

The following bugs have been detected in NONMEM 7.2. An interim version of NONMEM will not be issued as these bugs have work-arounds or are little used features. These issues are being corrected for version NONMEM 7.3, to be released in November 2013.

- 1) Some operating systems do not like the word 'nul' for a file name for FNULL. Work-around: 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 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 INTERRUPT" is misspelled.
- 4) SAEM terminates on some problems. Cause is access violation when `CONSTRAIN` is called. Work-around 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 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 `$MSFI`, NONMEM sometimes prevents its use because of a flag indicating it was not properly estimated. Work-around: use the record `$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. Is 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: 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: (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 in NONMEM 7.3
 - 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))
 - 17) Internal arrays in NMTRAN that process MTIME/MPAST etc., are not dynamically expanded with \$SIZES PCT= statement. They remain statically allocated with the sizes defined in sizes.f90. To use more than PCT MTIMEs, increase PCT in sizes.f90, and rebuild NONMEM. This is fixed in NONMEM 7.3
 - 18) A user variable from \$PK may be used in \$ERROR. If the first time the variable appears in \$ERROR is in a logical condition, then the variable name should be all uppercase characters. This bug exists in nm72, fixed in nm73.
 - 19) Two variables greater 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.

20) If consecutive power term statements are defined within an IF/THEN block, this may cause 0 gradients to be produced for some parameters. 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
```

Will cause theta(1) to have a 0 gradient. Modifying the code so that the power term is outside the IF/THEN block will resolve this, for example:

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

This bug occurs in nonmem 7.2.0. and nonmem 7.3. In nmvi and nm710, nm712, the bug reveals itself only with LAPLACE no-interaction problems as an incorrect evaluation of the objective function, but not as a 0 gradient.