value is available to the parafile interpreter by the time it needs to use it in the rest of the parafile.

In addition, if a file called defaults.pnm exists in the run directory, it may list alternative defaults that override those in the parafile, such as:

```
$DEFAULTS
[nodes]=2
```

The defaults.pnm file is expected to have only entries for $DEFAULTS, and no other parafile records. The order of override is:
Command line on nmfe75 script overrides
defaults.pnm, which overrides
defaults defined in parafile.

The advantage to this ordering is that a generic parafile file can be created for most environments. A user may then override defaults specified in this generic parafile with his own in defaults.pnm, that may be more suitable to his environment. Finally, a user can temporarily override his own defaults by giving an alternative value as an nmfe75 script command option. For example, the *8.pnm files listed in the NONMEM ..\run directory serve as generic parafiles that can be run for up to 8 nodes on a multi-core single computer system. Also in the NONMEM ..\run directory there is an example defaults.pnm file that has [nodes]=2 defined as a default. If this file were placed in the user's run directory, and the user used fpiliwini8.pnm as a parafile:
nmfe75 mycontrol.ctl mresults.res -parafile=fpiwini8.pnm

then the number of nodes would be that given in defaults.pnm, nodes=2. The user may override this by specifying an alternative number of nodes on the command line:

nmfe75 mycontrol.ctl mresults.res -parafile=fpiwini8.pnm [nodes]=4

in which case the first 4 nodes (or node numbers 1, 2, 3, 4) listed in $COMMANDS and $DIRECTORIES would be executed.

To also make distinct commands easy to write when launching many processes, number list substitution can also be performed. For example,

```
$GENERAL
NODES=8 PARSE_TYPE=4 PARSE_NUM=200 TIMEOUTI=600 TIMEOUT=1000 PARAPRINT=0
TRANSFER_TYPE=1

$NAMES ;Give a name to each node, which is displayed
1:MANAGER
2-8:WORKER{10-17}
```

```
$COMMANDS ;each node gets a command line, used to launch the node session
; %cd% refers to current directory
; Beyond the first position, a ; will not be interpreted as a comment for
; commands
1:mpiexec -wdir "%cd%"  -hosts 1 localhost 1 nonmem.exe %*
2-8:-wdir "%cd%\wk{#-1}"  -hosts 1 localhost 1 nonmem.exe

$DIRECTORIES
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY
2-8:wk{#-1} ; NEXT SET ARE THE WORKER directories
```

In the above example, the name of processes 2 through 8 are given as:

```
2-8:WORKER{10-16}
```

In this case, each number represented in the list within the braces {} is expanded and matched with the process number, so this line is equivalent to:

```
2:WORKER10
3:WORKER11
4:WORKER12
5:WORKER13
6:WORKER14
7:WORKER15
8:WORKER16
```

Make sure that the number of items represented in the number list in the braces is at least as many as the number list before the colon. Another example:

```
2,4,7:WORKER{1-3}
```

Expands to

```
2:WORKER1
4:WORKER2
7:WORKER3
```

Another method is to use the expression {#offset}, which directly substitutes the process number listed before the colon into the place at the braces, with an offset added to it. So,

```
2-8:-wdir "%cd%\wk{#-1}"  -hosts 1 localhost 1 nonmem.exe
```

Expands to

```
2:-wdir "%cd%\wk1"  -hosts 1 localhost 1 nonmem.exe
3:-wdir "%cd%\wk2"  -hosts 1 localhost 1 nonmem.exe
4:-wdir "%cd%\wk3"  -hosts 1 localhost 1 nonmem.exe
5:-wdir "%cd%\wk4"  -hosts 1 localhost 1 nonmem.exe
6:-wdir "%cd%\wk5"  -hosts 1 localhost 1 nonmem.exe
7:-wdir "%cd%\wk6"  -hosts 1 localhost 1 nonmem.exe
8:-wdir "%cd%\wk7"  -hosts 1 localhost 1 nonmem.exe
```

Similarly,

```
2,4,7:-wdir "%cd%\wk{#+11}"  -hosts 1 localhost 1 nonmem.exe
```

Expands to:

```
2:-wdir "%cd%\wk13"  -hosts 1 localhost 1 nonmem.exe
4:-wdir "%cd%\wk15"  -hosts 1 localhost 1 nonmem.exe
```

```
7:-wdir "%cd%\wk18"  -hosts 1 localhost 1 nonmem.exe
```

Easy to Use Parafiles

For easy use, there are a series of pnm files in the ..\run directory that can take any number of cores on a single computer. These are fpiwini8.pnm, mpiwini8.pnm, fpilinux8.pnm, and mpilinux8.pnm (for MAC OSX, use the *linux8.pnm files), located in the NONMEM ..\run directory. The 8 refers to the default number of nodes (processes) being 8, if it is not specified on the command line, or in a defaults.pnm file. An example of its use is as follows:

Nmfe75 foce_parallel.ctl foce_parallel.res -parafile=mpiwini8.pnm [nodes]=4

The example control stream file foce_parallel.ctl is in the ..\examples directory.

WINDOWS

Setting up a network drive on Windows for multiple Computers:

Both FPI and MPI methods require the user to set up network drives to pass files between manager and worker computers. If you are running your multiple process on multiple cores of just a single computer, then you may skip this section.

From the worker computer, select a directory (or create a directory) which you would like to have shared with the manager computer. Suppose it is called c:\share. On windows XP, open "my computers", or right click on Start ->Explore, go to directory tree, right click on c:\share, select properties, then select Sharing, and click on share this folder. On other Windows systems, there may be a different menu path to follow. A suggested share name will be given. You may keep this as is, or change to a name you prefer. Click on Permissions, for user Everyone select Full control, click on apply. Consult your IT representative if you are not able to obtain privileges.

From the manager computer, right click on the my computer icon and select map network drive. Select an available drive letter, which for this example will be w. Then enter \\, the computer name of the remote computer, or its IP address. This is followed by a \ and a share name of an accessible directory. For this example, the computer name is any_computer, and the share name of the directory is share, so enter

\\any_computer\share

Thus, from the manager side, drive w: will be associated with \\any_computer\share, which is in fact c:\share as seen by the worker computer. You may be asked to enter username and password.

Setting up FPI on Windows:

A versatile loading program called psexec.exe (freeware, from www.sysinternals.com), supplied with the NONMEM installation in the ..\run directory, can be used, that allows one to load processes locally or on other computers. You may choose alternative loading programs. Copy

psexec.exe from the NONMEM's ..\run directory to your managers run directory. From a DOS console window, type

Psexec

to see the parameters options for this launching program.

To test that your manager computer can load the NONMEM program on the worker computer (if different from manager), copy a computername.exe from NONMEM's .\run directory (we shall assume it is named NONMEM7.2.0) to the network mapped directory that is local to the worker.

Copy \nonmem7.2.0\run\computername.exe w:\share

Then type from the manager console window:

Psexec \\any_computer c:\share\computername.exe

(remember, these are just example names of computers and network share directories. Your particular environment will be different). The computer name of the worker computer should be displayed. You may be required to enter a user name and password. If this is the case, you should make sure that your user account and password on your manager computer is the same as on the worker computer, so that user name and password is not requested. Otherwise, when you run the NONMEM program, the run will be continually interrupted for this information.

During the parallelization process, NONMEM sends a copy of its program (nonmem.exe on Windows, nonmem on Linux) to the worker processes's directory, and then loads it there. Therefore, the worker computers must typically be of the same operating system (although not necessarily same version) as the manager computer (but see below to get around this). The worker computer does not have to have Intel or gfortran installed.

For a quick test on a single multi-core computer, try the following. Copy foce_parallel.ctl and example1.csv from the NONMEM ..\examples directory, fpiwini8.pnm from the NONMEM ..\run directory, and psexec.exe from the NONMEM ..\run directory, into your standard run directory. Then, execute the following from your standard run directory:

Nmfe75 foce_parallel.ctl foce_parallel.res -parafile=fpiwini8.pnm [nodes]=4

where the values of [nodes] should be no greater than the number of cores available on your computer.

A parafile example set up for FPI method on Windows is as follows (set TRANSFER_TYPE=0):

```
$GENERAL
NODES=2 PARSE_TYPE=3 PARSE_NUM=200 TIMEOUTI=60 TIMEOUT=10 PARAPRINT=0
TRANSFER_TYPE=0
; NODES=number of nodes (that is process, whether cores or computers)
```

```
; SINGLE node: NODES=1
; MULTI node (node means process, whether cores or computers): NODES>1
; WORKER node: NODES=0
;
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading time.
; This setting of parse_type will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU mode.
;
; timeouti=seconds to wait for node to start.  if not started in time,
; deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. As of nm75, only a warning
; is given if node takes longer than timeout minutes to return
; paraprint=1  print to console the parallel computing process.  Can be
; modified at run-time with ctrl-B toggle.
; Regardless of paraprint setting, <control_stream>.log always records
; parallelization progress.
;
; transfer_type=0 for file transfer, unloading and reloading workers with
; each estimation
; transfer_type=1 for mpi
; transfer type=2 for file transfer, maintaining a single loaded process
; throughout the run.

;THE EXCLUDE/INCLUDE may be used to selectively use certain nodes,
; out of a large list.
; $EXCLUDE 5-7 ; exclude nodes 5-7
; or
;$EXCLUDE ALL
;$INCLUDE 1,4-6

$NAMES ; Give a label to each node for convenience
1:MANAGER
2:WORKER1
3:WORKER2
4:WORKER3

$COMMANDS ;each node gets a command line, used to launch the node session.
; Command lines must be on one line for each process.  The following commands
; are for FPI method on Windows.

; First node is manager, so it does not get a command line when using FPI
1:NONE
;
; load on a core of the same computer as manager: Note that worker does not
; really need a control stream file, but something must be there as a place
; holder.  Also, for psexec, notice that the worker directories are named
; as the worker sees them, not as the manager sees them.  Very important
; distinction for remote worker computers.
; -wdir refers to working directory for particular process
; do not user %cd% with psexec.  Just user relative directory notation
2:psexec -d -w worker1 cmd.exe /C nonmem.exe
; load on a core of the same computer as manager:
```

```
3:psexec -d -w worker2\ cmd.exe /C nonmem.exe
; load on a core of a different computer than manager:
4:psexec \\any computer -d -w c:\share\worker3 cmd.exe /C nonmem.exe

$DIRECTORIES ; Names of directories as a manager sees them.
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY. Make it NONE if no
; common directory is to be used. This is the best option.
2:worker1\ ; NEXT SET ARE THE WORKER directories.
3:worker2\
4:w:\share\worker3\ ; This directory is on a different computer from manager

$IDRANGES ; USED IF PARSE_TYPE=3
1:1,50
2:51,100
```

After an estimation step is performed, the worker processes exit. For the next estimation step that follows (if there is one), the manager will reload the worker processes.

For the FPI method with TRANSFER_TYPE=0, a PARAFILE file name may be given specific to a $EST command:

$EST METHOD=IMP INTERACTION NITER=20 PARAFILE=myparallel_imp.pnm
$EST METHOD=1 INTERACTION PARAFILE=myparallel_foce.pnm

If no parallel file is given for an estimation method, it takes the PARAFILE name of the previous $EST command. If no PARAFILE option was given for the first $EST method, then it takes the value given in the command line switch –parafile. If no –parafile switch was given, then the default name parallel.pnm is assumed. If parallel.pnm file does not exist, then NONMEM runs on a single CPU.

If you want worker processes to remain resident until all estimations and problems listed in the control stream file are completed, then select TRANSFER_TYPE=2. In these cases, new PARAFILE settings at $EST steps within the control stream file will be ignored, except for PARAFILE=ON or PARAFILE=OFF.

Installing MPI on Windows

Go to the web site
https://www.mpich.org/downloads/

and select the suitable Windows version, with extension .msi. Or, select the mpich2-1.2.1p1-win-ia32.msi file listed in https://nonmem.iconplc.com/mpich2/ . Install the full version on the manager computer by double clicking on the .msi file, or running it from START->run. Follow the instructions in section 7 of mpich2-1.2.1-windevguide.pdf, and verify that the MPI system is working. Copy the program mpiexec.exe from the bin directory of the MPICH2 directory, to your manager NONMEM run directory.

NONMEM comes with the MPI library files (they are located in ..\mpi\MPI_WINI for Intel Fortran and ..\mpi\MPI_WING for gfortran). For communication across computers, make sure you also have a network file allocated, as described above. If the MPI library files do not match

the version which you downloaded, or there are linking difficulties when you run nmfe75.bat, then copy the appropriate .lib file from the MPICH2 installed directory mpich2\lib to ..\mpi\MPI_WINI directory. Keep in mind that we have supplied 32 bit versions of libraries. Environments with 64 bit processing may require libraries from the mpich2 web site.

The MPI Windows installation guide (section 9) may offer other ways to supply user name and password via the program mpiexec. For example, from the manager computer

mpiexec –register
Enter name
Enter password.

During the parallelization process, NONMEM sends a copy of its program (in nonmem.exe on Windows, nonmem on Linux) to the worker computer, and then loads it there. Therefore, generally, the worker computers must be of the same operating system (although not necessarily same version) as the manager computer. For Intel fortran or gfortran, the worker computer does not have to have the compiler installed.

In addition, the MPI system needs certain executable files available on the worker computer. A minimal installation on the worker computer can be implemented by copying smpd.exe (found in the bin directory of you manager's MPICH2 directory) to the worker computer, and executing
Smpd.exe –install

See section 9 of the MPI Windows installation guide about the full use of smpd.exe.

Also, the MPI system needs certain dll library files placed in each worker processor's directory of the worker computer, or in the windows\system32 directory (more generally, in %systemroot%\system32):
Fmpich2.dll (intel) or fmpich2g.dll (gfortran)
Mpich2.dll
Mpich2mpi.dll

The dll files are located in the manager's %systemroot%\system32 directory.

Next, make sure the library file in the NONMEM system is the same as that of your MPICH2 system. For example, for Intel fortran:

cd \nm750\mpi\mpi_wini
cp fmpich2.lib fmpich2_orig.lib
cp "\program files\MPICH2\lib\fmpich2.lib"

For gfortran:
cd \nm750\mpi\mpi_wing
cp libfmpich2g.a libfmpich2g_orig.a
cp "\program files\MPICH2\lib\libfmpich2g.a"

In addition, the file PNM_MPI.f90 may need to be newly compiled with the appropriate mpi.mod file present. Sometimes there will be an appropriate gfortran or intel fortran based mpi.mod file in the appropriate ..\lib directory of the MPI installation, and you just copy it over:

gfortran:
cd \nm75g65\mpi_mpi_wing\
copy "c:\program files (x86)\MPICH2\lib\mpi.mod"
copy mpi.mod ..\..\resource

intel fortran:
cd \nm75g64\mpi_mpi_wini\
copy "c:\program files (x86)\MPICH2\lib\mpi.mod"
copy mpi.mod ..\..\resource

Or you may have to build it from a file called mpi.h or mpi.f, probably located in the ..\include folder of the MPI installation.

gfortran:
gfortran -c -fno-range-check mpi.f90
copy mpi.mod ..\..\resource
gfortran -c -O3 -ffast-math -m64 -mpc64 -I../../resource -J../../resource PNM_MPI.f90
(note you may want to use whatever compile options you used for the main NONMEM installation).

Intel fortran:
ifort /c mpi.f90
copy mpi.mod ..\..\resource
ifort /c PNM_MPI.f90 /I:..\..\resource
(note you may want to use whatever compile options you used for the main NONMEM installation).

Other MPI software are available, such as OpenMpi, and Microsoft MPI for Windows environment. In fact, MPICH may no longer work for Windows 10 and above. Instructions and materials for Microsoft MPI installation are located at https://nonmem.iconplc.com/msmpi

As an added precaution that your NONMEM installation will use the correct MPI setup (and compiler), you may set up a bash script, called nmloc, in which you establish the following environment variables, and place nmloc.bat in the top NONMEM installation directory. See the end of section I.8 Invoking NONMEM regarding the nmloc system.

Once you have an MPI system set up, for a quick test on a single multi-core computer, try the following. Copy foce_parallel.ctl and example1.csv from the NONMEM ..\examples directory, mpiwini8.pnm from the NONMEM ..\run directory, and mpiexec.exe from the NONMEM ..\run directory, into your standard run directory. Then, execute the following from your standard run directory:

Nmfe75 foce_parallel.ctl foce_parallel.res -parafile=mpiwini8.pnm [nodes]=4

where the values of [nodes] should be no greater than the number of cores available on your computer.

For instructional purposes, a typical structure of a PARAFILE is listed below that would be used for NONMEM on Windows using MPI (note the setting of TRANSFER_TYPE=1):

```
$GENERAL
NODES=2 PARSE_TYPE=3 PARSE_NUM=200 TIMEOUTI=60 TIMEOUT=10 PARAPRINT=0
TRANSFER_TYPE=1 COMPUTERS=2

; NODES=number of nodes (that is process, whether cores or computers)
; SINGLE node: NODES=1
; MULTI node (node means process, whether cores or computers): NODES>1
; WORKER node: NODES=0
;
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading time.
; This setting of parse_type will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU mode.
;
; timeouti=seconds to wait for node to start.  if not started in time,
; deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. As of nm75, only a warning
; is given if node takes longer than timeout minutes to return
; paraprint=1  print to console the parallel computing process.  Can be
; modified at run-time with ctrl-B toggle.
; Regardless of paraprint setting, <control_stream>.log always records
; parallelization progress.
;
; transfer_type=0 for file transfer, unloading and reloading workers with
; each estimation
; transfer_type=1 for mpi
; transfer type=2 for file transfer, maintaining a single loaded process
; throughout the run.

;THE EXCLUDE/INCLUDE may be used to selectively use certain nodes,
; out of a large list.
 $EXCLUDE 5-7 ; exclude nodes 5-7
; or
;$EXCLUDE ALL
;$INCLUDE 1,4-6

$NAMES ; Give a name to each node, which is displayed
1:MANAGER
2:WORKER1
3:WORKER2

$COMMANDS ;each node gets a command line, used to launch the node session
```

```
; The first one launches the manager's NONMEM.
; -wdir refers to working directory for particular process
; %* mean to transfer all options from command line to
; manager process's nonmem.exe
1:mpiexec -wdir "%cd%"   -hosts 1 localhost 1 -noprompt nonmem.exe %*
; the next one launches a worker process on the manager's computer
; the worker only needs certain of the parameters from the command line.
2:-wdir "%cd%"\worker1  -hosts 1 localhost 1 -noprompt nonmem.exe
;
; This launches a worker process on a separate computer.
3:-wdir c:\share\worker3  -n 1 -host any_worker -noprompt
 (continued on same line)
c:\share\worker3\nonmem.exe

$DIRECTORIES
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY
2:worker1\ ; NEXT SET ARE THE WORKER directories
3:w:\share\worker3\

$IDRANGES ; USED IF PARSE_TYPE=3
1:1,50
2:51,100
```

By default, at the beginning of each step (estimation, covariance, wres diagnostics, etc.), the individuals are evenly distributed among the nodes. As of nm742, an option called PARSE_PRESERVE may be set to a non-zero value to have the most edfficient ID range settings that were determined from the previous step be used for the first iteration after the next step. For example, if the previous step was estimation, then the ID distributions among the nodes of the last iteration of estimation will be used as the start of covariance assessment. The supposition is that the load distribution that was assessed in estimation should be suitable for covariance assessment. Set the bit for the appropriate previous step:
If Previous estimation was estimation, use its last ID load distribution: set bit 0
If Previous estimation was chain method, use its last ID load distribution: set bit 1
If Previous estimation was covariance, use its last ID load distribution: set bit 2
If Previous estimation was wres assessment, use its last ID load distribution: set bit 3
If Previous estimation was final eta assessment, use its last ID load distribution: set bit 4
If Previous estimation was simulation, use its last ID load distribution: set bit 5
If Previous estimation was nonparametric, use its last ID load distribution: set bit 6

The most sensible is to set bits 0, 2, 3, and 4, as these algorithms tend to provide the most accurate load distribution assessments, suitable for launching the next step with the same initial load distribution (PARSE_PRESERVE=29). After several iterations, each step empirically adjusts its load distribution anyway, regardless of PARSE_PRESERVE setting.

An additional setting in $GENERAL is introduced, called COMPUTERS. By default COMPUTERS is equal to 1. However, if you are running MPI method on Windows, and you have at least one of the worker processes on another computer, and your LIM values are not maximized, so that some file buffers are being used, then you may need to set COMPUTERS=2. If you obtain a read/write error on FILE10, or other FILEXX error, then set COMPUTERS=2.

Unlike FPI, the MPI system can only use the starting parallel.pnm file specified at the command line, and it may not be easily switched later in the control stream. All processes remain resident throughout the entire job, although it will honor requests of parafile=off or parafile=on at individual $EST records, which allows you to have control of which estimation method will use parallel processing.

In the FPI method, the manager NONMEM process has total control of loading followed by implementing all the workers, and is in fact loaded before the pnm file is interpreted and acted upon. With MPI, the mpi system has control, and the manager NONMEM program is just the first of a set of processes. The mpi system is first loaded using a DOS batch file called nmmpi.bat (constructed by the nmfe75 script by a call to nonmem_mpi), and with commands constructed from the $COMMANDS entries in the pnm file. The mpi program loads all the processes, including the manager. Therefore the manager's $COMMANDS entry has to have all of the parameters passed to it that was entered at the nmfe75 command line by the user, as shown in the example above, by using %*.

For the Windows version of MPI, sometimes you have to specify the full file path of the nonmem.exe program when launching on a remote computer.

LINUX

Setting up share directory, and ssh on a Linux System

The ssh system and share directory used to pass files between worker and manager must be set up for FPI and MPI methods, if the worker computer differs from the manager computer. The following instructions serve only as a guide as to how to set up the ssh system. You may need to vary some of the commands to suit your environment. Consult your Linux user manual as well.

The network files system (NFS) is used for the manager computer to access a network drive that points to a worker computer's local drive. Consider the following example.

From the worker computer, create a share directory, such as:
mkdir /home/myself/share

Next, use your editor, and sudo privilege, to modify the /etc/hosts file,

sudo gedit /etc/hosts

And map IP address to computer names:

127.0.0.1 localhost
192.168.1.3 my_manager
192.168.1.2 any_computer

Then save and exit. Use your editor to edit /etc/exports:

sudo gedit /etc/exports

Add the following line:
/home/myself/share 192.168.1.0/24(rw,sync)

Which allows IP addresses 192.168.1.0 through 192.168.1.255 to access this share directory.

Then exit.

sudo exportfs –a

Stop and restart NFS system (this is for Ubuntu: the command may differ on your computer)
sudo /etc/init.d/nfs-kernel-server Stop

sudo /etc/init.d/nfs-kernel-server restart

Go to the manager computer, and also place computer names to IP address mapping in /etc/hosts:

127.0.0.1 localhost
192.168.1.3 my_manager
192.168.1.2 any_computer

Then, create a mount drive for the remote directory:
mkdir /mnt/share

sudo gedit /etc/fstab

Enter the mount drive entry for the remote directory:

any_computer:/home/myself/share /mnt/share nfs rw,sync 0   0

and exit the editor. Then,

sudo mount /mnt/share

Test by copying a file from the manager to the worker:
cp myfile /mnt/share

Next, the ssh component must be set up.

Check that you have ssh installed on both manager and worker computers:

From the manager, run the standard Linux date program on the worker computer:

ssh –n any_computer date
enter password

If the date is returned from the worker computer, you have ssh connection. You might have to enter user account name:

ssh –n my_account@any_computer date

For ssh to work in parallel computing, you need to set up ssh so it does not always ask for your password. From the manager computer:

ssh-keygen –t dsa

Respond yes to writing to ~/.ssh, and enter in a passphrase.

Copy id_dsa.pub from the manager to the worker computer (possibly via the share drive you had set up):

cp ~./ssh/id_dsa.pub /mnt/share

Then concatenate this manager created id_dsa.pub to the authorized_keys file on the worker computer:

```
cd $HOME
chmod +w .ssh/authorized_keys
touch .ssh/authorized_keys
cat id_dsa.pub >> .ssh/authorized_keys
chmod 400 .ssh/authorized_keys
```

From the manager computer, repeat the command

ssh –n any_computer date

it should ask you for the pass-phrase, then give you the date.

Do it again:
ssh –n any_computer date

the pass phrase should not be requested this time, nor should a password be requested, and a date from the worker computer should return.

During the parallelization process, NONMEM sends a copy of its program to the worker computer, and then loads it there. Therefore, the worker computers must be of the same operating system (although not necessarily same version) as the manager computer. For Intel fortran, the worker computer does not have to have Intel Fortran installed. For gfortran, –static option for the FPI is used in the nmfe75 script, which makes gfortran portable to the worker computer without requiring the gfortran share library (libgfortran.so.3). If for some reason you needed to remove the –static option, then gfortran requires its share library available for the

worker process, and in the path designated by the manager's LD_LIBRARY_PATH setting, such as:

LD_LIBRARY_PATH="$HOME/gcc-trunk/lib:$HOME/libgf:$LD_LIBRARY_PATH"
Export LD_LIBRARY_PATH

where $HOME/gcc-trunk/lib is the library path for the manager's gfortran, and $HOME/libgf is the path on the worker computer containing at least the file libgfortran.so.3. You may place these lines in the .bashrc file. Therefore, if upon loading NONMEM on the worker computer, a message is displayed indicating that certain share files are missing, etc., then you may need to either install gfortran, or selectively make the share file available.

Setting up FPI on Linux

For a quick test on a single multi-core computer, try the following. Copy foce_parallel.ctl and example1.csv from the NONMEM ..\examples directory, fpilinux8.pnm from the NONMEM ..\run directory, and beolaunch.sh from the NONMEM ..\run directory, into your standard run directory. Then, execute the following from your standard run directory:

Nmfe75 foce_parallel.ctl foce_parallel.res -parafile=fpilinux8.pnm [nodes]=4

where the values of [nodes] should be no greater than the number of cores available on your computer.

For instructional purposes, here is an example pnm file for FPI on Linux systems (note TRANSFER_TYPE=0):

```
$GENERAL
NODES=3 PARSE_TYPE=2 PARSE_NUM=50 TIMEOUTI=300 TIMEOUT=20 PARAPRINT=0
TRANSFER_TYPE=0

; NODES=number of nodes (that is process, whether cores or computers)
; SINGLE node: NODES=1
; MULTI node (node means process, whether cores or computers): NODES>1
; WORKER node: NODES=0
;
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading time.
; This setting of parse_type will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU mode.
;
; timeouti=seconds to wait for node to start.  if not started in time,
; deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. As of nm75, only a warning
; is given if node takes longer than timeout minutes to return
; paraprint=1  print to console the parallel computing process.  Can be
; modified at run-time with ctrl-B toggle.
; Regardless of paraprint setting, <control_stream>.log always records
```

```
; parallelization progress.
;
; transfer_type=0 for file transfer, unloading and reloading workers with
; each estimation
; transfer_type=1 for mpi
; transfer type=2 for file transfer, maintaining a single loaded process
; throughout the run.

;THE EXCLUDE/INCLUDE may be used to selectively use certain nodes,
; out of a large list.
 $EXCLUDE 5-7 ; exclude nodes 5-7
; or
;$EXCLUDE ALL
;$INCLUDE 1,4-6

$NAMES ; Give a label to each node for convenience
1:MANAGER
2:WORKER1
3:WORKER2

$COMMANDS ;each node gets a command line, used to launch the node session
; Command lines must be on one line for each process.
; command not needed for node 1, manager
1:NONE
;
; following is a launch on a core of the manager computer. Beolaunch.sh is a
; simple script available from the NONMEM ../run directory
2:./beolaunch.sh wrk_ftif/ ./nonmem >worker1.out
;
; following is a launch on a remote worker computer
3:ssh -n any_computer cd /home/myself/share/worker1';'./nonmem >worker1.out &

$DIRECTORIES
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY
2:wrk_ftif/ ; NEXT SET ARE THE WORKER directories.
3:/mnt/share/worker1/

$CONTROL
;MTOUCH=1 for manager to "touch" the worker directory to get
; up-to-date information
;WTOUCH=1 for worker to "touch" its directory;
;MSLEEP=milliseconds for manager to wait between writing its content files
; to the remote worker directory
;WSLEEP=milliseconds for worker to wait between writing its content files
; to the worker directory
3: MTOUCH=1 WSLEEP=5 WTOUCH=0 MSLEEP=0

$IDRANGES ; USED IF PARSE_TYPE=3
1:1,50
2:51,100
```

There is an additional record introduced here, called $CONTROL. When working between computers on Linux with FPI, some network file systems (such as NFS on Unix) may require that the manager 'touch' the remote worker directory for that directory to show the up-to-date file information to the manager. Also, the process may need a period of waiting time before the signal file is created. Hence the need for the $CONTROL statements.

After an estimation step is performed, the worker processes exit. For the next estimation step that follows (if there is one), the manager will reload the worker processes. If you want worker processes to remain resident until all estimations and problems listed in the control stream file are completed, then select TRANSFER_TYPE=2.

Running Parallel Processes in a Mixed Platform Environment.

Suppose the manager process may be a new Linux operating system with a GLIBC that is new, while a worker computer may be Linux with an older operating system with an old GLIBC. This typically is not an easy environment to set up, but if you wish to do so, it means that you would need to create the nonmem executable on the Linux machine ahead of time, name it nonmem2, or some other name, so it is not copied over with the nonmem executable of the manager process, and use that nonmem2 on the worker $COMMANDS line:

```
2:./beolaunch.sh wrk_ftif/ ./nonmem2 >worker1.out
```

One would do something similar if the manager were a Windows process, and the worker were a Linux process, for example, but it is up to the user to find a means of launching a remote Linux process. The psexec launcher only works between Windows computers.

Installing MPI on Linux

If you are communicating across computers, make sure you set up a share drive and the ssh system as described earlier. Go to the web site
http://phase.hpcc.jp/mirrors/mpi/mpich2/

and select the appropriate *.tar.gz file. Or, select the mpich2_1.2.1.1.orig.tar.gz file in the MPI directory given in the NONMEM installation disk. On the manager computer, unpack the tar.gz file:

tar xfz mpich2_1.2.1.orig.tar.gz

Follow the instructions in section 2.2 of mpich2-1.2.1-installguide.pdf, and verify that the MPI system is working. NONMEM comes with the MPI library files (they are located in ..\mpi\mpi_lini for Intel Fortran and ..\mpi\mpi_ling for gfortran). For communication across computers, make sure you also have a network file allocated, just as with the FPI method. If the MPI library files do not match the version which you downloaded, or there are linking difficulties when you run nmfe75, then copy the appropriate *.a file from the MPICH2 installed directory mpich2\lib to the ..\mpi\mpi_lini directory. Keep in mind that we have supplied 32 bit versions of libraries. Environments with 64 bit processing may require libraries from the mpich2 web site.

For easy access of the mpi utility programs, you should expand the $PATH to include the path to the bin directory of the MPICH2 system, if it is not there already. You can insert the following line in the manager's $HOME/.bashrc file, for example:

export PATH=$HOME/MPICH2_LINUX/mpich2-install/bin:$PATH

During the parallelization process, NONMEM sends a copy of its program (in nonmem.exe on Windows, nonmem on Linux) to the worker computer, and then loads it there. Therefore, the worker computers must be of the same operating system (although not necessarily same version) as the manager computer. For Intel fortran, the worker computer does not have to have Intel Fortran installed. For gfortran, –static option for the MPI method cannot be used in the nmfe75 script, as it prevents the MPI components from being properly linked. Thus the gfortran version of NONMEM with MPI requires its share library (libgfortran.so.3) available for the worker process, and in the path designated by the manager's LD_LIBRARY_PATH setting:

```
LD_LIBRARY_PATH="$HOME/gcc-trunk/lib:$HOME/libgf:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH
```

where $HOME/gcc-trunk/lib is the library path for the manager's gfortran, and $HOME/libgf is the path on the worker computer containing at least the file libgfortran.so.3. You may place these lines in the .bashrc file. Therefore, if upon loading NONMEM on the worker computer, a message is displayed indicating that certain share files are missing, etc., then you may need to either install gfortran, or selectively make the share file available.

In addition, the MPI system needs certain executable files available on the worker computer. These are (obtained from the bin directory of the MPICH2 system):
mpdlib.py
mpdman.py
mpd.py

Place these files in a directory on the worker computer that has the same path as MPICH2 is installed in the manager's computer. For example, if the manager's MPICH2 bin path is $HOME/MPICH2_LINUX/mpich2-install/bin, then this should be where the worker computer's *.py files are.

Upon booting up, before executing your first NONMEM run, load up the mpi system:

```
mpdboot –n <number_of_computers> -f mpd.hosts
```

as instructed in the install guide. The mpd.hosts file contains a list of IP addresses, one per line, of the worker and manager computers. They could be referenced symbolically in the mpd.hosts, for example, as:

```
MY_MANAGER_COMPUTER
WORKER_A_COMPUTER
WORKER_B_COMPUTER
```

So long as these symbolic names are listed in the /etc/hosts file with the IP address.

The number_of_computers is number of worker computers (not cores), plus the manager computer. If loading just on one computer, then

mpdboot –n 1

To unload MPI after your last NONMEM run,

mpdallexit

See section 5 of mpich2-1.2.1-userguide.pdf for a full description of using the man MPI program mpiexec or mpirun.

Next, replace libmpich.a, in the NONMEM 75 directory, with the one from the located in the MPICH2 directory, for example, for gfortran:
cd /opt/nm75/mpi/mpi_wing
cp libmpich.a libmpich.a.orig
cp /usr/local/mpi32/lib/libmpich.a libmpich.a

or for Intel fortran:
cd /opt/nm74/mpi/mpi_lini
cp libmpich.a libmpich.a.orig
cp /usr/local/mpi32/lib/libmpich.a libmpich.a

In addition, the file PNM_MPI.f90 may need to be newly compiled with the appropriate mpi.mod file present. Sometimes there will be an appropriate gfortran or intel fortran based mpi.mod file in the appropriate ..\lib directory of the MPI installation, and you just copy it over:

gfortran:
cd /nm75g64/mpi_mpi_ling/
cp /usr/bin/MPICH2/lib/mpi.mod .
cp mpi.mod ../../resource

intel fortran:
cd /nm75g64/mpi_mpi_lini/
cp /usr/bin/MPICH2/lib/mpi.mod .
cp mpi.mod ../../resource

Or you may have to build it from a file called mpi.h or mpi.f, probably located in the ..\include folder of the MPI installation.

gfortran:
gfortran -c -fno-range-check mpi.f90
cp mpi.mod ../../resource
gfortran -c -O3 -ffast-math -m64 -mpc64 -I../../resource -J../../resource PNM_MPI.f90
(note you may want to use whatever compile options you used for the main NONMEM installation).

Intel fortran:
ifort -c mpi.f90

cp mpi.mod ../../resource
ifort -c PNM_MPI.f90 /I:../../resource
(note you may want to use whatever compile options you used for the main NONMEM installation).

As an added precaution that your NONMEM installation will use the correct MPI setup (and compiler), you may set up a bash script, called nmloc, in which you establish the following environment variables, and place nmloc.bat in the top NONMEM installation directory. See the end of section I.8 Invoking NONMEM regarding the nmloc system.

Once you have an MPI system set up, for a quick test on a single multi-core computer, try the following. Copy foce_parallel.ctl and example1b.csv from the NONMEM ..\examples directory, mpilinux8.pnm from the NONMEM ..\run directory, and psexec.exe from the NONMEM ..\run directory, into your standard run directory. Then, execute the following from your standard run directory:

./mfe75 foce_parallel.ctl foce_parallel.res -parafile=mpilinux8.pnm [nodes]=4

where the values of [nodes] should be no greater than the number of cores available on your computer.

A typical structure of a pnm file for running NONMEM/MPI/Linux (note TRANSFER_TYPE=1) is as folllows:

```
$GENERAL
NODES=2   PARSE_TYPE=2   PARSE_NUM=50   TIMEOUTI=100   TIMEOUT=10   PARAPRINT=0
TRANSFER_TYPE=1

; NODES=number of nodes (that is process, whether cores or computers)
; SINGLE node: NODES=1
; MULTI node (node means process, whether cores or computers): NODES>1
; WORKER node: NODES=0
;
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading time.
; This setting of parse_type will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU mode.
;
; timeouti=seconds to wait for node to start.  if not started in time,
; deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. As of nm75, only a warning
; is given if node takes longer than timeout minutes to return
; paraprint=1  print to console the parallel computing process.  Can be
; modified at run-time with ctrl-B toggle.
; Regardless of paraprint setting, <control_stream>.log always records
; parallelization progress.
;
```

```
; transfer_type=0 for file transfer, unloading and reloading workers with
; each estimation
; transfer_type=1 for mpi
; transfer type=2 for file transfer, maintaining a single loaded process
; throughout the run.

;THE EXCLUDE/INCLUDE may be used to selectively use certain nodes,
; out of a large list.
 $EXCLUDE 5-7 ; exclude nodes 5-7
; or
;$EXCLUDE ALL
;$INCLUDE 1,4-6

$NAMES ; Give a name to each node, which is displayed
1:MANAGER
2:WORKER1
3:WORKER2

$COMMANDS ;each node gets a command line, used to launch the node session
; first one launches manager version
1:mpirun "$PWD" -n 1 ./nonmem  $*
;
; This launches a worker process on the manager's computer
2:-wdir "$PWD"/nonmem/wrk_mpi -n 1 ./nonmem
; This launches a worker process on a separate computer
;
3:-wdir /home/myself/share/worker1  -n 1 -host any_worker ./nonmem

$DIRECTORIES
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY
2:nonmem/wrk_mpi/ ; NEXT SET ARE THE WORKER directories
3:/mnt/share/worker1/

$IDRANGES ; USED IF PARSE_TYPE=3
1:1,50
2:51,100
```

You will want to modify the pnm file for your particular environment, and use some of the other options available in setting up the mpiexec/mpirun command line.

Unlike FPI, the MPI system can only use the starting PARFILE specified at the command line, and it may not be easily switched later in the control stream. All processes remain resident throughout the entire job, although it will honor requests of parafile=off or parafile=on individual $EST records, which allows you to have control of which estimation method will use parallel processing.

Earlier we show that the addresses to the worker computers listed in the file mpd.hosts could be loaded using the mpdboot –f command. The –f option is also available in mpirun, so this information may be supplied within the parafile, for example:

```
1:mpirun "$PWD" -n 1 0 –f mpd.hosts ./nonmem  $*
```

Some Advanced Technics For Defining the PARAFILE for an MPI System.

Because the MPI system communicates completely via ports, and not via file transfer as the FPI system does, one can set up a parafile in which an MPI command is repeated for several nodes, even though they may point to the same directory. Here is an example which makes creating a PARAFILE for an MPI system versatile:

```
$GENERAL
NODES=8 PARSE_TYPE=2 TRANSFER_TYPE=1 PARAPRINT=0 COMPUTERS=2

$COMMANDS
1:mpiexec -wdir "$PWD" -n 1 ./nonmem $*
2-4: -wdir "$PWD" -n 1 -host MY_MANAGER_COMPUTER ./nonmem -wnf
5-8: -wdir $HOME -n 1 -host MY_WORKER_COMPUTER ./nonmem  -wnf

$DIRECTORIES
1-8:NONE
5:/mnt/worker1
```

In this example, node 1 is defined as usual as the manager process. Then, processes 2 through 4 are defined using a command that is repeated for each of these processes (it is copied 3 times in the resulting nmmpi script file that is eventually executed). Yet processes 2-4 all point to the default current directory of the manager ("$PWD"). Furthermore, the $DIRECTORIES entries for these processes is NONE. That means the three worker processes which are loaded on the manager computer are sharing the same directory as the manager, and because of the NONE directory designation in $DIRECTORIES, the executable nonmem will not be copied, as it should not, since the worker processes are pointing to the manager directory, and therefore the nonmem executable in the manager directory is already available to worker processes as well. Furthermore, the option –wnf is given. This option tells the nonmem process that it is a worker, MPI method, and the nf tells it not to make any file buffers (nf=no files). The worker process has all the information it needs to launch without requiring any file based communication with the manager, and minimizes the footprint on the drive directory.

The next 4 processes are launched on a remote computer with similar settings. Notice that only one of the processes among the 5 to 8 had to have a $DIRECTORY defined, that of /mnt/worker1, which they all are pointing to. The $HOME directory of the worker computer is the directory /mnt/worker1 that the manager has a share connection to. This means that NONMEM has a path direction to copy the nonmem executable from its current directory to the $HOME directory on the worker computer. If all processes $DIRECTORIES entries were NONE, then the most recently built nonmem executable cannot be copied to the remote computer. You may want that, if for example, you have arranged for a nonmem executable to be there already that was previously built with the identical control stream file. Maybe the remote computer is a different platform than the manager computer, and needed a different executable. MPICH2 communication between a Linux and Windows operating system has not been attempted, so it is not known if this would work anyway.

Note that –host MY_MANAGER_COMPUTER had to be identified on the worker processes that were being launched locally. The mpiexec command gets confused if it has to deal with several lines containing different computer names. So it is best not to leave the –host switch up to default once you get past the manager processor line.

The –wnf switch must be carefully used. Make sure that LIM1, LIM3, LIM4, LIM13, and Lim15 are appropriately sized so that the buffer files (named FILEXX) do not have to be used. Or, as of NM73, you may set –maxlim=1 or higher on the nmfe75 command line. Then, LIM1, LIM3, LIM4, LIM13, and Lim15 (those used during estimation, and therefore by workers in a parallelization problem), will be set to the size needed to assure no buffer files are used, and everything is stored in memory, for the particular prolem. If you set –maxlim=2, then LIM1, LIM2, LIM3, LIM4, LIM5, LIM6, LIM7, LIM8, LIM10, LIM13, LIM15, and LIM16 are also sized to what is needed to assure that buffer files are not needed.

If the buffer files do need to be used, then use switch –wf. Each worker process will make a series of files named WK1_FILE* for worker 1, WK2_FILE* for worker 2, etc. This way, even if the workers and manager share the same directory as a scratch pad, their files will be uniquely named, and there won't be a file clobber.

An alternative method of launching mpi processes is to use its multiple process launch option –n xx, where xx is the number of processes to launch:

```
$GENERAL
NODES=8 PARSE_TYPE=2 TRANSFER_TYPE=1 PARAPRINT=0 COMPUTERS=2

$COMMANDS
1:mpiexec -wdir "$PWD" -n 1 ./nonmem $*
2: -wdir "$PWD" -n 3 -host MY_MANAGER_COMPUTER ./nonmem -wnf
3: -wdir $HOME -n 4 -host MY_WORKER_COMPUTER ./nonmem -wnf

$DIRECTORIES
1-8:NONE
3:/mnt/worker1
```

Command 2 launches 3 processes, and command 3 launches 4 processes, so there are still 8 processes launched.

As of nm75, any workers option switch that is prefixed with "a"(-aworker, -awnf etc), will be ignored by the first nonmem executable loaded by the MPI system (so it will be the manager), allowing the same arguments to be placed on all nodes. For example:

```
$DEFAULTS
[nodes]=8
GENERAL
NODES=[nodes] PARSE_TYPE=2 TRANSFER_TYPE=1 PARAPRINT=0 COMPUTERS=2

$COMMANDS
1:mpiexec -wdir "$PWD" -n [nodes] ./nonmem $* -awnf

$DIRECTORIES
1-[nodes]:NONE
```

Special Considerations for MAC OS X

The MAC OS X system uses a form of Linux as its operating system, but there are some differences in the environment compard to Linux on non-MAC systems.

Mounting file systems on MAC OS X

It is easier to use afp (Apple Filing Protocol) than nfs .

To export a file system or folder to another Mac:
Select the Apple menu / System Preferences / Sharing / File Sharing
Under “shared folders:” click + and select the folder e.g., mydir
Under “users:” click + and select the users.

To mount a file system or folder from another Mac:
Open a finder window.
You should see the hostname of the other computer listed under “Shared”
Click on it. Click on “connect as”
Enter the username and password.
Click on the folder, e.g., mydir
The file system or folder will be mounted as /Volumes/mydir

E.g., in a terminal window: % ls /Volumes/mydir

Enabling ssh with no password on MAC OS X

Select the Apple menu / System Preferences / Sharing / Remote Login
The instructions for Linux (using ssh-keygen) should work on Mac OS X.
There may be an interaction with keychain, and this may be problematic.

If “ssh –n” cannot be made to work, you can use the workaround for mpdboot described in the MPICH2 Installer’s Guide.
See ‘start the daemons “by hand”’ on page 7 of mpich2-1.2.1-installguide.pdf

Installing MPICH2 on MAC OS X

MPICH2 must be compiled and installed for Mac OS X.
Please look at mpich2/README_vin.mht and the other documents.

You may need to set CPATH as it was during the NONMEM install

First, see what kind of binaries have been installed, e.g.,
% cd /Users/Shared/nm750/mpi/mpi_ling (or mpi_lini, with ifort):
% file mpi.o

You will see either of the following:
mpi.o: Mach-O 64-bit object x86_64
mpi.o: Mach-O object i386
“i386” indicates 32 bit binaries.

Suggested options for the configure step with MacOS 10.15 Catalina:

If SETUP75 installed 64 bit binaries:
./configure --prefix=/usr/local/mpi64 CFLAGS="-m64" FFLAGS="-m64" --enable-f90

–disable-cxx --enable-timer-type=gettimeofday |& tee c.txt

If SETUP75 installed 32 bit binaries:
./configure --prefix=/usr/local/mpi32 --enable-f90
--enable-timer-type=gettimeofday |& tee c.txt

(With earlier releases than Catalina, option –enable-time-type may not be necessary)
(With ifort rather than gfortran, -disable-cxx may not be necessary)

Either way, continue with
make |& tee m.txt
make install |& tee mi.txt

Then replace libmpich.a, in the NONMEM 75 directory, e.g, if 64 bit was installed:
cd /Users/Shared/nm750/mpi/mpi_ling
cp libmpich.a libmpich.a.orig
cp /usr/local/mpi64/lib/libmpich.a libmpich.a

In addition, the file PNM_MPI.f90 may need to be newly compiled with the appropriate mpi.mod file present. Sometimes there will be an appropriate gfortran or intel fortran based mpi.mod file in the appropriate ..\lib directory of the MPI installation, and you just copy it over:

gfortran:
cd /nm75g64/mpi_mpi_ling/
cp /usr/local/mpi64/lib/mpi.mod .
cp mpi.mod ../../resource

intel fortran:
cd /nm75g64/mpi_mpi_lini/
cp /usr/local/mpi64/lib/mpi.mod .
cp mpi.mod ../../resource

Or you may have to build it from a file called mpi.h or mpi.f, probably located in the ..\include folder of the MPI installation.

gfortran:
gfortran -c -fno-range-check mpi.f90
cp mpi.mod ../../resource
gfortran -c -O3 -ffast-math -m64 -mpc64 -I../../resource -J../../resource PNM_MPI.f90
(note you may want to use whatever compile options you used for the main NONMEM installation).

Intel fortran:
ifort -c mpi.f90
cp mpi.mod ../../resource
ifort -c PNM_MPI.f90 /I:../../resource
(note you may want to use whatever compile options you used for the main NONMEM installation).

Using the Correct MPI commands on MAC OS X

The user's path should be set so that commands such as mpirun and mpf90 from MPICH2 are used instead of the corresponding Open MPI commands native to Mac OS X.
For example, if 64 bit was installed, the following is suggested prior to doing one or more NONMEM runs with MPI in a csh window:

```
% set path = (/usr/local/mpi64/bin $path)
```

If this is not done, the message may appear:

```
Unfortunately, this installation of Open MPI was not compiled with
Fortran 90 support.  As such, the mpif90 compiler is non-functional.
```

Disabling Open MPI commands on MAC OS X

The Open MPI commands that are supplied with Mac OS X must be disabled. The following is suggested:

```
% sudo -s
# cd /usr/bin
# mkdir default.mpi
# mv mpi* default.mpi
# exit
```

If this is not done, this message may appear:

```
Unfortunately, this installation of Open MPI was not compiled with
Fortran 90 support.  As such, the mpif90 compiler is non-functional.
```

I.74 Repeated Observation Records(NM72)

To assist in specialized methodologies such as stochastic differential equations ([15,16,17], and see ..\examples\sde_inline\readme.txt), a record in a data file may be set up for repeated calls to PK and ERROR. Each time, the same record is passed through PK and/or ERROR, but with a different EVID. The user's control stream model in $PK or $ERROR may then take advantage of executing certain code conditional on the EVID value. For this to occur, the user must introduce one or more of the following data items in the data file, with these names:

XVID1 XVID2 XVID3 XVID4 XVID5

These stand for "extra" EVID's. On the first call to PK/ERROR, the EVID is set to the value given in XVID1. On the second call, the EVID is set to that in column XVID2, etc. up to XVID5. Only as many XVID's as are required are needed to be defined. All the other items in the record do not change, except that if the present EVID used is not 0, then the MDV value is set to 1 for that call. If an XVID is -1, then the call to PK/ERROR for that XVID is not made,

nor for the remaining XVID's. If there is an EVID column, the value in this column is not passed to PK/ERROR unless XVID1=-1, in which case a "normal" call on that record occurs.

The following is a control stream file to a stochastic differential equation (SDE) problem (courtesy of Dr. Christoffer Tornoe), that uses the XVID data items (..\examples\sde\sde_ex2_foce_xvid.ctl in the examples):

```
; Based on sde_ex2_base.ctl, with SDE equations put in, and .csv file
modified.  From Christoffer Tornoe, example 2, and can work with NONMEM VI
; Using XVID data item method as data file type

$PROBLEM PK ODE HANDS ON ONE
$INPUT ID TIME DV AMT CMT FLAG MDV EVID SDE QA=XVID1 QB=XVID2 QZ=XVID3
$DATA   sde_ex2_xvid.dat
        IGNORE=@

$SUBROUTINE ADVAN6 TOL 10 DP
$MODEL
       COMP = (CENTRAL);
       COMP = (P1)

$THETA (0,10)               ;1 CL
$THETA (0,32)               ;2 VD
$THETA (0, 2)               ;4 SIGMA
$THETA (0,1) ; SGW1
$OMEGA 0.1                  ;1 CL
$OMEGA 0.01                 ;2 VD
$SIGMA 1 FIX                ; PK

$PK
  IF(NEWIND.NE.2) OT = 0
  TVCL  = THETA(1)
  CL    = TVCL*EXP(ETA(1))
  TVVD  = THETA(2)
  VD    = TVVD*EXP(ETA(2))
SGW1 = THETA(4)

IF(NEWIND.NE.2) THEN
  AHT1 = 0
  PHT1 = 0
ENDIF

IF(EVID.NE.3) THEN
  A1 = A(1)
  A2 = A(2)
ELSE
  A1 = A1
  A2 = A2
ENDIF

IF(EVID.EQ.0) OBS = DV

IF(EVID.GT.2.AND.SDE.EQ.2) THEN
  RVAR = A2*(1/VD)**2+ THETA(3)**2
  K1   = A2*(1/VD)/RVAR
```

```
  AHT1 = A1 + K1*(OBS -( A1/VD))
  PHT1 = A2 - K1*RVAR*K1
ENDIF

IF(EVID.GT.2.AND.SDE.EQ.3) THEN
  AHT1 = A1
  PHT1 = 0
ENDIF

IF(EVID.GT.2.AND.SDE.EQ.4) THEN
  AHT1 = 0
  PHT1 = A2
ENDIF

IF(A_0FLG.EQ.1) THEN
  A_0(1) = AHT1
  A_0(2) = PHT1
ENDIF

$DES
 DADT(1) = - CL/VD*A(1) ;+0
DADT(2) = (-CL/VD)*(A(2))+(-CL/VD)*(A(2))+SGW1*SGW1

$ERROR (OBS ONLY)
     IPRED = A(1)/VD
     IRES  = DV - IPRED
W=SQRT(A(2)*(1/VD)**2+ THETA(3)**2)
     IWRES = IRES/W
     Y     = IPRED+W*EPS(1)

$EST MAXEVAL=9999 METHOD=1 LAPLACE NUMERICAL SLOW INTER NOABORT SIGDIGITS=3
PRINT=1
$COV MATRIX=R
$TABLE ID TIME FLAG AMT CMT IPRED IRES IWRES EVID
       ONEHEADER NOPRINT FILE=sde_ex2_foce_xvid.tab
```

With the following fragment of the data file:

ID	TIME	DV	AMT	CMT	FLAG	MDV	EVID	SDE	XVID1	XVID1	XVID3
1	0	0	1000	1	0	1	1	2	-1	-1	-1
1	0.5	24.317	0	1	1	0	0	2	0	2	3
1	1	18.469	0	1	1	0	0	2	0	2	3
1	1.5	18.018	0	1	1	0	0	2	0	2	3
1	2	18.728	0	1	1	0	0	2	0	2	3
1	2.5	13.445	0	1	1	0	0	2	0	2	3
1	3	14.924	0	1	1	0	0	2	0	2	3
1	3.5	11.846	0	1	1	0	0	2	0	2	3
1	4	10.691	0	1	1	0	0	2	0	2	3
1	4.5	9.9394	0	1	1	0	0	2	0	2	3
1	5	9.9075	0	1	1	0	0	2	0	2	3
1	5.5	10.7	0	1	1	0	0	2	0	2	3
1	6	8.9861	0	1	1	0	0	2	0	2	3
1	7	7.2274	0	1	1	0	0	2	0	2	3
1	8	6.4909	0	1	1	0	0	2	0	2	3
1	9	3.7281	0	1	1	0	0	2	0	2	3
1	10	1.9238	0	1	1	0	0	2	0	2	3
1	11	2.172	0	1	1	0	0	2	0	2	3
1	12	1.0763	0	1	1	0	0	2	0	2	3
2	0	0	1000	1	0	1	1	2	-1	-1	-1
2	0.5	17.586	0	1	1	0	0	2	0	2	3
2	1	13.758	0	1	1	0	0	2	0	2	3
2	1.5	9.6241	0	1	1	0	0	2	0	2	3
2	2	9.6419	0	1	1	0	0	2	0	2	3
2	2.5	8.5945	0	1	1	0	0	2	0	2	3
2	3	6.3709	0	1	1	0	0	2	0	2	3
2	3.5	7.7656	0	1	1	0	0	2	0	2	3
2	4	4.5152	0	1	1	0	0	2	0	2	3
2	4.5	5.0167	0	1	1	0	0	2	0	2	3
2	5	4.6339	0	1	1	0	0	2	0	2	3
2	5.5	4.2107	0	1	1	0	0	2	0	2	3
2	6	3.1452	0	1	1	0	0	2	0	2	3
2	7	2.0888	0	1	1	0	0	2	0	2	3
2	8	2.4506	0	1	1	0	0	2	0	2	3
2	9	0.001	0	1	1	0	0	2	0	2	3
2	10	1.1174	0	1	1	0	0	2	0	2	3
2	11	0.001	0	1	1	0	0	2	0	2	3
2	12	0.001	0	1	1	0	0	2	0	2	3

Compare this data file with sde_ex2.dat with its repeated data record (and see its control stream file ..\examples\sde\sde_ex2_foce.ctl), which is the traditional way of programming an SDE problem in NONMEM. The ..\examples\sde\sde_ex2_base.ctl control stream file is the problem without an SDE component.

I.75 Stochastic Differential Equation Plug-In(NM72)

An alternative method to evaluating stochastic differential equation problems is to utilize the plug-in routine SDE.f90 in the NONMEM ..\examples\sde directory, which numerically evaluates the SDE equations, without requiring in-line coding into the control stream. An example control stream file is as follows (..\examples\sde\sde_ex2_impo.ctl):

```
; Based on sde_ex2_base.ctl, using SDE.f90 OTHER file, and .dat file
modified.  From Christoffer Tornoe, example 2

$PROBLEM PK ODE HANDS ON ONE

$ABBR DES=FULL ; Must have this, so DA array is simple-mapped.

$INPUT ID TIME DV AMT CMT FLAG MDV SDE ; Add SDE data item.  Should have
values between 0 and 5, as described in comments of SDE.f90
; The data item SDE must be added. It has values of 0,1,2,3,4, or 5:
; SDE (SDE data item)=0, BEGINNING OF SUBJECT, OR WHENEVER YOU WANT TO
INITIALIZE SDE SYSTEM (SOMETIMES YOU WANT TO DO THIS
; EVEN WITHIN A SUBJECT, LIKE A NEW OCCASION)
; SDE=1, FIRST OBSERVATION OF PRESENT TIME.
; SDE=2 LAST OBSERVATION FOR PRESENT TIME.
; SDE=5 FIRST AND LAST OBSERVATION FOR PRESENT TIME.
; SDE=6, MIDDLE OBSERVAION FOR PRESENT TIME.
; THUS:
; DOSE RECORD, TIME=0, THEN SDE=0
; PK OBSERVATION, TIME=0.1, SDE=1
; PD OBSERVATION, TIME=0.1, SDE=6
; EFFICACY OBSERVATION, TIME=0.1, SDE=2  (SDE=2 SINCE NEXT RECORD HAS A NEW
TIME)
; PK OBSERVATION, TIME=0.2, SDE=1
; PD OBSERVATION, TIME=0.2, SDE=2
; PK OBSERVATION, TIME=0.5, SDE=5
; PK OBSERVATION, TIME=1.0, SDE=5

$DATA    sde_ex2o.dat
         IGNORE=@

$SUBROUTINE ADVAN6 TOL=9 DP OTHER=SDE.f90
; nde=number of base equations, ncmt=number of observation compartments
$ABBR DECLARE SGW(3) ; need at least nde of these

$MODEL
        COMP = (CENTRAL); there are nde base states from original
sde_ex2_base.ctl
        COMP = (DFDX1)  ; need to add ncmt observation compartments for SDE
        COMP = (DPDT11) ; Will need (nde+1)*nde/2 of these for SDE

$PK
  IF(NEWIND.NE.2) OT = 0
  MU_1  = THETA(1)
  CL    = EXP(MU_1+ETA(1))
  MU_2  = THETA(2)
  VD    = EXP(MU_2+ETA(2))
  SGW1 = THETA(4) ;  Add estimable scalar for modeling the SDE noise.
  NCMT=1.0 ; number of compartments.  Added for calls to SDE_DER and SDE_CADD
  NDE=1.0 ; Number of original, base ODEs.  Added for calls to SDE_DER and
SDE_CADD



$DES
 FIRSTEM=1 ; MAke sure FIRSTEM=1 so that DA arrays (Derivatives of DADT() wrt
```

```
; A()), are calculated, even when IMP is done.
 DADT(1) = - CL/VD*A(1) ; Original base derivative from sde_ex2_base.ctl
; NEXT DERIVATIVES ARE ACUALLY PREDICTIVE VALUES FOR COMPARTMENTS 1 AND 2,
RESPECTIVELY
;  Derivatives of these with respect to A() will be calculated symbolically
by DES routine created by NMTRAN
 DADT(2) = A(1)/VD ;  Add output equations, required for each CMT value.
; DUMMY PLACEMENT FOR DERIVATIVES OF THE STOCHASTIC ERROR SYSTEM.  THESE ARE
FILLED OUT BY SDE_DER
SGW(1)=SGW1 ; Specify SGW with appropriate index , for appropriate DES
equation number
;  the DA() array THEN contains all derivatives of DADT (=DXDT) with respect
to A(=X).
; number of base model derivative equations (nde)=1, Number of compartments
(ncmt)=1.
; DA is a reserved array, dimensioned DA(IR,*)

"LAST
"      CALL SDE_DER(DADT,A,DA,IR,SGW,NDE,NCMT)

$ERROR (OBS ONLY)
     IPRED = A(1)/VD
     IRES  = DV - IPRED
     W     = THETA(3)
     IWRES = IRES/W
     WS=1000.0
; CENTRAL COMPARTMENT, PLASMA LEVELS
; EPS(1) = USER MODEL ERROR CONTRIBUTION
; EPS(2) = STOCHASTIC ERROR CONTRIBUTION.  THE WS IS JUST A PLACEHOLDER
COEFFICIENT.  SDE_CADD WILL REPLACE THIS
; WITH THE CORRECT VALUE
     Y     = IPRED+W*EPS(1) + WS*EPS(2)
; SDE_CADD WILL EVALUATE THE TRUE COEFFICIENTS (WS) TO THE STOCHASTIC
COMPONENTS.
;  In general, if you have nmcmt observation compartments, then first ncmt
EPS() will pertain to
; measurement error, and the second ncmt set of EPS() will pertain to
stochastic errors.
;  This means you cannot have L2 type correlations, and prop+additive should
be packaged into a single EPS().
;  For two obervations, you may have:
;  IF(CMT==1) THEN
;  IPRED=A(1)/V
;  W=SQRT(THETA((5)*THETA(5)*IPED*IPRED+THETA(6)*THETA(6))
;  Y=IPRED+W*EPS(1)+WS*EPS(3)
;  ENDIF
;  IF(CMT==2) THEN
;  IPRED=A(2)/V
;  W=SQRT(THETA((7)*THETA(7)*IPED*IPRED+THETA(8)*THETA(8))
;  Y=IPRED+W*EPS(2)+WS*EPS(4)
;  ENDIF

; Number of compartments=1, number of base model derivative equations=1

"LAST
"      CALL SDE_CADD(A,HH,TIME,DV,CMT,NDE,NCMT,SDE)
```

```
$THETA (0,2.3)                ;1 CL
$THETA (0,3.5)                ;2 VD
$THETA (0, 2)                ;4 SIGMA
$THETA (0,1) ; SGW1

$OMEGA 0.1                    ;1 CL
$OMEGA 0.01                   ;2 VD
$SIGMA (1 FIX) (1 FIX)                ; PK

$EST METHOD=ITS INTERACTION LAPLACE NUMERICAL SLOW NOABORT PRINT=1 CTYPE=3
SIGL=5
$EST METHOD=IMP INTERACTION NOABORT SIGL=5 PRINT=1 IACCEPT=1.0 CTYPE=3
$EST MAXEVAL=9999 METHOD=1 LAPLACE INTER NOABORT NUMERICAL SLOW NSIG=3
PRINT=1 SIGL=9

$COV MATRIX=R UNCONDITIONAL

$TABLE ID TIME FLAG AMT CMT IPRED IRES IWRES
       ONEHEADER NOPRINT FILE=sde_ex2_impo.tab
```

Please note that using this add-in requires $ABBR DES=FULL setting. This process works well with the methods such as importance sampling, SAEM, or BAYES, but works only partially for classical NONMEM methods or ITS. If using with classical NONMEM methods or ITS, it is better to set LAPLACE NUMERICAL, although it does not solve the problem perfectly. Classical methods rely on NMTRAN creating symbolic derivatives of the residual variance components with respect to eta, which is used to create the proper individual objective function. For this to occur, NMTRAN has to see all of the relevant equations in the control stream file, or the user must have the eta derivatives evaluated. This method has some of the SDE differential equations and RVAR components calculated in subroutines SDE_DER and SDE_CADD, "hidden" from NMTRAN. Despite this problem, classical NONMEM methods provide parameters using the SDE call routines that are similar, although not identical, to those when the SDE equations are placed in-line into the control stream file. To see how the SDE call routines work for each of the analysis methods, see sde_ex2_impo.res that uses SDE.f90, and compare the results with sde_ex2_imp_mu.res, which uses the in-line equations. The new methods (except ITS) do not need these NMTRAN constructed components, so they work with the SDE call routines quite well.

As of NM73, numerical eta derivatives are now available for FOCE/ITS, so that it is not necessary for NMTRAN to see all the code, or for the user to supply evaluation of the eta derivatives. In the following example, OPTMAP=1 is chosen to provide forward finite difference eta derivatives for the search, and ETADER=2 is chosen to provide numerically assessed central finite difference derivatives to the Hessian matrix of the posterior density (sde_ex2_impo2.ctl), allowing ITS and FOCE to obtain results similar to Importance sampling:

```
$EST METHOD=ITS INTERACTION NOABORT PRINT=1 CTYPE=3 OPTMAP=1 ETADER=2 SIGLO=6
 SIGL=6 MCETA=1

$EST METHOD=IMP INTERACTION NOABORT PRINT=1 IACCEPT=1.0 CTYPE=3 OPTMAP=0
```

```
 ETADER=0 SIGLO=6 SIGL=6 MCETA=1 MAPITER=0

$EST MAXEVAL=9999 METHOD=1 INTER NOABORT NSIG=1 PRINT=1
 OPTMAP=1 ETADER=2 SIGLO=6 SIGL=6 MCETA=1 SLOW

$COV MATRIX=R UNCONDITIONAL TOL=9 SIGL=8 SIGLO=8
$TABLE ID TIME FLAG AMT CMT IPRED IRES IWRES
       ONEHEADER NOPRINT FILE=sde_ex2_impo2.tab
```

Example 4 from Christoffer Tornoe's workshop has also been processed in several ways. They are as follows:

sde_ex4_base.ctl: User defined control stream, without SDE components.
sde_ex4_foce.ctl: SDE components added to sde_ex4_base.ctl, by Tornoe's R/Splus command, for example:

```
source("NONMEMSDEscript.R")
SDEmodel( CS="sde_ex4_base.ctl", datafile="sde_ex4_base.dat")
```

The output files created are
sde_ex4_base.ctlSDE
sde_ex4_base.datSDE

These were manually renamed to
sde_ex4_foce.ctl
sde_ex4.dat
for the present example.

sde_ex4_foceo.ctl: This problem was modified to use the SDE.f90 OTHER routine, and FOCE was used for estimation.

sde_ex4_foceo.ctl: This problem was modified to use the SDE.f90 OTHER routine, and FOCE was used for estimation.

sde_ex4_impo.ctl: This problem was modified to use the SDE.f90 OTHER routine, and IMP was used for estimation.

See ..\examples\sde_inline\readme.txt in using Tornoe's inline code enhancer method. The choice to use Tornoe's inline code enhancer (using NONMEMSDEscript.R) or whether to use the SDE.f90 OTHER routine method depends on the complexity of the model. The NONMEMSDEscript. can only enhance NMTRAN code for which there are observations from only one output compartment. Its advantage is the NMTRAN sees all of the ODE code, and provides appropriate analytical derivative components for random variables (ETAS). Typically this results in faster execution, especially for FOCE. The SDE.f90 OTHER routine can handle multiple output compartments, but does not provide derivatives to the additional ODE components, so finite difference evaluation of random eta variables (through OPTMAP=1 and ETADER=2) must be used, which can be 2-4 times slower in execution.

I.76 Expanded Syntax and Capacity for User-Defined Functions (FUNCA) (NM74)

For the several past versions to NONMEM, user-defined functions may be incorporated into the NONMEM problem, and made available particularly for use in the classical NONMEM estimation methods, so long as the function also returns first and second derivatives. More information can be obtained from Guide VIII as to how this is done. Before nm74, at most 9 functions could be used, they had to be called FUNCA through FUNCI, each of which could be specified at most 9 times, and each function could accommodate only a maximum of 9 input variables.

In addition to returning the main result as a function of these input arguments, the function is also to return up to 9 partial derivatives of the result with respect to each variable, and their second derivatives. For example, a function FUNCA may be defined in a file called myfunc.f90, with the following header information:

```
FUNCTION FUNCA(X,X1,X2)
REAL*8 FUNCA,X(9),X1(9),X2(9,9)
…
…
FUNCA=…
RETURN
END
```

and the user may reference this function in the control stream file, as:

```
$SUB OTHER=myfunc.f90
…
$PK
…
…
VECTRA(1)=CL
VECTRA(2)=V
VECTR(3)=Q
VECTR(4)=V2
…
W=FUNCA(VECTRA)

…
```

The user function FUNCA will accept the input vector VECTRA() as the first argument, which maps to X() in the FUNCA routine itself. The user function then calculates the result, as well as packages the first derivatives of the result with respect to each element in VECTRA(), placing them in the array X1(). So the partial derivative of result with respect to VECTRA(2) is to be placed in X1(2). Similarly, second partial derivative of result with respect to VECTRA(i), VECTR(j) is to be packaged in X2(i,j).

As of nm74, a function may have more than 9 input variables, each function may be specified in abbreviated code more than 9 times, and up to 100 different functions may be used (although this can be increased in SIZES by changing NFUNCX and NVECX). Furthermore, the functions need not be named FUNCA, FUNCB, etc. in the function file.

To use this more versatile feature, the function must be defined as follows:

$ABBR FUNCTION function_name(input_vector_name,dimension,usage)

where function_name is the name of the function as it appears in the source file specified in OTHER, input_vector_name is the name of the input vector you will use in the control stream file when passing the first argument, dimension is the vector and matrix sizes, and usage is the maximum number of times you will be calling this function in the control stream file. The maximum usage need not be entered, and is 999 by default. However, you may wish to set a lower maximum usage boundary for NMTRAN to flag, if you desire.

For example,

```
$ABBR FUNCTION BIVARIATE(VBI,5,3)
```

means there is a function in the OTHER source code file with the following header:

```
FUNCTION BIVARIATE(X,X1,X2,NDIM)
INTEGER NDIM
REAL*8 X(NDIM),X1(NDIM),X2(NDIM,NDIM)
```

In the control stream file, the input vector VBI is to be used to load into the function:

```
$PK
…
   VBI(1)=RHO
   VBI(2)=5
   VBI(3)=6
   VBI(4)=1    ;***0 = Upper tail as in Drezner & Wesolowsky; 1 = Bottom
tail***;
   VBI(5)=1  ;***0 = 3 pt approximation; 1 = 5 point approximation***;
   BV=BIVARIATE(VBI)
```

Notice that in the control stream file only the first argument is given, that of the input vector X. NMTRAN will add the additional arguments, for first derivative vector (X1), second derivative matrix (x2), and dimension (NDIM, in this case equaling, 5). The maximum number of times BIVARIATE may be specified is 3, for this example. Notice that functions declared in the $ABBR FUNCTION option must allow for passing the dimension number NDIM, which the function should use to dynamically size and shape the arguments.

Usually you want to associate a specific vector to each function such as:

```
$ABBR FUNCTION BIVARIATE(VBI,5)
$ABBR FUNCTION BIVARIATEQ(VQI,10)
```

to assure that each vector-function pair are set up with comparable dimensions. However, it is not essential to have vectors and functions paire up like this, it is more a convenience, and avoids confusion. The vectors and functions need not be defined on the same $ABBR FUNCTION line, and you can use different vectors for different functions in the abbreviated code. Use the asterisk as a place holder, for example:

```
$ABBR FUNCTION BIVARIATE(*,5)
$ABBR FUNCTION BIVARIATEQ(*,10)
$ABBR VECTOR VQI(15)
```

In the above code, functions BIVARIATE and BIVARIATEQ are defined separately from vector VQI, whose use will be shared with both functions. Notice VQI is dimensioned differently (15) from BIVARIATE (5) and BIVARATEQ (10), which as a precaution the dimension of VQI should be at least the dimension for all the functions for which it will be used. Then VQI may be used with BIVARIATE or BIVARIATEQ:

```
…
VQI(1)=RHO
VQI(2)=MX
BVAL=BIVARIATE(VQI)
RVAL=BIVARIATEQ(VQI)
VECTRA(1)=RHO
VECTRA(2)=MX2
YVAL=FUNCB(VECTRA)
QVAL=BIVARIATEQ(VECTRA)
```

Furthermore, you may still use the FUNCxyz and VECTRxyz nomenclature for functions and vectors not declared by $ABBR FUNCTION, as shown above, and use any vector with any function. Keep in mind, that vector VECTRA not pre-defined in $ABBR FUNCTION, and used in code, as shown above, will have the implicit dimension of 9, and the FUNCB defined in the OTHER source code file cannot have the additional NDIM argument, and must be set up with a dimension of 9.

I.77 First Derivative Assessments (NM72, NM74)

NONMEM 7.2.0 and higher versions normally calculates first derivatives in the FSUBS file for classical NONMEM methods, and does not evaluate them for IMP, SAEM, and BAYES methods. This improves the speed at which the problem is evaluated. However, on occasion such derivatives are needed, for example, when steady state values are to be calculated, or when stochastic differential equations are to be evaluated. In such cases, insert as the first line in a block of abbreviated code (such as $PK, $ERROR, $DES, etc):
FIRSTEM=1

Then, incidental derivatives will be evaluated for the new methods as well. For steady state and stochastic differential equation problems, FIRSTEM=1 needs to be inserted only at the beginning of the $DES block (when ADVANS 6,8,9,13,14,15 are used).

NMTRAN has been modified such that it collects all first derivative computations together, and performs them only if FIRSTEM=1. For example, in the PK subroutine, generated for ..\examples\example1.ctl:

```
      IF (FIRSTEM == 1) THEN
!                     A00033 = DERIVATIVE OF CL W.R.T. ETA(01)
      A00033=B00002
```

```
!                     A00038 = DERIVATIVE OF V1 W.R.T. ETA(02)
      A00038=B00004
!                     A00043 = DERIVATIVE OF Q W.R.T. ETA(03)
      A00043=B00006
!                     A00048 = DERIVATIVE OF V2 W.R.T. ETA(04)
      A00048=B00008
!                     A00051 = DERIVATIVE OF S1 W.R.T. ETA(02)
      A00051=A00038
      GG(01,1,1)=CL
      GG(01,02,1)=A00033
      GG(02,1,1)=V1
      GG(02,03,1)=A00038
      GG(03,1,1)=Q
      GG(03,04,1)=A00043
      GG(04,1,1)=V2
      GG(04,05,1)=A00048
      GG(05,1,1)=S1
      GG(05,03,1)=A00051
      ELSE
      GG(01,1,1)=CL
      GG(02,1,1)=V1
      GG(03,1,1)=Q
      GG(04,1,1)=V2
      GG(05,1,1)=S1
      ENDIF
```

Every effort has been made to assure that this new process by NMTRAN works for every type of model. However, it may occur that NMTRAN arranges the equations in the wrong order, and your problem may not work correctly, whereas it may have worked correctly in NONMEM 7.1.2 or earlier. Should this occur, the re-arrangement of equations by NMTRAN can be turned off by inserting

$ABBREVIATED NOFASTDER

in the control stream file. If the problem is resolved using this setting, please send your example control stream file to nmconsult, and we will fix the error for the next version.

As of nm75, another variable used to control derivative assessment is DES_DER, which should be placed in $PK:

$PK
…
DES_DER=1

and this will turn on first derivative assessment for the subsequent call to the DES subroutine ($DES block). This is required if it is desired that the Jacobian be analytically evaluated for ODE models (ADVAN8, 9, 13, 14, 15, 16, 17), for IMP, SAEM, BAYES problems that normally do not have first derivatives turned on.

The DES_DER maps to the MITER variable, and can be used in place of setting MITER to 1, as described in guide VIII, section “DIFF EQ SOLVER SETTINGS”.

For very large model problems, NMTRAN can take a long time to produce code and variable names of all of the first derivative components. As of NM74, If 1st derivatives are not required (such as when only simulations are performed, SAEM or BAYES is only performed, or

IMPMAP/ITS/FOCE are performed using OPTMAP>0 and ETADER>0), then you can turn off analytical eta first derivative and other first derivative code production with

$ABBREVIATED DERIV1=NO

If this is set by the user, then NMTRAN will generate code that will insert
NOFIRSTDERCODE=1
in the PK, ERROR, PRED, and DES routines. This will inform NONMEM that analytical first derivative code is not available. This reserved variable is defined in the module NMPRD_INT. If a user is writing a user-defined PRED or PK routine in which analytical first derivatives will not be included, whether by NMTRAN, or by the user, the statement NOFIRSTDERCODE=1 should be inserted in the PRED routine.

Please note that any steady state evaluations requested, along with one of the ODE solver ADVANs (6,8,9,13,14,15) requires first derivatives, whether estimation or simulation.

Here are some estimation options that can be executed with no first derivatives.

```
$EST METHOD=SAEM … ; SAEM has no first derivative requiring components
$EST METHOD=BAYES… ; BAYES has no first derivative requiring components
$EST METHOD=IMP MAPITER=0… ; IMP uses derivatives in MAP
                           ; estimation on first iteration, unless MAPITER=0
$EST METHOD=ITS OPTMAP=1 ETADER=2… ; necessary derivatives can be
                                   ; numerically determined
$EST METHOD=IMP OPTMAP=1 ETADER=2…
$EST METHOD=IMPMAP OPTMAP=1 ETADER=2…
```

METHOD=NUTS requires first derivatives.

I.78 Ignoring Non-Impact Records During Estimation (NM73)

Typically users may produce data files that are augmented with additional non-dose, non-observation records in order to output predicted values at additional times to create high resolution curves. However, too many of such records tend to slow down the estimation analysis. As of NM73, if an MDV is set to a value greater than or equal to 100, it is converted to that value minus 100 upon input, but will not be used during estimation or covariance assessment, only for table outputting. This option allows you to use the same file for estimation and table outputs, without significantly slowing down the estimation. So if MDV=101, it will be converted to 1 upon use for final evaluations, and the records will be ignored during estimation.

The subroutines in NONMEM that ignore MDV=100 and MDV=101 records are: OBJ (all estimation and covariance steps), OBJ2 (parametric), OBJ3 (non-parametric), and OS (initial estimates of omegas and sigmas). Care must be taken in using MDV>=100, in that during estimation, covariate data items of these records are not used, which can have a slightly different interpolation impact than what is finally recorded in the tables where they are used. You may specifically request that any one of these routines not ignore the MDV>=100 records, by setting MDVI1=1 (for OBJ to include MDV>=100 records), MDVI2=1 (for OBJ2 to include

MDV>=100 records), MDVI3=1 (for OBJ3 to include MDV>100 records), in a $PK or $PRED block, for example:

```
$PK
include nonmem_reserved_general
MDVI1=1
MDVI2=1
MDVI3=1
```

I.79 table_quant, and table_resample Utility Programs for Analyzing $COV Sampling-Importance-Resampling (SIR) data (NM74)

The sub-section *Importance Sampling of the Variance-Covariance of the Parameter Estimates (NM74)* in section I.56 $COV: Additional Options and Behavior describes how importance sampling may be used to obtain samples around the minimum of the FOCE/Laplace objective function. After random samples with WEIGHT values is recorded in the .ext file, these results may be further analyzed using two utility programs.

The utility ..\util\table_quant will transform the results in the raw output file, utilizing the WEIGHT column, into a table file with frequencies and cumulative values. The utility is to be executed on the command line as follows:

table_quant *root.ext _root.qnt delimiter start end*

where *delimiter* is that used in the input file *root.ext* (s for space, default), and *start* and *end* are the range of iterations to be quantized (default is all non-negative iterations). Note that, even Bayes results, which do not have the WEIGHT column, can be processed with this utility, but the weight is then assumed constant among all samples.

The resulting file, *root.qnt*, will contain for each item, the sorted value, its frequency (freq), and its quantile position or cumulative probability (cum). One can readily plot the quantile value against quantile position for a cdf plot, or sum(freq) by quantile value range for frequency plot. The R script quantplot.R, or Splus script, quantplot.ssc, available in Pdx-Pop 5.2, can be used to view histograms and cdf plots from these result files, and produce quantile tables, linearly-interpolating the results at the most interesting quantile positions (0.025, 0.5, 0.975, etc). To use these plotting scripts, make sure you modify the header information to point to the desired file name and extension (by default these scripts look for a qnt extension). Also, make sure a .qnt file contains just one table of information.

An alternative to viewing the $COV/SIR results via its weight information is to perform a resampling of the information in the .ext file, with samples weighted according to the WEIGHT column (if no WEIGHT column is present, WEIGHT is assumed to be equal among all samples). This is done using the utility table_resample:

table_resample *root.ext _root_new.ext delimiter newsize SEED start end*

where *delimiter* is that used in the input file *root.ext* (s for space, default), and *start* and *end* are the range of iterations to be quantized (default is all non-negative iterations). In addition:

seed=0: non-randomized expansion of the samples, based on WEIGHT column
seed>0: randomized starting at seed, with repeated samples allowed
seed<0: randomized starting at abs(seed), with repeated samples not allowed

If the user chooses seed=0, then *newsize* samples will be generated, each line of the original root.ext file being repeated in proportion to its WEIGHT value, and these repeated samples will be placed in root_new.ext. Thus the weight of each sample is physically expressed in the manner of repeated rows of that sample. To assure that integer truncation does not render the smaller weighted samples to be not at all expressed, newsize should be something like 10000, or even 100000. The resulting file, root_new.ext, will have the same structure as a BAYES result file, without the WEIGHT column, and the R script bayesplot.R or Splus script bayesplot.ssc, available in Pdx-Pop 5.2, may be used to view histograms, quantile plots, and quantile tables.

If the user chooses seed>0, then *newsize* samples will be generated randomly and with replacement, in proportion to the WEIGHT column, and placed in the *root_new.ext* file. If

newsize>min(oldsize of original file, end-start+1)

then it would make sense to choose this option. The seed<0 should be used only if

newsize<<min(oldsize of original file, end-start+1)

that is, you just want to pick a few samples.

I.80 table_compare Utility Program(NM72)

The utility program table_compare will compare the numerical values between two table files produced by the NONMEM $TABLE record, and the user may specify the tolerance for the comparison. The syntax is:

```
table_compare mytable1.tab mytable2.tab , myprecision.xtl >mydifferences.txt
```

where delimiter is {, t s} for {comma tab space}, and myprecision.xtl is a precision specification or control file. Default delimiter is space and default control file is table_compare.xtl.

```
table_compare mytable1.tab mytable2.tab , S myprecision.xtl >mydifferences.txt
```

In the above example, the first file is comma delimited, and the second one is space (S) delimited.

If a second character is given to a delimiter, then this is for detecting a continuation marker at the end of a line that is to be continued. If a third character is given as a delimiter, this for detecting a continuation marker at the beginning of the continuing line. Some examples are:

```
table_compare mytable1.tab mytable2.tab ",&" "S&" myprecision.xtl >mydifferences.txt
```

(double quotes may be needed for DOS commands). In the above example, the first file is delimited by commas between column items, and an & at the end of a line breaks the record across multiple lines. The second file is delimited by spaces between column items, and an & breaks a record across multiple lines.

```
table_compare mytable1.tab mytable2.tab ",&c" "S&c" myprecision.xtl >mydifferences.txt
```

In the above example, the first file is delimited by commas between column items, and an & at the end of a line breaks the record, with a c at the beginning of the next line. The second file is delimited by spaces between column items, and an & at the end of a continuing line, and a c at the beginning of the next line.

```
table_compare mytable1.tab mytable2.tab ",&" "SSc" myprecision.xtl >mydifferences.txt
```

In the above example, the first file is delimited by commas between column items, and an & at the end of a line breaks the record. The second file is delimited by spaces between column items, and no special character at the end of a continuing line (the S serves as a place-holder for line contination markers, since apace is too ambiguous as a continuator) and a c at the beginning of the next line.

It is useful to redirect difference results to a file, in this example mydifferences.txt. For example, the user may desire that only relative differences greater than 0.01 be reported. A very simple control file could be:

```
$PRECISION
ALL=0.01,0.003
```

stating that all columns be compared with a relative difference of 0.01, and absolute difference of 0.003. Precision crietria for specific columns in the tables may also be given:

```
$PRECISION
ALL=0.01,0.003 WRES=0.1,0.2
CL=0.05,0.02
```

The equation for comparison is, if
ABS(X-Y)>R*MAX(ABS(X),ABS(Y))+A

then the difference is reported, where R is relative difference tolerance, and A is absolute difference tolerance.

I.81 table_to_xml Utility Program(NM72)

The utility table_to_xml program in the NONMEM ..\util directory can be used to convert additional NONMEM output tables produced during the $EST step into XML formatted files. The syntax is as follows, as an example:

```
table_to_xml my_results.cov my_results_cov.xml ,
```

where the delimiter may be , t, or s for comma, tab, or space. Default delimiter is space. The rules (schema, document type definition) by which the xml file is constructed are given in tables.xsd and tables.dtd, which are in the ..\run or ..\util directory.

```
table_to_xml my_results.cov my_results_cov.xml ",&c"
```

specifies that the table file may have line continuator characters & and c, as described in the table_compare section.

I.82 xml_compare Utility Program and its Use for Installation Qualification (NM72)

The utility program xml_compare will compare the contents of two NONMEM report XML files that are produced by NONMEM. The syntax to the command line is:

xml_compare myresult1.xml myresult2.xml myprecision.xtl >mydifferences.txt

where myprecision.xtl is a precision specification or control file. Default delimiter is space and default control file is xml_compare.xtl. It is useful to redirect difference results to a file, in this example mydifferences.txt.

The control file can be quite elaborate, but it allows specification of various precision values for the many different types of values in the NONMEM report XML file, and to ignore certain entries as well. An example xml_compare.xtl file is in the ..\util directory, and has the following contents:

```
$IGNORE
monitor
elapsed_time
datetime
covariance_status
termination_status
nonmem(version)
parallel_est
parallel_cov

$PRECISION
GENERAL=0.2,0.2      OBJ_BAYES=2.0,0.0      OBJ_SAEM=0,100.0      OBJ_ITS=0,5.0
OBJ_IMP=0,10.0 OBJ_F=0,5.0
DIAG=0.3,0  OFFDIAG=0,0.5  COR=0.0,0.3  VAR=0.3,0.1  COV=-1.0  EIGENVALUES=2.0,0
OBJ_DIRECT=0,100.0
correlation_o=-1.0 INVCOVARIANCE_O=-1 INVCOVARIANCE_D=-1
etashrinksd=0,20 epsshrinksd=0,10 ebvshrinksd=0,20
etashrinkvr=0,20 epsshrinkvr=0,10 ebvshrinkvr=0,20


METHOD=DIRECT ALL=-1

METHOD=SAEM epsshrinksd=0,20
```

The $IGNORE record will ignore all elements with the substrings that are listed, or just a specific attribute of an element, such as nonmem(version).

Under the $PRECISION record, a
GENERAL=R,A

can be given for most items, where R is the relative tolerance, and A is the absolute tolerance. Following the GENERAL specification, tolerances may be specified for other items.

Two items of identical element and attributes are compared between the two files, where the equation for comparison is, between value X of xml file 1 and value Y of xml file 2,

```
ABS(X-Y)>R*MAX(ABS(X),ABS(Y))+A
```

The OBJ_BAYES is given a special test, as it has a standard deviation with it:

```
STD(X,Y)=SQRT(STD(X)²+ STD(Y)²)
ABS(X-Y)>R*STD(X,Y)+A
```

In the above example OBJ_BAYES=(2,0) means that if the Bayes objective functions in the two files differ by more than 2 standard deviations, then the difference is noted. Please note that while the above test is suitable for tolerance comparison in an installation qualification setting, this is not an appropriate statistical test for model comparisons.

To ignore an item for comparison, specify -1. To specify an exact comparison, use 0,0. To refer to a particular optimization method, then enter METHOD=SAEM for example, and thereafter, all entries of items pertain to that estimation method, until METHOD is changed. The METHOD attribute may have one of the following settings:

FOCE, ITS, IMP, SAEM, DIRECT, BAYES (for standard and NUTS)

The total list of items, and their scope, are as follows (R/2=1/2 of relative error):

NAME	DESCRIPTION	DEFAULT (R,A)
GENERAL	Default to most non-matrix items	0.2,0.2
DIAG	Diagonal elements of OMEGA/SIGMA estimates	0.1,0
OFFDIAG	Off-diagonal elements of OMEGA/SIGMA estimates	0.0,0.2
VAR	Diagonals of variance of estimates	0.2,0
COV	Off-diagonals of covariance of estimates	0,0.2
COR	Correlations	0,0.2
TABLE	Table items listed in NONMEM report file.	GENERAL
OBJ_BAYES	BAYES objective function	1,0
OBJ_SAEM	SAEM objective function	0,100
OBJ_ITS	ITS objective function	0,2
OBJ_IMP	IMP/IMPMAP objective function	0,5
OBJ_DIRECT	Direct sampling objective function	0,100
OBJ_F	FO/FOCE/Laplace objective function	0,0.5
EIGENVALUES	Eigenvalues	2,2
ETABAR	Etabar	GENERAL
ETABARSE	Etabar Se	GENERAL
ETABARPVAL	Etabar Pval	GENERAL
ETASHRINKSD	Eta shrinkage, SD type	GENERAL

NAME	DESCRIPTION	DEFAULT (R,A)
EPSSHRINKSD	EPS shrinkage, SD type	GENERAL
EBVSHRINKSD	ETA Empirical Bayes Variance shrinkage, SD type	GENERAL
ETASHRINKVR	Eta shrinkage, variance type	GENERAL
EPSSHRINKVR	EPS shrinkage, variance type	GENERAL
EBVSHRINKVR	ETA Empirical Bayes Variance shrinkage, variance type	GENERAL
THETA	Thetas	GENERAL
OMEGA_D	Omega diagonals	DIAG
OMEGA_O	Omega off-diagonals	OFFDIAG
SIGMA_D	Sigma diagonals	DIAG
SIGMA_O	Sigma off-diagonals	OFFDIAG
OMEGAC_D	Omega correlation diagonals	DIAG (R/2,A)
OMEGAC_O	Omega correlation off-diagonals	COR
SIGMAC_D	Sigma corrlation diagonals	DIAG (R/2,A)
SIGMAC_O	Sigma correlation off-diagonals	COR
THETASE	Theta standard errors	VAR(R/2,A)
OMEGASE_D	Omega diagonal standard errors	VAR(R/2,A)
OMEGASE_O	Omega off-diagonal standard errors	COV(R/2,A)
SIGMASE_D	Sigma digaonl standard errors	VAR(R/2,A)
SIGMASE_O	Sigma off-diagonals standard errors	COV(R/2,A)
OMEGACSE_D	Omega correlation diagonal standard errors	VAR(R/2,A)
OMEGACSE_O	Omega correlation off-diagonal standard errors	COV(R/2,A)
SIGMACSE_D	Sigma correlation diagonal standard errors	VAR(R/2,A)
SIGMACSE_O	Sigma correlation off-diagonal standard errors	COV(R/2,A)
THETANP	Nonparametric Thetas	GENERAL
EXNPETA	EX non-paramatric etas	GENERAL
COVNPETA_D	Covariance of nonparametric etas, diagonals	DIAG
COVNPETA_O	Covariance of nonparametric etas, off-diagonals	OFFDIAG
OMEGANP_D	Omega of nonparametric analysis diagonals	DIAG
OMEGANP_O	Omega of nonparametric analysis off-diagonals	OFFDIAG
COVNPETAC_D	Correlation of nonparametric etas, diagonals	DIAG (R/2,A)
COVNPETAC_O	Correlation of nonparametric etas, off-diagonals	COR
OMEGANPC_D	Omega correlation of nonparametric analysis diagonals	DIAG (R/2,A)
OMEGANPC_O	Omega correlation of nonparametric analysis off-diagonals	COR
COVARIANCE_D	Diagonals of variance-covariance of estimates	VAR
COVARIANCE_O	Off-Diagonals of variance-covariance of estimates	COV
CORRELATION_D	Diagonals of correlation of variance-covariance of estimates	VAR(R/2,A)
CORRELATION_O	Off-Diagonals of correlation of variance-covariance of estimates	COR
INVCOVARIANCE_D	Diagonals of inverse of variance-covariance of estimates	VAR
INVCOVARIANCE_O	Off-Diagonals of inverse of variance-covariance of estimates	COV
SMATRIX_D	Diagonals of S-MATRIX	VAR
SMATRIX_O	Off-diagonals of S-MATRIX	COV
RMATRIX_D	Diagonals of R-MATRIX	VAR
RMATRIX_O	Off-diagonals of R-MATRIX	COV

Because of the versatility of selecting which items are to be compared and with what precision, the xml_compare program can be used for batch processing installation qualification procedures, in comparing NONMEM results of a test run against a reference run. All results given in the standard NONMEM output file are also reported in the XML file.

For example, you may wish to compare your results for example1 against the results given in the ..\examples directory of your NONMEM installation, run from your run directory, or a special installation qualification directory you may have set up:

```
Nmfe75 example1.ctl example1.res
xml_compare \nonmem7.2.0\examples\examples1.xml example1.xml example1.xtl
>example1.dif
```

example1.xtl would be a file you may have modified from xml_compare.xtl to suit your installation qualification needs. These .xtl files are listed in the ..\examples directory, and are simply replicates of xml_compare.xtl. You may change these for each example problem as needed. The file example1.dif will contain a list of differences, if any.

Available in the ..\util directory are some example batch processing installation files, that will execute example1 through example10l, then perform an installation qualification on these results files, against the ones in NONMEM's ..\examples directory:

Call example.bat (this will take many hours)
Call iq.bat (this will take 10 minutes)

The iq.bat repeatedly calls dif.bat. Remember to modify the "dir" option in iq.bat to point to the actual NONMEM installed directory. Also, modify dif.bat and iq.bat as needed for your particular environment. The iq.bat script will return a total differences count among all the example files. This is a convenient way of automating an installation qualification.

I.83 finedata Utility Program(NM73)

The utility program finedata in the ..\util directory will augment an NM-TRAN data file to incorporate additional, non-observation, time values spaced at regular increments so that when a table is generated, NONMEM can fill these records with predicted values, from which smooth prediction curves may be plotted.

The syntax is as follows:
finedata fineplot.ctl
or you may use the re-direction syntax:
finedata <fineplot.ctl

where ..\util\fineplot.ctl is an example control stream file with special commands for the finedata program. The fineplot.ctl example is extracted from part of example6.ctl:

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6.csv IGNORE=C

$FINEDATA TSTART=0 TSTOP=50 NEVAL=100 AXIS=TIME(LIN) CMT=1,3
          FILE=example6b.csv
```

The only records that finedata pays attention to is $INPUT, from which it obtains the column names, $DATA, from which it obtains the input data file, $FINEDATA, which contains instructions of how to fill in with additional fine increment time records, and $PROB by which problems are separated. All other control stream records are ignored. Thus, a way to create a control stream is to copy the first records describing the data layout from an existing NONMEM control stream file, and then adding the $FINEDATA record. The options to $FINEDATA are as follows:

TSTART=start time (real number or integer) for creating incremental time records. If you specify FIRST, or do not specify a value for TSTART, then the time of the first record of the subject or occasion (see OCC below) is used, or when the time is less than that of the previous record, or when EVID=3 or EVID=4. If TSTART is not a number and is not FIRST, then it is interpreted as the column name in the original data set containing the start time. In such cases, the TSTART value of the first data record of the subject is used, or of the first data record, or upon occasion change (if OCC= was given), or if EVID=3 or 4, or after a re-initialization of time (indicated by the time in the data record being less than that of the previous record). Thus, TSTART could differ according to instance. The same holds true for TSTOP, TDELTA, or NEVAL (see below) if they are obtained from the data file.

OCC=name of occasion column. This is optional, and will restart the time incrementing when the occasion changes, in addition to the other conditions listed above.

NEVAL=number of incremental time records per subject (integer, or truncated if real). If not a number, then column name in the data set containing NEVAL value. If NEVAL=-1, then you wish to interpolate covariate values in the original data set, but not add any additional records.

TDELTA: Alternative to entering NEVAL, the increment in time may be entered. If not a number, then the column name in the original data set containing the TDELTA is used.

TSTOP=stop time (real number or integer) for creating incremental time records. IF TSTOP is not specified, then default is LAST, and the last record of the subject or occasion or time section is used. If TSTOP is not a number and is not LAST, then it is assumed be the column name in the original data set containing the stop time.

FILE=output data file name, to contain original data records interspersed with incremental time records.

AXIS=Name of column containing times, usually TIME. Optionally, designate (LIN) or (LOG) in parenthesis, to indicate linear or geometric time incrementing.
If LIN: additive time increment=(tstop-tstart)/(neval+1)
If LOG: multiplicative time increment=(tstop/tstart)**(1/(neval+1))

DELIM=delimiter of output data file, if it is to be different from the input data file. DLEIM=S is space, DELIM=t is tab.

ITEM=number list of values for data item *ITEM* for which there is to be a record at each time increment. This can be done for a series of data items. For example, if you enter

```
$FINEDATA CMT=1,3 EVID=2,2
```

then two records per time point are inserted, one with CMT=1, EVID=2, and the other with CMT=3, EVID=2.

Or,

```
$FINEDATA CMT=1,1,3,3 EVID=0,2,0,2
```

Inserts four records per time point, with the following CMT, EVID values, in the order specified:

CMT	EVID
1	0
1	2
3	0
3	2

MISSING=comma-delimted-list of missing symbols.
By default a period (.) and space (s) are considered missing values. Values such as 0 or -99 may be present in the data as symbols for missing values. They may be described with MISSING=0 or MISSING=-99. During interpolation, missing values will be skipped, and only records with non-missing values will be used for interpolation.

If NEVAL/=-1, only the inserted records will have filled in interpolated values, and the original records will remain untouched. When NEVAL=-1, then original records will be filled in for the specified items, but no inserted records will be added. Thus, filling missing values in original records is done as a separate action from inserting records. They may not be done simultaneously in finedata with a single $PROB, but these two actions can be accomplished by two sequential $PROB records. See finetest7.ctl to first fill in original records with interpolated values, followed by using the resulting data file as the input for the next $PROB, in which additional records are inserted:

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT WT
$DATA finetest.csv IGNORE=C
$FINEDATA NEVAL=-1 AXIS=TIME(LIN) MISSING=-99 WT=LIN
          file=finetest7.csv

$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT WT
$DATA finetest7.csv IGNORE=C
$FINEDATA tstart=0 TSTOP=50 NEVAL=250 AXIS=TIME(LIN) CMT=1,3 WT=LIN,PREV MISSING=-99
          file=finetest7a.csv
```

A scheme to determine how to supply values to various data items for these inserted records may also be given. For example, to specify that the value of the next original record should be used to supply the value for WT in the inserted record:

```
$FINEDATA WT=NEXT
```

The following values may be given:

NEXT: When inserting records between two consecutive original records of time t1 (PREV) and t2 (NEXT), the PREDPP's default of using the covariate value of the t2 (NEXT) record is used for the inserted records. NEXT is the default.

PREV: When inserting records between two consecutive original records of time t1 (PREV) and t2 (NEXT), the covariate value of the t1 (PREV) record is used for the inserted records. (LAST may be coded instead of PREV, to be consistent with the options of the $BIND record. Note that the $BIND record is not used by finedata.)

LIN, or LINLIN: A covariate-linear, time-linear interpolation is used for the covariate value for the inserted records. LINT or LINLINT (T for truncate) produces truncated integer values, LINR or LINLINR (R for round) produces values rounded to the nearest integer.

LOG, or LOGLIN: A covariate-logarithmic, time-linear interpolation is used for the covariate value for the inserted records. A T or R suffix results in truncated or rounded integer values, respectively.

LINLOG: A covariate-linear, time-logarithmic interpolation is used for the covariate value for the inserted records. A T or R suffix results in truncated or rounded integer values, respectively.

LOGLOG: A covariate-logarithmic, time-logarithmic interpolation is used for the covariate value for the inserted records. A T or R suffix results in truncated or rounded integer values, respectively.

Another example:

```
$FINEDATA CMT=3,3 EVID=NEXT,2
```

indicating to create two inserted records for a given fine time point. For the first inserted record, CMT=3, and EVID of the next original record. For the second inserted record, CMT=3 and EVID=2.

Inserted records will be given the following values by default (unless over-ridden by a data item specification, such as $FINEDATA EVID=2):
DV=.
EVID=0
MDV=1

Times may be entered as numerical values, or in hh:mm:ss format. Data sets with DATE/TIME records may also be processed (but then TSTART and TSTOP must be in numerical hours or hh:mm:ss format).

Once finedata produces the augmented data file, in this example example6b.csv, then, a suitable NM-TRAN control stream file that would take advantage of these augmented records would be (taken from example6b.ctl in the ..\util directory):

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6b.csv IGNORE=C
```

```
$SUBROUTINES ADVAN13 TRANS1 TOL=4
$MODEL NCOMPARTMENTS=3

$PK
...

$DES
...

$ERROR
CALLFL=0
ETYPE=1
IF(CMT.NE.1) ETYPE=0
IPRED=F
Y = F + F*ETYPE*EPS(1) + F*(1.0-ETYPE)*EPS(2)

…

$EST METHOD=ITS INTERACTION SIGL=4 NITER=25 PRINT=1 FILE=example6.ext NOABORT
$TABLE ID TIME CONC IPRED CMT MDV EVID NOAPPEND NOPRINT FILE=example6b.fin
FORMAT=,1PE12.5 ONEHEADER
```

Of importance here is the $TABLE record. The file example6b.fin is generated by NONMEM, providing individual predicted values for each incremental time because of their presence in the input data file example6b.csv. Because incremental time records have MDV=1, there will be no impact on the estimation results. The table structure and contents of example6b.fin is suitable for importing into plotting programs, which can present smooth prediction curves (choose connect-line and no symbol) superimposed on observed data (choose with symbol, and no connect-line).

Although the added MDV=1 fine-date lines do not impact the estimation results (except where NONMEM may utilize time-changing covariates, and pick up a covariate value from these new records), they can increase estimation time. It may therefore be of advantage to perform the estimation using the original data file, followed by table generation using the enhanced data file. The FNLETA=2 setting comes in handy for this purpose:

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6.csv IGNORE=C ; original data file used

$SUBROUTINES ADVAN13 TRANS1 TOL=4
$MODEL NCOMPARTMENTS=3

$PK
...

$DES
...

$ERROR
CALLFL=0
ETYPE=1
IF(CMT.NE.1) ETYPE=0
IPRED=F
Y = F + F*ETYPE*EPS(1) + F*(1.0-ETYPE)*EPS(2)

…

$EST METHOD=ITS INTERACTION SIGL=4 NITER=25 PRINT=1 FILE=example6.ext NOABORT
               MSFO=example6.msf ATOL=4 FNLETA=0

$PROB RUN# example6 (from r2compl)
```

```
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6b.csv IGNORE=C ; enchanced data file
$MSFI example6.msf
$EST METHOD=1 FNLETA=2 ATOL=4
; Because FNLETA=2, no estimation us actually done. The etas loaded from the MSF file
; are used without modification to compute individual model parameters.
; Since no analysis is performed, setting METHOD=1 is sufficient, regardless of
; what method was used in the earlier analysis.
; Because ATOL=4 in the previous analysis, good idea to retain this setting, to yield
; identical evaluations from the differential equation solver.
$TABLE ID TIME CONC IPRED CMT MDV EVID NOAPPEND NOPRINT FILE=example6b.fin
FORMAT=,1PE12.5 ONEHEADER
```

As of NM73, if an MDV is set to a value greater than or equal to 100, it is converted to that value minus 100 upon input, but will also not be used at all during estimation, only for table outputting. This option allows you to use the same enhanced data file for estimation and Table outputs, without significantly slowing down the estimation. So, the finedata control stream file would be:

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6.csv IGNORE=C

$FINEDATA TSTART=0 TSTOP=50 NEVAL=100 AXIS=TIME(LIN)
          CMT=1,3 MDV=101,101 FILE=example6b.csv
```

In the following example, TSTART, TSTOP, and NEVAL are obtained from columns TIMESTART, TIMESTOP, and NEVAL, respectively.

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT TIMESTART
 TIMESTOP NEVAL
$DATA example6c.csv IGNORE=C

$FINEDATA TSTART=TIMESTART TSTOP=TIMESTOP NEVAL=NEVAL AXIS=TIME(LIN) CMT=1,3
          FILE=example6d.csv
```

Multiple data sets may be processed by one finedata control stream file, by using $PROB records to separate the problems:

```
$PROB
$INPUT C=DROP ID TIME CMT OBSV DV COHT EVID AMT DOSE MDV
$DATA mydata.csv  IGNORE=C
$FINEDATA tstart=0 TSTOP=700 NEVAL=500 AXIS=TIME(LIN) CMT=1,4
          file=mydata_fine.csv

$PROB
$INPUT C=DROP ID TIME CMT OBSV DV COHT EVID AMT DOSE MDV
$DATA mydatab.csv  IGNORE=C
$FINEDATA tstart=0 TSTOP=700 NEVAL=500 AXIS=TIME(LIN) CMT=1,4
          file=mydatab_fine.csv
```

See also fine1, infn1, infn2 in the examples section of on-line help and guide VIII on using the INFN routine and finedata utility to create interpolated values.

As of NM74, additional records will be inserted at all end of infusion (AMT/RATE) and additional dose positions (II, ADDL), so that predicted values will appear in the tables at these positions of discontinuity. To prevent these additional records from being inserted, set FILLDOSE=0. The dose positions and end of infusion cannot take into account any dynamic model based time-lags (ALAG), fraction infused (Fn), model dependent rates (Rn), or other model dependent changes in the actual dose events. However, if you are using RATE=-2, so that you prefer to specify duration, and the duration value is static, then you may incorporate a data item that contains the static duration, and inform finedata of it, so it can calculate where end of infusion will be for a RATE=-2 condition. Furthermore, if your fraction infused/absorbed (Fn) is static, you can add a data item that contains the static fraction infused, so that finedata can calculate end of infusion. Keep in mind that the infusion period calculated by NONMEM is (AMT*Fn)/RATE, so finedata will use the same calculation. To specify these data items for finedata to use:
LAG=lag time data item
DUR=duration data item
BIO=fraction infused data item

Here is an example for giving additional information for dosing positions (multidose.ftl):

```
$PROB RUN# example6 (from r2compl)
$INPUT C SET ID OC2 TIME DV=CONC DOSE=AMT RATE EVID MDV CMT ADDL II PER FF
LAGTIME
$DATA multidose.csv IGNORE=C

$FINEDATA tstart=0 TSTOP=500 NEVAL=50 AXIS=TIME(LIN) CMT=1
          file=multidoseb.csv MDV=101 DUR=PER BIO=FF LAG=LAGTIME
```

A new option available as of NM74 is the EXTRADOSE record for finedata. EXTRADOSE adds additional non-observation dose records to those already existing, but it allows you to vary compartment number, or EVID, etc. For example, in example delayed.ftl:

```
$PROB Time delay problem
$INPUT ID TIME  DV AMT RATE CMT EVID MDV
$DATA delayed_pre.csv IGNORE=C

$EXTRADOSE  TSTART=0.0 TSTOP=50.0 AXIS=TIME CMT=3
          file=delayed.csv
```

all existing non-observation records (those for which EVID<>0) will be supplemented with additional records in which all items are identical, except the CMT will be 3, for the additional record. All other items in the record will remain unchanged from the original record for the four new records. Or,

```
$PROB Time delay problem
$INPUT ID TIME  DV AMT RATE CMT EVID MDV
$DATA delayed.csv IGNORE=C

$EXTRADOSE  TSTART=0.0 TSTOP=50.0 AXIS=TIME CMT=3,4,5,6 AMT=1,1,1,1
RATE=0,0,0,0 file=delayedb.csv
```

will set AMT to 1 and RATE to 0 for four new dose records with CMT numbers 3,4,5,6, consecutively. The EXTRADOSE has particular value when adding time-delay compartments to a model, to deal with time-delay problems.

I.84 doexpand Utility Program for Expanding Repetetive Code (NM74)

The utility program doexpand in the ..\util directory will expand an NM-TRAN control stream file that has been annotated with DOE (which stands for DO expand) and ENDDOE (which stands for ENDDO expand) directives. This is useful for repetitive code statements that differ only by increments of indices. For example, consider the following example for replicating derivative equations that differ only by their indices to DADT() and A(), when dealing with time delay problems (although see ddexpand utility next section for a complete delay equation system process):

```
$DES
DOE (I=1,3,2) (J=2,4,2) (K=11,14,3)
DADT(I) = -KEL[K]*A(I)
DADT(J) = K0*(1.0 + SMAX*A(I)/(SC50*V + A(I))) - K1*A(J) &
        + KS*A(J)*SIN(2.0d+00*PI*T/TDELAY)*TDELAY/(2.0d+00*PI)
ENDDOE
```

This is expanded by the doexpand utility to:

```
DADT(1) = -KEL11*A(1)
DADT(2) = K0*(1.0 + SMAX*A(1)/(SC50*V + A(1))) - K1*A(2) &
        + KS*A(2)*SIN(2.0d+00*PI*T/TDELAY)*TDELAY/(2.0d+00*PI)
DADT(3) = -KEL14*A(3)
DADT(4) = K0*(1.0 + SMAX*A(1)/(SC50*V + A(3))) - K1*A(4) &
        + KS*A(4)*SIN(2.0d+00*PI*T/TDELAY)*TDELAY/(2.0d+00*PI)
```

The DOE statement has index statements, of the form (VAR=START,END,STEP). In the above example, the variable I is replaced with starting value 1, ending value 3, by steps of 2. Index variable J is replaced with starting value 2, ending value 4, steps of 2, etc. The number of items in each list for each variable need not be the same. Thus, records between DOE and ENDOE are done twice, first for I=1, J=2, then for I=3,J=4. You may enter up to 200 index lists for a given DOE record. Furthermore, surround substitution variable with brackets [] when the variable is part of a variable name rather than an index, as shown in the above example for [K]. You may also nest DOE/ENDOE blocks. Negative steps are allowed, but END must then be less than START. You may use & to continue on the next line if needed. If END is not specified, END is assumed to be equal to START. IF STEP is not specified, it is assumed to be 1.

The command syntax is as follows:

doexpand original.ctl new.ctl

or

doexpand <original.ctl >new.ctl

It is new.ctl that is then to be submitted to NMTRAN, via the nmfe7* script. The do expand method may also be used when providing a series of transit compartment differential equations that differ only by their DADT() and A() indexing.

An alternative specification of indexing is to use key word REPS for number of repetitions, and STEP for step size, and then only the starting index need be specified for each index variable. For example,

```
DOE (I=1,3,2) (J=2,4,2)
```

could be written as

```
DOE (REPS=2) (STEP=2) (I=1) (J=2)
```

See the delayed* files in the example directory for a simple working example. The batch file delayed.bat contains the following commands:

```
doexpand <delayed_pre.ctl >delayed.ctl
finedata <delayed.ftl
call nmfe75 delayed.ctl delayed.res -prdefault
```

It first executes finedata on the instruction file delayed.ftl, to add an extra dose for the time delay compartment number 3 in data file delayed_pre.csv, produce the new data file delayed.csv. Next, doexpand acts on delayed_pre.ctl containing DOE/ENDDO records, to produce delayed.ctl. Then, the problem is executed using nonmem.

Yet another syntax that could be used is [I-1], [I+2], etc., to incorporate simple offsets of an index (up to +/-100), rather than having to specify additional indixes for each offset. For example, in the previouse example, J is equivalent to I+1, so J can be replaced with [I+1]:

```
$DES
DOE (I=1,3,2) (K=11,14,3)
DADT(I) = -KEL[K]*A(I)
DADT([I+1]) = K0*(1.0 + SMAX*A(I)/(SC50*V + A(I))) - K1*A([I+1]) &
        + KS*A([I+1])*SIN(2.0d+00*PI*T/TDELAY)*TDELAY/(2.0d+00*PI)
ENDDOE
```

Another use for DOEXPAND is to prepare a partial differential equations (PDE) problem into a pure ordinary differential equations (ODE) problem. For example, diffusion type components of the form

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

with a second independent integration variable *x* is performed numerically by dinite difference expanding the right hand term. Suppose you want to expand it so that there are 20 steps (the more steps, the greater the accuracy), and the starting compartment at the left boundary is 5, and the ending compartment is at the right side of the "diffusion membrane" is modeled as compartment 25. Compartments 6 through 24 will be the intermediary diffusion compartments

```
…
N=20
kdel=N/L ; L is "width" of membrane (or some other appropriate meaning)
kdel2=kdel*kdel
DADT(5)={user model code}+kdel2*(A(6)-A(5))
```

```
DOE (I=6,24)
DADT(I) = kdel2*(A([I+1])-2.0*A(I)+A([I-1]))
ENDDOE
DADT(25)=kdel2*(A(24)-A(25))+{user model code}
…
```

Notice that the equation template is simply the finite difference approximation for second derivative assessment with respect to *x*. This accounts for bi-directional movement of material across the diffusion barrier, where

`A(I-1)-A(I)`

Accounts for forward movement, and

`A(I+1)-A(i)`

Accounts for backward movement. Together they are

`A(I+1)-2*A(I)+A(I-1)`

consistent with the code portion above.

Notice the similarity of diffusion with transit compartments, but with transit compartments, there is only forward movement:

`A(I-1)-A(I)`

Which would be expanded as:

```
…
N=20
kdel=N/L
DADT(5)={user model code}-kdel*A(5)
DOE (I=6,24)
DADT(I) = kdel*(A([I-1])-A(I))
ENDDOE
DADT(25)=kdel*A(24)+{user model code}
…
```

If the transit compartment were concieved as a one-directional transfer across some physiological region of some width L, then as number of divisions N increases, the transit compartment paradigm approaches

$$\frac{\partial u}{\partial t} = -\frac{\partial u}{\partial x}$$

since the A(I-1)-A(I) term represents a backward finite difference approximation of (negative) first derivative assessment.

Because there can be a variety of PDE expansion algorithms, one may wish to retain a repository of files with these expansions. For example, file difffusion may contain the line

```
DADT([I])=KDEL*( A([I-1]) - 2.0*A([I]) + A([I+1]) )
```

Which may be “include”d into a script:

```
DOE (I=6,24)
include diffusion
ENDDOE
```

The include file may contain more than one line (it may even reference include files, up to 20 nested deep). Only include statements within a DOE..ENDDOE loop will be opened and expanded by DOEXPAND.

The DOPDE Method of Modeling PDE's.

One can insert code that DOEXPAND can then expand to create more elaborate and systematic PDE systems, to incorporate one or more dimensions of flow and diffusion spaces. The ideas are taken from Schiesser [28].

Here are some commands:

DOPDE
This starts the pattern setup
ENDPDE
This ends the pattern setup

SIZE:n

Sets the number of compartments to describe the PDE process. Typically, this is L/dx, where L is the total length, and dx is the width that each compartment represents.

START:m
Then m+1 is the starting compartment for the PDE expansion.

MVAR:name
name is the variable name of the rate constant of transfer. So, if it is KDEL, then KDEL should be defined in the control stream before the DOPDE..ENDPDE segment. Default is KDEL.

MVARX:name
name is the variable name of the rate constant of transfer for a derivative reference. So, if it is KDELX, then KDELX should be defined in the control stream before the DOPDE..ENDPDE segment. Default is KDELX.

OFFSET:n
n is the offset compartment to the pattern. (This is explained as other commands are described).

m:a,b,c,d,…

A list of index values for the mth PDE compartment's (that is, compartment START+m) coefficents.

R:a,b,c,d,…
A list of index values for the repeat compartments' coefficients.

NMj:a,b,c,d,e…
The list of indices for the N minus jth compartment

N:a,b,c,d,e…
The list of indices for the Nth compartment.

UX:a,b
The a is the coefficient for du/dt coefficient (to which MVARX associates as a rate constant) of the first compartment. Set to 0 if not needed. The b is the coefficient for du/dt coefficient (to which MVARX associates as a rate constant) of the nth compartment. Set to 0 if not needed.

PATTERN:dss044_00

A pattern file, in this case, dss044_00, is "included", which contains OFFSET, 1, 2, NM2. NM1, N, R, UX descriptions for a particular purpose. In the case of dss044_00, this contains the following pattern:

```
OFFSET:-5
1: 0,0,0,0,0,45,-154,214,-156,61,-10
2: 0,0,0,0,10.0,-15,-4,14,-6,1
R: 0,0,0,-1,16,-30,16,-1
NM1:0,1,-6,14,-4,-15,10
N:-10,61,-156,214,-154,45
```

This is the pattern for coefficients to A() states for the first DADT() term, second DADT() term, repeat DADT() terms (3rd through N-2), the N-1th (NM1) DADT() term, and Nth DADT() term, in the PDE expansion. This pattern matches what is done in the dss044.for file by Schiesser:

```
      SUBROUTINE DSS044(XL,XU,N,U,UX,UXX,NL,NU)
      DX=(XU-XL)/DFLOAT(N-1)
      R12DXS=1./(12.0D0*DX**2)
C...     WITHOUT UX (EQUATION (53))
         IF(NL.EQ.1)THEN
         UXX(1)=R12DXS*
     1                (   45.0D0*U(1)
     2                  -154.0D0*U(2)
     3                  +214.0D0*U(3)
     4                  -156.0D0*U(4)
     5                   +61.0D0*U(5)
     6                   -10.0D0*U(6))
C...
C...     WITH UX (EQUATION (36))
         ELSE IF(NL.EQ.2)THEN
         UXX(1)=R12DXS*
     1                (-415.0D0/6.0D0*U(1)
     2                        +96.0D0*U(2)
     3                        -36.0D0*U(3)
     4                  +32.0D0/3.0D0*U(4)
     5                   -3.0D0/2.0D0*U(5)
     6                        -50.0D0*UX(1)*DX)
         END IF
C...
```

```
C...  UXX AT THE RIGHT BOUNDARY
C...
C...     WITHOUT UX (EQUATION (54))
         IF(NU.EQ.1)THEN
         UXX(N)=R12DXS*
     1                (   45.0D0*U(N  )
     2                  -154.0D0*U(N-1)
     3                  +214.0D0*U(N-2)
     4                  -156.0D0*U(N-3)
     5                   +61.0D0*U(N-4)
     6                   -10.0D0*U(N-5))
C...
C...     WITH UX (EQUATION (37))
         ELSE IF(NU.EQ.2)THEN
         UXX(N)=R12DXS*
     1                (-415.0D0/6.0D0*U(N  )
     2                        +96.0D0*U(N-1)
     3                        -36.0D0*U(N-2)
     4                  +32.0D0/3.0D0*U(N-3)
     5                   -3.0D0/2.0D0*U(N-4)
     6                        +50.0D0*UX(N  )*DX)
         END IF
         UXX(2)=R12DXS*
     1                (   10.0D0*U(1)
     2                   -15.0D0*U(2)
     3                    -4.0D0*U(3)
     4                   +14.0D0*U(4)
     5                    -6.0D0*U(5)
     6                    +1.0D0*U(6))
         UXX(N-1)=R12DXS*
     1                  ( 10.0D0*U(N  )
     2                   -15.0D0*U(N-1)
     3                    -4.0D0*U(N-2)
     4                   +14.0D0*U(N-3)
     5                    -6.0D0*U(N-4)
     6                    +1.0D0*U(N-5))
         DO 1 I=3,N-2
         UXX(I)=R12DXS*
     1                (   -1.0D0*U(I-2)
     2                  +16.0D0*U(I-1)
     3                  -30.0D0*U(I  )
     4                  +16.0D0*U(I+1)
     5                   -1.0D0*U(I+2))
1        CONTINUE
       RETURN
       END
```

So let's consider the pattern for the first DADT() equation in the PDE pattern:

```
1: 0,0,0,0,0,45,-154,214,-156,61,-10
```

Let us suppose that START=0, then the first DADT() is one after that, DADT(1). Because OFFSET=-5, the first coefficient is for 5 compartments before 1, A(1-5+0). Since this compartment should not exist for the pattern, its coefficient is 0. Not until we get to the 6th coefficient, belonging to compartment A(1-5+5)=A(1), should the pattern for DADT(1) begin,

and it is the value 45, just as in the dss044.for source code. The coefficient to A(2) for the DADT(1) equation is then -154, etc. Thus, DADT(1) with its linear combination of A() states, will be constructed according to dss044.for (using the without UX(1) option). The 2: line contains coefficients appropriate for DADT(2).

```
2: 0,0,0,0,10.0,-15,-4,14,-6,1
```

The first coefficient in the list belongs to A(2-5+0), which of course should be 0. Not until the 5th coefficient, A(2-5+4)=A(1), with value of 10, should the A() state pattern for DADT(2) begin, following the coefficients for UXX() in dss044.for.

The following are coefficients for N-1 and Nth derivatives respectively (where nth DADT() would be that of SIZE+START):

```
NM1:0,1,-6,14,-4,-15,10
N:-10,61,-156,214,-154,45
```

The reason for OFFSET needing to be -5 is to accommodate the six non-zero coefficients required for describing the nth DADT (for A() states N-5, N-4, N-3, N-2, N-1, ande N).

The coefficeints to the internal derivatives (so, derivatives 3 through N-2) are described with the R: record

```
R: 0,0,0,-1,16,-30,16,-1
```

These are the coefficients in the DO LOOP in dss044.for.

If you want to use the pattern in which UX(1) and UX(N) are used (condition NL=1, and NU=1 in the dss044.for code), then this would be as shown in dss044_11:

```
OFFSET:-5
UX:-50,50.0
1: 0,0,0,0,0,69.1666666666667,96.0,-36.0,10.66666666667,-1.5
2: 0,0,0,0,10.0,-15,-4,14,-6,1
R: 0,0,0,-1,16,-30,16,-1
NM1:0,1,-6,14,-4,-15,10
N:0,-1.5,10.66666666667,-36.0,96.0,69.1666666666667
```

Notice the coefficients for the UX(1) and UX(N) term are listed as first and second coefficients listed in the UX: record, respectively.

The various PDE patterns from Schiesser [28] are in the files in ..\util\pde directory, given as dss020, dss036, etc. Please read the comment in the respect .for files and reference [28] to understand the basis of each of these patterns.

Let's consider some combination patterns. Consider the file flowxdiffy.txt, so named because it describes flow (transit compartment transfer) in the x direction, and no cross transiting of diffusion along the y (or radial direction if a tube), but just parallel flow:

```
; flow through a tube (or blood vessel), in x direction, with no difussion
; in y (or radial direction from center).  In x direction, there is
```

```
; a pre-compartment 1, and postcompartment 52.
DOPDE
; X direction, non-boundaries, so connects with compartment 1, ends
; with compartment 2+10*5=52
; Classic diffusion dudt=d^2u/dx^2
SIZE:10
START:1
MVAR:KR
PATTERN:dss012nb ; transit compartment pattern

;y DIRECTION
; Classic diffusion dudt=d^2u/dx^2, boundaried
; No pattern, but there is number of compartments (SIZE) along
; y directionto be defined
SIZE:5
ENDPDE
```

To run this problem.

```
Doexpand flowxnodiffy.txt flowxnodiffy.out
```

And you will see the resulting code in flowxnodiffy.out, which could be cut and pasted into the $DES section of a control stream.

Another example is flowxdiffyz.txt, which has transit compartment type flow in the x direction, and diffusions along the y and z axes:

```
; flow through a tube, in x direction, with difussion
; in y and z directions also.  In x direction, there is
; a pre-compartment 1, and postcompartment 252.
; In y and z direction, there are boundaries.
DOPDE
; X direction, non-boundaries, so connects with compartment 1, ends
; with compartment 2+10*5*5=252
; Classic diffusion dudt=d^2u/dx^2
SIZE:10
START:1
MVAR:KR
PATTERN:dss012nb ; transit compartment pattern

;y DIRECTION
; Classic diffusion dudt=d^2u/dx^2, boundaried
SIZE:5
PATTERN:dss042_00

;z DIRECTION
; Classic diffusion dudt=d^2u/dx^2, boundaried
SIZE:5
PATTERN:dss042_00
ENDPDE
```

Finally, an example of diffusion in the x, y, and z directions is diffxyz.txt:

```
; diffusion through a tube, in x direction, with difussion
; in y and z directions also.  In x direction, there is
```

```
; a pre-compartment 1, and postcompartment 252.
; In y and z direction, there are boundaries.
DOPDE
; X direction, non-boundaries, so connects with compartment 1, ends
; with compartment 2+10*5*5=252
; Classic diffusion dudt=d^2u/dx^2
SIZE:10
START:1
MVAR:KR
PATTERN:dss042nb

;y DIRECTION
; Classic diffusion dudt=d^2u/dx^2, boundaried
SIZE:5
PATTERN:dss042_00

;z DIRECTION
; Classic diffusion dudt=d^2u/dx^2, boundaried
SIZE:5
PATTERN:dss042_00
ENDPDE
```

With the dss* pattern files, one can construct an elaborate combination of diffusion/flow patterns with any number of dimensions which can be inserted into the NONMEM control stream. As you can see, however, PDE equations can add many compartments to a problem, and therefore solving PDE's can potentially be very time consuming.

I.85 ddexpand Utility Program for Modeling Discrete Time Delays (NM74)

The utility program ddexpand in the ..\util directory will expand an NM-TRAN- template control stream to propagate and incoropporate delay differential equations (dde) by method of steps (MOS). The template should contain the following type of information, as exampled here with ra2.dde. The comments explain the different components of the dde system.

```
; Pre-Control stream template ra2.dde used by ddexpand program to
; form functional NMTRAN control stream ra2.ctl
$PROB DDE Problem: RA2

; the data file should have only DOSE input records pertaining to base
equations.
; Also, the CMT must be a data item
; The ddexpand program, using finedata's EXTRADOSE facility, will add doses
for
; additional compartments and call the new data file ra2_dde.csv
$INPUT ID TIME    AMT    RATE   CMT   EVID MDV  DV
$DATA ra2.csv IGNORE=C
$SUBROUTINES ADVAN13 TRANS1 TOL=12
$MODEL NCOMPARTMENTS=8 ; number of compartments will be adjusted by ddexpand

$PK
CEVID=EVID
IF(CMT/=1) CEVID=1
K10=THETA(1)+ETA(1)
K12=THETA(2)+ETA(2)
```

```
K21=THETA(3)+ETA(3)
V1=THETA(4)+ETA(4)
K1=THETA(5)+ETA(5)
K2=THETA(6)+ETA(6)
K4=THETA(7)+ETA(7)
K5=THETA(8)+ETA(8)
SIG1=THETA(9)+ETA(9)
SIG2=THETA(10)+ETA(10)
SIG3=THETA(11)+ETA(11)
;  TAU1, TAU2, TAU3,etc. are time delays.  This sample has one time delay,
TAU1
TAU1=THETA(12)+ETA(12)
I0=THETA(13)+ETA(13)
K3=5.0
AA=1.0
BB=0.5
; Set initial conditions for Base equations
A_0(1)=AA
A_0(2)=I0
A_0(6)=AA
A_0(7)=I0
;Any propagations of initial conditions and ALAG's will be placed here by
ddexpand program.

$DES
; AD_x_y is the State value of A(x) delayed for time TAUy.
; These are used in the differntial equations later on.
; AP_x_y is the State value of A(x) in the past, for time delay TAUy.
; That is, when T<Tauy, the AP_x_y defines A(x)
; For every AD_x_y used in the differential equations, there must be an
AP_X_Y defined.
; If past is constant, then it can be as simple as AP_x_y=Initial condition
constant
; (same value as A_0(x) is set to).
; Make sure AP_x_y is a function of T: do not use T-TAUy,
; as this will be done by the ddexpand program.
AP_1_1=AA*EXP(BB*T)
AP_6_1=AA*EXP(BB*T)
; BASE EQUATIONS ENTERED BY USER.  Note use of AD_1_1 and AD_6_1, which
warrants an expansion.
DADT(1)=K3-(K1/K2)*(1.0-EXP(-K2*T))*A(1)
DADT(2)=K4*A(1)-K4*AD_1_1
DADT(3)=K4*AD_1_1-K5*A(3)
CC=A(4)/V1
EFFECT=CC*(SIG1*EXP(-SIG2*CC)+SIG3)
DADT(4)=-K10*A(4)-K12*A(4)+K21*A(5)
DADT(5)=K12*A(4)-K21*A(5)
DADT(6)=K3-EFFECT*A(6) - K1/K2* &
(1.0-EXP(-K2*T))*A(6)
DADT(7)=K4*A(6)-K4*AD_6_1
DADT(8)=K4*AD_6_1-K5*A(8)
;Any delay equations necessary are placed here by the ddexpand program.

$ERROR
A1=A(1)
A2=A(2)
A3=A(3)
```

```
A4=A(4)
A5=A(5)
A6=A(6)
A7=A(7)
A8=A(8)
A9=A(9)
A10=A(10)
A11=A(11)
A12=A(12)

Y1=A(2)+A(3)
Y2=A(7)+A(8)
Y3=A(3)
Y4=A(8)
IF(CMT==1) IPRED=Y1
IF(CMT==2) IPRED=Y2
IF(CMT==3) IPRED=Y3
IF(CMT==4) IPRED=Y4
IF(CMT==1) Y=IPRED*(1.0+EPS(1))
IF(CMT==1) Y=IPRED*(1.0+EPS(2))
IF(CMT==2) Y=IPRED*(1.0+EPS(3))
IF(CMT==4) Y=IPRED*(1.0+EPS(4))

$THETA
0.32544   ; 1: K10
2.6496    ; 2: K12
2.5944    ; 3: K21
0.02645   ; 4: V
0.456     ; 5: K1
0.169     ; 6: K2
0.185     ; 7: K4
0.031     ; 8: K5
0.328     ; 9: SIG1
0.328     ; 10: SIG2
0.025     ; 11: SIG3
10.6      ; 12: TAU1
2.83      ; 13: I0

$OMEGA (0.0 FIXED)X13
$SIGMA (0.04)X4
$SIMULATION (567811 NORMAL) (2933012 UNIFORM) ONLYSIMULATION SUBPROBLEMS=1
$TABLE TIME Y1 Y2 Y3 Y4 EXCLUDE_BY CEVID NOAPPEND NOPRINT FILE=ra2.tab
```

Thus, the user need only populate the template file with the base equations above, then execute ddexpand as follows:

ddexpand ra2.dde ra2.ctl

The destination file ra2.ctl will be produced, and a finedata control stream file (fine.ftl) will be produced and submitted to finedata utility to add extra doses to the data file identified in the ra2.dde control stream (make sure finedata is in the execution directory or in the PATH system). The resulting nmtran ready file will be produced (ra2.ctl) for this example:

; Pre-Control stream template ra2.dde used by ddexpand program to form functional NMTRAN control stream ra2.ctl

```
; Pre-Control stream template ra2.dde used by ddexpand program to
; form functional NMTRAN control stream ra2.ctl

$PROB DDE Problem: RA2

; the data file should have only DOSE input records pertaining to base
equations.
; Also, the CMT must be a data item
; The ddexpand program, using finedata's EXTRADOSE facility, will add doses
for
; additional compartments and call the new data file ra2_dde.csv

$INPUT ID TIME    AMT    RATE   CMT   EVID MDV  DV

$DATA ra2_dde.csv IGNORE=C

$SUBROUTINES ADVAN13 TRANS1 TOL=12

$MODEL NCOMPARTMENTS=12 ; number of compartments will be adjusted by ddexpand


$PK
CEVID=EVID
IF(CMT/=1) CEVID=1
K10=THETA(1)+ETA(1)
K12=THETA(2)+ETA(2)
K21=THETA(3)+ETA(3)
V1=THETA(4)+ETA(4)
K1=THETA(5)+ETA(5)
K2=THETA(6)+ETA(6)
K4=THETA(7)+ETA(7)
K5=THETA(8)+ETA(8)
SIG1=THETA(9)+ETA(9)
SIG2=THETA(10)+ETA(10)
SIG3=THETA(11)+ETA(11)
;  TAU1, TAU2, TAU3,etc. are time delays.  This sample has one time delay,
TAU1
TAU1=THETA(12)+ETA(12)
I0=THETA(13)+ETA(13)
K3=5.0
AA=1.0
BB=0.5
; Set initial conditions for Base equations
A_0(1)=AA
A_0(2)=I0
A_0(6)=AA
A_0(7)=I0
;Any propagations of initial conditions and ALAG's will be placed here by
ddexpand program.


; INITIALIZING EQUATIONS FOR DDE COMPARTMENTS
```

```
  A_0(9)=AA
  A_0(12)=AA
  TAU_1=1*TAU1
  MTDIFF=1
  MTIME(1)=PASTZERO+TAU_1
    DTAU_1=0.0
    IF((TSTATE-PASTZERO)>=TAU_1) DTAU_1=1.0
    CTAU_1=0.0
    IF(TSTATE-PASTZERO>=TAU_1) CTAU_1=1.0
  ALAG9=TAU_1
  ALAG10=TAU_1
  ALAG11=TAU_1
  ALAG12=TAU_1


$DES
  ; AD_x_y is the State value of A(x) delayed for time TAUy.
  ; These are used in the differntial equations later on.
  ; AP_x_y is the State value of A(x) in the past, for time delay TAUy.
  ; That is, when T<Tauy, the AP_x_y defines A(x)
  ; For every AD_x_y used in the differential equations, there must be an
AP_X_Y defined.
  ; If past is constant, then it can be as simple as AP_x_y=Initial condition
constant
  ; (same value as A_0(x) is set to).
  ; Make sure AP_x_y is a function of T: do not use T-TAUy,
  ; as this will be done by the ddexpand program.
   AP_1_1=AA*EXP(BB*(T-TAU_1))
   AP_6_1=AA*EXP(BB*(T-TAU_1))
    AD_1_1=(1.0-DTAU_1)*AP_1_1+DTAU_1*A(9)
    AD_6_1=(1.0-DTAU_1)*AP_6_1+DTAU_1*A(12)
; BASE EQUATIONS ENTERED BY USER.  Note use of AD_1_1 and AD_6_1, which
warrants an expansion.
 DADT(1)=K3-(K1/K2)*(1.0-EXP(-K2*T))*A(1)
 DADT(2)=K4*A(1)-K4*AD_1_1
 DADT(3)=K4*AD_1_1-K5*A(3)
 CC=A(4)/V1
 EFFECT=CC*(SIG1*EXP(-SIG2*CC)+SIG3)
 DADT(4)=-K10*A(4)-K12*A(4)+K21*A(5)
 DADT(5)=K12*A(4)-K21*A(5)
 DADT(6)=K3-EFFECT*A(6)-K1/K2*(1.0-EXP(-K2*T))*A(6)
 DADT(7)=K4*A(6)-K4*AD_6_1
 DADT(8)=K4*AD_6_1-K5*A(8)
;Any delay equations necessary are placed here by the ddexpand program.

   DADT(9)=CTAU_1*(K3-(K1/K2)*(1.0-EXP(-K2*(T-TAU_1)))*A(9))
 CC_1=A(10)/V1
 EFFECT_1=CC_1*(SIG1*EXP(-SIG2*CC_1)+SIG3)
   DADT(10)=CTAU_1*(-K10*A(10)-K12*A(10)+K21*A(11))
   DADT(11)=CTAU_1*(K12*A(10)-K21*A(11))
   DADT(12)=CTAU_1*(K3-EFFECT_1*A(12)-K1/K2*(1.0-EXP(-K2*(T-TAU_1)))*A(12))
  ; FOR FINEDATA $EXTRADOSE: CMT=1:,9,4:,10,5:,11,6:,12


$ERROR
A1=A(1)
A2=A(2)
```

```
A3=A(3)
A4=A(4)
A5=A(5)
A6=A(6)
A7=A(7)
A8=A(8)
A9=A(9)
A10=A(10)
A11=A(11)
A12=A(12)

Y1=A(2)+A(3)
Y2=A(7)+A(8)
Y3=A(3)
Y4=A(8)
IF(CMT==1) IPRED=Y1
IF(CMT==2) IPRED=Y2
IF(CMT==3) IPRED=Y3
IF(CMT==4) IPRED=Y4
IF(CMT==1) Y=IPRED*(1.0+EPS(1))
IF(CMT==1) Y=IPRED*(1.0+EPS(2))
IF(CMT==2) Y=IPRED*(1.0+EPS(3))
IF(CMT==4) Y=IPRED*(1.0+EPS(4))


$THETA
0.32544    ; 1: K10
2.6496     ; 2: K12
2.5944     ; 3: K21
0.02645    ; 4: V
0.456      ; 5: K1
0.169      ; 6: K2
0.185      ; 7: K4
0.031      ; 8: K5
0.328      ; 9: SIG1
0.328      ; 10: SIG2
0.025      ; 11: SIG3
10.6       ; 12: TAU1
2.83       ; 13: I0


$OMEGA (0.0 FIXED)X13

$SIGMA (0.04)X4

$SIMULATION (567811 NORMAL) (2933012 UNIFORM) ONLYSIMULATION SUBPROBLEMS=1

$TABLE TIME Y1 Y2 Y3 Y4 EXCLUDE_BY CEVID NOAPPEND NOPRINT FILE=ra2.tab
```

Notice the expansion of appropriate equations. Also, notice that only the necessary equations are propagated, allowing conservation of memory and computation.

Delay differential equations and their propagation for their use with ODE's is a studied art, and the above example is not to be considered to be a complete description of this process. The user is expected to be knowledgeable in this art in order to use the utility properly.

There are some models, with feedback loops for example, in which the time delay process and their equations must be propagated or recursively applied to cover a certain simulation time. In this case, the user should supply commented lines containing the minimal TAU1, TAU2, etc., that are used in the base equations, and the maximal simulation time TSTOP to be used. For the example EPO, the epo.dde file contains at the beginning the expected tine values for several TAUs, and TSTOP time :

```
; Pre-Control stream template
;TAU1=75.0
;TAU2=42.0
;TAU3=147.0
;TAU4=114.0
;TAU5=1587.0
;TAU6=1554.0
;TSTOP=200.0
```

When the ddexpand utlity is applied, the original 6 differential equations will be expanded to 72 differential equations. It would be very tedious for the user to have to type in 66 additional equations that ddexpand does automatically, and with no typographical errors.

Certain sizing parameters may need to be modified to accommodate NMTRAN and NONMEM in producing and executing code (located in FSUBS.f90) of such extensive content. Some control stream statements to consider are as follows:

Second derivatives need not be produced if Laplace method is not used, or Laplace method NUMERICAL is used, so the following line may be added in the control stream:

```
$ABBR DERIV2=NO DERIV2=NOCOMMON
```

Furthermore, if only simulations are produced, and there are no steady-state conditions requested, the production of first derivatives may also be prevented, as follows:

```
$ABBR DERIV2=NO DERIV2=NOCOMMON DERIV1=NO
```

For sizing, the number of compartments may need to be increased

```
$SIZES PC=73
```

where PC must at least equal the number of compartments+1

The number of lag times and additional doses may need to be increased (as a rule of thumb, make it at least the number of compartments), for example:

```
$SIZES PAL=100
```

The number of basic plus additional PK parameters (for ODE ADVAN's, the number of user-defined parameters may be used) should be accounted for:
```
$SIZES PG=150
```

Finally, there are a few parameters that must be increased to allow NMTRAN enough memory to build the FSUBS.f90 routine. The size requirements for these are reduced when you turn off DERIV2 and/or DERIV1 in the manner shown above with the $ABBR statements. The default values of these parameters are listed in ..\resource\SIZES.f90.

```
$SIZES DIMNEW=1000
```
Related to number of intermediate variables, and can be incremented by 1000's, until NMTRAN no longer issues an error.

```
$SIZES DIMTMP=500
```
Related to the number of user-defined variables, and can be incremented by 1000's, until NMTRAN no longer issues an error.

```
$SIZES LNP4=4000
```
Related to generating the NMPRD4P common block module, and can be incremented by 1000's starting from its default of 4000, until NMTRAN no longer issues an error.

Please see ddexpand-page-2017.pdf for further descriptions of the example problems.

I.86 Using the Delay differential equation Solvers with the ddexpand program for Discrete Delay Problems (nm75)

The advantage to using ddexpand to add the additional ODE's inline in the NMTRAN control stream is that NMTRAN can create proper partial derivative terms that will make it useful for FOCE, Laplace, and ITS. The disadvantage is that if there is a recursive process, as with the EPO model, then there can be a large number of equations added, and this may slow down an estimation process. An alternative is to use a delay differential equation (DDE) solver, such as RADAR5 (jointly developed by Ernst Hairer and Nicola Guglielmi [30]) (ADVAN16) or DDE_SOLVER (ADVAN18) delay differential equation solvers (or RADAR5/ADVAN17 if there are equilibrium compartments). To prepare your control stream for these solvers, select ADVAN16, ADVAN17, or ADVAN18 in the $SUB record, then execute ddexpand (using example ..\examples\epod.dde):
ddexpand epod.dde epod.ctl

Results in the addition of the following setup equations in $DES

```
; Pre-Control stream template
;DDE
;TAU1=75.0
;TAU2=42.0
;TAU3=147.0
;TAU4=114.0
;TAU5=1587.0
```

```
;TAU6=1554.0
;TSTOP=200.0
;$SIZES MAXNDRS=3 ; SUGGESTED VALUE FOR MAXNRDS.  UNCOMMENT TO USE

$SIZES PC=80 DIMNEW=3000 DIMTMP=1000 PG=200
$PROB EPO
$ABBR FUNCTION DDEFUNC(VDDE,7)
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME DV EVID MDV  CMT
$DATA EPO.csv IGNORE=C
$SUBROUTINES ADVAN16  TOL=9 ATOL=9
$MODEL NCOMPARTMENTS=6

$PK
KON  =THETA(1)+ETA(1)
KOFF =THETA(2)
KEL  =THETA(3)
KPT  =THETA(4)
KTP  =THETA(5)
VP   =THETA(6)
KINT =THETA(7)
SMAX =THETA(8)
SC50 =THETA(9)
IMAX =THETA(10)
IC50 =THETA(11)
MCH  =THETA(12)
C0   =THETA(13)
RR0   =THETA(14)
KDEG =THETA(15)
RBC0 =THETA(16)
TP1  =THETA(17)
TP2  =THETA(18)
TRET =THETA(19)
TRBC =THETA(20)

;TAUy
TAU1=TP1+TP2
TAU2=TP2
TAU3=TP1+TP2+TRET
TAU4=TP2+TRET
TAU5=TP1+TP2+TRET+TRBC
TAU6=TP2+TRET+TRBC

RET0=TRET*RBC0/(TRET+TRBC)
RBCM0=RBC0-RET0
HB0=MCH*RBC0
AT0=KPT*C0*VP/KTP
RC0=KON*RR0*C0/(KOFF+KINT)
KEPO=KEL*C0*VP+KINT*RC0*VP
KSYN=KDEG*RR0+KINT*RC0
KIN=RET0/(TRET*(1+SMAX*RC0/(SC50+RC0))**2)

; Initial conditions for Base equations
A_0(1)=C0*VP
A_0(2)=AT0
A_0(3)=RR0
A_0(4)=RC0
```

```
A_0(5)=RET0
A_0(6)=RBCM0

  ETZ_1=ETA(1)

TSTOP=200.0
; INITIALIZING EQUATIONS FOR DDE COMPARTMENTS

$DES
; AD_x_y is the State value of A(x) delayed for time TAUy.
; AP_x_y is the State value of A(x) in the past, for time delay TAUy.
; PASTS
  TAU_1=TAU1
   AP_4_1=RC0
    DTAU_1=0.0
     IF((T-PASTZERO)>=TAU_1) DTAU_1=1.0
  VDDE(1)=TAU_1
  VDDE(2)=T
    VDDE(3)=4
  VDDE(4)=1
  VDDE(5)=1
  VDDE(6)=ETZ_1
  VDDE(7)=0
  AZ_4_1=DDEFUNC(VDDE)
    AD_4_1=(1.0-DTAU_1)*AP_4_1+DTAU_1*AZ_4_1
  TAU_14=TAU2
   AP_4_2=RC0
    DTAU_14=0.0
     IF((T-PASTZERO)>=TAU_14) DTAU_14=1.0
  VDDE(1)=TAU_14
  VDDE(2)=T
    VDDE(3)=4
  VDDE(4)=2
  VDDE(5)=1
  VDDE(6)=ETZ_1
  VDDE(7)=0
  AZ_4_2=DDEFUNC(VDDE)
    AD_4_2=(1.0-DTAU_14)*AP_4_2+DTAU_14*AZ_4_2
  TAU_15=TAU3
   AP_4_3=RC0
    DTAU_15=0.0
     IF((T-PASTZERO)>=TAU_15) DTAU_15=1.0
  VDDE(1)=TAU_15
  VDDE(2)=T
    VDDE(3)=4
  VDDE(4)=3
  VDDE(5)=1
  VDDE(6)=ETZ_1
  VDDE(7)=0
  AZ_4_3=DDEFUNC(VDDE)
    AD_4_3=(1.0-DTAU_15)*AP_4_3+DTAU_15*AZ_4_3
  TAU_21=TAU4
   AP_4_4=RC0
    DTAU_21=0.0
     IF((T-PASTZERO)>=TAU_21) DTAU_21=1.0
  VDDE(1)=TAU_21
  VDDE(2)=T
```

```
  VDDE(3)=4
VDDE(4)=4
VDDE(5)=1
VDDE(6)=ETZ_1
VDDE(7)=0
AZ_4_4=DDEFUNC(VDDE)
  AD_4_4=(1.0-DTAU_21)*AP_4_4+DTAU_21*AZ_4_4
TAU_22=TAU5
 AP_4_5=RC0
  DTAU_22=0.0
   IF((T-PASTZERO)>=TAU_22) DTAU_22=1.0
VDDE(1)=TAU_22
VDDE(2)=T
  VDDE(3)=4
VDDE(4)=5
VDDE(5)=1
VDDE(6)=ETZ_1
VDDE(7)=0
AZ_4_5=DDEFUNC(VDDE)
  AD_4_5=(1.0-DTAU_22)*AP_4_5+DTAU_22*AZ_4_5
TAU_23=TAU6
 AP_4_6=RC0
  DTAU_23=0.0
   IF((T-PASTZERO)>=TAU_23) DTAU_23=1.0
VDDE(1)=TAU_23
VDDE(2)=T
  VDDE(3)=4
VDDE(4)=6
VDDE(5)=1
VDDE(6)=ETZ_1
VDDE(7)=0
AZ_4_6=DDEFUNC(VDDE)
  AD_4_6=(1.0-DTAU_23)*AP_4_6+DTAU_23*AZ_4_6
TAU_1=TAU1
 AP_5_1=RET0
  DTAU_1=0.0
   IF((T-PASTZERO)>=TAU_1) DTAU_1=1.0
VDDE(1)=TAU_1
VDDE(2)=T
  VDDE(3)=5
VDDE(4)=1
VDDE(5)=1
VDDE(6)=ETZ_1
VDDE(7)=0
AZ_5_1=DDEFUNC(VDDE)
  AD_5_1=(1.0-DTAU_1)*AP_5_1+DTAU_1*AZ_5_1
TAU_15=TAU3
 AP_5_3=RET0
  DTAU_15=0.0
   IF((T-PASTZERO)>=TAU_15) DTAU_15=1.0
VDDE(1)=TAU_15
VDDE(2)=T
  VDDE(3)=5
VDDE(4)=3
VDDE(5)=1
VDDE(6)=ETZ_1
VDDE(7)=0
```

```
 AZ_5_3=DDEFUNC(VDDE)
   AD_5_3=(1.0-DTAU_15)*AP_5_3+DTAU_15*AZ_5_3
 TAU_22=TAU5
  AP_5_5=RET0
   DTAU_22=0.0
    IF((T-PASTZERO)>=TAU_22) DTAU_22=1.0
 VDDE(1)=TAU_22
 VDDE(2)=T
   VDDE(3)=5
 VDDE(4)=5
 VDDE(5)=1
 VDDE(6)=ETZ_1
 VDDE(7)=0
 AZ_5_5=DDEFUNC(VDDE)
   AD_5_5=(1.0-DTAU_22)*AP_5_5+DTAU_22*AZ_5_5
 TAU_1=TAU1
  AP_6_1=RBCM0
   DTAU_1=0.0
    IF((T-PASTZERO)>=TAU_1) DTAU_1=1.0
 VDDE(1)=TAU_1
 VDDE(2)=T
   VDDE(3)=6
 VDDE(4)=1
 VDDE(5)=1
 VDDE(6)=ETZ_1
 VDDE(7)=0
 AZ_6_1=DDEFUNC(VDDE)
   AD_6_1=(1.0-DTAU_1)*AP_6_1+DTAU_1*AZ_6_1
 TAU_15=TAU3
  AP_6_3=RBCM0
   DTAU_15=0.0
    IF((T-PASTZERO)>=TAU_15) DTAU_15=1.0
 VDDE(1)=TAU_15
 VDDE(2)=T
   VDDE(3)=6
 VDDE(4)=3
 VDDE(5)=1
 VDDE(6)=ETZ_1
 VDDE(7)=0
 AZ_6_3=DDEFUNC(VDDE)
   AD_6_3=(1.0-DTAU_15)*AP_6_3+DTAU_15*AZ_6_3
 TAU_22=TAU5
  AP_6_5=RBCM0
   DTAU_22=0.0
    IF((T-PASTZERO)>=TAU_22) DTAU_22=1.0
 VDDE(1)=TAU_22
 VDDE(2)=T
   VDDE(3)=6
 VDDE(4)=5
 VDDE(5)=1
 VDDE(6)=ETZ_1
 VDDE(7)=0
 AZ_6_5=DDEFUNC(VDDE)
   AD_6_5=(1.0-DTAU_22)*AP_6_5+DTAU_22*AZ_6_5

CC=A(1)/VP
AT=A(2)
```

```
 RR=A(3)
 RC=A(4)
 RET=A(5)
 RBCM=A(6)

 X1=1+SMAX*AD_4_1/(SC50+AD_4_1)
 X2=1+SMAX*AD_4_2/(SC50+AD_4_2)
 X3=1+SMAX*AD_4_3/(SC50+AD_4_3)
 X4=1+SMAX*AD_4_4/(SC50+AD_4_4)
 X5=1+SMAX*AD_4_5/(SC50+AD_4_5)
 X6=1+SMAX*AD_4_6/(SC50+AD_4_6)
 X0=1+SMAX*RC0/(SC50+RC0)

 I1=1-IMAX*(MCH*(AD_5_1+AD_6_1)-HB0)/(IC50+(MCH*(AD_5_1+AD_6_1)-HB0))
 I3=1-IMAX*(MCH*(AD_5_3+AD_6_3)-HB0)/(IC50+(MCH*(AD_5_3+AD_6_3)-HB0))
 I5=1-IMAX*(MCH*(AD_5_5+AD_6_5)-HB0)/(IC50+(MCH*(AD_5_5+AD_6_5)-HB0))



; BASE EQUATIONS.

 DADT(1)=KEPO-KON*CC*VP*RR+KOFF*RC*VP-(KEL+KPT)*CC*VP+KTP*AT
 DADT(2)=KPT*CC*VP-KTP*AT
 DADT(3)=KSYN-KON*CC*RR+KOFF*RC-KDEG*RR
 DADT(4)=KON*CC*RR-(KOFF+KINT)*RC
 DADT(5)=KIN*X1*X2*I1-KIN*X3*X4*I3
 DADT(6)=KIN*X3*X4*I3-KIN*X5*X6*I5
```

And, there are no new ODE equations added to the control stream. Instead, the calls to DDEFUNC provide the time-delays for the required state variables. The DDEFUNC routine calls algorithms in the DDE solver system to create the delayed versions of state variable given by VDDE(3), for delay time given by VDDE(1). The advantage to using RADAR5 is its use in those problems that would normally create a recursive scheme if they were expanded by method of steps, potentially resulting in hundreds of additional ODE's. With the RADAR5/ADVAN16 method, no additional ODE's are added to the control stream.

The ..\pr\RADAR5U.f90 (for ADVAN16 and ADVAN17) and ..\pr\DDESLVU.f90 (for ADVAN18) routines offer additional control. Please read ..\guides\manrad5-v2.pdf for RADAR5, and ..\guides\ddes_f90.pdf for DDE_SOLVER for details about the parameter settings, and optional routines that may be incorporated.

For example, to use DDESLVu.f90, make a copy of it in your run directory, and rename it, such as DDESLVU2.f90. Make the modifications, and then reference it in the $SUBR record:

$SUBR … OTHER=DDESLVU2.f90

Some parameters of interest regarding ADVAN16(17)/RADAR5 include those controlled by $SIZES record:
PAST_SIZE=4000 by default, which determines the number of detailed times points (and hence resolution) of the delay equation storage

MAXNRDS=PC by default, but can be set to actual number of delay compartments used, in oder to save memory.

These parameters can be set with $SIZES in the control stream file:
$SIZES MAXNRDS=2 PAST_SIZE=6000

ADVAN18/DDE_SOLVER operates with allocatable arrays, and does not require pre-sizing by the user.

If you are not interested in viewing the processed control stream file with the additional records added to interface with the DDE solver system, you can directly send your control stream file to the nmfe75 script, but add the -dde option. Examples in ..\examples\dde directory, such as epodd.ctl, and rad.ctl, can be directly submitted, as follows:

nmfe75 epodd.ctl epodd.res -prdefault –dde

and the problem will run. If you wish to see the processed control stream after the execution, this is file epodd.ctl_dde in your run directory. See ..\guides\advan16_dde.pdf for more information.

Sometimes you may wish to have the equations transition from the past to the present other than at time T=0. In this case, set the reserved variable PASTZERO to a non-zero (including negative) value:

```
$PK
PASTZERO=-10.0
```

For example, suppose you wanted to have 30 additional doses (for a total of 31) every 8 hours, followed by records sampling the decline in concentration after the last dose. Suppose also that you wanted the time of sampling begin at TIME=0, therefore the beginning of the dose would be at -240.0 hours (see example simpledii16_2):

CID	TIME	AMT	RATE	II	ADDL	CMT	EVID	MDV	DV
100	-240	100	0	8	30	1	1	1	0
100	-1.00E-06	0	0	0	0	1	0	0	1
100	0	0	0	0	0	1	0	0	1
100	1	0	0	0	0	1	0	0	1
100	2	0	0	0	0	1	0	0	1
100	3	0	0	0	0	1	0	0	1
100	4	0	0	0	0	1	0	0	1
100	5	0	0	0	0	1	0	0	1
100	6	0	0	0	0	1	0	0	1

When using ADVAN16, ADVAN17, and ADVAN18 the past would be any time before the first dose, so it would be -240.0 hours as well. In $PK, you specify this using the reserved variable PASTZERO:

```
$PK
PASTZERO=-240.0
```

As of nm75, the traditional ODE solvers ADVAN9, 13, 14, and 15 have a superimposed algorithm that stores state variables of past times, and interpolates them for the present time (using cubic hermite interpolation, as is done by Petzoldt and Soetaert for the R DeSolve packages), which allows you to use these in a similar manner as ADVAN16 to ADVAN18, without using method of steps. These superimposed algorithms on traditional ODE solvers are not as efficient as the specially designed solvers ADVAN16 to ADVAN18, and may require an additional increase in TOL of 1 or 2 in order to obtain the desired precision. In turn the ATOL may be increased by 1 or 2, or you may increase it even to ATOL=12-15, and this may provide significant improvement in the TOL as well, without too much increase in computation time. For example, suppose that you wish to have at least 6 digits precision in your results, so that normall you would set TOL=6 ATOL=6. For a -dde problem using ADVAN 9,13,14 or 15, you could set

```
$SUBROUTINES ADVAN13 TOL=8 ATOL=14
```

to achieve a 6 digit precision. A little trial and error is warranted with one iteration of an estimation step, to ensure there is not too much of a speed sacrifice.

To ensure that ddexpand does not create a method of steps (MOS) expansion, but a dde (DDE) expansion when using ADVAN 9, 13, 14, or 15, insert ;DDE as a comment line near the top of the control stream:

```
;DDE
…
$SUBROUTINES ADVAN13 TRANS1 TOL=8 ATOL=8
```

Then, execute in the same manner as you would with the dde solvers ADVAN16 to ADVAN18:

nmfe75 epodd.ctl epodd.res -prdefault –dde

Similarly, if you wish to have ADVAN16 to 18 utilize the MOS method instead of its native DDE method, you would insert ;MOS near the top of the control stream:

;MOS

Otherwise, if ;DDE or ;MOS is not placed in the control stream, the default action will be MOS for ADVAN's 9,13,14,15, and DDE for ADVAN's 16, 17, 18.

When using ADVAN14 or ADVAN15 in their DDE mode, and when analytical derivatives are being used (such as during ITS, FOCE, Laplace), and there is a random eta effect (inter-subject variability) on a TAU, then it is sometimes best to uncomment the lines

```
      NBASE=NTOTAL
      NSENSE=0
```

in CVODEU.f90 (for ADVAN14) or IDAU.f90 (for ADVAN15) to turn off the sensitivity equation facility, and add in the control stream an OTHER statement for this modified code:

```
$SUBROUTINES ADVAN14 TOL=6 ATOL=6 OTHER=CVODEU.F90
or
$SUBROUTINES    ADVAN15    TOL=6    ATOL=6         OTHER=IDAU.F90
```

Otherwise, the derivatives with respect to eta relating to TAU may be less accurately evaluated to a sufficient degree to affect optimization.

For the equilibrium compartment ADVAN's (ADVAN9, ADVAN15, ADVAN17), even when they are doing a non-equilbirum compartment problem, when ddexpand inserts code for the DDE mode, the start of the delay time is signaled to the solver with an addition MTIME() statement, such as:

```
$PK
…
MTDIFF=1
MTIME(1)=PASTZERO+TAU_1
 DTAU_1=0.0
 IF((TSTATE-PASTZERO)>=TAU_1) DTAU_1=1.0
```

For the other ADVAN's the following code is inserted in the $DES section instead:

```
$DES
DTAU_1=0.0
 IF((T-PASTZERO)>=TAU_1) DTAU_1=1.0
```

These decisions were made based on trial and error assessments of each of these ADVAN's abilities to properly calculate the derivative of TAUx with respect to ETAs. If the analysis, particularly FOCE/Laplace/ITS, tends to not converge efficiently, you may wish to over-ride these default decisions with ;MTDIFF or ;NOMTDIFF comment inserted near the top of the control stream. For example, to have the MTDIFF/MTIME() code used for ADVAN18, then add

```
;MTDIFF
```

And for quilibrium compartment ADVAN's, to not have the MTDIFF/MTIME() code inserted, add

```
;NOMTDIFF
```

The ddexpand program, when it adds code for DDE solvers, needs to know the number of etas used in the model. Usually it can determine this on its own, but if you are using symbolic names for eta indices (such as ETA(CL)), in the code, then it cannot interpret this. To assist ddexpand

in this, add the following commented line, preferably right after the ;DDE comment line, for example:

```
;DDE
;NETAS=6
```

When using delay solvers (;DDE), do not use the INITIALOFF feature in $MODEL compartment descriptions, unless the compartment remains off throughout for an individual for a given occasion, or the compartment is turned on (by an input dose) at the start (usually time=0) of the numerical integration, and stays on. For example, you may use it in data sets in which an individual and occasion (EVID=1 or EVID=4) is given a non-IV dose at time 0, or the individual/occasion is never given a non-IV dose.

I.87 Modeling Discrete Delays on General Variables Using the ddexpand program (nm751)

Sometimes you want to impose a direct delay on a general variable, rather than a state variable A(*). Consider the following case, where it is desired that kin be delayed by an amount TAU. You can readily delay the states that KIN depends on, in this case A(1) (distrib_delay\smax_discrete.ctl), creating kindelay as the TAU delayed version of kin:

```
;DDE
$PROBLEM Discrete Delay on Smax Drug Responsive Input function
$ABBR DERIV2=NO DERIV2=NOCOMMON DERIV1=NO
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_discrete.dat IGNORE=C
$SUBROUTINES ADVAN18 TOL=10 ATOL=10
$MODEL NCOMPARTMENTS=2

$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
TAU1 = aa*dexp(gamln(1.0+1.0/bb))
Vmax = 2
V50 = 3
V = 1

; Initial conditions
A_0(1)=0.0
A_0(2)=5.0

$DES
AP_1_1=0.0
 c = A(1)/V
 kin = kin0*(1.0+(Smax*c)/(SC50+c))
 cdelay=AD_1_1/V
 kindelay=kin0*(1.0+(Smax*cdelay)/(SC50+cdelay))
```

```
 DADT(1) = -(Vmax*A(1))/(V50+A(1))
 DADT(2) = kin - kindelay
$ERROR
A1=A(2)
Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))

$THETA
0.5
10.0
0.8
7.0
6.5

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1

$TABLE TIME KIN IPRED NOPRINT NOAPPEND FILE=smax_discrete.tab ONEHEADER
FORMAT=S1PE20.13
```

Alternatively, you could create an additional state variable that represents kin, such as A(3), but you would need to represent the derivative of kin with respect T, DADT(3). The following example does this (smax_discrete_kin.ctl):

```
;DDE
$PROBLEM Discrete Delay on Smax Drug Responsive Input function kin
$ABBR DERIV2=NO DERIV2=NOCOMMON
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_discrete.dat IGNORE=C
$SUBROUTINES ADVAN16 TOL=10 ATOL=10
$MODEL NCOMPARTMENTS=3

$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
Vmax = 2
V50 = 3
V = 1
TAU1 = aa*dexp(gamln(1.0+1.0/bb))
; Initial conditions
A_0(1)=0.0
A_0(2)=5.0
; KIN at time 0 actually contains the bolus dose contribution into A(1).
; So it is not kin0, but rather;
CCC = AMT/V
A_0(3)=kin0*(1.0+(Smax*ccc)/(SC50+ccc))

$DES
```

```
 c = A(1)/V
; DERIVATIVE OF A WRT T, WHICH EQUALS V TIMES DERIVATIVE OF C WRT T
 DADT1=-(Vmax*A(1))/(V50+A(1))
; DERIVATIVE OF KIN WRT C
 DKINC=KIN0*SMAX*SC50/(SC50+C)/(SC50+C)

; The past, is when T<0, so before the bolus dose into A(1) is given, which
is:
 AP_3_1=KIN0

 DADT(1) = DADT1
; DERIVATIVE OF KIN WRT C TIMES DERIVATIVE OF C WRT T YIELDS DERIVATIVE OF
KIN WRT T
 DADT(3) = DKINC*DADT1/V
; SUBMIT STATE VARIABLE A(3), WHICH REPRESENTS KIN, FOR WEIBULL-DISTRIBUTED
DELAY
 DADT(2) = A(3) - AD_3_1

$ERROR
CC = A(1)/V
kin = kin0*(1.0+(Smax*cc)/(SC50+cc))
A1=A(2)
KINCALC=A(3)

Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))

$THETA
0.5
10.0
0.8
7.0
6.5

$OMEGA (0.0 fixed)x5
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME KIN KINCALC IPRED NOPRINT NOAPPEND FILE=smax_discrete_kin.tab
ONEHEADER FORMAT=S1PE20.13
```

As a third option. if we use the DAE ADvans, such as ADVAN15 or ADVAN17, then we can equate the A(3) with kin, and have A(3) solved via an equilibrium status, and we will not need to manually create the derivative of kin with respect to T (smax_discrete_kin_dae.ctl):

```
;DDE
$PROBLEM Discrete Delay on Smax Drug Responsive Input function kin using dae
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_discrete.dat IGNORE=C
$SUBROUTINES ADVAN17 TOL=10 ATOL=10 ; OTHER=ddeslvu.f90
$MODEL NCOMPARTMENTS=3
       COMP(COMP1)
       COMP(COMP2)
       COMP(COMP3 EQUILIBRIUM)
$PK
```

```
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
TAU1 = aa*dexp(gamln(1.0+1.0/bb))

Vmax = 2
V50 = 3
V = 1

; Initial conditions
A_0(1)=0.0
A_0(2)=5.0



$DES
AP_3_1=kin0
 DADT(1) = -(Vmax*A(1))/(V50+A(1))
 DADT(2) = A(3) - AD_3_1

$AESINIT
INIT=0

$AES
 c = A(1)/V
 kin = kin0*(1.0+(Smax*c)/(SC50+c))
E(3)=A(3)-kin

$ERROR
A1=A(2)

Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))

$THETA
0.5
10.0
0.8
7.0
6.5

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1

$TABLE TIME KIN IPRED NOPRINT NOAPPEND FILE=smax_discrete_kin_dae.tab
ONEHEADER FORMAT=S1PE20.13
```

Although for the above simple example it was sufficiently easy to delay the source state variable A(1) to kin, there are problems in which the general variable relies on several state variables, each of which are to be delayed the same amount. Rather than evaluating for each of their

delayed values individually, and then constructing the final delayed variable, in such cases, using the DAE method to equate that general variable to an equilibirium compartment allows the DDE solver to calculate that delayed variable.

These techniques will also come in handy when we use the discrete differential equation solvers to solve distributed delay prtoblems, shown in subsequent sections.

I.88 Solving Distributed Delayed Output Rate Prolems Using Repetition Variables RPTO/RPTI (nmVI)

There are conditions in which one would like to model a delay in a distributed manner. In one special type of such problems, the derivative state variable is governed by a simple form of input rate minus output rate, in which the output rate is a delayed form of the input rate. The distribution of the delay times is governed by a probability density function *l*, so this can be mathematically described as

$$\frac{dN(t)}{dt} = k_{in}(t) - \int_0^\infty l(s)k_{in}(t-s)ds$$

The second term on the right hand side is the convolution integral, and it weights input rate delayed by a time s, kin(t-s), with the pdf function l(s), and then sums over all possible time delays, from s=0 to s=infinity. This differs from the discrete delay, where only a single delay time s=tau is considered. Mathematically, such a delay at a single time tau would be represented by the delta pdf, which equals 1 at s=tau, and 0 otherwise:

$$l(s) = \delta(s-\tau)$$
$$\delta(z) = 1, z = 0$$
$$= 0, z \neq 0$$

So that

$$\frac{dN(t)}{dt} = k_{in}(t) - \int_0^\infty \delta(s-\tau)k_{in}(t-s)ds = k_{in}(t) - k_{in}(t-\tau)$$

While this convolution method potentially can be used to solve discrete delay problems it would be far less efficient then using discrete delay solvers.

The state variable to a distributed delayed output rate problem type can be shown to be solved in the form of:

$$N(t) = \int_0^\infty \big(1 - L(s)\big)k(t-s)ds$$

where *L* is the cumulative distribution function (CDF), or integral, of *l*. In guide VIII, the Repetition Parameters I example shows how a convolution for this type of problem can be

performed in NONMEM, using RPTO and RPTI variables. Appendix M of the NONMEM Technical Guide further explains how the NONMEM RPTO/RPTI method can use repeated use of an ODE solver to evaluate the above equation. Here we show an example that uses the WEIBULL distribution on an input function governed by an Smax relation to the drug level in the central compartment (examples\distrib_delay\smax_weibull_rep.ctl):

```
$PROBLEM Weibull Distributed Delay on Smax Drug Responsive Input function
using NONMEM's repetition variables
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_weibull.dat IGNORE=C
$SUBROUTINES ADVAN13 TOL=6 ATOL=6 ; OTHER=ddeslvu.f90
$MODEL NCOMPARTMENTS=2

$INFN
 IF (ICALL.EQ.0) RPTO=1      ;enables use of repetition feature
$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
Vmax = 2
V50 = 3
V = 1
 IF (RPTI.EQ.0) TI=TIME
 IF (NEWIND.EQ.2) RPTO=-1

; Initial conditions
A_0(1)=0.0
A_0(2)=5.0

$DES
 c = A(1)/V
 kin = kin0*(1.0+(Smax*c)/(SC50+c))
  DD=0.0
  if(ti-t>=0.0) DD=EXP(-((TI-T)/BB)**AA) ; DD=1-WEIBULLCDF(TI-T)

 DADT(1) = -(Vmax*A(1))/(V50+A(1))
 DADT(2) = DD*(KIN-KIN0) ; A(2)=N(t)

$ERROR
A1=A(2)

Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))
A3=A(3)

$THETA
0.5
10.0
```

```
0.8
7.0
6.5

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)

$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME KIN IPRED NOPRINT NOAPPEND FILE=smax_weibull_rep.tab ONEHEADER
FORMAT=S1PE20.13
```

The advantage to using the NONMEM repetition variables method is that the distrubtional convolution of kin is exactly evaluated, whereas the distributed transit or distributed discrete methods described in the next sections only approximates these convolutions. The repetition variables method does add computation time, in that the numerical integration by the ODE solver is repeated from T=0 to T=TIME of each data record. The more data records in the data set, the greater the computation time. Appendix M of the NONMEM Technical Guide discusses this further, and why the disadvantage of the RPTO/RPTI method is that it cannot be used for general convolution problems, and therefore cannot be used for more general distributed delay problems, whereas distributed transit and distributed discrete delay methods, shown in the next section, can be used to solve more general convolution problems.

I.89 Transit Compartments for Gamma distributed Time Delays Using the ddexpand program (nm751)

There are conditions in which one would like to model a delay in a distributed manner for more general problems than the delayed output rate problem. One distribution that is popular is the gamma distribution that can be emulated by the the transit compartment method. This is done by adding several ODE's to the NMTRAN control stream, so that a state variable A(x) is filtered through a series of NU transit compartments in which the rate constant of transfer KTR is in one direction, much as is done when filtering a dose input through a series of transit compartments. The KTR_y is related to the mean transit time (MTT) of TAUy and NUy transit compartments as:

KTR_y=NUy/TAUy

This is modeled in the same way as the discrete delay, but instead of specifying AP_x_y and AD_x_y, you specify APT_x_y and ADT_x_y. The transit compartment method is equivalent to a gamma distribution when NUy is an integer. If NUy is a non-integer, the ddexpand utility will evaluate CEILING(NUy) transit compartments, and provide the linear weighted average of the CEILING(NUy) transit compartment and the (CEILING(NUy)-1)th transit compartment. For example, if NUy=2.75, then 0.25 of state variable to transit compartment (CEILING(NUy)-1) and 0.75 of transit compartment CEILING(NUy) will be added together, and serve as state variable ADT_x_y. While this is strictly speaking not the gamma distribution of NUy=2.75, the difference is only a few percent, and is very easy to calculate. Consider the following user-created nmtran control stream (distrib_delay\logt.ctl):

```
;DDE
;Define as a comment the largest size for NU1 that one is likely to need.
;ddexpand will create up to 10 transfer compartments for this example.
; Specify as ;NN or ;NU
;NN1=10

$PROBLEM LOGISTIC with fractional transit compartments; turn off
second derivative assessments, sometimes even 1st derivatives if only
simulating
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME PRDV DV EVID MDV
$DATA logt.dat IGNORE=C
$SUBROUTINES ADVAN15 TOL=10 ATOL=10 ; OTHER=ddeslvu.f90
$MODEL NCOMPARTMENTS=1

$PK
CALLFL=-2
MXSTEP=2000000000
KG=THETA(1)
Y0=THETA(2)
YSS=THETA(3)
TAU1=THETA(4)
; The NU1 specified here should be no greater than the NN1 transit
compartments
; defined in the comment earlier.
NU1=2.5

; Initial conditions
A_0(1)=Y0

$DES
; AD_1_1 is the State value of A(1) delayed for time TAU1.
; AP_1_1 is the State value of A(1) in the past, for time delay TAU1.

 APT_1_1=Y0
 DADT(1)=KG*(1.0-ADT_1_1/YSS)*A(1)

$ERROR
A1=A(1)

Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))

$THETA
0.2D+00
1.0D+00
10.0D+00
5.0D+00

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)

$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME IPRED NOPRINT NOAPPEND FILE=logt.tab ONEHEADER FORMAT=S1PE20.13
```

The NN1 defines the number of compartments for the expansion, and TAU1 defines the time delay. The APT_x_y must be defined for transit compartments, even though the transit compartment system does not use past information (other than a constant past), because ddexpand uses the APT_x_y entry to identify the type of delay as the simple transit compartment system. The expansion by ddexpand results in:

```
;DDE
;Define as a comment the largest size for NU1 that one is likely to need.
;ddexpand will create up to 10 transfer compartments for this example.
; Specify as ;NN or ;NU
;NN1=10

$PROBLEM LOGISTIC with fractional transit compartments
; turn off second derivative assessments, sometimes even 1st derivatives if
only simulating

$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME PRDV DV EVID MDV
$DATA logt.dat IGNORE=C
$SUBROUTINES ADVAN15 TOL=10 ATOL=10 ; OTHER=ddeslvu.f90
$MODEL NCOMPARTMENTS=11


$PK
CALLFL=-2
MXSTEP=2000000000
KG=THETA(1)
Y0=THETA(2)
YSS=THETA(3)
TAU1=THETA(4)
; The NU1 specified here should be no greater than the NN1 transit
compartments
; defined in the comment earlier.
NU1=2.5

; Initial conditions
A_0(1)=Y0

  NU_1=NU1
  KTR_1=NU_1/TAU1
 XNU_1=MAX(INT(NU_1)+1,0)
 YNU_1=NU_1-XNU_1+1.0


$DES
  ; AD_1_1 is the State value of A(1) delayed for time TAU1.
  ; AP_1_1 is the State value of A(1) in the past, for time delay TAU1.


; DELAY SETUP FOR EQUATION SET 1
 APT_1_1=Y0
  TZ_GD=T-PASTZERO
  TZ_G=TZ_GD+TG_DEL
 T_O=0.0
  IF(TZ_GD>=0.0) T_O=1.0
```

```
  IF(T_O==0.0) TZ_G=1.0
 IX_=1
    SUM_1=0.0
IF(IX_>XNU_1) THEN
    DADT(2)=0.0
ENDIF
IF(IX_==XNU_1) THEN
   SUM_1=T_O*((1.0-YNU_1)*A(1)+YNU_1*(A(2)+APT_1_1))
ENDIF
IF(IX_<=XNU_1) THEN
  IX_=IX_+1
 DADT(2)=T_O*KTR_1*(A(1)-A(2)-APT_1_1)
ENDIF
IF(IX_>XNU_1) THEN
    DADT(3)=0.0
ENDIF
IF(IX_==XNU_1) THEN
   SUM_1=T_O*((1.0-YNU_1)*A(2)+YNU_1*A(3)+APT_1_1)
ENDIF
IF(IX_<=XNU_1) THEN
  IX_=IX_+1
    DADT(3)=T_O*KTR_1*(A(2)-A(3))
ENDIF
IF(IX_>XNU_1) THEN
    DADT(4)=0.0
ENDIF
…
IF(IX_>XNU_1) THEN
    DADT(11)=0.0
ENDIF
IF(IX_==XNU_1) THEN
   SUM_1=T_O*((1.0-YNU_1)*A(10)+YNU_1*A(11)+APT_1_1)
ENDIF
IF(IX_<=XNU_1) THEN
  IX_=IX_+1
    DADT(11)=T_O*KTR_1*(A(10)-A(11))
ENDIF
    ADT_1_1=(1.0-T_O)*APT_1_1+SUM_1

; DELAY EQUATIONS FOR EQUATION SET 0 (BASE EQUATIONS)
 DADT(1)=KG*(1.0-ADT_1_1/YSS)*A(1)


$ERROR
A1=A(1)

Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))

$THETA
0.2D+00
1.0D+00
10.0D+00
5.0D+00

$OMEGA (0.0 fixed)x4
```

```
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME IPRED NOPRINT NOAPPEND FILE=logt.tab ONEHEADER FORMAT=S1PE20.13
```

I.90 Creating Transit Compartments for Time Delays of any distribution, Using the ddexpand program (nm751)

A general method of modeling distributed delays (without having to use discrete delay solvers, which will be shown in the next example), in which any distribution function may be used, is the general distributed delay method, developed by Koch and Schropp [32]. In this case instead of using APT_x_y for the past, and ADT_x_y for the delayed state, the user should define in $DES the variables APG_x_y and ADG_x_y, respectively, for mean transit time TAUy and state variable A(x), and the ddexpand utility will then make the appropriate expansions.

Consider an example for a Webull distribution. The user-written control stream may look like this (distrib_delay\logdt.ctl):

```
;DDE
;NN1=28
;PRC1=0.01
;DISTRIB1=WEIBULLCDF(VG)

$PROBLEM LOGISTIC with distributed transit compartments
; turn off second derivative assessments, sometimes even 1st derivatives if
only simulating

$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME PRDV DV EVID MDV
$DATA logt.dat IGNORE=C
$SUBROUTINES ADVAN15 TOL=10 ATOL=10 ; OTHER=ddeslvu.f90
$MODEL NCOMPARTMENTS=1

$PK
CALLFL=-2
MXSTEP=2000000000
KG=THETA(1)
Y0=THETA(2)
YSS=THETA(3)
TAU1=THETA(4)
NU1=2.5

; Initial conditions
A_0(1)=Y0

$DES
; Define parameters to Weibull distribution again, for $DES record
VG(2)=NU1
VG(3)=TAU1/NU1
; ADG_1_1 is the State value of A(1) delayed for time TAU1, using a general
distribution.
; APG_1_1 is the State value of A(1) in the past, for time delay TAU1.
```

```
 APG_1_1=Y0
 DADT(1)=KG*(1.0-ADG_1_1/YSS)*A(1)

$ERROR
A1=A(1)

Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))

$THETA
0.2D+00
1.0D+00
10.0D+00
5.0D+00

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)

$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME IPRED NOPRINT NOAPPEND FILE=logdt.tab ONEHEADER FORMAT=S1PE20.13
```

Notice the beginning comment lines:

```
;NN1=27
;PRC1=0.01
;DISTRIB1=WEIBULLCDF(VG)
```

The number of transit equations to expand is NNy, for time delay y. Next, the precision PRCy is specified. If it is not specified, then 1/NNy precision is assumed. If ;PRC is specified without the y index, then the one specified by the most recent NNy is used. Finally the distribution function is specified (;DISTRIBy, again, with y optional, and by default that specified by the most recent NNy), along with an arbitrary vector name to use with it, in this case the vector name is VG. The user may use any of the following distribCDF functions, as defined in [32]:

WEIBULLCDF
GAMMACDF
LOGNORMALCDF
GOMPMAKECDF (Gompertz-Makeham)
WEIBULLCCDF (combined Weibull)
UNIFORMCDF

In theory, one can use any of the CDF versions of the distributions listed in section I.32 Probability Density Functions (NM742), but the CDF inverse functions (distribCDFINV) to many of these have not yet been constructed, which is required for this process.

An approximation of the convulution of the distribution with the input state variable A(x) is intended to be calculated. The integral of the CDF is to be evaluated at evenly spaced time intervals from s=0 to s= distribCDFINV(1-PRC1), spaced aprt by delta-s= distribCDFINV(1-PRC1)/NN1. Thus, the imprecision is about (MAX(1/NN1,PRC1). The lower the PRC1, and the higher the number of transit compartments in the expansion, the more precisely the convolution process is calculated. If you wish to minize the computational expense, and still

have a reasonable rough approximation to the distribution (and remember, the selected distribution function is in itself often arbitrary or empirically assessed and rarely associated with a mechanistic basis, so purity should not be a priority here), one could select NN1=10 and PRC1=0.01.

The user must have in his code the placement of the appropriate parameters to the distribution, into the vector name allocated to that distribution. In the above example, the vector is VG. As these CDF functions are structured for use with $ABBR FUNCTION (such as described for the probability densities listed in section I.32 Probability Density Functions (NM742)), indices starting at position 2 should have the appropriate parameter values specified (the first index is for the variable of integration t, and is filled in by ddexpand, as needed):

```
; Define parameters to Weibull distribution
VG(2)=NU1
VG(3)=TAU1/NU1
```

The above lines must be placed at beginning of $DES. The parameter locations for the distributions are ordered as follows, starting with index 2, using the parameter names listed in [32]:
WEIBULLCDF: a,b
GAMMACDF: a,b
LOGNORMALCDF: b,sigma
GOMPMAKECDF (Gompertz-Makeham): a,b,c
WEIBULLCCDF (combined Weibull): a1,b1,a2,b2,d
UNIFORMCDF: a,b

When the above control stream is submitted to ddexpand (using the -dde option to the nmfe script), we get the resulting control stream, which carries out the afore-mentioned convolution of the distribution with the intended state variable, with the histogramming resolution of NN1, and up to the time determined by PRC1:

```
;DDE
;NN1=27
;PRC1=0.01
;DISTRIB1=WEIBULLCDF(VG)

$PROBLEM LOGISTIC with distributed transit compartments; turn off
second derivative assessments, sometimes even 1st derivatives if only
simulating

$ABBR FUNCTION WEIBULLCDF(VG,10)

$ABBR FUNCTION WEIBULLCDFINV(*,10)

$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO

$INPUT ID AMT TIME PRDV DV EVID MDV

$DATA temp.dat IGNORE=C
```

```
$SUBROUTINES ADVAN15 TOL=10 ATOL=10 ; OTHER=ddeslvu.f90

$MODEL NCOMPARTMENTS=28


$PK
CALLFL=-2
MXSTEP=2000000000
KG=THETA(1)
Y0=THETA(2)
YSS=THETA(3)
TAU1=THETA(4)
NU1=2.5
; Define parameters to Weibull distribution
VG(2)=NU1
VG(3)=TAU1/NU1

; Initial conditions
A_0(1)=Y0


IF(NEWIND/=2) THEN
NNU_1=27
VG(1)=1.0-0.01
TTEND_1=WEIBULLCDFINV(VG)
IF(TTEND_1==-1.0D-300) EXIT 2
KK_1=NNU_1/TTEND_1
DTT_1=1.0/KK_1
ENDIF

; INITIALIZING EQUATIONS FOR DDE COMPARTMENTS


$DES
  ; Define parameters to Weibull distribution again, for $DES record
   VG(2)=NU1
   VG(3)=TAU1/NU1

; DELAY SETUP FOR EQUATION SET 1
 APG_1_1=Y0
  TZ_GD=T-PASTZERO
 T_O=0.0
  IF(TZ_GD>=0.0) T_O=1.0
  SUM_1=0.0
  TTAU_1=0.0
    DADT(2)=KK_1*(A(1)-A(2)-APG_1_1)
TTAU_1=TTAU_1+DTT_1
VG(1)=TTAU_1
ADDG_1=WEIBULLCDF(VG)
…
    DADT(27)=KK_1*(A(26)-A(27))
SUM_1=SUM_1+ADDG_1*(A(26)-A(27))
TTAU_1=TTAU_1+DTT_1
VG(1)=TTAU_1
ADDG_1=WEIBULLCDF(VG)
    DADT(28)=KK_1*(A(27)-A(28))
SUM_1=SUM_1+ADDG_1*(A(27)-A(28))
```

```
 ADG_1_1=SUM_1+A(28)+APG_1_1

 DADT(1)=KG*(1.0-ADG_1_1/YSS)*A(1)



$ERROR
A1=A(1)


Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))


$THETA
0.2D+00
1.0D+00
10.0D+00
5.0D+00

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME IPRED NOPRINT NOAPPEND FILE=logdt.tab ONEHEADER FORMAT=S1PE20.13
```

Sometimes you want to impose a distributed delay on a general variable, rather than a state variable A(*). To do this, set x=variablename for APG_x_y and APG_x_y. For example, in the following example, kin is to be convolved with a Weibull distribution (distrib_delay\smax_weibulldt.ctl):

```
;DDE
;NN1=100
;PRC=0.01
;DISTRIB=WEIBULLCDF(VG)

$PROBLEM LOGISTIC Distributed transit compartments on Smax Drug Responsive
Input function
…
$DES
VG(2)=AA
VG(3)=BB

 c = A(1)/V
 kin = kin0*(1.0+(Smax*c)/(SC50+c))
 APG_KIN_1=KIN0

DADT(1) = -(Vmax*A(1))/(V50+A(1))
DADT(2) = kin - ADG_KIN_1
…
```

The ADG_KIN_1 is the result of convolution of kin with pdf of Weibull.

You can optionally use a probability density function:

```
;DDE
;NN1=100
;PRC=0.01
;DISTRIB=WEIBULL(VG)
```

but generally results are better when using the CDF.

You may have a mixture of some states be general-distributed delayed (using APG_x_y and ADG_x_y), some states be discrete delayed (using AP_x_y and AD_x_y), and some states transit compartment delayed (using APT_x_y and ADT_x_y). The following is a portion of the epo problem, highlighting the delay modeling portion, in which ADG_4_1 is disributed delayed, and the others are discrete delayed.

```
;DDE
;NU1=10
;DISTRIB=WEIBULLCDF(VG)
$PROB EPO
$ABBR DERIV2=NO DERIV2=NOCOMMON DERIV1=NO

$INPUT ID AMT TIME DV EVID MDV  CMT
$DATA test_epo.csv IGNORE=C
$SUBROUTINES ADVAN16 TOL=10 ATOL=10
$MODEL NCOMPARTMENTS=6

$PK
...
TP1  =THETA(17)
TP2  =THETA(18)
TRET =THETA(19)
TRBC =THETA(20)
NUG=5
; All Nuy are defined here, but only NU1 was needed and needs to be defined.
NU1=NUG
NU2=NUG
NU3=NUG
NU4=NUG
NU5=NUG
NU6=NUG

;TAUy
TAU1=TP1+TP2
TAU2=TP2
TAU3=TP1+TP2+TRET
TAU4=TP2+TRET
TAU5=TP1+TP2+TRET+TRBC
TAU6=TP2+TRET+TRBC
...

$DES
 APG_4_1=RC0; Note the APG definition for distributed delay
 AP_4_2=RC0
 AP_4_3=RC0
 AP_4_4=RC0
 AP_4_5=RC0
```

```
 AP_4_6=RC0
 AP_5_1=RET0
 AP_5_3=RET0
 AP_5_5=RET0
 AP_6_1=RBCM0
 AP_6_3=RBCM0
 AP_6_5=RBCM0
 CC=A(1)/VP
 AT=A(2)
 RR=A(3)
 RC=A(4)
 RET=A(5)
 RBCM=A(6)
; Note the accompanying ADG usage for distributed delay
 X1=1+SMAX*ADG_4_1/(SC50+ADG_4_1)
 X2=1+SMAX*AD_4_2/(SC50+AD_4_2)
 X3=1+SMAX*AD_4_3/(SC50+AD_4_3)
 X4=1+SMAX*AD_4_4/(SC50+AD_4_4)
 X5=1+SMAX*AD_4_5/(SC50+AD_4_5)
 X6=1+SMAX*AD_4_6/(SC50+AD_4_6)
 X0=1+SMAX*RC0/(SC50+RC0)
 I1=1-IMAX*(MCH*(AD_5_1+AD_6_1)-HB0)/(IC50+(MCH*(AD_5_1+AD_6_1)-HB0))
 I3=1-IMAX*(MCH*(AD_5_3+AD_6_3)-HB0)/(IC50+(MCH*(AD_5_3+AD_6_3)-HB0))
 I5=1-IMAX*(MCH*(AD_5_5+AD_6_5)-HB0)/(IC50+(MCH*(AD_5_5+AD_6_5)-HB0))
 DADT(1)=KEPO-KON*CC*VP*RR+KOFF*RC*VP-(KEL+KPT)*CC*VP+KTP*AT
 DADT(2)=KPT*CC*VP-KTP*AT
 DADT(3)=KSYN-KON*CC*RR+KOFF*RC-KDEG*RR
 DADT(4)=KON*CC*RR-(KOFF+KINT)*RC
 DADT(5)=KIN*X1*X2*I1-KIN*X3*X4*I3
 DADT(6)=KIN*X3*X4*I3-KIN*X5*X6*I5
```

Although the discrete delays may be expanded by MOS (;MOS comment) or DDE (;DDE comment), expanding by ;DDE is generally more efficicent.

I.91 Using Discrete Delay Solvers for Time Delays of any distribution, Using the ddexpand program (nm751) and Built-IN DDEFUNCG Routine

So far, transit comaprtments were used to approximate delay compartments for use in distributed delays. You can use the direct delay DDE algorithms to perform a distributed delay as well, which create perfect discrete delay evaluations of the past on state variables A(), instead of using the transit compartment method. The advantage to using the DDE solvers for distributional delays is that no additional differential equations are added to the user's model system, and particularly with the built-in DDE solvers ADVAN16 to ADVAN18, are more efficient than the transit compartment distributed method. The user models direct delay solver method the same way you would distributed delays via transit comaprtments, with only one change required, and that is to specify the number of parameters to the distribution in the comment line of ;DISTRIB, for example:

```
;DISTRIB=WEIBULLCDF(VG,2)
```

And of course you must use one of the ODE solvers that are capable of discrete delay solving (ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAn17, ADVAN18). Doing so will allow the use of the discrete DDE solvers on NN1 evenly spaced times t-i*delta_tau, for delay times i=1 to NN1, weighted according to the CDF of the distribution. The advantage to this method is that there will not be NN1 differential equations added to the control stream. Instead, a built-in routine called DDEFUNCG will perform the calculations for the NN1 discrete delay states, and no added memory (except for ADVAN18) or added differential equations are needed. For ADVAN18, you may need to add $SIZES MAXNRDS=x, where x is the total delay times, which would be NN1 in this example.

One down-side to using the built-in DDEFUNCG routine and non-dae discrete delay solvers is that variables such as KIN cannot be directly delayed, and require access via an equivalent state variable A(), in the manner we showed earlier. For example, you would add something like:

DADT(x)={expression for derivative of kin with respect to time}.

In a manner that was shown earlier. In the example above, kin could be coded by the user as an additional state variable, A(3) (distrib_delay\smax_weibulldd.ctl):

```
;DDE
;NN1=100
;DISTRIB=WEIBULL(VG,2)

$PROBLEM LOGISTIC Distributed discrete delay on Smax Drug Responsive Input
function
; turn off second derivative assessments, sometimes even 1st derivatives if
 only simulating

$ABBR DERIV2=NO DERIV2=NOCOMMON
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_weibull.dat IGNORE=C
$SUBROUTINES ADVAN16 TOL=10 ATOL=10 OTHER=DDEFUNCG.F90

$MODEL NCOMPARTMENTS=3

$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
Vmax = 2
V50 = 3
V = 1

; Initial conditions
A_0(1)=0.0
A_0(2)=5.0
; KIN at time 0 actually contains the bolus dose contribution into A(1).
; So it is not kin0, but rather;
CCC = AMT/V
```

```
A_0(3)=kin0*(1.0+(Smax*ccc)/(SC50+ccc))

$DES
VG(2)=AA
VG(3)=BB
 c = A(1)/V
; DERIVATIVE OF A WRT T, WHICH EQUALS V TIMES DERIVATIVE OF C WRT T
 DADT1=-(Vmax*A(1))/(V50+A(1))
; DERIVATIVE OF KIN WRT C
 DKINC=KIN0*SMAX*SC50/(SC50+C)/(SC50+C)

; The past, is when T<0, so before the bolus dose into A(1) is given, which
is:
 APG_3_1=KIN0

 DADT(1) = DADT1
; DERIVATIVE OF KIN WRT C TIMES DERIVATIVE OF C WRT T YIELDS DERIVATIVE OF
KIN WRT T
 DADT(3) = DKINC*DADT1/V
; SUBMIT STATE VARIABLE A(3), WHICH REPRESENTS KIN, FOR WEIBULL-DISTRIBUTED
DELAY
 DADT(2) = A(3) - ADG_3_1


$ERROR
CC = A(1)/V
kin = kin0*(1.0+(Smax*cc)/(SC50+cc))
A1=A(2)
KINCALC=A(3)

Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))


$THETA
0.5
10.0
0.8
7.0
6.5

$OMEGA (0.0 fixed)x5
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1

;kin and kincalc will have identical values, as expected.
$TABLE TIME KIN KINCALC IPRED NOPRINT NOAPPEND FILE=smax_weibulldd.tab
ONEHEADER FORMAT=S1PE20.13
```

By default, the ddexpand routine will establish the most suitable quadrature method for performing the convolution integration. The default is as follows:

If DISTRIB/=GAMMA and DISTRIB/=UNIFORM, then the Gaussian-Laguerre Gaussian Quadrature (GQ) method with parameter SHP=-0.5 is used, in the gamma weight function

X**SHP*exp(-x). Furthermore, the PRC (see below) is used to determine the best scaling to cover the appropriate integral range. This has been found to be the most robust and efficient for most distributions. The default PRC is 0.0001, unless it is specified with ;PRCx

If DISTRIB=UNIFORM, then the Gauss=Legendre GQ method is used, with lower bound and upper bound being that of the (a,b) parameters of the uniform distribution modeled.

If DISTRIB=GAMMA, then the Gaussian-Laguerre GQ method, with SHP=NU-1 (where NU is the shape partameter of the modeled Gamma distribution), if NU>=0.1. The PRC precision is not used for specifying integral range, and instead the scaling using SCL=VG(3) is done on the default time positions (and therefore integral range) of the GQ method. The relationship between VG(3) and the resulting effective PRC is complex, and may result in insufficient integral range and therefore accuracy. It may be best to specify ;ALG=GQ(G,Y), where Y is a specified NU value, as described below, and PRC (default 0.0001 or that specified by ;PRC) will then be used to determine the best scaling. If NU<0.1, then this GQ method performs poorly, and instead the Gauss-Legendre method with domain range (0,1) is used on the quantile domain, using GAMMACDFINV for the re-mapping of the absicissas that the Gauss-Legendre algorithm produced.

Rarely does a user need to modify the default behavior. However, should it be necessary, the following Gaussian-Quadrature technigues are available, by specifying the algorithm as follows (to be placed after the ;DISTRIB line):

;ALGx=GQ({U/N/G},NU)

where x refers to the xth index TAU of the total model. By default, x is the most recent index specified by ;NNx. GQ refers to using a Gaussian Qudrature method. The arguments can be given in any order, as there is no risk of confusion. The Weighting function of the GQ method could be:

U=uniform=Gauss-Legendre
N=normal=Gauss-Hermite
G=Gamma=Gauss-Lagerre

The second aregument NU is optional, and would be valid only when G weighting function is used. In this case:

If NU is not specified, the default SHP is determined in the manner described earlier.

If NU is specified>0, then it is used.

If NU=0, then when DISTRIB/=GAMMA, the default alpha=-0.5 (NU=0.5) is used.

If U is selected, then the Gauss-Legendre->DISTRIBCDFINV(abscissas) process is used, where abscissas is the vector of positions calculated by Gauss-Legendre.

If N is selected, then the Gauss-Hermite->DISTRIBCDFINV(exp(abscissas)) is used. This provides poor results, even for the lognormal distribution, so it should not be used.

From experience, it appears that the default GQ method (so, do not specify ALG), with NNx=100, tends to very often provide a precision of the convolution equivalent to RTOL=0.01, ATOL=0.01, that is, 1% imprecision. An exception is for GAMMA with NU<=0.1, in which case NNx=150 may be required to get a 1% imprecision. A few trial and error simulations with varying NNx is sufficient to verify the imprecision, and that NNx can then be used for the entire analysis.

The Newton-Cotes quadratures may also be used, but they are not as efficient as the GQ methods. You may want to use Newton-Cotes to have more standard histogrammic representation of the convolution of limited resolution (NNx=10, say), to purposely have a discretized rendition of the convolution, perhaps to replicate a result from the literature, in which the convolution analysis may have been simplified (as mentioned earlier, if there is no good mechanistic reason to have a specific distribution, but merely to represent some empirical average life-span with some degree of variability, this could be sufficient). This syntax is as follows:

ALGx=NC(n,{T/Q},PRC,{P/C})

The arguments may be given in any order, as there is no risk of confusion. The Newton Cotes algorithms are as follows:
n=1: rectangular integration of steps of integration
n=2: trapezoidal rule
n=3: Simpson's rule
n=4: Simpson's 3/8 rule
n=5: Bode's rule
6<=n<=35: Newton-Cotes

{T/Q}: Select T for equally spaced time values throughout the integration, select Q for equally spaced quantile values (that is, for which CDF values are equally spaced (find taui such that CDF(taui)=i/m)). Default is Q.

PRC: Value between 0 and <1, to indicate that the integration should be performed from 0 to the 1-PRC quantile position, similar to what is done for the transit compartment distributed delay method described in the previous section. Default is 0.1/NNx for Newton-Cotes. The PRC may also be specified on a separate line with ;PRC.

{P/C}: Select to use CDF (C) of the distribution, or PDF (P). Default is C, and generally gives the best results. P can only be selected with T. Selecting P is risky, for pdf's that have a non-defined value, such as NU<1 for a GAMMA desnity.

When Q is selected and n is not specified, then default is n=5, as it appears to be the most accurate for most problems. When T is selected and n is not specified, then default n=3.

So:

ALGx=NC
chooses n=5, Q, C

ALGx=NC(T,0.001)
chooses 3, T, C, and PRC=0.001

ALGx=NC(2,T)

selects trapezoidal rule on equally spaced times, using CDF in the integrand.

ALGx=NC(3,T,P)
Selects Simpson's rule on equally spaced times, using pdf in the integrand.

We can alternatively use the equilibrium compartment method (using a dae dde solver) that we had shown earlier for a simple discrete delay problem, but now apply it for distributed delay as follows (distrib_delay\smax_weibulldd_dae.ctl):

```
;DDE
;NN1=100
;DISTRIB=WEIBULLCDF(VG,2)

$PROBLEM LOGISTIC Distributed discrete delay on Smax Drug Responsive Input
function using dae
$ABBR DERIV2=NO DERIV2=NOCOMMON
$INPUT ID AMT TIME DV EVID MDV
$DATA smax_weibull.dat IGNORE=C
$SUBROUTINES ADVAN17 TOL=10 ATOL=10
$MODEL NCOMPARTMENTS=3
       COMP(COMP1)
       COMP(COMP2)
       COMP(COMP3 EQUILIBRIUM)

$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
Vmax = 2
V50 = 3
V = 1

; Initial conditions
A_0(1)=0.0
A_0(2)=5.0
; KIN at time 0 actually contains the bolus dose contribution into A(1).
; So it is not kin0, but rather;

$DES
```

```
VG(2)=AA
VG(3)=BB
 APG_3_1=kin0

 DADT(1) = -(Vmax*A(1))/(V50+A(1))
 DADT(2) = A(3) - ADG_3_1

$AESINIT
INIT=0

$AES
 c = A(1)/V
 kin = kin0*(1.0+(Smax*c)/(SC50+c))
E(3)=A(3)-kin

$ERROR
A1=A(2)
Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))

$THETA
0.5
10.0
0.8
7.0
6.5

$OMEGA (0.0 fixed)x4
$SIGMA (0.0 fixed)
$SIML (122345) ONLYSIM SUBP=1

$TABLE TIME KIN IPRED NOPRINT NOAPPEND FILE=smax_weibulldd_dae.tab ONEHEADER
FORMAT=S1PE20.13
```

I.92 General Quadrature Integration Using ddexpand, and Routines GQ and NC

In the previous section, we used a packaged routine called DDEFUNCG that efficiently evaluates quadrature integration of distributed time delays using pre-packaged distribution functions. The advantage to this method is that the DDEFUNCG routine retains up to 1000 distribution values that needed to be calculated once for every delta-tau position, and the values were then stored for repeated used. This allows for more rapid evaluation. Furthermore, the coding of the convolution process is done for you. The down-side is that 1) only convolution integrations using pre-packaged distribtutions could be used as the kernel function, 2), the past (AP_x_y) could only be constant, and 3) the function being convolved needed to be defined in some way as a state variable A(x) (such as creating a state variable that represents kin in the previous section by specifying dA(x)/dt=dkin/dt, or using equilibrium compartment equations and solving a DAE, such as E(x)=kin). A more versatile method that can perform any quadrature integration of any equation set and has none of the afore-mentoned limitations, is to have the user code within NMTRAN the specific equations to be integrated, and using the DOWHILE feature, along with calling the quadrature routines GQ or NC directly.

Revisting the smax_weibulldd problem, we noticed that the state variable A(3) needed to be created that ultimately represented kin, since DDEFUNCG can only utilize time-delay state variables convolved with distribution functions. We can alternatively code the necessary integration more directly within NMTRAN, as shown in this example, smax_weic.ctl, but requiring the user to code more of the equations:

```
;DDE
$PROBLEM Discrete Delay on Smax Drug Responsive Input function
; turn off second derivative assessments, sometimes even 1st derivatives if
only simulating

$ABBR DERIV2=NO DERIV2=NOCOMMON DERIV1=NO

$INPUT ID AMT TIME DV EVID MDV

$DATA smax_gammat.dat IGNORE=C

$SUBROUTINES ADVAN16 TOL=14 ATOL=14

$MODEL NCOMPARTMENTS=2


$PK
CALLFL=-2
MXSTEP=2000000000
SMAX=THETA(1)
SC50=THETA(2)
KIN0=THETA(3)
AA=THETA(4)
BB=THETA(5)
Vmax = 2
V50 = 3
V = 1
TEND=BB*(-LOG(0.0001))**(1.0D+00/AA) ; Have TEND cover most of the integral

; Initial conditions
A_0(1)=0.0
A_0(2)=0.0


$DES
DADT(1)=0.0
DADT(2)=0.0


SUM=0.0
SUMW=0.0

;DOC 25 TS CS GQ(G,0.5,TEND)
TAU1=TS
AP_1_1=0.0
LL=AA/BB*((TS/BB)**(AA-1.0))*EXP(-((TS/BB)**AA))
CDELAY=AD_1_1/V
FF=KIN0*(1.0+(SMAX*CDELAY)/(SC50+CDELAY))
SUM=SUM+CS*LL*FF
```

```
SUMW=SUMW+CS*LL
;ENDDOC

ADC=SUM/SUMW


 C=A(1)/V
 DADT(1)=-(VMAX*A(1))/(V50+A(1))
 KINW=KIN0*(1.0+(SMAX*C)/(SC50+C))
 DADT(2)=KINW-ADC


$ERROR
CC=A(1)/V
A1=A(2)
Y1=1.0
IPRED=A(2)
Y=IPRED*(1.0+EPS(1))
kin = kin0*(1.0+(Smax*cc)/(SC50+cc))


$THETA
0.5
10.0
190.0
7.0
6.5


$OMEGA (0.0 fixed)x4

$SIGMA (0.0 fixed)

$SIML (122345) ONLYSIM SUBP=1

$TABLE TIME KIN IPRED NOPRINT NOAPPEND FILE=smax_weic.tab ONEHEADER
FORMAT=S1PE20.13
```

The ;DOC and :ENDDOC comment lines are interpreted by ddexpand (remember to have the –dde option switch on the nmfe7x script command line) as follows:

;DOCy tells ddexpand that this is the beginnning of DOWHILE loop y (if y is not present, it is assumed 1), and it ends at ;ENDDOCy

The next item is the number of points for the integration, which is 25 in this example.
The next item is the variable of quadrature integration, which is TS in this example.
The next item is the coefficient of quadrature integration, which is CS in this example.
The next item is the qudaraute method to be used, along with arguments. GQ or NC may be selected, with the following arguments:

GQ(L,A,B):
L=U=gaussian-legendre (uniform weighting) quadrature

A=beginning Time, B=ending time
L=G=gaussian-laguerre (gamma weigthing) quadrature
A=NU, B=ending time TEND. Beginning time is always 0.
L=N=gaussian-Hermite (normal function, times then exponentiated).
A=beginning Time, B=ending time

NC(x,A,B):
X=x-point Newton-Cotes method
A=beginning Time, B=ending time

Notice how the variable of integration TS and coefficient of integration CS are used in the accumulator variables SUM and SUMW. The FF variable is kin calculated using the state variable A(1) delayed by an amount TS (notice how TAU1 needs to be defined and associated with TS), LL is the Weibull density evaluated at time TS. The FF*LL is therefore the convolution summand, weighted with CS, the quadrature integration coefficient. Althoughg we are using Gauss-Laguerre, with gamma function weighting, the gamma function weighting is removed from the CS, so that the user can replace it with whatever density or kernel function LL is needed (in this case the Weibull density).

It is not essential to normalize the result by divding by SUMW, as SUMW should be nearly 1 if LL is properly a pdf, but it sometimes helps in accuracy.

If you are specifically convolving with the GAMMA distribution, you can use the Gauss-Laguerre's gamma function weighting to its full advantage as shown in the following example (logits_ddg.ctl):

```
;DDE
$PROBLEM LOGISTIC
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$INPUT ID AMT TIME PRDV DV EVID MDV
$DATA logistic6.csv IGNORE=C
$SUBROUTINES ADVAN16 TOL=6 ATOL=6
$MODEL NCOMPARTMENTS=1

$PK
MXSTEP=2000000000
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
MU_5=THETA(5)
KG=EXP(MU_1+ETA(1))
Y0=EXP(MU_2+ETA(2))
YSS=EXP(MU_3+ETA(3))
KK=EXP(MU_4+ETA(4))
NU1=EXP(MU_5+ETA(5))
A_0(1)=Y0

$DES
ADC=0.0
;DOC1 10 TS CS GQ(GAM,NU1,KK)
```

```
TAU1=TS
AP_1_1=Y0
ADC=ADC+CS*AD_1_1
;ENDDOC1

 DADT(1)=KG*(1.0-ADC/YSS)*A(1)

$ERROR
A1=A(1)
Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))

$THETA
-1.61764E+00
-2.40835E-02
 2.29145E+00
 -0.5
 1.098612289

$OMEGA BLOCK(3)
 8.45335E-03
 6.59591E-04  7.39132E-03
 1.01727E-03 -1.73680E-03  6.86871E-03

$OMEGA 0.01 0.01

$SIGMA
 3.23154E-03

;$SIML (122345) SUBP=1
;$EST METHOD=IMP AUTO=1 NOABORT PRINT=1

$EST METHOD=CHAIN FILE=temp14.ext ISAMPLE=-1000000000 TBLN=1 NSAMPLE=0

$EST METHOD=ITS INTERACTION NOHABORT SIGL=5 MCETA=10 NSIG=2 PRINT=1 NITER=0
CTYPE=3 OPTMAP=0 ETADER=0 FNLETA=0

$EST METHOD=ITS INTERACTION NOHABORT SIGL=5 MCETA=10 NSIG=2 PRINT=1 NITER=0
CTYPE=3 OPTMAP=1 ETADER=2 FNLETA=0
```

Make sure you specify “GAM” instead of “G” for the GC function. Notice that the CS coefficients already contain the gamma distribuition weighting, appropriate specifically for parameters NU1 and KK, so that the density need not be multiplied into the summand. Note also that the appropriate TEND will be determined based on NU1 and KK.

Supose it is desired to create quadrature positions along quantile positions of the distribution. This might be particularly useful if there may be singularities in the the pdf. The following does this for the gamma density (distrib_delay\logits_ddcdfinv.ctl):

```
;DDE
$PROBLEM LOGISTIC
$ABBR DERIV2=NO DERIV2=NOCOMMON ; DERIV1=NO
$ABBR FUNCTION GAMMACDFINV(VG,10)
```

```
$INPUT ID AMT TIME PRDV DV EVID MDV
$DATA logistic6.csv IGNORE=C
$SUBROUTINES ADVAN16 TOL=6 ATOL=6
$MODEL NCOMPARTMENTS=1

$PK
MXSTEP=2000000000
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
MU_5=THETA(5)
KG=EXP(MU_1+ETA(1))
Y0=EXP(MU_2+ETA(2))
YSS=EXP(MU_3+ETA(3))
KK=EXP(MU_4+ETA(4))
NU1=EXP(MU_5+ETA(5))
DEN=KK/EXP(GAMLN(NU1))
A_0(1)=Y0


$DES
VG(2)=NU1
VG(3)=KK
ADC=0.0
;DOC1 10 TSS CS GQ(U,0.0,1.0)
VG(10)=II_1
VG(1)=TSS
TS=GAMMACDFINV(VG)
TAU1=TS
AP_1_1=Y0
ADC=ADC+CS*AD_1_1
;ENDDOC1

 DADT(1)=KG*(1.0-ADC/YSS)*A(1)

$ERROR
A1=A(1)
Y1=1.0
IPRED=A(1)
Y=IPRED*(1.0+EPS(1))

$THETA
-1.61764E+00
-2.40835E-02
 2.29145E+00
 -0.5
 1.098612289

$OMEGA BLOCK(3)
 8.45335E-03
 6.59591E-04  7.39132E-03
 1.01727E-03 -1.73680E-03  6.86871E-03

$OMEGA 0.01 0.01
$SIGMA
 3.23154E-03
```

```
$EST METHOD=CHAIN FILE=temp14.ext ISAMPLE=-1000000000 TBLN=1 NSAMPLE=0
$EST METHOD=ITS INTERACTION NOHABORT SIGL=5 MCETA=10 NSIG=2 PRINT=1 NITER=0
CTYPE=3 OPTMAP=0 ETADER=0 FNLETA=0
$EST METHOD=ITS INTERACTION NOHABORT SIGL=5 MCETA=10 NSIG=2 PRINT=1 NITER=0
CTYPE=3 OPTMAP=1 ETADER=2 FNLETA=0
```

In the above example, the Gaussian-Legendre guadrature is selected, with returned time values TSS scaled between 0.0 and 1.0, the domain of a CDF inverse (or range of the CDF). These TSS are transformed into time positions TS using the GAMMACDFINV function. Because GAMMACDFINV is an expensive calculation, the routine has the option to store up to 1000 values, so every unique value needs to be calculated only once, and then recalled upon repeated use. Each unique result needs to be given a unique index, and a convenient one is the DOWHILE index II_y (ddexpand creates II_y for a given ;DOCy). IF NDIM of the VG() vector is at least 10, then the tenth argument (VG(10)) must contain an index number. Under these consitions, the GAMMACDFINV routine will test if any of the input variables for a given index II_y changed, and will re-calculate if it has changed, or recall from the internal array if the input variables have not changed. Other CDFINV functions such LOGNORMALCDFINV, GOMPMAKECDFINV, and WEIBULLCCDFINV, also have this efficiency method, with the unique index expected to be entered in the 10th element of the input array.

The integral need not be a classic convolution with a distribution. The following example shows the considerablke versatility of this method. The problem lympho.ctl is as follows:

```
;DDE
$SIZES MAXNRDS=2 PAST_SIZE=20000

$PROBLEM DDDE Model of Lymphocyte Trafficking
; turn off second derivative assessments, sometimes even 1st derivatives if
only simulating

$ABBR DERIV2=NO DERIV2=NOCOMMON DERIV1=NO
$ABBR FUNCTION GAMMACDF(GCDF,4)
$INPUT ID AMT TIME DV EVID MDV
$DATA lympho.dat IGNORE=C
$SUBROUTINES ADVAN16 TOL=6 ATOL=6
$MODEL NCOMPARTMENTS=3

$PK
CALLFL=-2
MXSTEP=2000000000
AB0=THETA(1)
SHAZ0=THETA(2)
KOUT=THETA(3)
U1SHAZ0=THETA(4)
NU=THETA(5)
IMAX=THETA(6)
IC50=THETA(7)
CL=THETA(8)
VP=THETA(9)

U2=(1-U1SHAZ0)/SHAZ0
KIN=AB0*KOUT*U2*SHAZ0
GNU=EXP(GAMLN(NU))
```

```
QQ=(U1SHAZ0)**(1.0/NU)
BETA=U2*QQ/(1.0-QQ)

; Initial conditions
A_0(1)=0.0
A_0(2)=AB0
A_0(3)=KIN/U2
TEND=-LOG(0.0001)/U2 ; TEND is calculated to be when exp(-u2*t)<=0.001, so
should be enough


$DES
DADT(1)=0.0
DADT(2)=0.0
DADT(3)=0.0
 CP=A(1)/VP
 EMAX=1.0-IMAX*CP/(IC50+CP)

ADC1=0.0
ADC2=0.0

;DOC1 30 TA CA GQ(U,0.0,TEND)
IUU=0.0 ; IUU=INTEGRATED U=U1+U2
DEN=GNU
;DOC2 30 TZ CZ GQ(U,0.0,TA)
TAU2=TA-TZ
IF(TAU2<0.0) TAU2=0.0
AP_1_2=0.0
CPD2=AD_1_2/VP
EMAX2=1.0-IMAX*CPD2/(IC50+CPD2)
U1TZ=BETA*((BETA*TZ)**(NU-1.0))*EXP(-BETA*TZ)
DEN=DEN-U1TZ
GCDF(1)=TZ
GCDF(2)=NU
GCDF(3)=BETA
DENZ=GNU-GAMMACDF(GCDF)
IUU=IUU+CZ*(EMAX2*U1TZ/DENZ+U2)
;ENDDOC2

GCDF(1)=TA
GCDF(2)=NU
GCDF(3)=BETA
DENZ2=GNU-GAMMACDF(GCDF)
U1=BETA*((BETA*TA)**(NU-1.0))*EXP(-BETA*TA)/DENZ2
SHAZ=EXP(-IUU)
TAU1=TA
AP_2_1=AB0
CPD=AD_2_1
ADC1=ADC1+CA*CPD*U1*SHAZ
ADC2=ADC2+CA*CPD*(U1*EMAX+U2)*SHAZ
;ENDDOC1

 DADT(1)=-CL*CP
 DADT(2)=KIN-KOUT*A(2)+KOUT*EMAX*ADC1
 DADT(3)=KOUT*A(2)-KOUT*ADC2

$ERROR
```

```
CC=A(1)/VP
Y1=1.0
IPRED=CC
A2=A(2)
A3=A(3)
Y=IPRED*(1.0+EPS(1))

$THETA
31.4
48.0
0.137
0.5
1.0
1.0
4.34
9471.0
70961.0

$OMEGA (0.0 fixed)x5
$SIGMA (0.0 fixed)

$SIML (122345) ONLYSIM SUBP=1
$TABLE TIME IPRED A2 A3 NOPRINT NOAPPEND FILE=lympho.tab ONEHEADER
FORMAT=S1PE20.13
```

It is beyond the present scope to explain the above model, but there are several features used here that have general applicability. Notice that Do while loop 2 (;DOC2) is nested within DO while loop 1 (;DOC1). When ddexpand expands these comment directives, it defines index variable II_y for DO While loop DOCy, and termination index NN_y. To distinguish the various do while activities, always have a unique y value for every ;DOCy definition in the control stream. Notice that each ;DOC has its own integration variable and coefficient of integration variable.

One caveat should be noted when using ;DOC (or more generally, DOWHILE). Only accumulator variables (those defined as value=value+…) have their analytical derivative dependencies carried over properly from one DOWHILE iteration to the next. So any dependency on a value of the previous iteration, such as TSPRE AND CSPRE in this example:

```
;DOC1 13 TS CSS NC(3,0.0,TEND)
…
DELT=TS-TSPRE
DELC=CS-CSPRE
IF(DELT/=0.0) ADC=ADC+CSS*DELC*AD_1_1/DELT
TSPRE=TS
CSPRE=CS
;ENDDOC1
```

will not have their derivatives properly calculated. Model instead as follows:

```
TSPRE=0.0
CSPRE=0.0
;DOC1 13 TS CSS NC(3,0.0,TEND)
…
```

```
DELT=TS-TSPRE
DELC=CS-CSPRE
IF(DELT/=0.0) ADC=ADC+CSS*DELC*AD_1_1/DELT
TSPRE=DELT+TSPRE
CSPRE=CSPRE+DELC
;ENDDOC1
```

I.93 lpcc: LinePrinter Controls Converter Program (NM75)

The utility program lpcc.f90 in the ..\util directory will convert the line printer control characters in the NONMEM report file into their proper actions. The line printer control characters are in the first character in each line, and at one time was interpreted by old 132 column line-printers.

The conversion is acted upon only between the lines
NM-TRAN MESSAGES
and just before
Stop Time:

as the lines before and after these do not use column 1 as a line-printer control column.

The code command characters are interpreted by lpcc as follows:

Code Option	Acting on Control character	Action
A0	+	Advance 0 lines (overprint on (merge with) previous line)
A1	space	Advance 1 line (no action is taken)
A2	0	Advance 2 lines (insert a blank line)
A3	-	Advance 3 lines (insert 2 blank lines) Not used in NONMEM
AS		Remove action character column once code is acted upon (But will not remove character in col 1 if not one of the above)
FF	1	Advance page (insert Form feed character)
FFA	1	Advance page and add blank line (insert Form feed with a next line)
FFA2	1	Advance page and add 2 blank lines
FF1	1	Replace ‘1’ as ‘ ‘ (interpret FF as A1)
FF2	1	Replace ‘1’ as ‘0‘ (interpret FF as A2)
FF3	1	Replace ‘1’ as ‘-‘ (interpret FF as A3)
ALL		FF,A0,A1,A2,A3
ALLS		FF,A0,A1,A2,A3,AS

After an action, the control character is replaced with a space. That space will be printed unless AS is requested. So AS shifts the line one column to the left.

An N followed by the code prevents that action from occurring.

Examples (options are acted upon in order):

lpcc example1.res example1.lst FF A0
Only replace '1' in column 1 with form feed, and merge line with '+' with previous line.

lpcc example1.res example1.lst ALL FFA
Act on all codes, and replace '1' with FF/next line.
lpcc example1.res example1.lst ALLS FFA
Act on all codes, replace '1' with FF/next line, and shift line one column to the left.

lpcc example1.res example1.lst ALLS NFF
Act on all codes except '1' (FF), and shift line one column to the left.

I.94 lblr: Insert User-Defined Labels in Additional Files (NM75)

The label substitution process (*NOSUB=0 (Default) (NM74)*in section I.30 General New Options for $ESTIMATION Record (NM73)., and section I.7 Expansions on Abbreviated and Verbatim Code and Other Items (NM72,NM73,NM74,NM75)) is normally done only in the NONMEM report file, but not in the additional files, such as the .ext, .cov, .coi, .phi, etc. These additional files need often be read by 3rd party software, which may rely on traditional names for thetas, omegas, sigmas, phi, eta, etc.

Nonetheless, one may wish to make label substitutions on these additional files, which can be done after the NONMEM analysis, using the label-replace utility, lblr.

The lblr utility is used as follows:
lblr myresult.ext myresult.res myresult_new.ext {S/T/C} {0/1/2}

where
myresult.ext =input file to have label substitutions done.
myresult.res=NONMEM report file containing label substitution patterns, or any file containing label substitution patterns such as:

```
THETA(1)=THETA(CL)
ETA(2)=ETA(ETV1)
EPS(1)=EPS(RSW)
```

myresult_new.ext is the new file with substitutions

Option 1 settings:
S=Expand and adjust with spaces (default)
T=Do not expand, and truncate label if necessary
C=Expand, and compress so there are no spaces (suitable only for comma delimited files)

Option 2 settings:
0=Full name of parameter type (THETA/OMEGA/SIGMA/ETA/EPS/PHI/ETC/PHC) (default)
1=two letter truncation of parameter type name (TH/OM/SG/ET/EP/PH/EC/PC)
2=single letter truncation of parameter type name (T/O/S/E/P/H/C/C)

Truncated parameter type name (1 or 2) works well with T, when no label expansion is permitted, and you want to make the label as compact as possible.

If the user specified NOSUB=1, then the thetas, omegas/etas, and sigmas/eps will not have been label substituted in the report file during the NONMEM execution. The user may wish to make the label substitution afterword with lblr. For example,

lblr myresult.res myresult.res myresult_new.res T 2
Notice in this example that the report file myresult.res serves as the input file, and the file from which to get label substitution pattern. If you also want to perform an lpcc, do lblr first, then the lpcc utility.

I.95 csvalign: Pad a csv File with Spaces to align columns (NM75)

The utility program csvalign.f90 in the ..\util directory will pad a csv file with spaces so that the columns align, and more easily readable.

csvalign source.csv output.csv {number of header lines to exclude}

For example, consider file 501.csv:

```
C,DataDesc:Problem3,,,,,,,
C,ID,TIME,DV,AMT,RATE,WT,AGE,SEX
.,1,0,0,1000,200,58.4,51,1
.,1,1,6.47,0,0,58.4,51,1
```

You may not want to include the header line in considering the maximum spacing needed, so execlude it:

csvalign 501.csv 504_new.csv 1

The result is:

```
C,DataDesc:Problem3,,,,,,,
C ,ID ,TIME ,DV     ,AMT  ,RATE ,WT   ,AGE ,SEX
. ,1  ,0    ,0      ,1000 ,200  ,58.4 ,51  ,1
. ,1  ,1    ,6.47   ,0    ,0    ,58.4 ,51  ,1
```

I.96 priorget: Transfer Results of an Analysis to NMTRAN Prior Information (NM75)

This utility extracts results from a raw output file (.ext) and .cov file into $PRIOR NWPRI code, for easy transfer of results of a previous analysis into informative prior values for a subsequent analysis.

Usage:
priorget root.ext root.cov myprior.ctl TBLN s/,/t

where root.ext and root.cov are input the files from a previous analysis.
TBLN is the table number, as these input files may have several tables to choose from.
Delimiter can be , t for tab, or s for space. Optional if delimiter is space.

The order of the parameters in the input files must be in the default TSOL (lower triangular) order.

Example:
priorget example3.ext example3.cov prior3.ctl 5 s

extracts information from table 5 of the population mixture problem example3, and places it in NMTRAN $PRIOR control stream format in the output file prior3.ctl

```
$PRIOR NWPRI

$THETAP
 ( 4.24157E+00 FIXED)
 (-2.29640E+00 FIXED)
 ( 4.26189E+00 FIXED)
 (-6.75464E-01 FIXED)
 ( 6.66665E-01 FIXED)

$SIGMAP BLOCK(1)
  1.02333E-02 FIXED

$OMEGAP BLOCK(2)
  4.61841E-02 FIXED
 -8.57594E-03  2.72391E-02

$OMEGAP BLOCK(2)
  4.93083E-02 FIXED
 -1.10257E-02  5.99622E-02

$THETAPV BLOCK(5)
  2.40775E-04 FIXED
 -5.00646E-05  1.51280E-04
  1.10075E-07  1.90557E-08  5.12763E-04
 -2.07126E-09  1.87141E-08 -1.13186E-04  6.00993E-04
 -1.01416E-09  5.36580E-09  2.29895E-09  2.38694E-08  7.40750E-04

;DF from OMEGA(1,1): 184.063854545076
;DF from OMEGA(2,2): 162.626638917147
$OMEGAPD 162.626638917147 FIXED

;DF from OMEGA(3,3): 92.4955765039376
;DF from OMEGA(4,4): 99.3586543336461
$OMEGAPD 92.4955765039376 FIXED

$SIGMAPD 1780.99669390981 FIXED
```

You can then patch (or "include") this code into your control stream for the subsequent analysis. Any fixed thetas will have $THETAPV variances with value 0, so you will need to modify this as needed. The degrees of freedom for Omegas and Sigmas ($OMEGAPD, $SIGMAPD) are calculated according to the formula described in section I.41 A Note on Setting up Prior Information. Be aware that the degrees of freedom may fall below the block dimension, or be above the total number of subjects, and the user may wish to modify the results accordingly before using them in an analysis. Results from priorget should always be treated as informational first, and then inspected and modified according to your intended purpose.

I.97 nmtemplate Utility Program (NM73)

The utility program nmtemplate in the ..\util directory will perform variable substitution on appropriately tagged control stream template files, and produce executable control stream files. The syntax is as follows:

nmtemplate source-template-file destination-file var1=val1 var2=val2 var3=val3 …

where var1=val1 is the variable name, and value to substitute in the template file. The variable var1 must in turn appear as <var1> in the template file, and is case sensitive. For example, consider the template file ..\util\nmtemp.nmt:

```
$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X SDIX SDSX
$DATA nmtemp2.csv IGNORE=C ACCEPT=(ID.EQ.<NMID>)


$SUBROUTINES ADVAN3 TRANS4
$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
IPRED=F
Y = F + F*EPS(1)

; Initial values of THETA
$THETA <TH1> <TH2> <TH3> <TH4>
$OMEGA BLOCK(4)
0.15
0.01 0.15
0.01 0.01 0.15
0.01 0.01 0.01 0.15
$SIGMA
(0.06 )

$ETAS (0)x4
$EST METHOD=1 INTERACTION FNLETA=2 MAXEVAL=0
$TABLE ID TIME DV IPRED CMT EVID MDV ETA1 ETA2 ETA3 ETA4 NOAPPEND NOPRINT NOTITLE FILE=nmtemp.tab
```

Note that <NMID> is to be replaced with a particular NONMEM ID number by nmtemplate, and the <THX> are to be replaced with specific values of thetas:

nmtemplate nmtemp.nmt nmtemp.ctl NMID=47 TH1=1.7 TH2=1.4 TH3=0.8 TH4=2.0

The resulting file nmtemp.ctl will have the various values substituted into the various <> placeholders, and is ready to be read by NMTRAN:
nmfe75 nmtemp.ctl nmtemp.res

In the above nmtemp.nmt example, because FNLETA=2, then NONMEM will simply evaluate the IPRED values using the inputted etas from the $ETAS record without performing an estimation. Another example template file is example6.nmt listed in the ..\util directory, that you may inspect for other ideas.

Actually, nmtemplate is a general variable substitution program, and can process any text file in the manner shown above. Consider a FINEDATA control stream file template (..\util\nmtemp.fnt):

```
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X
SDIX SDSX
$DATA nmtemp.csv IGNORE=C

$FINEDATA   AXIS=TIME(LIN)   TSTOP=<TSTOP>   TSTART=<TSTART>   NEVAL=<NEVAL>
FILE=nmtemp2.csv
```

in which the tstart, tstop, and neval parameters are to be inserted:

nmtemplate nmtemp.fnt nmtemp.fnd TSTART=0 TSTOP=100 NEVAL=200

resulting in the FINEDATA control stream file nmtemp.fnd:

```
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X
SDIX SDSX
$DATA nmtemp.csv IGNORE=C

$FINEDATA AXIS=TIME(LIN) TSTOP=100 TSTART=0 NEVAL=200 FILE=nmtemp2.csv
```

Note that only words that match the variable list at the nmtemplate command line, and have enclosing brackets <>, will be replaced with the suggested values. The values may also be text with no spaces in them.

These two scripts could be combined to provide a means of creating individual simulated curves. Consider the following DOS patch script (which could also be converted to an R/S-PLUS script or function), nmtemp.bat:

```
nmtemplate.exe nmtemp.fnt nmtemp.fnd TSTART=%1 TSTOP=%2 NEVAL=%3
finedata.exe nmtemp.fnd
nmtemplate.exe nmtemp.nmt nmtemp.ctl NMID=%4
$nmfe75.bat nmtemp.ctl nmtemp.res -prdefault
```

Where %1 through %4 are the DOS command line substitution parameters. So the script could be executed as follows:
Call nmtemp.bat 0 100 200 34

Then, a program such as R, S-PLUS, or S-ADAPT, can read in the results from nmtemp.tab, and plot them.

Another feature of nmtemplate is that the user may request a random number to be generated to serve as a value, by referring to ~R(a1,a2,a3). R(a1,a2,a3) is a special function of nmtemplate, which obtains a uniform random variate between a1 and a2. If a seed a3 is given that is not 0, it means to initialize the seed. The initialization should be done once in a series. For example:

The following line sets the seed:
```
nmtemplate wexample12.nmt dummy.ctl SAMPLE=~R(1,10000,113345)
```

with a throw-away result file dummy.ctl. Then one could perform a for loop in a DOS batch file to generate a series of control stream files with different starting seeds:
```
for /l %%n in (1,1,9) do nmtemplate wexample12.nmt wexample12_%%n.ctl SAMPLE=~R(%%n000,%%n999,0)
```

where `for /l %%n in (1,1,9)` is a DOS command generating n starting at 1, incrementing by 1, and ending at 9. When n=3, for example, `~R(%%n000,%%n999,0)` will be `~R(3000,3999,0)`, generating a random number between 3000 and 3999, to be substituted wherever <SAMPLE> shows up in the template file wexample12.nmt.

The template file wexample12.nmt may contain:
```
$EST METHOD=CHAIN FILE=wexample12.txt NSAMPLE=0 ISAMPLE=<SAMPLE>
```

and the resulting files wexample12_1.ctl through wexample12_9.ctl will contain random ISAMPLE values, such as:

wexample12_1.ctl:
```
$EST METHOD=CHAIN FILE=wexample12.txt NSAMPLE=0 ISAMPLE=1345
```

wexample12_2.ctl:
```
$EST METHOD=CHAIN FILE=wexample12.txt NSAMPLE=0 ISAMPLE=2456
```
wexample12_3.ctl:
```
$EST METHOD=CHAIN FILE=wexample12.txt NSAMPLE=0 ISAMPLE=3089
```

etc. It should be pointed out that this example, in which nmtemplate is used to create a random variable for substitution into ISAMPLE, can easily be done in NM73 using the ISAMPEND and SELECT=3 options for $EST METHOD=CHAIN or $CHAIN (see I.64 Method for creating several instances for a problem starting at different randomized initial positions: $EST METHOD=CHAIN and $CHAIN Records).

You may use STDIN for console input, and STDOUT for console output, useful for piping commands together, for example:

```
nmtemplate doetest.ctl STDOUT IMAX=3 JMAX=4 KMAX=14 | doexpand >doetestq.ctl
```

I.98 NEFF and NEFFI Utility Programs (NM74)

In MCMC sampling, often, the number of effective samples is desired to be determined, which takes into account the correlation between samples. You may collect 10000 samples during the stationary phase, but these may be only equivalent to 100 independent samples. The NEFF utility analyzes EXT files containing Bayesian samples generated from NONMEM. Samples from all trables in one or more files can be collected, and overall NEFF statistics will be generated. For each file, enter the minimum and maximum iteration number value. When finished entering file names, enter the word STOP (in uppercase). For example:

```
c:\nm74a6\util>neff
 ENTER EXT FILE name, MIN, MAX
stanrb_169.ext,0,10000
 stanrb_169.ext           0       10000
 ENTER EXT FILE name, MIN, MAX
stanrb_171.ext,0,10000
 stanrb_171.ext           0       10000
 ENTER EXT FILE name, MIN, MAX
STOP

PARAMETER     MEAN          STD            N         NEFF1   %NEFF1   NEFF2   %NEFF2 RHAT
THETA1       -2.505708E-03  4.006207E-03      4002    4002.0 100.00    4002.0 100.00 1.000
THETA2         3.68305      2.921375E-02      4002    2655.4  66.35    2801.9  70.01 1.000
THETA3        -5.01367      4.326550E-02      4002     368.8   9.22     406.7  10.16 1.006
THETA4       -0.988017      0.142392          4002    1843.7  46.07    1866.7  46.65 1.000
THETA5        -1.12213      0.162227          4002    1843.5  46.06    1885.3  47.11 1.000
SIGMA(1,1)     15.8272      0.265440          4002    4002.0 100.00    4002.0 100.00 1.000
OMEGA(1,1)    1.028170E-02  6.782039E-04      4002    2274.6  56.84    2492.6  62.28 1.000
OMEGA(2,1)    2.716656E-02  2.568944E-03      4002     303.5   7.58     512.5  12.81 1.009
OMEGA(2,2)    0.175062      1.291445E-02      4002     296.2   7.40     558.0  13.94 1.010
OMEGA(3,1)    7.886005E-04  1.690585E-03      4002     226.1   5.65     296.1   7.40 1.007
OMEGA(3,2)   -4.639637E-02  5.949785E-03      4002       7.7   0.19     153.9   3.85 1.087
OMEGA(3,3)    4.143005E-02  5.804516E-03      4002       5.2   0.13     149.4   3.73 1.134
MCMCOBJ        26650.1       473.698          4002       2.1   0.05      76.6   1.91 1.481
```

You may add a delimiter to the command line of neff, if for example, the input files are comma delimited:
neff ,

The NEFF1 statistics are based on the method of evaluation in the appendices of the STAN reference manual [20]. The NEFF2 statistics are based on Appendix A of [19].

If you wish to analyze individual parameters, for example phi values, then you will need to add additional lines in $PK or $PRED, using the BAYES_EXTRA_REQUEST signal (see section I.68 Obtaining individual predicted values and individual parameters during MCMC Bayesian Analysis, and example8 as well, near the end of this document), and output the desired individual parameters. For example:

```
$PRED
include nonmem_reserved_general
BAYES_EXTRA_REQUEST=1

…
(code here)
```

```
…
PHI1=ETA(1)+MU_1
PHI2=ETA(2)+MU_2
PHI3=ETA(3)+MU_3
IF(ICALL==1) THEN
" OPEN(UNIT=50,FILE='stanb_new296.eta')
" WRITE(50,'(A)') 'TABLE NO.     2: NUTS Bayesian Analysis: Phi Values'
" WRITE(50,'(A12,1X,A14,3(1X,A12))') 'ITERATION','ID','PHI1','PHI2','PHI3'
ENDIF
; GET THE FIRST RECORD OF EACH SUBJECT, AND ONLY WHEN IBMETHOD=EST_BAYES
; AND BAYES_METHOD=1 (NUTS),
; AND DURING THE STATIONARY ITERATIONS (ITER_REPORT>0)
IF(BAYES_EXTRA==1 .AND.IBMETHOD==EST_BAYES.AND.BAYES_METHOD==1. &
AND.NEWIND/=2.AND.ITER_REPORT>=0) THEN
" WRITE(50,'(I12,1X,F14.0,3(1X,1PG12.5))') ITER_REPORT,ID,PHI1,PHI2,PHI3
ENDIF
```

This will produce a file of the following structure:

```
TABLE NO.     2: NUTS Bayesian Analysis: Phi Values
   ITERATION             ID         PHI1         PHI2         PHI3
           0             1.  4.55347E-02 -0.10868      -4.4562
           0             2.  3.51615E-02 -0.38840      -4.4000
           0             3.  8.05289E-02 -0.18820      -4.2769
           0             4. -0.10113      -1.1717      -4.3652
           0             5. -6.03206E-02 -0.58953      -4.3818
           0             6. -9.53758E-02  -1.3678      -4.2215
           0             7. -6.60832E-02 -0.67833      -4.3358
           0             8. -7.32508E-02  -1.0046      -4.2898
           0             9.  2.12179E-02 -0.41888      -4.5667
           0            10.  0.13819     -0.20922      -4.3962
           0            11.  0.16236     -0.81138      -4.1605
           0            12.  5.78395E-02 -0.30828      -4.5920
           0            13.  5.00854E-02  -1.0830      -4.2070
```

Notice that there needs to be a table header with “TABLE NO. “ specified, followed by the column hear , the first two items begin iteration, then ID (or subject number NIREC), followed by a list of desired items, in accordance with the write statement in the control stream file.

A file of this structure can be read by the utility program NEFFI (for NEFF individual), for example (a list of table files may be given as with NEFF, ending with STOP):

```
neffi
 ENTER EXT FILE name, MIN, MAX
stanb_new296.ext,0,10000
STOP
```

The efficiency and correlation statistics will be printed out, with labels of type XXXX_itemname, where XXXX is the subject number.

I.99 Single-Subject Analysis using Population with Unconstrained ETAs (NM73)

By default, NONMEM performs single-subject analysis by supposing that the data of the entire data file is from one subject, implied by the lack of an ID item, and lack of a $SIGMA record, but presence of a $OMEGA record. The help manual demonstrates another means by which one

data file may contain data from all subjects to be separately analyzed, using ID item as a parsing parameter over multiple single-subject problems. The RECS=ID option is used for this purpose, as given by the following example, ..\examples\indestb.ctl:

```
$PROB  THEOPHYLLINE POPULATION DATA; Analysis of Individuals
; Modification of CONTROL5 control steam
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP RECS=ID
;RECS=ID:  Data set will be read until ID changes or end-of-file

$SUBROUTINES  ADVAN2

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=MEAN ELIMINATION RATE CONSTANT (1/HR)
;THETA(3)=SLOPE OF CLEARANCE VS WEIGHT RELATIONSHIP (LITERS/HR/KG)
;SCALING PARAMETER=VOLUME/WT SINCE DOSE IS WEIGHT-ADJUSTED
   CALLFL=1
   KA=THETA(1)
   K=THETA(2)
   CL=THETA(3)
   SC=CL/K

$THETA  (0.001,3) (0.001,.2) (0.001,.1)
$OMEGA .2
;For single subject data OMEGA is residual variance.

$ERROR
   Y=F+ERR(1)
;ERR must be used instead of EPS.

$EST MAXEVAL=450  PRINT=5

$COV SPECIAL MATRIX=R PRINT=E
;SPECIAL is required to obtain the variance-covariance matrix for single-subject data.

$TABLE ID DOSE WT TIME NOPRINT ONEHEADER FILE=indestb.tab NOTITLE

$TABLE ID KA K CL SC NOPRINT FIRSTONLY NOAPPEND FILE=indestb.par NOTITLE ONEHEADER

INCLUDE indestb.txt 11
; INCLUDE: Inserts copies of the file named indestb.txt for each additional individual.
```

which performs the analysis for the first subject, and the accompanying include file performs analysis on the subsequent subjects:

```
$PROB  THEOPHYLLINE POPULATION DATA; Analysis of Individuals
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP RECS=ID NOREWIND
;NOREWIND: data set will be read starting after the previous individual

$THETA  (0.001,3) (0.001,.2) (0.001,.1)
$OMEGA .2

;For single subject data OMEGA is residual variance

$EST MAXEVAL=450  PRINT=5

$COV SPECIAL MATRIX=R PRINT=E
;SPECIAL is required to obtain the variance-covariance matrix for single-subject data

$TABLE ID DOSE WT TIME NOPRINT FORWARD NOHEADER FILE=indestb.tab

$TABLE ID KA K CL SC NOPRINT FIRSTONLY FORWARD NOAPPEND NOHEADER
FILE=indestb.par
```

Another method now available in NM73 is for NONMEM to treat all the subjects as part of a population analysis, but if all OMEGA diagonals are set to 1.0E+06 FIXED, this is a key value to indicate to NONMEM that there is no population density constraint for etas associated with the posterior density, effectively making the posterior density strictly a data likelihood. In the following example, the indestb problem was restructured to implement this method, as shown here in ..\examples\indestm.ctl:

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP

$SUBROUTINES  ADVAN2

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=MEAN ELIMINATION RATE CONSTANT (1/HR)
;THETA(3)=SLOPE OF CLEARANCE VS WEIGHT RELATIONSHIP (LITERS/HR/KG)
;SCALING PARAMETER=VOLUME/WT SINCE DOSE IS WEIGHT-ADJUSTED
   CALLFL=1
   KA=THETA(1)+ETA(1)
   K=THETA(2)+ETA(2)
   CL=THETA(3)+ETA(3)
   SC=CL/K

$THETA (0.0 FIXED)X4
$OMEGA (1.0E+06 FIXED)X4
$ETAS  3 .08 .04 0.2

$ERROR
   W1=SQRT(ABS(THETA(4)+ETA(4)))
   IPRED=F
   Y=F+W1*EPS(1)

$SIGMA  (1.0 FIXED)

$EST  METHOD=1 INTERACTION LAPLACE MAXEVAL=0 PRINT=5 NOHABORT FNLETA=0 MCETA=1 NONINFETA=1
$TABLE          ID DOSE TIME DV IPRED W1 NOAPPEND NOPRINT FILE=INDESTM.TAB
$TABLE          ID KA K CL NOAPPEND FIRSTONLY NOPRINT FILE=INDESTM.PAR
```

Notice in the above example that OMEGA diagonals are set to 1.0E+06, telling NONMEM to report the objective function of each subject as a data likelihood, without an eta population density or an integral over all etas component added. This is called POPULATION WITH UNCONSTRAINED ETAS analysis, versus the standard SINGLE-SUBJECT or POPULATION, and will be labeled as such in the NONMEM report file under ANALYSIS TYPE. For this example, all thetas are fixed to 0 as well, so that the etas contain the full values of the individual parameters to which they are associated (KA, K, CL, and residual variance W1 squared). Since thetas are no longer in play in indestm, initial etas become relevant, so the $ETAS record is used to introduce them, and MCETA=1 assures that these initial etas (as well as etas=0) are tested at the beginning of the etas curve fitting (the MAP estimation) as viable starting positions. Also, since all of the traditional population parameters THETAS, SIGMAS, and OMEGAS are fixed, only a single evaluation (MAXEVAL=0) is necessary. To compare the results of indestm with those of indestb, note that the four etas in indestm.phi match with the final three theta parameters and OMEGA(1,1) listed in indestb.ext or indestb.res, and notice that the individual objective functions of subjects listed in indestm.phi match with the final objective function of each of the 12 single-subject analyses in indestb.ext. Furthermore, the variance-covariance etas (ETC(*,*)) listed in indestm.phi match with the variance-covariance of the thetas and OMEGA(1,1) in indestb.cov. The perfect match of the variance between indestm and

indestb was done by ensuring both performed 2nd derivative information matrix analyses, in indestm by selecting LAPLACE in the $EST step, and in indestb by selecting MATRIX=R in the $COV step.

One can also use the equivalent $EST statement (which can sometimes provide more accurate results, particularly if residual error is heteroscedastic):

```
$EST  METHOD=1 INTERACTION MAXEVAL=0 PRINT=5 NOHABORT FNLETA=0 MCETA=1 ETADER=3
```

What adds power to this technique over the typical single-subject analysis method is that some of the parameters may be shared. For example, in ..\examples\indestms.ctl, instead of each subject finding its own residual variance coefficient, a shared SIGMA(1,1) is estimated:

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME CP=DV WT
$DATA       THEOPP

$SUBROUTINES  ADVAN2

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=MEAN ELIMINATION RATE CONSTANT (1/HR)
;THETA(3)=SLOPE OF CLEARANCE VS WEIGHT RELATIONSHIP (LITERS/HR/KG)
;SCALING PARAMETER=VOLUME/WT SINCE DOSE IS WEIGHT-ADJUSTED
   CALLFL=1
   KA=THETA(1)+ETA(1)
   K=THETA(2)+ETA(2)
   CL=THETA(3)+ETA(3)
   SC=CL/K

$THETA (0.0 FIXED)X3
$OMEGA (1.0E+06 FIXED)X3
$ETAS  3 .08 .04

$ERROR
   IPRED=F
   Y=F+EPS(1)

$SIGMA 0.2

$EST  METHOD=1 INTERACTION LAPLACE MAXEVAL=9999 PRINT=1 NOHABORT FNLETA=0 MCETA=1
$TABLE          ID DOSE TIME DV IPRED NOAPPEND NOPRINT FILE=INDESTMS.TAB
$TABLE          ID KA K CL NOAPPEND FIRSTONLY NOPRINT FILE=INDESTMS.PAR
$COV MATRIX=R
```

Thus, while each subject finds its own K, KA, and CL in the form of unconstrained etas as is done in indestm.ctl, a single residual variance as SIGMA(1,1) is estimated across subjects for indestms. For this analysis, a re-iterative analysis to improve SIGMA must be performed, so MAXEVAL>0 must be set. Non-zero THETAS may also be introduced to provide additional shared parameters, as is done in standard population analysis.

Please note that when using this POPULATION WITH UNCONSTRAINED ETAS analysis, NM-TRAN still sees the data as population, and will declare it as such in its warning statements. NMTRAN/NONMEM process the problem as population, while the statistical algorithms treat the data as single-subject (at least concerning unconstrained etas), offering the best of both worlds. Thus, NONMEM is capable of parallelizing these problems. The traditional single-subject analysis, however, cannot be parallelized because NONMEM processes each subject in sequence.

I.100 References

[1] Hooker AC, Staatz CE, Karlsson MO. Conditional weighted residuals (CWRES): a model diagnostic for the FOCE method. Pharmaceutical research 2007; 24: 2187-97.
[2] Comets E, Brendel K, Mentre F. Computing normalized prediction distribution errors to evaluate nonlinear mixed effects models: the npde add-on package for R. Computer Methods and Programs in Biometrics 2008; 90:154-166.
[3] Brendel K, Comets E, Laffont C, Laveille C, Mentre´ F. Metrics for External Model Evaluation with an Application to the Population Pharmacokinetics of Gliclazide. Pharmaceutical Research, 2006; 23(9): 2036-2049.
[4] Nguyen THT, Comets E. Mentre´ F. Extension of NPDE for evaluation of nonlinear mixed effect models in presence of data below the quantification limit with applications to HIV dynamic model. J Pharmacokinet Pharmacodyn (2012) 39:499–518
[5] Press WH, Teukolsky SA, Vettering WT, Flannery BP. Numerical Recipes, The Art Of Scientifc Programming. 2nd Edition, Cambridge University Press, New York, 1992, pp. 269-305.
[6] Press WH, Teukolsky SA, Vettering WT, Flannery BP. Numerical Recipes, The Art Of Scientifc Programming. 2nd Edition, Cambridge University Press, New York, 1992, pp. 180-184.
[7] Almquist J, Leander J, Jirstrand M. Using sensitivity equations for computing gradients of the FOCE and FOCEI approximations to the population likelihood. J Pharmacokinetics and Pharmacodynamics. (2015) 42:191–209.
[8] Savic RM, Karlsson MO. Evaluation of an extended grid method using nonparametric distributions. AAPS Journal. 2009; 11(3): 615-627.
[9] Baverel PG, Savic RM, Karlsson MO. Two bootstrapping routines for obtaining imprecision estimates for nonparametric parameter distributions in nonlinear mixed effects models. J. Pharmacokinetics and Pharmacodynamics 2011; 38(1):63-82.
[10] Hee Sun Hong And Fred J. Hickernell. Algorithm 823: Implementing Scrambled Digital Sequences. ACM Transactions on Mathematical Software, Vol. 29, No. 2, June 2003, Pages 95–109.
[11] Lavielle, M. *Monolix Users Manual* [computer program]. Version 2.1. Orsay, France: Laboratoire de Mathematiques, U. Paris-Sud; 2007.
[12] Bennett, Racine-Poone, and Wakefield. MCMC for non linear hierarchical models. In: Markov Chain Monte Carlo in Practice. W.R. Gilks et al., Chapman & Hall (1996), chapter 19, pp 341-342.
[13] Gilks, Richardson and Spiegelhalter. Introducing Markov chain Monte Carlo. In: Markov Chain Monte Carlo in Practice. W.R. Gilks et al., Chapman & Hall (1996), chapter 1, pp 5-8.
[14] Karlsson MO and Savic RM. Diagnosing Model Diagnostics. Clinical Pharmacology and Therapeutics, 2007; 82(1): 17-20.
[15] Overgarrd RV, Jonsson N, Tornoe CW, and Madsen H. Non-Linear Mixed Effects Models with Stochastic Differential Equations: Implementation of an Estimation Algorithm. J. Pharmacokinetics and Pharmacodynamics, 2005; 32(1): 85-107.
[16] Predictive Performance for Population Models Using Stochastic Differential Equations Applied on Data From an Oral Glucose Tolerance Test. Moller JB, Overgaard RV, Madsen H, Hansen T, Pedersen O, and Ingwersen SH. J. Pharmacokinetics and Pharmacodynamics 2010; 37:85-98.

[17] Tornoe CW, Overgaard RV, Agerso H, Nielsen H, Madsen H, and Jonsson EN. Stochastic Differential Equations in NONMEM: Implementation, Application, and Comparison with Ordinary Differential Equations. Pharmaceutical Research, 2005; 22(8): 1247-1258.
[18] Bauer RJ, Guzy S, Ng CM. A survey of population analysis methods and software for complex pharmacokinetic and pharmacodynamic models with examples. AAPS Journal 2007; 9(1):E60-83.
[19] Hoffman MD, Gelman A. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. J. of Machine Learning Research 2014; 15: 1593-1623.
[20] Stan Modeling Language. User's Guide and Reference Manual. Stan Version 2.6.3.
[21] Lavielle M and Ribba B. Enhanced Method for Diagnosing Pharmacometric Models: Random Sampling from Conditional Distributions. Pharmaceutical Research Sept. 2016.
[22] Foracchia M, Hooker A, Vicini P, Ruggeri A. POPED, a Software for Optimal Experiment Design in Population Kinetics. Computer Methods and Programs in Biomedicine (2004) 74:29-46.
[23] Ogungbenro K, Graham G, Gueorguieva I, Aarons L. The Use of a Modified Fedorov Exchange Algorithm to Optimize Sampling Times for Population Pharmacokinetic Experiments. Computer Methods and Programs in Biomedicine (2005) 80:115-125.
[24] Nyberg J, Ueckert S, Stroemberg EA, Hennig S, Karlsson MO, Hooker AC. PopED: An Extended, Parallelized, Nonlinear Mixed Effects Models Optimal Design Tool. Computer Methods and Programs in Biomedicine (2012) 108:789-805.
[25] Nyberg J, Bazzoli C, Ogungbenro K, Aliev A, Leonov S, Duffull S, Hooker AC, Mentre F. Methods and Software Tools for Design Evaluation in Population Pharmacokinetics-Pharmacodynamics Studies. British J. Clinical Pharmacology (2014). 79:6-17.
[26] Lestini G, Dumont C, Nguyen TT. PFIM Interface 4.0. User Guide. May 2015. INSERM and U. Paris Diderot, Paris, France.
[27] Dumont C, Lestini G, Nagard H, Mentre F, Comets E, Nguyen TT, PFIM group. PFIM 4.0, an Extended R Program for Design Evaluation and Optimization in Nonlinear Mixed-Effect Models. Computer Methods and Programs in Biomedicine (2018). 156:217-229.
[28]. Schiesser WE. The Numerical Method of Lines: Integration of Partial Differential Equations. Academic Press, Inc. 1991.
[29] Wang Y. On Fast Computation of the Non-Parametric Maximum Likelihood Estimate of a Mixing distribution. J. R. Statist. Soc. B (2007). 79(2): 185-198.
[30] Guglielmi, Nicola & Hairer, Ernst. (2001). Implementing Radau IIA Methods for Stiff Delay Differential Equations. Computing. 67. 1-12.
[31] Nyberg J, Hoglund R, Bergstrand M, Karlsson MO, Hooker AC. Serial correlation in optimal design for nonlinear mixed effects models. J Pharmacokinet Pharmacodyn (2012) 39:239-249.
[32] Koch G. and Schropp J. Distributed transit compartments for arbitrary lifespan distributions in aging populations. J. Theoretical Biology (2015) 380:550-558.
[33] Krzyzanski W. Ordinary Differential Equation Approximation of Gamma Distributed Delay Model. J Pharmacokinet Pharmacodyn (2019) 46:53-63.

I.101 Example 1: Two compartment Model, Using ADVAN3, TRANS4.

```
;Model Desc: Two compartment Model, Using ADVAN3, TRANS4
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# Example 1 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX
       V1X QX V2X SDIX SDSX
$DATA example1.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4

$PK
; The thetas are MU modeled.
; Best that there is a linear relationship between THETAs and Mus
; The linear MU modeling of THETAS allows them to be efficiently
; Gibbs sampled.

MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
Y = F + F*EPS(1)

; Initial values of THETA
$THETA
(0.001, 2.0) ;[LN(CL)]
(0.001, 2.0) ;[LN(V1)]
(0.001, 2.0) ;[LN(Q)]
(0.001, 2.0) ;[LN(V2)]

;INITIAL values of OMEGA
$OMEGA BLOCK(4)
0.15   ;[P]
0.01  ;[F]
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]

;Initial value of SIGMA
$SIGMA
(0.6 )   ;[P]
```

```
;Prior information is important for MCMC Bayesian analysis,
;not necessary for maximization methods
;Note the syntax used for defining priors that is available
;as of NONMEM 7.3
$PRIOR NWPRI

; Prior information of THETAS
$THETAP (2.0 FIX)X4

; Variance to prior information of THETAS.
; Because variances are very large, this means that the prior
; information to the THETAS is highly uninformative.
$THETAPV BLOCK(4) FIX VALUES(10000,0.0)

; Prior information to the OMEGAS.
$OMEGAP BLOCK(4) FIX VALUES(0.2,0.0)
; Degrees of freedom to prior OMEGA matrix.
; Because degrees of freedom is very low, equal to the
; the dimension of the prior OMEGA, this means that the
; prior information to the OMEGAS is highly uninformative
$OMEGAPD (4 FIX)

; Prior information to the SIGMAS
$SIGMAP 0.06 FIX
; Degrees of freedom to prior SIGMA matrix.
; Because degrees of freedom is very low, equal to the
; the dimension of the prior SIGMA, this means that the
; prior information to the SIGMA is highly uninformative
$SIGMAPD (1 FIX)

; The first analysis is iterative two-stage,
; maximum of 500 iterations (NITER), iteration results
; are printed every 5 iterations, gradient precision (SIGL) is 4.
; Termination is tested on all of
; the population parameters (CTYPE=3),
; and for less then 2 significant digits change (NSIG).
; Prior information is not necessary for ITS, so NOPRIOR=1.
; The intermediate and final results of the ITS method will be
; recoded in row/column format in example1.ext

$EST METHOD=ITS MAPITER=0 INTERACTION FILE=example1.ext NITER=500
     PRINT=5 NOABORT SIGL=4 CTYPE=3 CITER=10
     CALPHA=0.05 NOPRIOR=1 NSIG=2

; The results of ITS are used as the initial values for the
; SAEM method. A maximum of 3000 ; stochastic iterations (NBURN)
; is requested, but may end early if statistical test determines
; that variations in all parameters is stationary
; (note that any settings from the previous $EST
; carries over to the next $EST statement, within a $PROB).
; The SAEM is a Monte Carlo process,
; so setting the SEED assures repeatability of results.
; Each iteration obtains only 2 Monte Carlo samples ISAMPLE),
;  so they are very fast.
; But many iterations are needed, so PRINT only
; every 100th iteration.
```

```
; After the stochastic phase, 500 accumulation iterations will be
; Performed (NITER), to obtain good parameters estimates with
; little stochastic noise.
; As a new FILE has not been given, the SAEM results will append to
; example1.ext.

$EST METHOD=SAEM INTERACTION NBURN=3000 NITER=500 PRINT=100
     SEED=1556678 ISAMPLE=2

; After the SAEM method, obtain good estimates of the marginal
; density (objective function),
; along with good estimates of the standard errors.
; This is best done with importance sampling ; (IMP),
; performing the expectation step only (EONLY=1), so that
; final population parameters remain at the final SAEM result.
; Five iterations (NITER) should allow the importance sampling
; proposal density to become stationary.
; This is observed by the objective function settling
; to a particular value (with some stochastic noise).
; By using 3000 Monte Carlo samples
; (ISAMPLE), this assures a precise assessment of standard errors.

$EST METHOD=IMP  INTERACTION EONLY=1 NITER=5 ISAMPLE=3000 PRINT=1
     SIGL=8 NOPRIOR=1

; The Bayesian analysis is performed.
; While 10000 burn-in iterations are requested as a maximum,
; because the termination test is on (CTYPE<>0, set at the
; first $EST statement), and because the initial parameters are at
; the SAEM result, which is the maximum likelihood position,
; the analysis should settle down to a stationary distribution in
; several hundred iterations.
; Prior information is also used to facilitate Bayesian analysis.
; The individual Bayesian iteration results are important,
; and may be need for post-processing analysis.
; So specify a separate FILE for the Bayesian analysis.

$EST METHOD=BAYES INTERACTION FILE=example1.txt NBURN=10000
     NITER=10000 PRINT=100 NOPRIOR=0

; Just for old-times sake, let's see what the traditional
; FOCE method will give us.
; And, remember to introduce a new FILE, so its results won't
; append to our Bayesian FILE.
; Appending to example1.ext with the EM methods is fine.

$EST METHOD=COND INTERACTION MAXEVAL=9999 NSIG=3 SIGL=10
     PRINT=5 NOABORT NOPRIOR=1
     FILE=example1.ext

; Time for the standard error results.
; You may request a more precise gradient precision (SIGL)
; that differed from that used during estimation.

$COV MATRIX=R PRINT=E UNCONDITIONAL SIGL=12

; Print out results in tables. Include some of the new weighted
```

```
; residual types

$TABLE ID TIME PRED RES WRES CPRED CWRES EPRED ERES EWRES NOAPPEND
       ONEHEADER FILE=example1.TAB NOPRINT
$TABLE ID CL V1 Q V2 FIRSTONLY NOAPPEND NOPRINT FILE=example1.PAR
$TABLE ID ETA1 ETA2 ETA3 ETA4 FIRSTONLY NOAPPEND
        NOPRINT FILE=example1.ETA
```

I.102 Example 2: 2 Compartment model with Clearance and central volume modeled with covariates age and gender

```
;Model Desc: Two Compartment model with Clearance and
; central volume modeled with covariates age and gender
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# example2 (from sampc)
$INPUT C SET ID JID TIME DV=CONC AMT=DOSE RATE EVID MDV CMT GNDR AGE
$DATA example2.csv IGNORE=C
$SUBROUTINES ADVAN3 TRANS4

$PK
; LCLM=log transformed clearance, male
LCLM=THETA(1)
;LCLF=log transformed clearance, female.
LCLF=THETA(2)
; CLAM=CL age slope, male
CLAM=THETA(3)
; CLAF=CL age slope, female
CLAF=THETA(4)
; LV1M=log transformed V1, male
LV1M=THETA(5)
; LV1F=log transformed V1, female
LV1F=THETA(6)
; V1AM=V1 age slope, male
V1AM=THETA(7)
; V1AF=V1 age slope, female
V1AF=THETA(8)
; LAGE=log transformed age
LAGE=DLOG(AGE)

;Mean of ETA1, the inter-subject deviation of Clearance,
; is ultimately modeled as linear function of THETA(1) to THETA(4).
; Relating thetas to Mus by linear functions is not essential for
; ITS, IMP, or IMPMAP methods, but is very helpful for MCMC methods
; such as SAEM and BAYES.

MU_1=(1.0-GNDR)*(LCLM+LAGE*CLAM) + GNDR*(LCLF+LAGE*CLAF)

; Mean of ETA2, the inter-subject deviation of V1,
; is ultimately modeled as linear function of THETA(5) to THETA(8)

MU_2=(1.0-GNDR)*(LV1M+LAGE*V1AM) + GNDR*(LV1F+LAGE*V1AF)
MU_3=THETA(9)
MU_4=THETA(10)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
CALLFL=0
```

```
; Option to model the residual error coefficient in THETA(11),
; rather than in SIGMA.
SDSL=THETA(11)
W=F*SDSL
Y = F + W*EPS(1)
IPRED=F
IWRES=(DV-F)/W

;Initial THETAs
$THETA
( 0.7 ) ;[LCLM]
( 0.7 ) ;[LCLF]
( 2 )   ;[CLAM]
( 2.0);[CLAF]
( 0.7 ) ;[LV1M]
( 0.7 ) ;[LV1F]
( 2.0 )   ;[V1AM]
( 2.0 )   ;[V1AF]
( 0.7 ) ;[MU_3]
(  0.7 );[MU_4]
( 0.3 )     ;[SDSL]

;Initial OMEGAs
$OMEGA BLOCK(4)
0.5  ;[p]
0.001  ;[f]
0.5  ;[p]
0.001 ;[f]
0.001 ;[f]
0.5  ;[p]
0.001 ;[f]
0.001 ;[f]
0.001 ;[f]
0.5 ;[p]

; SIGMA is 1.0 fixed, serves as unscaled variance for EPS(1).
; THETA(11) takes up the residual error scaling.
$SIGMA
(1.0 FIXED)

;Prior information is important for MCMC Bayesian analysis,
; not necessary for maximization methods
; In this example, only the OMEGAs have a prior distribution,
; the THETAS do not.
; For Bayesian methods, it is most important for at least the
; OMEGAs to have a prior, even an uninformative one,
; to stabilize the analysis. Only if the number of subjects
; exceeds the OMEGA dimension number by at least 100,
; then you may get away without priors on OMEGA for BAYES analysis.
$PRIOR NWPRI
; Prior OMEGA matrix
$OMEGAP BLOCK(4) FIX VALUES(0.01,0.0)
; Degrees of freedom to OMEGA prior matrix:
$OMEGAPD 4 FIX

; The first analysis is iterative two-stage.
; Note that the GRD specification is THETA(11) is a
```

```
; Sigma-like parameter.  This will allow NONMEM to make
; efficient gradient evaluations for THETA(11), which is useful
; for later IMP,IMPMAP, and SAEM methods, but has no impact on
; ITS and BAYES methods.

$EST METHOD=ITS INTERACTION FILE=example2.ext NITER=1000 NSIG=2
     PRINT=5 NOABORT SIGL=8 NOPRIOR=1 CTYPE=3 GRD=TS(11)

; Results of ITS serve as initial parameters for the IMP method.

$EST METHOD=IMP INTERACTION EONLY=0 MAPITER=0 NITER=100 ISAMPLE=300
     PRINT=1 SIGL=8

; The results of IMP are used as the initial values for the SAEM method.

$EST METHOD=SAEM NBURN=3000 NITER=2000 PRINT=10 ISAMPLE=2
     CTYPE=3 CITER=10 CALPHA=0.05

; After the SAEM method, obtain good estimates of the marginal density
; (objective function),
; along with good estimates of the standard errors.

$EST METHOD=IMP INTERACTION EONLY=1 NITER=5 ISAMPLE=3000
     PRINT=1 SIGL=8 SEED=123334
     CTYPE=3 CITER=10 CALPHA=0.05

; The Bayesian analysis is performed.

$EST METHOD=BAYES INTERACTION FILE=example2.TXT NBURN=10000
     NITER=3000 PRINT=100 NOPRIOR=0
     CTYPE=3 CITER=10 CALPHA=0.05

; Just for old-times sake, lets see what the traditional
; FOCE method will give us.
; And, remember to introduce a new FILE, so its results wont
; append to our Bayesian FILE.

$EST  METHOD=COND INTERACTION MAXEVAL=9999 FILE=example2.ext NSIG=2
  SIGL=14 PRINT=5 NOABORT NOPRIOR=1

$COV MATRIX=R UNCONDITIONAL
```

I.103 Example 3: Population Mixture Problem in 1 Compartment model, with Volume and rate constant parameters and their inter-subject variances modeled from two sub-populations

```
;Model Desc: Population Mixture Problem in 1 Compartment model,
; with Volume and rate constant parameters and their inter-subject
; variances modeled from two sub-populations
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# example3 (from ad1tr1m2s)
$INPUT C SET ID JID TIME CONC=DV DOSE=AMT RATE EVID MDV CMT VC1 K101
   VC2 K102 SIGZ PROB
$DATA example3.csv IGNORE=C

$SUBROUTINES ADVAN1 TRANS1

; The mixture model uses THETA(5) as the mixture proportion parameter,
; defining the proportion of subjects in sub-population 1 (P(1),
; and in sub-population 2 (P(2)

$MIX
P(1)=THETA(5)
P(2)=1.0-THETA(5)
NSPOP=2


$PK
;  The MUs should always be unconditionally defined, that is,
;  they should never be defined in IF/THEN blocks
; THETA(1) models the Volume of sub-population 1
MU_1=THETA(1)
; THETA(2) models the clearance of sub-population 1
MU_2=THETA(2)
; THETA(3) models the Volume of sub-population 2
MU_3=THETA(3)
; THETA(4) models the clearance of sub-population 2
MU_4=THETA(4)
VCM=DEXP(MU_1+ETA(1))
K10M=DEXP(MU_2+ETA(2))
VCF=DEXP(MU_3+ETA(3))
K10F=DEXP(MU_4+ETA(4))
Q=1
IF(MIXNUM.EQ.2) Q=0
V=Q*VCM+(1.0-Q)*VCF
K=Q*K10M+(1.0-Q)*K10F
S1=V

$ERROR
Y = F + F*EPS(1)

; Initial THETAs
$THETA
(-1000.0  4.3 1000.0) ;[MU_1]
(-1000.0 -2.9 1000.0) ;[MU_2]
```

```
(-1000.0 4.3 1000.0)  ;[MU_3]
(-1000.0 -0.67 1000.0) ;[MU_4]
(0.0001 0.667 0.9999)   ;[P(1)]

;Initial OMEGA block 1, for sub-population 1
$OMEGA BLOCK(2)
 .04 ;[p]
 .01 ; [f]
 .027; [p]

;Initial OMEGA block 2, for sub-population 2
$OMEGA BLOCK(2)
 .05; [p]
 .01; [f]
 .06; [p]

$SIGMA
0.01 ;[p]

; Prior information setup for OMEGAS only
$PRIOR NWPRI

; Prior OMEGA block 1.  Note that because the OMEGA is separated
; into blocks, so their priors should have the same block design.

$OMEGAP BLOCK(2)
 0.05 FIX
 0.0 0.05

; Prior OMEGA block 2

$OMEGAP BLOCK(2)
0.05 FIX
0.0 0.05

; Degrees of Freedom defined for Priors.
; One for each OMEGA block defining each sub-popluation
$OMEGAPD (2 FIX) (2 FIX)

$EST METHOD=ITS INTERACTION NITER=20 PRINT=1 NOABORT SIGL=8
     FILE=example3.ext CTYPE=3 CITER=10
     CALPHA=0.05 NOPRIOR=1

$EST NBURN=500 NITER=500 METHOD=SAEM INTERACTION PRINT=10 SIGL=6
     ISAMPLE=2

$EST METHOD=IMP INTERACTION NITER=5 ISAMPLE=1000 PRINT=1 NOABORT
     SIGL=6 EONLY=1 MAPITER=0

$EST METHOD=BAYES INTERACTION NBURN=2000 NITER=1000 PRINT=10
     FILE=example3.txt SIGL=8 NOPRIOR=0

$EST MAXEVAL=9999 NSIG=3 SIGL=12 PRINT=1 FILE=example3.ext
     METHOD=CONDITIONAL INTERACTION NOABORT
     NOPRIOR=1
$COV MATRIX=R UNCONDITIONAL
```

I.104 Example 4: Population Mixture Problem in 1 Compartment model, with rate constant parameter and its inter-subject variances modeled as coming from two sub-populations

```
;Model Desc: Population Mixture Problem in 1 Compartment model,
; with rate constant parameter and its inter-subject variances
; modeled as coming from two sub-populations
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# example4 (from ad1tr1m2t)
$INPUT C SET ID JID TIME CONC=DV DOSE=AMT RATE EVID MDV CMT VC1
       K101 VC2 K102 SIGZ PROB
$DATA example4.csv IGNORE=C

$SUBROUTINES ADVAN1 TRANS1

$MIX
P(1)=THETA(4)
P(2)=1.0-THETA(4)
NSPOP=2


$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
V=DEXP(MU_1+ETA(1))
K10M=DEXP(MU_2+ETA(2))
K10F=DEXP(MU_3+ETA(3))
Q=1
IF(MIXNUM.EQ.2) Q=0
K=Q*K10M+(1.0-Q)*K10F
S1=V

$ERROR
Y = F + F*EPS(1)

$THETA
(-1000.0  4.3 1000.0) ;[MU_1]
(-1000.0 -2.9 1000.0) ;[MU_2]
(-1000.0 -0.67 1000.0) ;[MU_3]
(0.0001 0.667 0.9999)   ;[P(1)]


$OMEGA BLOCK(3)
 .04  ;[p]
 0.01  ;[f]
 .027 ;[p]
 0.01  ;[f]
 0.001 ;[f]
 0.06 ;[p]
```

```
$SIGMA
 0.01 ;[p]

; Prior information setup for OMEGAS only
$PRIOR NWPRI

; Prior OMEGA
$OMEGAP BLOCK(3)
 0.05 FIX
 0.0 0.05
 0.0 0.0  0.05

; Degrees of Freedom defined for Priors.
$OMEGAPD (3 FIX)

$EST METHOD=ITS INTERACTION NITER=30 PRINT=5 NOABORT SIGL=6
     FILE=example4.ext NOPRIOR=1 CTYPE=3 CITER=10 CALPHA=0.05

$EST METHOD=IMP INTERACTION NITER=20 ISAMPLE=300 PRINT=1
     NOABORT SIGL=6 NOPRIOR=1

$EST NBURN=500 NITER=500 METHOD=SAEM INTERACTION PRINT=10 SIGL=6
     ISAMPLE=2 NOPRIOR=1 MAPITER=0

$EST METHOD=IMP INTERACTION EONLY=1 NITER=20 ISAMPLE=3000 PRINT=1
     NOABORT SIGL=6 NOPRIOR=1

$EST METHOD=BAYES INTERACTION NBURN=2000 NITER=5000 PRINT=10
     FILE=example4.txt SIGL=6 NOPRIOR=0

$EST MAXEVAL=9999 NSIG=3 SIGL=12 PRINT=1
     METHOD=CONDITIONAL INTERACTION
     NOABORT FILE=example4.ext NOPRIOR=1

$COV MATRIX=R UNCONDITIONAL SIGL=10
```

I.105 Example 5: Population Mixture Problem in 1 Compartment model, with rate constant parameter mean modeled for two sub-populations, but its inter-subject variance is the same in both sub-populations.

```
;Model Desc: Population Mixture Problem in 1
; Compartment model, with rate constant parameter
; mean modeled for two sub-populations, but its inter-subject
; variance is the same in both sub-populations
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# example5 (from ad1tr1m4t)
$INPUT C SET ID JID TIME CONC=DV DOSE=AMT RATE EVID MDV CMT
       VC1 K101 VC2 K102 SIGZ PROB
$DATA example5.csv IGNORE=C

$SUBROUTINES ADVAN1 TRANS1

$MIX
P(1)=THETA(4)
P(2)=1.0-THETA(4)
NSPOP=2


$PK
Q=1
IF(MIXNUM.EQ.2) Q=0
MU_1=THETA(1)
; Note that MU_2 can be modeled as THETA(2) or THETA(3),
; depending on the MIXNUM value.
; Also, we are avoiding IF/THEN blocks.
MU_2=Q*THETA(2)+(1.0-Q)*THETA(3)
V=DEXP(MU_1+ETA(1))
K=DEXP(MU_2+ETA(2))
S1=V

$ERROR
Y = F + F*EPS(1)

$THETA
(-1000.0  4.3 1000.0)  ;[MU_1]
(-1000.0 -2.9 1000.0)  ;[MU_2-1]
(-1000.0 -0.67 1000.0) ;[MU_2-2]
(0.0001 0.667 0.9999)  ;[P(1)]

$OMEGA BLOCK(2)
0.04 ;[p]
0.01  ;[f]
0.04 ;[p]

$SIGMA
0.01 ;[p]

$EST METHOD=ITS INTERACTION NITER=100 PRINT=1 NOABORT SIGL=8
     FILE=example5.ext CTYPE=3
```

```
$EST METHOD=IMPMAP INTERACTION NITER=20 ISAMPLE=300 PRINT=1 NOABORT SIGL=8
$EST METHOD=IMP INTERACTION NITER=20 MAPITER=0 ISAMPLE=1000 PRINT=1 NOABORT
SIGL=6
$EST NBURN=500 NITER=500 METHOD=SAEM INTERACTION PRINT=10 SIGL=6 ISAMPLE=2
$EST METHOD=IMP INTERACTION NITER=5 ISAMPLE=1000 PRINT=1
     NOABORT SIGL=6 EONLY=1
$EST METHOD=BAYES INTERACTION NBURN=2000 NITER=5000 PRINT=10
FILE=example5.txt SIGL=8
$EST MAXEVAL=9999 NSIG=2 SIGL=8 PRINT=10 FILE=example5.ext
     METHOD=CONDITIONAL INTERACTION NOABORT
$COV MATRIX=R
```

I.106 Example 6: Receptor Mediated Clearance model with Dynamic Change in Receptors

```
;Model Desc: Receptor Mediated Clearance model with Dynamic Change
;            in Receptors
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# example6 (from r2compl)
$INPUT C SET ID JID TIME DV=CONC DOSE=AMT RATE EVID MDV CMT
$DATA example6.csv IGNORE=C

; The new numerical integration solver is used, although ADVAN=9
; is also efficient for this problem.

$SUBROUTINES ADVAN13 TRANS1 TOL=4
$MODEL NCOMPARTMENTS=3

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
MU_5=THETA(5)
MU_6=THETA(6)
MU_7=THETA(7)
MU_8=THETA(8)
VC=EXP(MU_1+ETA(1))
K10=EXP(MU_2+ETA(2))
K12=EXP(MU_3+ETA(3))
K21=EXP(MU_4+ETA(4))
VM=EXP(MU_5+ETA(5))
KMC=EXP(MU_6+ETA(6))
K03=EXP(MU_7+ETA(7))
K30=EXP(MU_8+ETA(8))
S3=VC
S1=VC
KM=KMC*S1
F3=K03/K30

$DES
DADT(1) = -(K10+K12)*A(1) + K21*A(2) - VM*A(1)*A(3)/(A(1)+KM)
DADT(2) = K12*A(1) - K21*A(2)
DADT(3) =  -VM*A(1)*A(3)/(A(1)+KM) - K30*A(3) + K03

$ERROR
CALLFL=0
ETYPE=1
IF(CMT.NE.1) ETYPE=0
IPRED=F
Y = F + F*ETYPE*EPS(1) + F*(1.0-ETYPE)*EPS(2)


$THETA
;Initial Thetas
( 4.0 )  ;[MU_1]
( -2.1 ) ;[MU_2]
( 0.7 )  ;[MU_3]
( -0.17 );[MU_4]
( 2.2 ) ;[MU_5]
( 0.14 )  ;[MU_6]
( 3.7 )  ;[MU_7]
( -0.7) ;[MU_8]


;Initial Omegas
$OMEGA BLOCK(8)
0.2 ;[p]
```

```
-0.0043  ;[f]
0.2 ;[p]
0.0048   ;[f]
-0.0023  ;[f]
0.2 ;[p]
0.0032   ;[f]
0.0059   ;[f]
-0.0014  ;[f]
0.2 ;[p]
0.0029   ;[f]
0.0027 ;[f]
-0.00026 ;[f]
-0.0032  ;[f]
0.2 ;[p]
-0.0025  ;[f]
0.00097  ;[f]
0.0024   ;[f]
0.00197  ;[f]
-0.0080  ;[f]
0.2 ;[p]
0.0031   ;[f]
-0.00571 ;[f]
0.0030   ;[f]
-0.0074  ;[f]
0.0025   ;[f]
0.0034   ;[f]
0.2 ;[p]
0.00973  ;[f]
0.00862  ;[f]
0.0041   ;[f]
0.0046   ;[f]
0.00061  ;[f]
-0.0056  ;[f]
0.0056   ;[f]
0.2 ;[p]

$SIGMA
0.1 ;[p]
0.1 ;[p]

$PRIOR NWPRI
; Omega prior
$OMEGAP BLOCK(8)
0.2 FIX
0.0 0.2
0.0 0.0 0.2
0.0 0.0 0.0 0.2
0.0 0.0 0.0 0.0 0.2
0.0 0.0 0.0 0.0 0.0 0.2
0.0 0.0 0.0 0.0 0.0 0.0 0.2
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2
; degrees of freedom for OMEGA prior
$OMEGAPD
(8 FIXED)           ;[dfo]

; Starting with a short iterative two stage analysis brings the
; results closer so less time needs to be spent during the
; burn-in of the BAYES analysis

$EST METHOD=ITS INTERACTION SIGL=4 NITER=15 PRINT=1
     FILE=example6.ext NOABORT NOPRIOR=1

$EST METHOD=BAYES INTERACTION NBURN=4000 SIGL=4 NITER=10000
     PRINT=10 CTYPE=3 FILE=example6.txt NOABORT NOPRIOR=0

; By default, ISAMPLE_M* are 2.  Since there are many data points
; per subject, setting these to 1 is enough, and it reduces the
; time of the analysis
     ISAMPLE_M1=1 ISAMPLE_M2=1 ISAMPLE_M3=1 IACCEPT=0.4
$COV MATRIX=R UNCONDITIONAL
```

I.107 Example 7r: Inter-occasion Variability

```
;Model Desc: Interoccasion Variability
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB run# example7r
$INPUT C SET ID  TIME  AMT RATE EVID MDV CMT DV OCC
$ABBR REPLACE ETA(OCC_CL)=ETA(3,4,5)
$DATA example7r.csv IGNORE=C

$SUBROUTINES ADVAN1 TRANS2

$PK
MU_1=THETA(1)
MU_2=THETA(2)
V=DEXP(MU_1+ETA(1))
S1=V
VC=V
CL=DEXP(MU_2+ETA(2))*EXP(ETA(OCC_CL))

$ERROR
IPRED=F
Y = F+F*EPS(1)

;Initial Thetas
$THETA
 2.0  ;[MU_1]
 2.0  ;[MU_2]

;Initial omegas
$OMEGA BLOCK(2)
 .3 ;[p]
 -.01  ;[f]
 .3 ;[p]
$OMEGA BLOCK(1)
 .1  ;[p]
$OMEGA BLOCK(1) SAME(2)

$SIGMA
 0.1 ;[p]

$PRIOR NWPRI
; Degrees of freedom for Prior Omega blocks
$OMEGAPD (2.0 FIXED) (1.0 FIXED)
; Prior Omegas
$OMEGAP BLOCK(2)
 .14 FIX
 0.0 .125
$OMEGAP BLOCK(1) .0164 FIX
$OMEGAP BLOCK(1) SAME(2)

$EST METHOD=ITS INTERACTION FILE=example7r.ext   NITER=10000
     PRINT=5 NOABORT SIGL=8 CTYPE=3 CITER=10
     NOPRIOR=1 CALPHA=0.05 NSIG=2
```

```
$EST METHOD=SAEM INTERACTION NBURN=30000 NITER=500 SIGL=8
     ISAMPLE=2 PRINT=10 SEED=1556678 CTYPE=3
     CITER=10 CALPHA=0.05 NOPRIOR=1

$EST METHOD=IMP  INTERACTION EONLY=1 NITER=4 ISAMPLE=3000
     PRINT=1 SIGL=10 NOPRIOR=1 MAPITER=0

$EST METHOD=BAYES INTERACTION FILE=example7r.txt NBURN=10000
     NITER=10000 PRINT=100 CTYPE=3 CITER=10
     CALPHA=0.05 NOPRIOR=0

$EST METHOD=COND INTERACTION MAXEVAL=9999 NSIG=3 SIGL=10 PRINT=5
     NOABORT NOPRIOR=1
     FILE=example7r.ext

$COV MATRIX=R PRINT=E UNCONDITIONAL
```

I.108 Example 8: Sample History of Individual Values in MCMC Bayesian Analysis

```
;Model Desc: Two compartment Model, Using ADVAN3, TRANS4
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# Example 8 (from samp51)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT
       CLX V1X QX V2X SDIX SDSX
$DATA example8.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4


$PK
include nonmem_reserved_general
; Request extra information for Bayesian analysis.
; An extra call will then be made for accepted samples
BAYES_EXTRA_REQUEST=1
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
; When Bayes_extra=1, then this particular set of individual
; parameters were "accepted" So you may record them if you wish
  IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 .AND. TIME==0.0) THEN
"  WRITE(51,98) ITER_REPORT,ID,CL,V1,Q,V2
" 98 FORMAT(I12,1X,F14.0,4(1X,1PG12.5))
ENDIF

$ERROR
include nonmem_reserved_general
BAYES_EXTRA_REQUEST=1
Y = F + F*EPS(1)
IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 ) THEN
" WRITE(52,97) ITER_REPORT,ID,TIME,F
" 97 FORMAT(I12,1X,F14.0,2(1X,1PG12.5))
ENDIF

; Initial values of THETA
$THETA
(2.0) ;[LN(CL)]
(2.0) ;[LN(V1)]
(2.0) ;[LN(Q)]
(2.0) ;[LN(V2)]
;INITIAL values of OMEGA
$OMEGA BLOCK(4)
0.15   ;[P]
0.01  ;[F]
```

```
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]
;Initial value of SIGMA
$SIGMA
(0.6 )   ;[P]


$PRIOR NWPRI
; Prior information to the Thetas.
$THETAP (2.0 FIX)x4
$THETAPV BLOCK(4) FIX VALUES(10000.0,0.0)

; Prior information to the OMEGAS.
$OMEGAP BLOCK(4)
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2
$OMEGAPD (4 FIX)

$EST METHOD=BAYES INTERACTION FILE=example8.ext NBURN=10000
     NITER=1000 PRINT=100 NOPRIOR=0 CTYPE=3 CINTERVAL=100
```

Note that the contents is written to file fort.51 and fort.52. If parallelization is used, then fort.51 and fort.52 files in each of the worker directories will be created, and must be collected after the run to obtain records for all of the subjects. Alternatively, specific file names may be given, the names being created according to the node number. However, care must be given the specific directory location is valid for a given run (example8b):

```
;Model Desc: Two compartment Model, Using ADVAN3, TRANS4
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# Example 8b (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X
SDIX SDSX
$DATA example8.csv IGNORE=C
$abbr DECLARE INTEGER FIRST_WRITE INTEGER FIRST_WRITE2

$SUBROUTINES ADVAN3 TRANS4


$PK
include nonmem_reserved_general
; Request extra information for Bayesian analysis.  An extra call will
; then be made for accepted samples
BAYES_EXTRA_REQUEST=1
MU_1=THETA(1)
MU_2=THETA(2)
```

```
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
; When Bayes_extra=1, then this particular set of individual parameters
; were "accepted"
; So you may record them if you wish
IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 .AND. TIME==0.0) THEN
IF(FIRST_WRITE==0) THEN
" OPEN(unit=53,FILE='C:\NONMEM\WORKA_'//TRIM(TFI(PNM_NODE_NUMBER)))
FIRST_WRITE=1
ENDIF
" WRITE(53,'(I12,1X,F14.0,5(1X,1PG12.5))')
ITER_REPORT,ID,CL,V1,Q,V2,OBJI(NIREC,1)
ENDIF

$ERROR
include nonmem_reserved_general
BAYES_EXTRA_REQUEST=1
Y = F + F*EPS(1)
IF(BAYES_EXTRA==1 .AND. ITER_REPORT>=0 ) THEN
IF(FIRST_WRITE2==0) THEN
"OPEN(UNIT=54,FILE='C:\NONMEM\WORKB_'//TRIM(TFI(PNM_NODE_NUMBER)))
FIRST_WRITE2=1
ENDIF
" WRITE(54,'(I12,1X,F14.0,2(1X,1PG12.5))') ITER_REPORT,ID,TIME,F
ENDIF

; Initial values of THETA
$THETA
(2.0) ;[LN(CL)]
(2.0) ;[LN(V1)]
(2.0) ;[LN(Q)]
(2.0) ;[LN(V2)]
;INITIAL values of OMEGA
$OMEGA BLOCK(4)
0.15   ;[P]
0.01  ;[F]
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]
0.01  ;[F]
0.01  ;[F]
0.01  ;[F]
0.15   ;[P]
;Initial value of SIGMA
$SIGMA
(0.6 )   ;[P]

$PRIOR NWPRI
; Prior information to the THETAS.
$THETAP (2.0 FIX) (2.0 FIX) (2.0 FIX) (2.0 FIX)
$THETAPV BLOCK(4)
```

```
10000 FIX
0.00 10000
0.00  0.00 10000
0.00  0.00 0.0 10000

; Prior information to the OMEGAS.
$OMEGAP BLOCK(4)
0.2 FIX
0.0  0.2
0.0  0.0 0.2
0.0  0.0 0.0 0.2
$OMEGAPD (4 FIX)

$EST METHOD=BAYES INTERACTION FILE=example8b.ext NBURN=10000 NITER=1000
PRINT=100 NOPRIOR=0
     CTYPE=3 CINTERVAL=100
```

Note the use of the include file nonmem_reserved_general, which for purposes of this example contain the following declarations of reserved variables:

```
"C ITER_REPORT: Iteration number that is reported to output
"C (can be negative, if during a burn period).
"C BAYES_EXTRA, BAYES_EXTRA_REQUEST, used in example 8
" USE NMBAYES_REAL, ONLY: OBJI
" USE NMBAYES_INT, ONLY: ITER_REPORT,BAYES_EXTRA_REQUEST,BAYES_EXTRA
" USE PNM_CONFIG, ONLY: PNM_NODE_NUMBER
‟ USE NM_INTERFACE, ONLY: TFI,TFD
```

I.109 Example 9: Simulated Annealing For Saem using Constraint Subroutine

```
;Model Desc: Two compartment Model, Using ADVAN3, TRANS4
;Project Name: nm7examples
;Project ID: NO PROJECT DESCRIPTION

$PROB RUN# Example 9 (from samp5l)
$INPUT C SET ID JID TIME  DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX V2X
SDIX SDSX
$DATA example9.csv IGNORE=C

$SUBROUTINES ADVAN3 TRANS4 OTHER=ANEAL.F90

$PK
MU_1=THETA(1)
MU_2=THETA(2)
MU_3=THETA(3)
MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1

$ERROR
Y = F + F*EPS(1)

; Initial values of THETA
$THETA
(0.001, 2.0) ;[LN(CL)]
(0.001, 2.0) ;[LN(V1)]
(0.001, 2.0) ;[LN(Q)]
(0.001, 2.0) ;[LN(V2)]
;INITIAL values of OMEGA
$OMEGA BLOCK(4)
0.05   ;[P]
0.01  ;[F]
0.05   ;[P]
0.01  ;[F]
0.01  ;[F]
0.05   ;[P]
0.01  ;[F]
0.01  ;[F]
0.01  ;[F]
0.05   ;[P]
;Initial value of SIGMA
$SIGMA
(0.6 )   ;[P]

$EST METHOD=SAEM INTERACTION FILE=example9.ext NBURN=5000 NITER=500 PRINT=10
NOABORT SIGL=6
    CTYPE=3 CINTERVAL=100 CITER=10 CALPHA=0.05

File Aneal.f90
```

```
      SUBROUTINE
CONSTRAINT(THETAS,NTHETAS,SIGMA2,NSIGMAS,OMEGA,NOMEGAS,ITER_NO)
      USE SIZES, ONLY: ISIZE,DPSIZE
      INCLUDE '..\nm\TOTAL.INC'
      INTEGER(KIND=ISIZE) NTHETAS,NSIGMAS,NOMEGAS,ITER_NO
      INTEGER I,J,ITER_OLD
      DATA ITER_OLD /-1/
      REAL(KIND=DPSIZE) ::
OMEGA(MAXOMEG,MAXOMEG),THETAS(MAXPTHETA),SIGMA2(MAXPTHETA)
      REAL(KIND=DPSIZE) :: OMEGO(MAXOMEG)
      SAVE
!-------------------------------------------------------------------------
-
      IF(SAEM_MODE==1 .AND. IMP_MODE==0 .AND. ITS_MODE==0 .AND. ITER_NO<200)
THEN
      IF(ITER_NO/=ITER_OLD .OR. ITER_NO==0) THEN
! During burn-in phase of SAEM, and when a new iteration occurs
! (iter_old<>iter_no)
! store the present diagonals of omegas
      ITER_OLD=ITER_NO
      DO I=1,NOMEGAS
      OMEGO(I)=OMEGA(I,I)
      ENDDO
      ENDIF
      IF(ITER_NO /=0) THEN
      DO I=1,NOMEGAS
! Use whatever algorithm needed to "slow down" the reduction of Omega
! The expansion of Omega should be less with each iteration.
      OMEGA(I,I)=OMEGO(I)*(1.0D+00+10.0D+00/ITER_NO)
      ENDDO
      ENDIF
      ENDIF
      RETURN
!
      END SUBROUTINE CONSTRAINT
```

I.110 Example 10: One Compartment First Order Absorption Pharmaokinetics with Categorical Data

```
$PROB  F_FLAG04est2a.ctl
$INPUT C ID DOSE=AMT TIME DV WT TYPE
$DATA example10.csv IGNORE=@

$SUBROUTINES  ADVAN2 TRANS2


$PK
   CALLFL=1
   MU_1=DLOG(THETA(1))
   KA=DEXP(MU_1+ETA(1))
   MU_2=DLOG(THETA(2))
   V=DEXP(MU_2+ETA(2))
   MU_3=DLOG(THETA(3))
   CL=DEXP(MU_3+ETA(3))
   SC=V/1000

$THETA  5.0 10.0 2.0 0.1 0.1

$OMEGA BLOCK (3)
0.5
0.01 0.5
0.01 0.01 0.5


; Because THETA(4) and THETA(5) have no inter-subject variability
; associated with them, the algorithm must use a more computationally
; expensive gradient evaluation for these two parameters

$SIGMA 0.1


$PRIOR NWPRI
; Priors to Omegas
$OMEGAP BLOCK (3)
0.09 FIX
0.0 0.09
0.0 0.0 0.09
$OMEGAPD (3 FIX)

$ERROR
    EXPP=THETA(4)+F*THETA(5)
IF (TYPE.EQ.0) THEN
; PK Data
    F_FLAG=0
    Y=F+F*ERR(1) ; a prediction
 ELSE
; Categorical data
    F_FLAG=1
; Use protected exponent PEXP, to avoid numerical overflow
    A=PEXP(EXPP)
    B=1+A
```

```
    Y=DV*A/B+(1-DV)/B      ; a likelihood
 ENDIF


$EST METHOD=ITS INTER LAP NITER=1000 PRINT=5 SIGL=6 NSIG=2
     NOABORT NOPRIOR=1 CTYPE=3 CITER=10 CALPHA=0.05
     FILE=example10.ext
; Because of categorical data, which can make conditional density highly
; non-normal, select a t-distribution with 4 degrees of freedom for
; importance sampling proposal density
$EST METHOD=IMP INTER LAP NITER=1000 PRINT=1 ISAMPLE=300 DF=4
     IACCEPT=1.0
$EST METHOD=IMP EONLY=1 NITER=5 ISAMPLE=1000 PRINT=1 DF=4
     IACCEPT=1.0 MAPITER=0

$EST METHOD=SAEM EONLY=0 INTER LAP NBURN=2000 NITER=1000 PRINT=50
     DF=0 IACCEPT=0.4
$EST METHOD=IMP EONLY=1 NITER=5 ISAMPLE=1000 PRINT=1 DF=4
     IACCEPT=1.0 MAPITER=0

$EST METHOD=BAYES NBURN=3000 NSAMPLE=3000 PRINT=100
     FILE=example10.txt DF=0 IACCEPT=0.4 NOPRIOR=0

$EST METHOD=COND LAP INTER MAXEVAL=9999 PRINT=1 FILE=example10.ext
     NOPRIOR=1 NOHABORT

$COV UNCONDITIONAL PRINT=E MATRIX=R SIGL=10
$TABLE ID DOSE WT TIME TYPE DV A NOPRINT FILE=example10.tab
```