Pre-Conference Workshop

Steve Dirkse

GAMS Development Corporation

SDirkse@gams.com

www.gams.com

Minneapolis, 5 Oct 2013
Introduction

- **Workshop is aimed at novice users**
  - What are the core features and strengths of GAMS?
  - What does effective GAMS use look like?

- **Visit us at the GAMS booth**
  - Bring your questions, comments, etc.
  - Learn about the latest features, solvers, etc.

- **COIN-OR Cup celebration**
  - Monday night, 8:30-??
  - The Loon (500 N. 1st St)
Outline

• GAMS
  – GAMS at a Glance
  – Simple Example
  – GAMS/Base
  – Using GAMS Effectively
• Advanced features
  – Data Import/Export
  – Advanced Use of GAMS Solver Links
  – Extending the GAMS Syntax
Outline

- GAMS
  - GAMS at a Glance
    - Simple Example
    - GAMS/Base
    - Using GAMS Effectively
- Advanced Features
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
GAMS at a Glance

Algebraic Modeling System

- Facilitates formulation of mathematical optimization problems expressed in an algebraic notation
  - Simplified model building

- Provides links to appropriate state-of-the-art external algorithms
  - Efficient solution process
GAMS at a Glance

General Algebraic Modeling System

- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corp.
- GAMS Software GmbH

- Broad academic & commercial user community and network
GAMS’ Fundamental concepts

- **Platform independence**
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements

![Supported Platforms Diagram]
GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements

30+ Integrated Solvers

- ALPHAEC
- MOSEK
- XPRESS
- XA
- MINOS
- CONOPT
- BARON
- LINDOLOBAL
- CPLEX
- BDMLP
- GUROBI
- COIN-OR
- DICOPT
Binary Data Exchange

- Fast exchange of data
- Syntactical check on data before model starts
- Data Exchange at any stage (Compile and Run-time)
- Platform Independent
- Direct GDX interfaces and general API
- Scenario Management Support
- Full Support of Batch Runs

GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements
GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements

Declaration of..
- Sets
- Parameters
- Variables
- Equations
- Models
- …

Procedural Elements like…
- loops
- if-then-else
- …
Outline

- **GAMS**
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
  - Using GAMS Effectively
- **Advanced Features**
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
A Transportation Model

Seattle (350)
San Diego (600)
Chicago (300)
New York (325)
Topeka (275)
Minimize Transportation cost
subject to Demand satisfaction at markets
Supply constraints

A Transportation Model

San Diego 600
Seattle 350

New York 325
Topeka 275
Chicago 300

Transportation cost:
- San Diego to New York: 2.5
- San Diego to Topeka: 1.4
- San Diego to Chicago: 1.8
- Seattle to New York: 2.5
- Seattle to Topeka: 1.8
- Seattle to Chicago: 1.7
Model Formulation

Indices: $i$ (Canning plants)

$\quad j$ (Markets)

Decision variables: $x_{ij}$ (Number of cases to ship)

Parameter: $c_{ij}$ (Transport cost per case)

\[ \text{min } \sum_i \sum_j c_{ij} \cdot x_{ij} \] (Minimize total transportation cost)

subject to

\[ \sum_j x_{ij} \leq sup_i \quad \forall i \] (Shipments from each plant $\leq$ supply capacity)

\[ \sum_i x_{ij} \geq dem_j \quad \forall j \] (Shipments to each market $\geq$ demand)

\[ x_{ij} \geq 0 \quad \forall i, j \]

\[ i, j \in \mathbb{N} \]
GAMS Algebra

Variables
\[ x(i,j) \quad \text{shipment quantities in cases} \]
\[ z \quad \text{total transportation costs in thousands of dollars} \]

Positive Variable \( x \);

Equations
\[ \text{cost} \quad \text{define objective function} \]
\[ \text{supply}(i) \quad \text{observe supply limit at plant } i \]
\[ \text{demand}(j) \quad \text{satisfy demand at market } j \]

\[ \text{cost} \quad z = \sum (i,j), c(i,j) \times x(i,j) \]
\[ \text{supply}(i) \quad \sum (j, x(i,j)) = l = a(i) \]
\[ \text{demand}(j) \quad \sum (i, x(i,j)) = g = b(j) \]

Model transport /all/ ;
# Outline

**GAMS**
- GAMS at a Glance
- Simple Example
- GAMS/Base
  - Using GAMS Effectively

**Advanced Features**
- Data Import/Export
- Advanced Use of GAMS Solver Links
- Extending the GAMS Syntax
GAMS at a Glance

The GAMS/BASE Module

• Compiler and Execution System

• GAMS IDE (Windows)

• Documentation + Model libraries

• GDX Utilities

• Free Solvers/Solver Links
GAMS at a Glance

The GAMS/BASE Module

• Compiler and Execution System
• GAMS IDE (Windows)
• Documentation + Model libraries
• GDX Utilities
• Free Solvers/Solver Links
Integrated Development Environment

- Project management
- Editor / Syntax coloring / Spell checking
- Launching and monitoring of (multiple) GAMS processes
- Listing file / Tree view / Syntax-error navigation
- Solver selection / Option selection
- GDX viewer
  - Data cube
  - Data export (e.g. to MS Excel)
  - Charting facilities
- Model libraries
- Documentation
- Diff for GDX and Text
GAMS at a Glance

The GAMS/BASE Module

- Compiler and Execution System
- GAMS IDE (Windows)
- Documentation + Model libraries
- GDX Utilities
- Free Solvers/Solver Links
Documentation

• **Distributed Documentation**
  – GAMS Users Guide
  – Expanded GAMS Users Guide (McCarl)
  – Solver Manuals
  – GAMS Utility Manuals

• **Wikis**
  – Support Wiki  [http://support.gams-software.com](http://support.gams-software.com)
  – Interfaces Wiki  [http://interfaces.gams-software.com](http://interfaces.gams-software.com)
• **Groups**
  – Google Group [http://groups.google.de/group/gamsworld](http://groups.google.de/group/gamsworld)

• **Newsletter**
  – Release List

• **Search all GAMS Websites**
  [http://www.gams.com/search.htm](http://www.gams.com/search.htm)
Distributed Model Libraries

- **GAMS Model Library**
  - Example and user-contributed models
  - Very often used as templates
  - Tests for
    - Solver robustness and correctness
    - Backward compatibility

- **GAMS Test Library**
  - Transparent and reproducible Quality Assurance Tests
  - Tests for
    - Solver correctness
    - Special functions
    - GAMS utilities
Distributed Model Libraries

• **GAMS Data Utilities Library**
  – Demonstration of the various utilities interfacing GAMS with other applications
  – E.g. gdxxrw, mdb2gms, sql2gms

• **GAMS EMP Library**
  – Examples for the use of Extended Mathematical Programming

• **Practical Financial Optimization Models**
  Models of the book
  
  “PRACTICAL FINANCIAL OPTIMIZATION – A Library of GAMS Models”

  by Consiglio, Nielsen and Zenios
GAMS at a Glance

The GAMS/BASE Module

- Compiler and Execution System
- GAMS IDE (Windows)
- Documentation + Model libraries
- GDX Utilities
- Free Solvers/Solver Links
Gams Data eXchange

Binary Data Exchange

- Fast exchange of data
- Syntactical check on data before model starts
- Data Exchange at any stage (Compile and Run-time)
- Platform Independent
- Direct GDX interfaces and general API
- Scenario Management Support
- Full Support of Batch Runs

GDX Tools

- Invert
- IDE
- GDX Viewer
- GDXrank
- GDX2HAR/HAR2GDX
- GDXmerge
- GDXdump
- GDXcopy
- GDXdiff
- MDB2GMS
- GDX2XLS
- GDXxrw
- GDXAPI
GAMS at a Glance

The GAMS/BASE Module

- Compiler and Execution System
- GAMS IDE (Windows)
- Documentation + Model libraries
- GDX Utilities
- Free Solvers/Solver Links
GAMS at a Glance

The GAMS/BASE Module

Free Solvers:

• Convert
• EMP/JAMS, DE, NLPEC
• BENCH, EXAMINER, GAMSCHK
• BDMLP, LS, and MILES
• KESTREL (Remote Solver Execution on NEOS Servers)
• COIN-OR: Cbc, IpOpt, BonMin, Couenne, …
• Soplex, Scip (academic only)
• All other solvers in limited versions
## Outline

<table>
<thead>
<tr>
<th>GAMS</th>
<th>Advanced Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>– GAMS at a Glance</td>
<td>– Data Import/Export</td>
</tr>
<tr>
<td>– Simple Example</td>
<td>– Advanced Use of GAMS Solver Links</td>
</tr>
<tr>
<td>– GAMS/Base</td>
<td>– Extending the GAMS Syntax</td>
</tr>
<tr>
<td>– Using GAMS Effectively</td>
<td></td>
</tr>
</tbody>
</table>
## Outline

<table>
<thead>
<tr>
<th>GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>– GAMS at a Glance</td>
</tr>
<tr>
<td>– Simple Example</td>
</tr>
<tr>
<td>– GAMS/Base</td>
</tr>
<tr>
<td>– Using GAMS Effectively</td>
</tr>
</tbody>
</table>

### Advanced Features

<table>
<thead>
<tr>
<th>Advanced Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Data Import/Export</td>
</tr>
<tr>
<td>– Advanced Use of GAMS Solver Links</td>
</tr>
<tr>
<td>– Extending the GAMS Syntax</td>
</tr>
</tbody>
</table>
Then …

In Table 17.1 we list sizes and attributes of representative models that are “large” in the sense that they are near the limit of what is practical on a personal computer, along with the model generation time (GAMS) and solution time (solver), both in minutes. These examples were run on an 8 MHz AT with an 80287 coprocessor and 640K of RAM. The times shown are to give you a rough idea of what is possible: these are not precisely controlled benchmarks, and we have a host of performance improvements in mind for the near future.

Table 17.1: Problem Characteristics

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Rows</th>
<th>Number of Columns</th>
<th>Number of Nonzeros</th>
<th>Generation Time *</th>
<th>Solution Time *</th>
<th>Iterations</th>
<th>Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>DINAMICO</td>
<td>318</td>
<td>425</td>
<td>4156</td>
<td>3.0</td>
<td>30.1</td>
<td>628</td>
<td>MINOS</td>
</tr>
<tr>
<td>SARF</td>
<td>532</td>
<td>542</td>
<td>3949</td>
<td>37.7</td>
<td>115.8</td>
<td>2775</td>
<td>MINOS</td>
</tr>
<tr>
<td>FERTD</td>
<td>458</td>
<td>2968</td>
<td>7252</td>
<td>11.4</td>
<td>28.3</td>
<td>1368</td>
<td>ZOOM</td>
</tr>
<tr>
<td>CAMCGE</td>
<td>243</td>
<td>280</td>
<td>1356</td>
<td>0.8</td>
<td>7.0</td>
<td>189</td>
<td>MINOS</td>
</tr>
<tr>
<td>GANGES</td>
<td>274</td>
<td>357</td>
<td>1405</td>
<td>1.8</td>
<td>7.3</td>
<td>187</td>
<td>MINOS</td>
</tr>
<tr>
<td>VENECM</td>
<td>168</td>
<td>258</td>
<td>953</td>
<td>0.9</td>
<td>7.6</td>
<td>600</td>
<td>ZOOM</td>
</tr>
<tr>
<td>EITPP</td>
<td>281</td>
<td>618</td>
<td>3168</td>
<td>4.0</td>
<td>25.3</td>
<td>1551</td>
<td>ZOOM</td>
</tr>
</tbody>
</table>

* Measured in minutes.

* The problem is too big for MINOS. ZOOM was used instead.

* A nonlinear problem. 63% of the non-zeroes are nonlinear.

* A nonlinear problem. 58% of the non-zeroes are nonlinear.

* A mixed binary problem, with 55 binary variables (solved with a relative termination criterion of 10%).

* A linear problem, solved using XMP which is contained within ZOOM.
... and now

<table>
<thead>
<tr>
<th></th>
<th>Type</th>
<th>s in 1988</th>
<th>s in 2013</th>
<th>Improvement Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>camcge</td>
<td>NLP</td>
<td>468</td>
<td>0.031</td>
<td>15097</td>
</tr>
<tr>
<td>dinamico</td>
<td>LP</td>
<td>1986</td>
<td>0.125</td>
<td>15888</td>
</tr>
<tr>
<td>egypt*</td>
<td>LP</td>
<td>1758</td>
<td>0.015</td>
<td>117200</td>
</tr>
<tr>
<td>fertd*</td>
<td>MIP</td>
<td>2382</td>
<td>0.062</td>
<td>38419</td>
</tr>
<tr>
<td>ganges</td>
<td>NLP</td>
<td>546</td>
<td>0.109</td>
<td>5009</td>
</tr>
<tr>
<td>sarf</td>
<td>LP</td>
<td>9210</td>
<td>0.139</td>
<td>66259</td>
</tr>
<tr>
<td>yemcem*</td>
<td>MIP</td>
<td>510</td>
<td>0.140</td>
<td>3643</td>
</tr>
</tbody>
</table>

* 1988 solver ZOOM, 2008 solver CPLEX 11.0.1
Improvements on all Frontiers

- **Solver Technology**
  - Updates for existing solver
  - New solvers

- **Productivity Tools**
  - Databases, spreadsheets
  - Specialized visualization tools
  - IDE improvements
  - Grid computing

- **Interfaces**
  - Gams Data eXchange
  - Using GAMS from other applications
## Outline

<table>
<thead>
<tr>
<th>GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>- GAMS at a Glance</td>
</tr>
<tr>
<td>- Simple Example</td>
</tr>
<tr>
<td>- GAMS/Base</td>
</tr>
<tr>
<td>- Using GAMS Effectively</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Data Import/Export</td>
</tr>
<tr>
<td>- Advanced Use of GAMS Solver Links</td>
</tr>
<tr>
<td>- Extending the GAMS Syntax</td>
</tr>
</tbody>
</table>
GDXXRW

- Read and write Excel spreadsheet data
- Can read multiple ranges in a spreadsheet and write the data to a GDX file
- Can read from a GDX file and write the data to different ranges in a spreadsheet
- Examples in the GAMS Data Library

Hands-On
GDXMRW

- Import/export data between GAMS and MATLAB
- Call GAMS models from MATLAB
- Get results back in MATLAB
- Gives MATLAB users the ability to use all the optimization capabilities of GAMS
- Allows visualization of GAMS models directly within MATLAB
**GDXRRW**

- GDXRRW bridges the gap between R and GAMS (import/export data between GAMS and R)

- Fits into the ecosystem of existing GDX utilities

- Presents data in a natural form for R users

- More information:  

Source: [http://blog.modelworks.ch](http://blog.modelworks.ch)
Load from GDX

Compile Time:

$gdxIn transSol.gdx  // open file for reading
$load              // list file content
$load i            // load symbol i
$load jj=j          // load symbol j as jj
$loadDC a b         // load a & b domain controlled
$load[DC]M k        // load symbol k, merge content
$load[DC]R l        // load symbol l, replace content
$gdxIn              // close open file

Execution Time:

execute_load 'transSol.gdx' a;

put_utility 'gdxin' / 'transSol.gdx';
execute_load b;
# Outline

- **GAMS**
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
  - Using GAMS Effectively
- **Advanced Features**
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
Model transport /all/ ;
Option solvelink = { %Solvelink.ChainScript%,
                  %Solvelink.CallScript%,
                  %Solvelink.CallModule%,
                  %Solvelink.AsyncGrid%,
                  %Solvelink.AsyncSimulate%,
                  %Solvelink.LoadLibrary%};

solve transport using lp minimizing z;

• ChainScript [0]: Solver process, GAMS vacates memory
  + Maximum memory available to solver
  + protection against solver failure (hostile link)
  - swap to disk
Solvelink Option – cont.

• Call{Script [1]/Module [2]}: Solver process, GAMS stays live
  + protection against solver failure (hostile link)
  + no swap of GAMS database
  - file based model communication

• LoadLibrary [5]: Solver DLL in GAMS process
  + fast memory based model communication
  + update of model object inside the solver (hot start)
  - not supported by all solvers
Solving Scenarios

transport.gms (LP) solved 500 times with CPLEX:

```gams
Loop(s,
    d(i,j) = dd(s,i,j);
    f = ff(s);
    solve transport using lp minimizing z;
    rep(s) = transport.objval;
);
```

<table>
<thead>
<tr>
<th>Setting</th>
<th>Solve time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solvelink=%Solvelink.ChainScript%</td>
<td>52.221</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.CallModule%</td>
<td>37.366</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.LoadLibrary%</td>
<td>03.252</td>
</tr>
</tbody>
</table>
Gather-Update-Solve-Scatter (GUSS)

<table>
<thead>
<tr>
<th>Setting</th>
<th>Solve time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solvelink=%Solvelink.ChainScript%</td>
<td>52.221</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.CallModule%</td>
<td>37.366</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.LoadLibrary%</td>
<td>03.252</td>
</tr>
<tr>
<td>GUSS</td>
<td>01.046</td>
</tr>
</tbody>
</table>

- Update model data instead of matrix coefficients/rhs
- Hot start (keep the model hot inside the solver and use solver’s best update mechanism)
- Save model generation and solver setup time
- Model rim unchanged from scenario to scenario
- Apriori knowledge of all scenario data

Hands-On
Solution Pool

- Several solver links can write out alternative solutions to GDX: AlphaECP, ANTIGONE, BARON, CBC, CPLEX, GloMIQO, Gurobi, SCIP, Xpress
- BARON, CPLEX, and Xpress also offer functionality to explicitly search for alternative solutions
- See GAMS Model Library model solnpool

----
142 PARAMETER xcostX cost structure by solution

totcost     tcost     fcost

file1  499.000  219.000  280.000
file2  512.000  212.000  300.000
file3  985.000  355.000  630.000
Outline

<table>
<thead>
<tr>
<th>• GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>– GAMS at a Glance</td>
</tr>
<tr>
<td>– Simple Example</td>
</tr>
<tr>
<td>– GAMS/Base</td>
</tr>
<tr>
<td>– Using GAMS Effectively</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>• Advanced Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Data Import/Export</td>
</tr>
<tr>
<td>– Advanced Use of GAMS Solver Links</td>
</tr>
<tr>
<td>– Extending the GAMS Syntax</td>
</tr>
</tbody>
</table>
Function Libraries

- Allows users to import functions from an external library into a GAMS model
- Imported functions can be used in the same way as intrinsic GAMS functions
- Some function libraries are included in the GAMS distribution
- Users can create their own libraries using an open programming interface (simple examples written in C, Delphi and Fortran come with every GAMS system)
- To make a library available call
  \[
  $\text{FuncLibIn} \ <\text{IntLibName}> \ <\text{ExtLibName}>
  \]
- Declare functions similar to sets, parameters, ..., :
  \[
  \text{Function} \ <\text{IntFuncName}> \ /<\text{IntLibName}>.\<\text{FuncName}>/;
  \]
Function Libraries – Included Examples

- FITfclib
  - FITPACK from P. Dierckx
  - One and two dimensional spline interpolation
- LSAdclib
  - Use sampling routines from Lindo inside GAMS
  - Requires GAMS/Lindo license (or runs in limited demo mode)
- PWPcclib
  - Piecewise polynomial function evaluation
- STOdclib
  - Random deviates, probability density functions, cumulative density functions and inverse cumulative density functions
  - E.g., ChiSquare, Gumbel, Logistic, Rayleigh, …
- TRIcclib, TRIdclib, TRIfclib
  - Simple examples compiled and as source code written in C, Delphi and Fortran respectively
Function Libraries – Interface

- `int LibInit(
  abcRec_t *abc,    // in handle
  const int version, // in library version
  char *msg)          // out message
```

- `int <FUNCTIONNAME>(
  abcRec_t *abc,    // in handle
  const int DR,     // in derivative request
  const int args,   // in number of arguments
  const double x[], // in arguments
  double *f,        // out function value
  double g[],       // out gradient
  double h[],       // out hessian
  void *cb,         // in error callback
  void *usermem)    // in user memory for error callback
```
**Stochastic Programming in GAMS**

- The Extended Mathematical Programming (EMP) framework is used to replace parameters in the model by random variables.

- Support for Multi-stage recourse problems and chance constraint models.

- Easy to add uncertainty to existing deterministic models, to either use specialized algorithms or create Deterministic Equivalent (new free solver DE).
Excursus: EMP, what?

With new modeling and solution concepts do not:

- overload existing GAMS notation right away!
- attempt to build new solvers right away!

But:

- Use existing language features to specify additional model features, structure, and semantics
- Express extended model in symbolic (source) form and apply existing modeling/solution technology
- Package new tools with the production system

→ Extended Mathematical Programming (EMP)
JAMS: a GAMS EMP Solver

- EMP Information
- Original Model
- Reformulated Model
- Solving using established Algorithms
- Solution

Mapping Solution into original space

Translation

Viewable