Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 48

May 2023

This Newsletter addresses the following

1 Releases 39-43

Since the last newsletter we have new releases 39 through 43 with release notes being at https://www.gams.com/latest/docs/RN_MAIN.html . When I peruse these notes things that strike of broad interest are:

1.1 GAMS system

- GAMS is now also available as a native build for macOS 13 on ARM64 CPUs (Apple M1/M2). However some solvers and utilities are not available for that platform.
- Added new dollar control options which allow one to load all the contents of a GDX file into GAMS. There are two variants of this.
    - Use of **$gdxLoadAll** loads all symbols present in the target GDX file that have been declared in the GAMS program. Note this does not need or allow a list of the symbol to load.. When symbol definitions are not consistent with what is in

GAMS an error message is issued. Any symbols present in the GDX that are not declared in the GAMS program are ignored.
    - Use of **$declareAndLoad** functions much as does **$gdxLoadAll** but also will declare any new symbols resident in the GDX file. In particular it uses information that is resident in the GDX file on to declare sets, explanatory text etc for symbols that were not previously declared in the GAMS program.
- GAMS is introducing the concept of Tools which will be used in links to other programs (Soon to be released versions of CONNECT and Transfer) that we discuss below.
    - For now there are available tools for Rank, ExcelDump, GDXEncoding, GDXRename, Cholesky, Eigenvalue, Eigenvector, Invert, OLS (Ordinary Least Squares), ExcelMerge, ExcelTalk, MSAppAvail, and ShellExecute. All of these are defined in the release notes.
    - These tools can be accessed at execute or compile time. The execute time commands are executeTool and put_utility 'execTool' and the compile time ones are $callTool and $hiddenCallTool.
    - Cursory treatment on using tools appears between the Users Guide and the release notes.
- In the reference file the contents of the tab that lists all files used has now been augmented. Previously it only listed files that had gams keyword identified commands within them. Now it lists all files including those that just include data entries.
- Added two new logical constructs that evaluate the presence of conjunctions (sAND) and disjunctions (sOR)
- Added information on largest and smallest values found in model setup in the RHS, Bounds and constraint matrix.

1.2 CONNECT

GAMS has been working on a platform independent means of interchanging data from and to other sources that it calls CONNECT. This allows data interchanges with spreadsheets, SQL databases, GDX and CSV files. The CONNECT framework consists of a database and then context specific agents that move data between a variety of file types and the database. One can also manipulate items within the CONNECT database.

To use CONNECT users must develop instructions to call CONNECT agents for some combination of actions involving a) reading data from various file types into the CONNECT database, b) transforming data in the CONNECT database, and c) writing data from the CONNECT database to several file types. Instructions are passed using YAML syntax.

According to the CONNECT write up there are available agents to:

- Read items from a GAMS run into the CONNECT database and pass database items into a GAMS run;

- read items from several file formats into the CONNECT database and also pass items back. The formats as of now are GDX, EXCEL, SQL and CSV;
- Change some domain information or deal with domain violations within an item in the CONNECT database;
- Filter the data eliminating selected set elements and or numerical entries within specified limits;
- Reorder sets when moving data from one item to another including possibly eliminating some sets (doing things defining y(I,j,k) from x(k,j,i) or constructing y(I,k) as the sum(j,x(I,j,k)) ;
- Execute custom python code over database items.

CONNECT is currently in Beta and it is likely that a non Beta version will not be released for at least 6 months.

CONNECT code can be embedded for execution in a GAMS program. For example the GAMS developed code to extract parameters dat and pro from an excel sheet and move into GAMS is as follows

```
Set i 'rows'
    j 'columns'
    k 'planes';
Parameter
   dat(k<,i<,j<) 'unprotected data table'
   pro(k,i,j)    'information sensitive cells';
* extract data from Excel workbook
$onEmbeddedCode CONNECT:
- PandasExcelReader:
    file: cox3.xlsx
    symbols:
      - name: dat
        range: Sheet1!A1
        rowDimension: 2
        columnDimension: 1
      - name: pro
        range: Sheet2!A1
        rowDimension: 2
        columnDimension: 1
- GAMSWriter:
    writeAll: True
$offEmbeddedCode
```

CONNECT is implemented in a manner requiring more programming than most other GAMS features. GAMS is working on adding some TOOL linkages for select purposes that might perform something like Rutherford's XLIMPORT or my PUT_toExcel. There is a connect file editor that appears within gams studio. One can find the instructions by searching the studio

documentation for the word connect. I found this instructions referred to things that didn't exist in terms of creating the initial file but if you create a file with the extension YAML then when you click on that file it automatically opens up the editor. In turn the editor offers you a choice of keywords and will lead you through the creation of a version of the YAML file.

A number of examples are included in the CONNECT write up.

1.3 TRANSFER

GAMS has also been working on a GAMS to programming language linking product called TRANSFER which is also available in Beta. It provides a way to exchange, augment and maintain data in a programming language like Python, R or Matlab. Using TRANSFER one can manipulate GAMS symbols(Sets, Aliases, Parameters, Variables and Equations), add new ones and read/write symbols. GAMS indicates in the writeup that TRANSFER's main focus is highly efficient data transfer between GAMS and the target programming language, while keeping those operations as simple as possible for the user.

Data are stored in the Pandas DataFrame within the programming language. This approach is used to accommodate users familiar with that data structure and also to provide access to large tool boxes resident in PYTHON and MATLAB for data operations. Transfer contains highly efficient bulk data transfer methods for reading from and writing to GAMS. There is a writeup that describes TRANSFER usage mainly within a PYTHON environment.

TRANSFER is currently in Beta and it is likely that a non Beta version will appear in the next major release.

1.4 STUDIO

Studio has been augmented in a number of ways.

- One can now pin a second view of a file either to the right or below of an edit window with the file in it. This is done through the View tab.
- Studio also now contains a feature called NAVIGATOR that allows you to navigate through a list of files in the project or in specified areas the computer and also narrow down the contents of that list. By default navigator shows a list of all files that are open in the project Explorer. Therein the first column contains the file name, and the second the path relative to the current project's directory if it differs. The third column contains additional information. Typing something into the input field filters this list. Using the arrow keys a result can be selected and confirmed with the Enter key. The filter also supports wildcards: to e.g. find all ".gdx" files one can input *.gdx. Navigator is accessed through the edit menu or through crtl+k. To me it is not all that intuitive ti juest start

using but there is a section on its use in the STUDIO writeup at https://www.gams.com/latest/docs/T_STUDIO.html#STUDIO_NAVIGATOR .

- Studio now contains an export feature wherein in the GDx viewer one can choose to export a viewed symbol to Excel.
- Since the last time we looked the concept of searching in a folder and included subdirectories has been embraced
- One not very well documented thing a user can do in STUDIO involves the ability to add a library of example files for use in a class or a modeling group. I covered this a long time ago the second newsletter and it involves something called a GLB file. It used to be that any user-defined files were put at the end of the list of things accessible in the library but recently Studio has now been altered so they appear first.

2 GAMSCHK-ANALYSIS

GAMS is devoting resources to building a model inspector which likely will result in some automatic and user driven procedures that will do things like portray model structure solution properties and identify obvious flaws. This would likely embody some of the GAMSCHK features. In discussions I found they had asked at least one GAMSCHK user about features they find useful. ANALYSIS was identified as one of these.

ANALYSIS works on a model that has been numerically specified and prepared for passage to a solver. It checks such a model for obvious flaws and that capability is quite valuable to us as it finds about 75% of the issues in models we see from students and others without ever having to numerically solve that model. In doing that it identifies obvious flaws that cause outcomes like infeasibility, zero variable values, unboundedness, along with warning about redundant constraints, variable that obviously cause constraints to be nonbinding and variables that will always have a zero solution value. Here we review the basic concepts used and some simple examples. We will do this in a more simple case that ANALYSIS can treat where for simplicity all the variables are nonnegative (note ANALYSIS also accommodates nonpositive and unrestricted variables).

So for our discussion consider the problem:

$$
\begin{aligned}
\text{Max} \quad & \sum_j c_j X_j & & & \\
\text{s.t.} \quad & \sum_j a_{\text{ij}} X_j & \leq & b_i & \text{for all i} \\
& \sum_j e_{\text{nj}} X_j & = & d_n & \text{for all n} \\
& \sum_j f_{\text{mj}} X_j & \geq & g_m & \text{for all m} \\
& X_j & \geq & 0 & \text{for all j}
\end{aligned}
$$

Certain values of these parameters can cause the model to:

1) be infeasible,

2) contain a set of variables that must be zero,

3) contain redundant constraints,

4) yield an unbounded solution, or

5) contain variables that are always zero.

Also note that ANALYSIS is programmed to first inspect each constraint in the model on an individual basis and then later each variable individually. Thus interactions between constraints and variables are not considered.

2.1 Errors it will find

2.1.1 Constraints that will be infeasible – a messaged error

Constraints can be setup so they are never going to be possible to satisfy causing infeasibility. Suppose that in the above model structure we have constraints like the following

```
4x1   +     2x2   <=  -2
6x1   +     x2    =   -1
-5x1  -     3x2   =   1
-x1   -     6x2   >=  2
x1    ,     x2    >=  0
```

In each of these cases there are no values of x1 and x2 that can make the model feasible. What ANALYSIS does is counts the incidence of signs on the nonnegative variables in an equation and when for a less than or equal to constraint all signs on the nonzero coefficients for variables are positive but the right hand side is negative then then it messages an error that the constraint will be infeasible. Similarly, for an equality constraint if all signs on the nonzero coefficients for the variables are positive and the RHS is negative again we have infeasibility. The same

happens if all the nonzero coefficient signs are negative and the RHS is positive. Finally for a greater than or equal to if all the nonzero coefficients are negative and the RHS is positive again we have infeasibility.

In terms of the algebraic problem above the tests are

If all nonzero a_{ij} >0 and b_i < 0 then the ith $\geq$ constraint will be infeasible

If all nonzero e_{nj} > 0 and d_n < 0 then the nth = constraint will be infeasible or if all nonzero e_{nj} < 0 and d_n > 0 then the nth = constraint will be infeasible

If all nonzero f_{mj} < 0 and g_m > 0 then the mth $\leq$ constraint will be infeasible.

2.1.2 A constraint that will be redundant – a messaged warning

Constraints can be setup so they are never going to be binding. Suppose that in the above model structure we have constraints like the following

```
-4x1  -     2x2   <  2
+x1   +     6x2   >  -2
x1    ,     x2    >  0
```

In each of these cases the any values of x1 and x2 can satisfy the constraint. What ANALYSIS does is counts the incidence of signs on the nonnegative variables in such an equation and when for a less than or equal to constraint all signs on the nonzero coefficients for variables are negative but the right hand side is positive then it messages a warning that the constraint is redundant. Similarly, for a greater than or equal to if all the nonzero coefficients are positive and the RHS is negative again we have redundancy.

In terms of the algebraic problem above the tests are

If all nonzero a_{ij} <0 and b_i > 0 then the ith $\leq$ constraint will be redundant

If all nonzero f_{mj} > 0 and $g_m \leq 0$ then the mth $\geq$ constraint will be redundant.

2.1.3 A constraint forcing variables to be zero – a messaged warning

Constraints can be setup so that all variables with nonzero coefficients in them must take on a value of zero. Suppose that in the above model structure we have constraints like the following

```
4x1   +     2x2   <  0
6x1   +     x2    =  0
-5x1  -     3x2   =  0
-x1   -     6x2   >  0
x1    ,     x2    >  0
```

In each of these cases there are no values of x1 and x2 other than zero that are feasible. What ANALYSIS does is counts the incidence of signs on the nonnegative variables in an equation

and when for a less than or equal to constraint all signs on the nonzero coefficients for variables are positive but the right hand side is zero then then it messages an warning that the constraint forces all variables appearing with nonzero coefficients in it to be zero. Similarly, for an equality constraint if all signs on the nonzero coefficients for the variables are positive and the RHS is zero again the constraint forces included variables to be zero. The same happens if all the nonzero coefficient signs are negative and the RHS is zero. Finally for a greater than or equal to if all the nonzero coefficients are positive and the RHS is zero again the constraint forces variables to zero.

In terms of the algebraic problem above the tests are

> If all nonzero $a_{ij} > 0$ & $b_i = 0$ then the ith $\geq$ constraint will force all included variables to be zero.
>
> If all nonzero $e_{nj} > 0$ & $d_n = 0$ then the nth = constraint will force all included variables to be zero or
>
> if all nonzero $e_{nj} < 0$ & $d_n = 0$ then the nth = constraint will force all included variables to be zero
>
> If all nonzero $f_{mj} < 0$ & $g_m = 0$ then the mth $\leq$ constraint will force all included variables to zero.

2.1.4 Variables that will be unbounded – a messaged error

Variables can be setup so that the model will be unbounded. Suppose that we have a variable like the following

```
Max    2x1                (or   Min -3x1)
      -4x1        <=  10
      +7x1        >=  12
        x1        >=  0
```

In this case the x1 can be made infinitely large with the objective function getting better. What ANALYSIS does is consider each nonnegative model variable identifying ones that: 1) as they increase contribute to a better value of the objective function; 2) have negative or zero coefficients in all less than or equal to constraints; 3) have zero coefficients in all equalities and 4) have positive or zero coefficients in all greater than or equal to constraints. When it finds such a case it issues an error that the variable will be unbounded.

In terms of the algebraic problem above the tests are

> If in a maximization problem for variables with $c_j>0$ that simultaneously have all nonzero $a_{ij} < 0$, $e_{nj} = 0$ and $f_{mj} > 0$ then the jth variable will be unbounded

Also in a minimization problem for variables with $c_j<0$ that simultaneously have all nonzero $a_{ij} < 0$, $e_{nj} = 0$ and $f_{mj} > 0$ then the jth variable will be unbounded

2.1.5 Variables that will be zero – a messaged warning

Variables can be setup so that they will always have an optimal value of zero. Suppose that we have a variable like the following

```
Max   -2x1                (or   Min 3x1)
      +4x1        <=  10
      -7x1        >=  12
        x1        >=  0
```

In this case making the x1 greater than zero causes the objective function to get worse and use up resources or increase requirements. What ANALYSIS does is consider each nonnegative model variable identifying ones that: 1) as they increase worsen the value of the objective function; 2) have positive or zero coefficients in all less than or equal to constraints; 3) have zero coefficients in all equalities; and 4) have negative or zero coefficients in all greater than or equal to constraints. When it finds such a case it issues a warning that the variable will always be zero.

In terms of the algebraic problem above the tests are

In a maximization problem for variables with $c_j<0$ that simultaneously have all nonzero $a_{ij} > 0$, $e_{nj} = 0$ and $f_{mj} < 0$ then the jth variable will always be zero at optimality

Also in a minimization problem for variables with $c_j>0$ that simultaneously have all nonzero $a_{ij} > 0$, $e_{nj} = 0$ and $f_{mj} < 0$ then the jth variable will always be zero at optimality

2.2 Why we use ANALYSIS or the equivalent features for blocks

These structural checks allow one to examine the algebraic formulation or its numerical counterpart without ever solving it. This finds well over half of the issues we have seen with new users setting up models. Generally the errors indicate bad data or algebra as do the warnings. While models with cases leading to warnings will solve in many cases the ANALYSIS warning reveal omissions or errors in the GAMS code.

While we presented the above in terms of ANALYSIS we use the same riles at the equation or variable bloc level in both BLOCKPIC and BLOCKLIST. The block level usage generally finds algebraic errors as the messages only come out when the flaws are present in all instances of an equation or variable (ie for all variables defined over the set j or for all equations defined over i in the above algebraic model).

In concluding note this does not discover flaws caused by interactions between variables and equations - ANALYSIS only finds flaws contained in individual variables or equations. In such

more complex cases one must refer to the unbounded and infeasibility treatments discussed in earlier newsletters and the POSTOPT procedure as discussed in chapter 17 of McCarl and Spreen.

3 Basic, Advanced and Combined courses offered soon

This year McCarl will again be teaching GAMS courses for basic and advanced users. These courses will be offered in early June via ZOOM. Dates, times and content are

- Basic to Advanced GAMS class June 5, 2023- June 12, 2023 (6 sessions of 6 hour days via ZOOM with the weekend off). The course spans from Basic topics to an Advanced GAMS class. Details are found at https://www.gams.com/courses/2023-06-basic-to-advanced-gams-modeling-mccarl/ .
- Basic GAMS class June 5, 2023- June 8, 2023 (4 sessions of 6 hours per days via ZOOM). The course starts assuming no GAMS background. Details are given at https://www.gams.com/courses/2023-06-basic-gams-course-mccarl/
- Advanced GAMS class June 7, 2023- June 12, 2023 (4 sessions of 6 hours per days via ZOOM with the weekend off). The course is for users who have a GAMS background. Details are found at https://www.gams.com/courses/2023-06-advanced-gams-modeling-mccarl/ .

Further information and other courses are listed on http://www.GAMS.com/courses.htm . Note I also give custom courses for individual groups a couple of times a year.

4 Unsubscribe or subscribe to future issues of this newsletter

Please unsubscribe through the web form available at: https://gams.us18.list-manage.com/unsubscribe?u=f1497c76718404eae593beb11&id=45ccea2ee0

Those who wish to subscribe to future issues can do this through the newsletter section of http://www.GAMS.com/maillist/index.htm.

This newsletter is not a product of GAMS Corporation although it is distributed with their cooperation.

May 5 , 2023

*