

Good Modeling Practices

Enhancing GAMS Model Self Documentation

Bruce A. McCarl

Specialist in Applied Optimization
Professor of Agricultural Economics, Texas A&M
Principal, McCarl and Associates

mccarl@tamu.edu
mccarl@bihs.net
agrinet.tamu.edu/mccarl

979-693-5694
979-845-1706

Good Modeling Practices

Why Worry and What can be done?

Relatively frequently I am involved with an exercise where I need to work with and thus understand a model built by someone else. Often I am struck by the **obscurity** with which some modelers code GAMS instructions (perhaps it's a move toward job security).

I feel it's largely **in a modelers hands as how much self documentation a piece of GAMS code contains**. I think there are **purposeful actions** one can undertake to improve documentation. This effort often **determines how easy it is to reuse or repair a model at a later time** or how easily a colleague (or in my case a consultant) can work with that code.

Several actions are possible

1. Use longer names and descriptions
2. Include comments on nature and sources of data
3. Include as much raw data as possible as opposed to externally calculated data.
4. Don't use an * as a set specification for input data.
5. Use sets to aid in readability
6. Format files to improve model readability.

Below I cover each of these.

Good Modeling Practices - Enhancing Documentation

Naming conventions - Use longer names and descriptions

One can radically affect readability by altering names of parameters, sets etc. Consider the example of Robert from the GAMS model library

```
Variables    x(p,tt)  production and sales
             s(r,tt)  opening stocks
             profit
Positive variables x, s;
Equations    cc(t)    capacity constarint
             sb(r,tt) stock balance
             pd       profit definition ;
cc(t)..      sum(p, x(p,t)) =l= m;
sb(r,tt+1).. s(r,tt+1) =e= s(r,tt) - sum(p, a(r,p)*x(p,tt));
pd.. profit =e= sum(t, sum(p, c(p,t)*x(p,t))
                 -sum(r, misc("storage-c",r)*s(r,t)))
                 + sum(r, misc("res-value",r)*s(r,"4")));
s.up(r,"1") = misc("max-stock",r);
```

After an hour of looking at it I reformatted it as follows

```
Variables    production(process,Quarters)  production and sales
             openstock(rawmaterial,Quarters)  opening stocks
             profit ;
Positive variables production, openstock;
Equations    capacity(quarter)              capacity constarint
             stockbalan(rawmaterial,Quarters)  stock balance
             profitacct                      profit definition ;
capacity(quarter)..
             sum(process, production(process,quarter)) =l= mxcapacity;
stockbalan(rawmaterial,Quarters+1)..
             openstock(rawmaterial,Quarters+1) =e=
             openstock(rawmaterial,Quarters)
             - sum(process, usage(rawmaterial,process)
                 *production(process,Quarters));
profitacct.. profit =e=
             sum(quarter,
                 sum(process, expectprof(process,quarter)
                     *production(process,quarter))
                 -sum(rawmaterial, miscdata("store-cost",rawmaterial)*
                     openstock(rawmaterial,quarter)))
             + sum(rawmaterial, miscdata("endinv-value",rawmaterial)
                 *openstock(rawmaterial,"winter"));
openstock.up(rawmaterial,"spring") = miscdata("max-stock",rawmaterial);
```

They have same algebra but different names. Which tells you more and which would you want to face in 5 years?

Good Modeling Practices - Enhancing Documentation

Naming conventions

Longer set names and set element names also help.

Bad Model– More Robert from model library

```
Sets  p      products      / low, medium, high /
      r      raw materials / scrap, new /
      tt     long horizon  / 1*4 /
      t(tt) short horizon  / 1*3 /
Table a(r,p) input coefficients
      low  medium  high
scrap   5      3      1
new     1      2      3
Table c(p,t) expected profits
      1      2      3
low    25    20    10
medium 50    50    50
high   75    80   100
```

Can be improved by

```
Sets  process  production processes available
      / low uses a low amount of new materials,
      medium uses a medium amount of new materials,
      high uses a high amount of new materials/
      rawmateral  source of raw materials / scrap, new /
      Quarters   long horizon / spring, summer, fall ,winter /
      quarter(Quarters) short horizon / spring, summer, fall /
Table  usage(rawmateral,process) input coefficients
      low  medium  high
scrap   5      3      1
new     1      2      3
Table  expectprof(process,quarters) expected profits
      spring summer fall
low    25    20    10
medium 50    50    50
high   75    80   100
```

Again which one makes more sense to you?

Good Modeling Practices - Enhancing Documentation

Naming conventions - Basic Point

GAMS allows 10 character long names and 80 characters of explanatory text defining each of the following items

sets	parameters	tables
scalars	variables	equations
models		

Use descriptive 10 character names. X is a wonderful variable name in theory, yet is lousy in practice as is A(i,j). Type a little more and be able to figure out your model later. Half the models in the GAMS model library have such failings. A little more typing always helps and it is **just not that hard**.

Let no item go undefined. Enter units, sources and descriptions.

Check for completeness with **onsymlist** making sure all names are somewhat apparent and all items have explanatory text.

Associate text with set element definitions and use the up to 31 character set element name capability. (Note names longer than 10 characters do not work well in multi column displays)

Remember only **You** can cause your GAMS code to be self documenting.

Good Modeling Practices - Enhancing Documentation

Setting up data – Include Comments on Nature and Sources

Questions I often ask when looking at a set of model data are:

Where did they come from? and
What characteristics such as units, and year of
applicability do those data possess?

Such questions certainly apply to tables of data in GAMS code. Often it's nice to go beyond the GAMS 80 character description of an item to **put in several lines of description identifying exactly what document a data set is from including page number, table number, year of applicability, units etc.** You can do this with comments (* in column 1) statements or \$ontext \$offtext sequences.

Again it's largely in your hands as to how much documentation a GAMS code contains. The documentation effort often determines how easy it is to reuse or repair a model at a later time or how easily a colleague can work with that code.

Good Modeling Practices - Enhancing Documentation

Setting up data - Raw versus calculated data

Modelers often face two choices with respect to data.
Enter **raw data into GAMS and transform** it to the extent needed inside GAMS

or

Externally process data entering the final results in GAMS.

The latter choice is often motivated by the availability of the raw data in a spreadsheet where those data could be manipulated before being put into GAMS.

My recommendation (as one who often updates models and needs to figure out the embodied assumptions)

put data in as close to the form it is collected into GAMS
(often from a spreadsheet ASCII text file save or print)
and then manipulate that data in the GAMS code

Justification for viewpoint. Over time spreadsheets and other data manipulation programs change, or get lost. Also often internal documentation in such programs is weak.

Good Modeling Practices - Enhancing Documentation

Setting up data

Don't use an * as a Set specification for input data

Another excerpt from the GAMS library code Robert

```
Table misc(*,r) other data
      scrap new
max-stock    400 275
storage-c     .5  2
res-value    15  25
...
pd.. profit =e= sum(t, sum(p, c(p,t)*x(p,t))
                -sum(r, misc("storage-c",r)*s(r,t)))
                + sum(r, misc("res-val",r)*s(r,"4"));
```

Here an * is used in specifying the first index position for the misc table. This tells GAMS anything goes in that position suppressing “domain checking”. However, suppose we mistyped the objective function reference to “res-value” as “res-val”. The GAMS code would compile and execute without error but would create a garbage result. On the other hand if we have entered another set and the table as follows

```
Set miscitem misc. input items
Table misc(miscitem,r) other data
```

then GAMS would have given error messages.

I say don't use * for set references in input data specifications.

I learned this through the school of hard knocks when a research associate used such an input strategy and had some small typing errors which caused important data to be ignored.

Good Modeling Practices - Enhancing Documentation

Setting up data – Make Sets work for you

Sets are used to address a family of similar items. You need to decide when to use a single versus multiple sets. There are cases when it is convenient to have an element in such a family and cases when it is not.

Suppose we have three grades of oil and three processes to crack it. Should we have one set with nine entries or make the item two dimensional. I would do the latter. Suppose we have a budget using fertilizer, seed, labor by month and water by month. Do we have a set with 26 items or two sets one with fertilizer, seed, labor and water and the other with month names plus annual. I would do the latter. I err on the side of being more extensive with set definitions.

Sets should contain items treated similarly in the problem (i.e., resources like fertilizer, seed, and energy), but when there are two items crossed (i.e., monthly availability of land, labor, and water involves month and resource) one should have **two sets**.

Good Modeling Practices - Enhancing Documentation

Setting up data – Make Sets work for you

One other set definition consideration involves the use of subsets. Sometimes it is desirable to have items which can be treated simultaneously in some places, but separately elsewhere. For example, when entering crop budgets one might wish to enter yield along with usage of inputs, land, labor, and water, in one spot yet treat those differently elsewhere (i.e., where variable inputs might be in one equation, yield balance in another, with water and labor availability in yet a third and fourth equation). Subsets allow this. Consider two models

In the Egypt model from the GAMS model library we have yields and input costs in two tables some 50 lines apart

```
Table yield (c,r)  yield for different commodities
      u-egypt  m-egypt  e-delta  m-delta  w-delta
*      ton      ton      ton      ton      ton
  wheat      1.29      1.36      1.39      1.404      1.36
  barley      1.41      1.26      1.33      .984      .96
```

```
Table cropdat(c,*)  seed protein starch misc costs and pestic data
      protein  starch  seed  misc  pest  n-fer  p-fer
*      %      %      ton  le   le   ton   ton
  wheat  .1    23.3  .075  12.0  .    0.054  0.015
  barley .1    23.3  .060   8.0  .    0.045  0.015
```

Good Modeling Practices - Enhancing Documentation

Setting up data - Make Sets work for you

(continued)

In my ASM model (**crop.10**) we have

```
TABLE CCCBUDDATA(ALLI, SUBREG, CROP, WTECH, CTECH, TECH) REGIONAL CROP  
BUDGET
```

```
                northeast.corn.DRYLAND.BASE.0  
corn                115.87  
CROPLAND            1.00  
LABOR                4.34  
nitrogen            24.69  
potassium            15.17  
phosporous          7.34
```

Where everything for a crop budget is together in concurrence with practices in data sources because we use subsets (set.10) to unravel the data

```
SET ALLI                ALL BUDGET ITEMS  
  / corn                , soybeans ,  
  cropland              , pasture  , labor,  
  nitrogen              , potassium, phosporous, trancost /  
set PRIMARY(ALLI)      PRIMARY PRODUCTS  
  / cotton              , soybeans /  
set INPUT(ALLI)        NATIONAL INPUTS  
  / nitrogen            , potassium , trancost , phosporous/  
set LANDTYPE(ALLI)     LAND TYPES  
  / cropland            , pasture  /
```

Subsets and a general set like ALLI allow one to both organize the input according to convenience with data sources and then deal with it efficiently in the model and report writer statements. You can also avoid the * in the input data sets.

Good Modeling Practices - Enhancing Documentation

Typing – Format files to improve model readability.

Enter the code in a fixed order

Data related sets

Data definitions organized by type of data possibly
intermixed with calculations

Variable and equation definitions

Algebraic equation definitions

Model and solve

Report writing sets and parameter definitions

Report writer calculations

Report writers displays

Keep the sections together

Format the code for readability using spacing and indents

Align item names, descriptions and definitions

Indent in sums, loops and ifs to delineate terms

Use blank lines to set things off

Don't split variables between lines in equations, but rather
keep them together with all their index positions.

Good Modeling Practices - Enhancing Documentation

Typing – Format files to improve model readability.

By typing you can make a file more readable using spacing, indents, set labels

Case A

```
Sets products available production process / low uses a low amount of new
materials, medium uses a medium amount of new materials, high uses a high amount
of new materials/
rawmaterial source of raw materials / scrap, new /
Quarters long horizon / spring, summer, fall ,winter /
quarter(Quarters) short horizon / spring, summer, fall /
Scalar mxcapacity maximum production capacity/ 40 /;
Variables production(products,Quarters) production and sales
openstock(rawmaterial,Quarters) opening stocks, profit ;
Positive variables production, openstock;
Equations capacity(quarter) capacity constraint, profitacct.. profit
=e= sum(quarter, sum(products, expectprof(
products,quarter) *production(products,quarter))-sum(
rawmaterial,miscdata("store-cost",rawmaterial)*openstock(rawmaterial
,quarter)))+ sum(rawmaterial, miscdata("endinv-value",rawmaterial)
*openstock(rawmaterial,"winter")));
```

Case B

```
Sets products available production process
/ low uses a low amount of new materials,
medium uses a medium amount of new materials,
high uses a high amount of new materials/
rawmaterial source of raw materials / scrap, new /
Quarters long horizon / spring, summer, fall ,winter /
quarter(Quarters) short horizon / spring, summer, fall /
Variables production(products,Quarters) production and sales
openstock(rawmaterial,Quarters) opening stocks
profit ;
Positive variables production, openstock;
Equations capacity(quarter) capacity constarint
stockbalan(rawmaterial,Quarters) stock balance
profitacct profit definition ;
profitacct.. profit =e=
sum(quarter,
sum(products, expectprof(products,quarter)
*production(products,quarter))
-sum(rawmaterial, miscdata("store-cost",rawmaterial)*
openstock(rawmaterial,quarter)))
+ sum(rawmaterial, miscdata("endinv-value",rawmaterial)
*openstock(rawmaterial,"winter")));
```

You may like case A, I don't. I find B much more readable. This matters when you have 1000 lines or more.