

The following bugs have been detected in NONMEM 7.3. At present, a corrected version of NONMEM will not be issued for these bugs as they have work-arounds or are little used features.

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.

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 to 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.