

GAMS Newsletter Number 16

Bruce McCarl

GAMS 21.4 release

GAMS Development Corporation has recently released version 21.4. Version 21.4 expands some language capabilities. Release notes are on the GAMS web site at <http://www.gams.com/docs/release/release.htm#21.4> . Some of the new features within this release are discussed below. The software can be downloaded from <http://www.gams.com/download/> .

Speed Enhancements

Execution time characteristics have been improved. In particular, as discussed in the Execution speed chapter of the McCarl Guide GAMS statements can execute unnecessarily slowly if one is not careful about index ordering. For example $x(i,j,k)=y(k,j,i)$ can be slower than $x(i,j,k)=y(i,j,k)$ for exactly the same contents of y just different index ordering. Version 21.4 has features to speed up execution in such cases with the release notes showing 90% reductions in execution time. Work has also been done on speeding up lead and lag operators.

GAMS use of these speedups (that involve automatic reordering of data arrays during execution) is controlled by the command line parameter `sys11`, where by default GAMS uses the new procedures but if the user sets `sys11=1` GAMS reverts to the old procedures.

Solvers

Solver capability upgrades, solver availability and pricing developments are present.

- BARON, CONOPT, MINOS, OQNLP and XPRESS have new features.
- NLPEC and PATHNLP are upgraded from beta versions.
- New solvers are contained
- GAMS/COINGLPK and GAMS/COINSBB, two free LP/MIP solvers from the Free Computational Infrastructure Operations Research (COIN-OR) Initiative.
- MSNLP a stochastic search algorithm from Optimal Methods, Inc for global optimization problems.
- A pricing discount is available for those purchasing at least two of the global solvers BARON, LGO, or MSNLP, amounting to 25%. Package discounts are also present regarding the solvers CONOPT, MINOS, and SNOPT. See the sales page for details.

McCarl User Guide

The release contains the newest version of the McCarl GAMS user guide current through GAMS 21.3. This is now automatically installed by the Windows installation procedure.

Unfortunately the current release is missing the document chapter Variables, Equations, Models and Solves. For now this can be fixed by going to the web and downloading that chapter then placing it in the subdirectory.\docs\bigdocs\gams2002 of your GAMS system which on my machine is C:\Program Files\GAMS21.4\docs\bigdocs\gams2002. A soon to arrive maintenance release will fix this.

Other features

The release notes indicate 21.4 also contains

- New entries in the model and test model libraries that illustrate new features.
- Procedures to rank order data in parameters (GDXRANK) as documented here.
- IDE features regarding licenses.
- New functions regarding time of day, execution time, execution error codes, beta distributions and gamma distributions.
- Features that enable the use of embedded values in parameter and set data statements.
- Suffixes Grad and Hess that get exact point derivatives from any function.
- Procedures to reformat dates and insert tabs in put files.
- An option suppressing the logging of put files.
- A \$IF option that allows testing for the solver being used at compile time:\$IF SOLVER command
- \$if NOT solver baron \$goto nobaron
- Improved underflow control.

GDX, Timing of \$ commands versus EXECUTE_UNLOAD

When one is using \$ commands in column 1 of a GAMS program (not \$ conditionals) one must remember that the only data available to the commands is that in the set or parameter statements not any of the data developed by program execution. Consider the example

```
Set          i          /a1,a2,a3,a4/;
Parameter    x(i)       /a1 2,a4 12/;
x(i)$x(i)=0)=5;
$gdxout ss.gdx
$unload x
$gdxout
execute_unload "ss1.gdx" x
```

In this example the files ss.gdx has 2 entries for x while the file ss1.gdx has 4 entries. Why because the \$ commands are all executed at compile time and the statement redefining x will not be performed until execution time.

\$UNLOAD, and \$CALL GDXRW or any other form of \$CALL can never use data computed during the GAMS run. If one wishes to use such data then

EXECUTE_UNLOAD and EXECUTE GDXXRW must be used. I think use of \$UNLOAD and \$CALL GDXXRW should be avoided if at all possible. The McCarl guide available through the web or the IDE on Windows machines elaborates in the chapters on GDX file usage, and Links to Other Programs.

Transferring data via TXT and GDX files

Erwin Kalvelagen has recently created programs that will pass information to and from TXT files at time of program execution. These are open source programs called gdx2txt.exe and txt2gdx.exe described in Erwin's webpage at <http://www.gams.com/~erwin/> through Interfacing GAMS with other applications (<http://www.gams.com/~erwin/interface/interface.html>): Tutorial and examples specifically in the section on the GDX API under language bindings and are available for download through that document. These permit data exchange with user written programs. Consider the following suggestive example:

```
Set      i      /a1,a2,a3,a4/;
Parameter  x(i)  /a1 2,a4 12/;
alias(i,j);
parameter y(i,j);
parameter z(i,j);
x(i)$x(i)=0=5;
execute_unload "ss1.gdx" x;
execute "gdx2txt.exe in=ss1.gdx id=x out=mytext.txt"
y(i,j)=ord(i)*ord(j);
execute_unload "ss1.gdx" y;
execute "gdx2txt.exe in=ss1.gdx id=y out=mytext2.txt"
execute "del mytext3.txt";
execute "copy mytext2.txt mytext3.txt";
*execute "myfavoriteprogram.exe";
execute "txt2gdx.exe out=ss1.gdx parameter id=z in=mytext3.txt";
execute_load "ss1.gdx" z;
display z;
```

Herein the file mytext.txt will contain the data for the parameter x in a file looking as follows

```
i1 2
i2 5
i3 5
i4 12
```

while the first 6 lines of mytext2.txt contain the information for the two dimensional parameter y

```
i1.j1 1
```

i1.j2 2
i1.j3 3
i1.j4 4
i2.j1 2
i2.j2 4

finally the entries in the GAMS parameter z at the end of the program is the information ultimately in y but after

- unloading it to a GDX
- then moving it to a text file
- then copying it to another text file
- then loading to a GDX file
- then bringing it back into GAMS.

Here one could replace the copy command with an Execute that runs of a user written program that reads the TXT file passed from GAMS then creates a TXT file for passage back to GAMS. The formatting requirements for such files involve using set names consistent with the GAMS program then for multidimensional items separating the various set elements with periods.

Courses offered

I teach Basic GAMS in College Station Nov 30-Dec 3, 2004. An Advanced GAMS class will be offered January 4-7, 2005. Further information and other courses are listed on <http://www.gams.com/courses.htm>.