OR2006  
GAMS  
Workshop  

Michael R. Bussieck  
MBussieck@gams.com  

Franz Nelißen  
F Nelissen@gams.com  

GAMS Software GmbH  
www.gams.de  

OR 2006  
Karlsruhe, Germany,  September 5, 2006
Welcome/Agenda

GAMS Development / GAMS Software
Working with GAMS – A Guided Tour
Model Development
Model Deployment and Maintenance
Agenda

GAMS Development / GAMS Software

- Working with GAMS – A Guided Tour
- Model Development
- Model Deployment and Maintenance
GAMS Development / GAMS Software

- Roots: Research project World Bank 1976
- Pioneer in Algebraic Modeling Systems used for economic modeling
- Went commercial in 1987
- Offices in Washington, D.C and Cologne

- Professional software tool provider, not a consulting company
- Operating in a segmented niche market
- Broad academic & commercial user base and network
## Typical Application Areas *

<table>
<thead>
<tr>
<th>Agricultural Economics</th>
<th>Applied General Equilibrium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Engineering</td>
<td>Economic Development</td>
</tr>
<tr>
<td>Econometrics</td>
<td>Energy</td>
</tr>
<tr>
<td>Environmental Economics</td>
<td>Engineering</td>
</tr>
<tr>
<td>Finance</td>
<td>Forestry</td>
</tr>
<tr>
<td>International Trade</td>
<td>Logistics</td>
</tr>
<tr>
<td>Macro Economics</td>
<td>Military</td>
</tr>
<tr>
<td>Management Science/OR</td>
<td>Mathematics</td>
</tr>
<tr>
<td>Micro Economics</td>
<td>Physics</td>
</tr>
</tbody>
</table>

* Illustrative examples in the GAMS Model Library
Network of Application Partners
Agenda

- GAMS Development / GAMS Software
- Working with GAMS – A Guided Tour
- Model Development
- Model Deployment and Maintenance
GAMS at a Glance


Design Principles:
- Balanced mix of declarative and procedural elements
- Open architecture and interfaces to other systems
- Different layers with separation of:
  - model and data
  - model and solution methods
  - model and operating system
  - model and interface
More GAMS Features

- State of art professional modeling technology
- Increased productivity
- Robust and scalable
- Rapid development
- Broad Network
- Large model libraries with templates
- Multiple Model Types
- Platform / Solver independence:
  - Protection of investments
  - Maintainable models
System Overview

Connectivity Tools
• Uniform Data Exchange:
  – ASCII
  – GDX (ODBC, SQL, XLS, XML)
• GDX Tools
• Data API
• Ext. programs
  – EXCEL
  – MATLAB
  – GNUPLOT, ...
  – C, Delphi, ...

Interactive API/Batch

User Interfaces

GAMS Language Compiler and Execution System

Solvers
LP-MIP-QCP-MIQCP-NLP-MINLP-CNS-MCP-MPEC
MPSGE, global, and stochastic optimization

Productivity Tools
• Integrated Development Environment (IDE)
• Model Debugger and Profiler
• Model Libraries
• Data Browser
• Charting Engine
• Benchmarking
• Deployment System
• Quality Assurance and Testing

BARON, COIN, CONOPT, CPLEX, DECIS, DICOPT, KNITRO, LGO, MINOS, MOSEK, OQNLP, PATH, SNOPT, XA, XPRESS, ...
Hands-on! Installing GAMS

Welcome to the GAMS 22.2 Setup Wizard

Select the components you want to install. Clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

McCarty User Guide

Copy license file

installing: gamsinst
installing: gamslc
installing: gamspack
installing: gamsunzip
installing: gamslice.txt

Unpacking GAMS ...

- estimated disk blocks needed: 20480, available: 923181
- executing: ./gamsunpack

Exporting PATH=PATH=/home/susanne/euro2006::PATH
Hands-on! Testing the installation

This file contains the basic data and definition of the surface water system. Data is complete for year 1988. Some parameters could be computed for future years using growth rates provided in this file, others had to be estimated and entered. Enter the year for which the setup is desired in set isr (set isr should have only one entry).
Hands-on! Testing the installation

GAMS/Cplex  Apr 21, 2006 LNX.CP.CP 22.2 031.034.041.LX3 For Cplex 10.0
Cplex 10.0.1, GAMS Link 31
Cplex licensed for 1 use of lp, qp, mip and barrier, with 4 parallel threads.

Reading data...
Starting Cplex...
Tried aggregator 1 time.
LP Presolve eliminated 280 rows and 805 columns.
Aggregator did 652 substitutions.
Reduced LP has 1794 rows, 5113 columns, and 33006 nonzeros.
Presolve time = 0.04 sec.
Initializing dual steep norms . . .

Iteration log . . .
Iteration:  1  Scaled dual infeas = 2955567.467575
A few Words about GAMS Syntax

Minimize Transportation cost
subject to Demand satisfaction at markets
Supp supply constraints
\[ \sum_{c,p: (c,p) \in \mathcal{N}} t_{\text{cost}} \cdot d_{\text{ist}}(c, p) \cdot x^c_p \rightarrow \min \]

\[ \sum_{c,p: (c,p) \in \mathcal{N}} x^c_p \leq sup(c) \quad \forall c \]

\[ \sum_{c,p: (c,p) \in \mathcal{N}} x^c_p \geq dem(p) \quad \forall p \]

\[ x^c_p \geq 0 \quad \forall c, p : (c, p) \in \mathcal{N} \]
### GAMS Syntax – GAMS Algebra

#### Variables

```plaintext
x(i,j)  shipment quantities in cases
z      total transportation costs in thousands of dollars;
```

#### Positive Variable `x`;

#### Equations

```plaintext
cost       define objective function
supply(i)   observe supply limit at plant i
demand(j)   satisfy demand at market j;
```

```plaintext

cost ..    z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..  sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..  sum(i, x(i,j))  =g=  b(j) ;
```

#### Model `transport /all/`;

1: 1 | Insert
### GAMS Syntax – cont.

#### Symbols:
- **Sets**
  - `Set I /cat,dog,ding1*ding10/`

- **Parameters**
  - `Parameter life(I) life count / cat 7 /

- **Variables**
  - `Integer Variable x(I) number to purchase;`

- **Equations**
  - `Equation e(I) relate something;`

- **Models**
  - `Model animallife / e, some, more;`

- **ASCII Output Files**
  - `File fx some file / 'c:\text.txt';`

#### Statements:
- **Declaration + Data statement**
  - `Set I /cat,dog/;`

- **Data Assignments**
  - `life('dog')=life('cat')-1; x.lo(I)=1;`

- **Equation Definition**
  - `e(I).. Sqr(x(I)) =l= log(life(I));`

- **Programming Flow Control**
  - `loop(I, put fx I.tl);`

- **Option statement**
  - `Option reslim=10;`
Hands-On! Inspect trnsport.gms

• IDE:
  File → Model Library trnsport
  Hit F9 or Click

• Unix:
  $ gamslib trnsport
  $ vi trnsport.gms
  $ gams trnsport
  $ vi trnsport.lst
Hands-on! IDE - A Guided Tour

- IDE Project Management
- Documentation
- Model Library
- Editor
- Solver Selection
- Option Selection
- Listing file/Tree view/Error navigation
- GDX Viewer
  - Data cube
  - Export to Excel
  - Graphs
Hands-on! Create result file

```gams
file fx / result.txt /;
loop((i,j),
    put fx i.tl j.tl x.l(i,j) /;
);
file fx / result.txt /;
loop((i,j)$(x.l(i,j)>0),
    put fx i.tl:0 ',', j.tl:0 ',', x.l(i,j):10:4 /;
);
file fx / result.txt /;
fx.pc=5; fx.lw=0; fx.nw=10; fx.nd=4;
loop((i,j)$(x.l(i,j)>0),
    put fx i.tl j.tl x.l(i,j) /;
);
```
The Mean-Variance Model

**Markowitz (1952), Nobel prize 1990**

**Given**

Some investments $x_i$ with historical data
- **Rewards** = Expected returns of investments: $\mu_i$
  (Mean of historical returns)
- **Risk**: Variance of investments $Q_{i,j}$

**Goal**

Balance risk $r$ of portfolio against expected returns of portfolio

Minimize variance $\nu$ of portfolio for a given target return $r$

Talk: Wednesday, 18:00, Geb. 11.40, R. 202
Mean-Variance Model Algebra

<table>
<thead>
<tr>
<th><strong>Variance of Portfolio</strong></th>
<th>$\text{Min } \sum_{i=1}^{I} \sum_{j=1}^{J} x_i Q_{i,j} x_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target return</strong></td>
<td>$\text{s.t. } \sum_{i=1}^{I} \mu_i x_i \geq r$</td>
</tr>
<tr>
<td><strong>Budget constraint</strong></td>
<td>$\sum_{i=1}^{I} x_i = 1$</td>
</tr>
<tr>
<td><strong>No short sales</strong></td>
<td>$x_i \geq 0$</td>
</tr>
</tbody>
</table>
### Hands-On: Inspect qcp1.gms

<table>
<thead>
<tr>
<th>IDE:</th>
<th>Unix:</th>
</tr>
</thead>
</table>
| File → Model Library  
qcp1             | $ gamslib qcp1             |
| File → Open in project dir  
“qpdata.inc”    | $ vi qcp1.gms              |
| Select qcp1.gms  
Hit F9 or Click  | $ vi qpdata.inc            |
|                | $ gams qcp1                |
|                | $ vi qcp1.lst              |
Hands-on! Modifications to qcp1

- Make return a variable
  
  ```gams
  Variable ret;
  retcon.. sum(s, mean(s)*x(s)) =e= ret;
  ret.lo = totmean*1.25;
  ```

- Select a different solver
  
  ```gams
  Option qcp=conopt;
  ```

- Minimize/Maximize return
  
  ```gams
  solve qcp1 min ret using qcp;
  solve qcp1 max ret using qcp;
  ```

- Select QCP=cplex and primal simplex for solving the QCP
  
  ```gams
  qpmethod 1 (in file cplex.opt)
  ```

- Complete model in qcpeuro06.gms
Solver Option Files

- Pass solver specific options
  - E.g. tolerances, limits, algorithm selection

- Solver option file `solver.opt` e.g. `cplex.opt` with solver specific options (one per line)

- Activate solver option file
  - `optfile=1` on command line/parameter window
  - `<modelname>.optfile=1;` before solve

- Multiple option files:
  - `solver.opt` optfile=1
  - `solver.op2` optfile=2
  - ...
  - `solver.999` optfile=999
Special Solvers

- Solvers that do not solve the problem:
  - CONVERT
    - Converts the model into different formats
  - AMPL/LINGO
    - Converts model into AMPL/LINGO syntax and calls the other system to solve the problem
  - EXAMINER
    - Checks the quality of a solution found by a different solver
  - BENCH
    - Benchmarking solver
Instructions

In order to use the GMS2XX translation service which is based on the "solver" GAMS/CONVERT you have to attach your model to an email and send it to our translation server at gms2xx@gamsworld.org. You specify the language in the subject line, for example

Subject: GAMS

At the moment we support the following languages:

- AMPL
- BARON
- CplexLP
- CplexMPS
- GAMS
- LGO
- LINGO
- MINOPT
- ALL (this creates scalar versions of all supported languages, listed above)
• Translation of MP Model into Scalar Model  
  – List of Variables/Equations  
• Advantages:  
  – Syntax for Scalar Models almost identical for different Modeling Languages (easy Translation)  
  – Hides proprietary Information  
• Seamless Modeling System Connection  
  – For example: GAMS/AMPL with Kestrel (NEOS)
Set I Products /P1*P2/
J Cutting Patterns /C1*C2/;

Parameter c(J)    cost of raw material /C1 1, C2 1/
    cc(J)   cost of change-over of knives /C1 0.1, C2 0.2/
    b(I)    width of product roll-type I /P1 460, P2 570/
    nord(I) number of orders of product type I /P1 8, P2 7/
    Bmax    width of raw paper roll /1900/
    Delta   tolerance for width / 200/
    Nmax    max number of products in cut /  5/
    bigM    max number of repeats of any pattern / 15/;

Variable    y(J)    cutting pattern
            m(J)    number of repeats of pattern j
            n(I,J) number of products I produced in cut J
            obj objective variable;

Binary Variable y; Integer Variable m, n;

Equation defobj, max_width(J), min_width(J), max_n_sum(J),
    min_order(I), cut_exist(J), no_cut(J);

defobj..      sum(j, c[j]*m[j] + cc[j]*y[j]) =e=  obj;
max_width(j).. sum(i, b[i]*n[i,j])                =l=  Bmax;
min_width(j).. sum(i, b[i]*n[i,j]) + Delta       =g=  Bmax;
max_n_sum(j).. sum(i, n[i,j])                    =l=  Nmax;
min_order(i).. sum(j, m[j]*n[i,j])               =g=  nord[i];
cut_exist(j).. y[j]                              =l=  m[j];
no_cut(j) ..  m[j]                               =l=  bigM*y[j];

m.up[j] =    bigM; n.up[i,j] =    nmax;

model trimlosstrimloss /all/;
solve trimloss minimize obj using minlp;
* MINLP written by GAMS Convert
Variables b1,b2,i3,i4,i5,i6,i7,i8,x9;
Binary Variables b1,b2;
Integer Variables i3,i4,i5,i6,i7,i8;
e1.. 0.1*b1 + 0.2*b2 + i3 + i4 - x9 =E= 0;
e2.. 460*i5 + 570*i7 =L= 1900;
e3.. 460*i6 + 570*i8 =L= 1900;
e4.. 460*i5 + 570*i7 =G= 1700;
e5.. 460*i6 + 570*i8 =G= 1700;
e6.. i5 + i7 =L= 5;
e7.. i6 + i8 =L= 5;
e8.. i3*i5 + i4*i6 =G= 8;
e9.. i3*i7 + i4*i8 =G= 7;
e10.. b1 - i3 =L= 0;
e11.. b2 - i4 =L= 0;
e12.. -15*b1 + i3 =L= 0;
e13.. -15*b2 + i4 =L= 0;

* set non default bounds
i3.up = 15; i4.up = 15; i5.up = 5;
i6.up = 5; i7.up = 5; i8.up = 5;
Model m / all /;
Solve m using MINLP minimizing x9;

# MINLP written by GAMS Convert
var b1 binary;
var b2 binary;
var i3 integer >= 0, <= 15;
var i4 integer >= 0, <= 15;
var i5 integer >= 0, <= 5;
var i6 integer >= 0, <= 5;
var i7 integer >= 0, <= 5;
var i8 integer >= 0, <= 5;

minimize obj:
    0.1*b1 + 0.2*b2 + i3 + i4;

subject to

e2: 460*i5 + 570*i7 <= 1900;
e3: 460*i6 + 570*i8 <= 1900;
e4: 460*i5 + 570*i7 >= 1700;
e5: 460*i6 + 570*i8 >= 1700;
e6: i5 + i7 <= 5;
e7: i6 + i8 <= 5;
e8: i3*i5 + i4*i6 >= 8;
e9: i3*i7 + i4*i8 >= 7;
e10: b1 - i3 <= 0;
e11: b2 - i4 <= 0;
e12: -15*b1 + i3 <= 0;
e13: -15*b2 + i4 <= 0;
Hands-on! Running Special Solvers

- Run *Convert* and inspect `gams.gms` and `dict.txt`

- Run *Convert* with the following option file `convert.opt`
  ```
  ampl qcp1.mod
  ```
  and inspect `qcp1.mod`

- Run *Examiner* with the following option file `examiner.opt`
  ```
  subsolver cplex
  primalfeastol 1e-30
  ```

- Run *Bench* with the following option file `bench.opt`
  ```
  solvers cplex cplex.1 conopt minos snopt xpress mosek
  ```
Solver Links

- **Supported Solvers:**
  - GAMS Support includes Solver Support
  - Standardized Solver Interface
    - Return Codes, Limits, Interrupts, …
    - Common Solver Attributes (e.g. time) through GAMS options
  - Specific Solver Options through Option”file”
- Standardized Solver Interface allows “hassle free” replacement of solvers:
  
  ```
  option nlp=conopt;
  ```
- IO Library (C, Fortran, Delphi) provides access to
  - Matrix, Function/Derivative Evaluator, …
Linking your Solver to GAMS
THE COMPLETE NOTES

Don't Panic !!
Hands-on! More Modeling

• Minimum Investment in stock: $x(s) \geq stockmin$ or $x(s)=0$
  Scalar stockmin /0.05/; Binary Variable bx(stocks);
  equations bxup(stocks), bxlo(stock);  
  bxup(s).. x(s) =l= bx(s); 
  bxlo(s).. x(s) =g= bx(s)*stockmin;
  model /... ,bxup,bxlo/;
  solve qcp1 min z using miqcp;

• Dealing with infeasibilities:
  – Add slack variables
    Variable sl; sl.lo=0;
    retcon.. sum(s, mean(s)*x(s)) + sl =e= ret;
    solve qcp1 min sl using miqcp;
  – Use solver (e.g. Cplex) that supports feasibility mode feasopt 1
Hands-on! More Modeling

- Efficient Frontier

```gams
scalar step;
solve qcp1 max ret using miqcp;
set scen / s1*s10 /;
step = ret.l/(card(scen)-1);
parameter rep, repx; ret.lo = 0;
loop(scen,
solve qcp1 min z using miqcp;
   rep(scen, s) = x.l(s);
   repx(scen,'ret') = ret.l;
   repx(scen,'var') = z.l;
   ret.lo = ret.lo+step);
```

Efficient Frontier
Imagine…

.. you have to solve 1,000’s of independent scenarios..

.. and you can do this very rapidly for little additional money…

.. without having to do lots of cumbersome programming work..

Grid Computing
What is Grid Computing?

A pool of connected computers managed and available as a common computing resource

• Effective sharing of CPU power
• Massive parallel task execution
• Scheduler handles management tasks
• E.g. Condor, Sun N6 Grid Engine, Globus
• Can be rented or owned in common
• Licensing & security issues
Economics of Grid Computing

- Yearly cost, 2-CPU workstation: $5200
  - Hardware: $1200
  - Software: $4000

- Hourly cost on the grid: $2/cpu
  - $1/hour for CPU time (to grid operator)
  - $1/hour for software (GAMS, model owner)

- 1 workstation:
  - ~ 2600 hrs grid time or
  - ~ 50 hrs/week grid time

- Up-front vs. deferred, as-needed costs
GAMS & Grid Computing

• **SOLVE Statement Grid enabled**

• **Scalable:**
  – support of massive grids, **but also**
  – multi-cpu / multiple cores desktop machines
  – “1 CPU - Grid”

• (Almost) **Platform independent**

• **Only minor changes** to model required

• **Separation** of model and solution method
  → Model stays **maintainable**
Simple Serial Solve Loop

```
loop(scen,
    solve qcp1 min z using miqcp;
    rep(scen, s) = x.l(s);
    repx(scen,'ret') = ret.l;
    repx(scen,'var') = z.l;
    ret.lo = ret.lo+step);
```

How do we get to parallel and distributed computing?
GRID specific enhancements

1. Submission of Jobs

2. “Grid Middleware”
   – Distribution of Jobs
   – Job execution

3. Collection of Solutions

4. Processing of results
Job Submission Loop

Parameter h(scen) store the instance handle;
* turn on grid option
qcp1.solvelink = 3;
loop(scen,
    solve qcp1 min z using miqcp;
* save instance handle
    h(scen) = qcp1.handle
    ret.lo = ret.lo+step);

LOG:
--- LOOPS scen = s1
--- 323 rows 322 columns 1,094 non-zeroes
--- 237,702 nl-code 147 nl-non-zeroes
--- 160 discrete-columns
--- qcp1.gms(16371) 5 Mb
--- Submitting model qcp1 with handle grid135000002
“Grid”- Middleware (PC)

: gams grid submission script
: arg1 solver executable
: 2 control file
: 3 scratch directory
: gmscr_nx.exe processes the solution and produces 'gmsgrid.gdx'
: note: %3 will be the short name, this is needed because
:       the START command cannot handle spaces or "..."
:       before we use %~3 will strip surrounding "...
:       makes the name short
: gmsrerun.cmd will resubmit runit.cmd

```cmd
echo @echo off            > %3runit.cmd
echo %1 %2               >> %3runit.cmd
echo echo gmscr_nx.exe %2     >> %3runit.cmd
echo mkdir %3finished    >> %3runit.cmd
echo exit                >> %3runit.cmd
```

```cmd
@start /b /BELOWNORMAL %3runit.cmd ^> nul > %3gmsrerun.cmd
start /b /BELOWNORMAL %3runit.cmd > nul
exit
```
Solution Collection Loop

Repeat
  loop(scen$h(scen),
    if(handlestatus(h(scen))=2,
      qcp1.handle = h(scen); execute_loadhandle qcp1;
      rep(scen, s) = x.l(s);
      repx(scen,'ret') = ret.l; repx(scen,'var') = z.l;
      h(scen) = 0));
    if(card(h), execute 'sleep 1');
  until card(h) = 0 or timeelapsed > 100;

LOG: ...
--- GDXin=c:\work\mod\225b\grid137000002\gmsgrid.gdx
--- grid_qmeanvar.gms(154) 3 Mb
--- Removing handle grid137000002
--- GDXin=c:\work\mod\225b\grid137000003\gmsgrid.gdx
--- Removing handle grid137000003
--- GDXin=c:\work\mod\225b\grid137000007\gmsgrid.gdx
--- ...
Results for 4096 MIPS on Condor Grid

- Submission started Jan 11, 16:00
- All jobs submitted by Jan 11, 23:00
- All jobs returned by Jan 12, 12:40
  - 20 hours wall time, 5000 CPU hours
  - Peak number of CPU's: 500

Talk: Thursday, 08:30
“Chemie-Hörsaal 1”
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMS Development / GAMS Software</td>
</tr>
<tr>
<td>Working with GAMS – A Guided Tour</td>
</tr>
<tr>
<td>Model Development</td>
</tr>
<tr>
<td><strong>Model Deployment and Maintenance</strong></td>
</tr>
</tbody>
</table>
Important Principles

- Deployed models have often 15+ years lifecycle
  - Changing environment:
    - hardware, operating system, interface (GUI/data)

- Backward compatibility
- Platform/Solver/Interface Independence
  - Model benefits from
    - Advanced hardware
    - Advanced solver technology

- Reduced Total Cost of Ownership (TCO)
Flow of Data

Data Model I
- Application in control of data processing
- No direct data access

Data Model II
- Large Scale/Raw data exchange GAMS↔DB
- Control Data GAMS↔Application
Input/Output through ASCII Files

• ASCII Input Data
  – Part of model input ($\texttt{include file.txt}$)
  – Posix Utilities are part of GAMS Windows System
    • Platform independent data file preparation
    • \texttt{sed, awk, grep, cut, …}
      $\texttt{call cut -d, -f1,3- file.txt > filenew.txt}$

• ASCII File Output
  – GAMS Put Facilities

• Hands-on! qcpascii.gms.
GAMS Data eXchange

- **GAMS Data eXchange (GDX):**

  ![Diagram of data exchange](image)

  - Complements the ASCII text data input
  - Advantages:
    - Fast exchange of data (factor >20)
    - Syntactical check on data before model starts
    - Compile-time and Run-time Data Exchange
    - Platform Independent

- Hands-on! qcpgdx.gms.
GDX Tools

- GAMS
- GDX
- API
- IDE
- GDX Viewer
- gdxdiff
- gdxmerge
- gdxsplit
- gdxdump
- gdxxxrw (MS Office)
Calling GAMS from an Application

Creating Input for GAMS Model
Callout to a GAMS Process/Executable
Reading Output from GAMS Model

- Works from basically every environment
  - Web application (server side)
  - Application Builder
    - Oracle, Eclipse, .NET, …
    - Regular Programming language C++, Java, VB, …
  - MS Office Application / VBA

- Hands-on! qcpexcel.xls
Data Contracts

• Application provides data in GAMS readable format (DM I)
• **Data Contract** between GAMS and Data Source (DM II)
  – Responsibility for Model and Data often in different hands
  – Format, location, layout
  – Data Transfer Process (SQL query) relies on Data Contract

• Hands-on! qcpdc.gms
• Data Contract in GAMS
  $call gdxr stockprice.xls o=sp.gdx dset=days rdim=1 rng=sp!a2 dset=stocks cdim=1 rng=sp!b1 par=val dim=2 rng=sp!a1

• Data Contract in Excel
  $call gdxr stockprice.xls o=sp.gdx index=index!a1
Object Code for GAMS Models

• GAMS Save and Restart Facility
  – Store (parts of a) model in an *object* file
  – Complements Source Code Model distribution
  – Convenient packaging
  – Protection of Intellectual Property (code/data)
  – Attractive deployment prices

• Hands-on!
• Protect Model Equations: qcpsave.gms qcprestart.gms
• Protect “Algorithm”: qcpsaveX.gms qcprestartX.gms
A few Words about Maintenance

**Optimization**
- Takes Longer than one is willing to wait
- It will eventually fail

**Application**
- Real Time
- Always need a *Solution* to Problem

- Key for support/maintenance
  - Catch problems before a model is solved
    - Implement Data Error checks
  - Reproduce the problem offline
    - Get hold of Instance (`dumpopt=11`)  
    - Solver related problems in confidential models
      - Get scalar Model using solver `CONVERT` 

- Hands-on! Error checking `qc perror.gms`
Summary

• 30+ Years Experience in Modeling
  – Strong views on modeling process (*The GAMS Way*)
    • Development
    • Deployment
    • Maintenance
  – Less than 5% of modeling/optimization projects do not fit the GAMS way
  – Use of GAMS and its productivity tools (after potentially steep learning curve)
    • Increases productivity of model building
    • Reduces total cost of ownership for model client
    • Opens doors to a large network of GAMS developers and clients with modeling needs
Contacting GAMS

Europe:
GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne
Germany
Phone: +49 221 949 9170
Fax: +49 221 949 9171
http://www.gams.de

USA:
GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA
Phone: +1 202 342 0180
Fax: +1 202 342 0181
http://www.gams.com