GAMS Pre Conference Workshop

Lutz Westermann LWestermann@gams.com

Clemens Westphal CWestphal@gams.com

GAMS Software GmbH

GAMS Development Corporation www.gams.com

EuroOR 2013 IMPACT

GAMS

Rotterdam, September 3, 2013

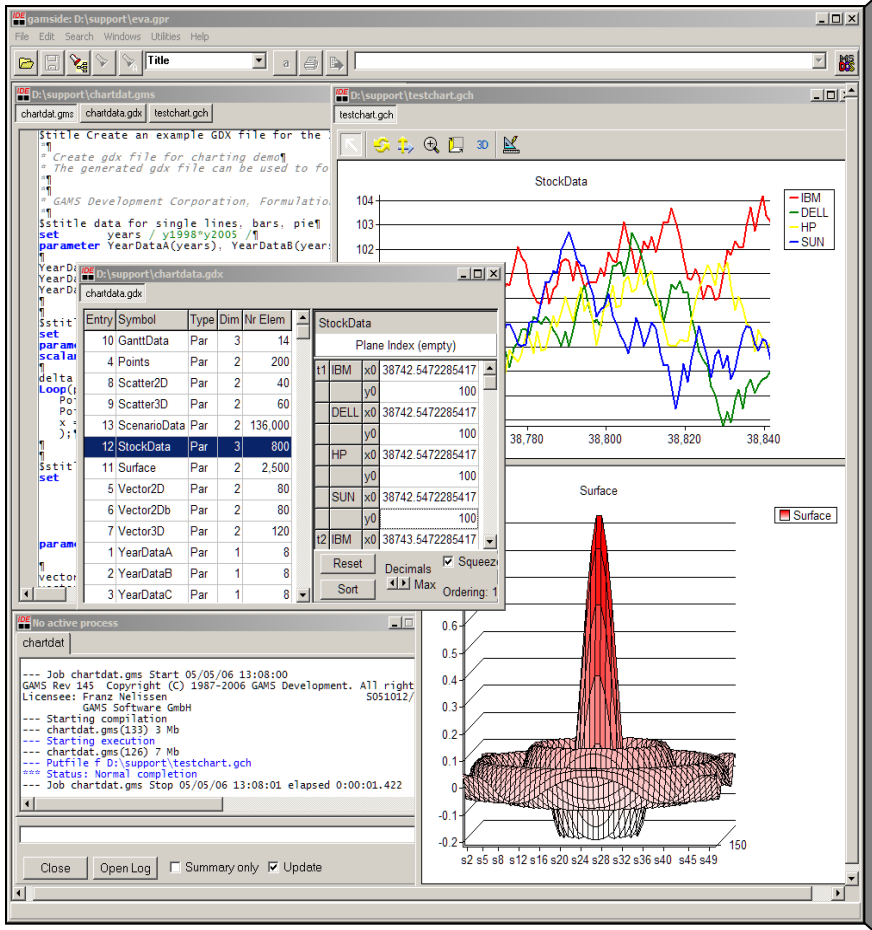
Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

GAMS at a Glance



Algebraic Modeling System

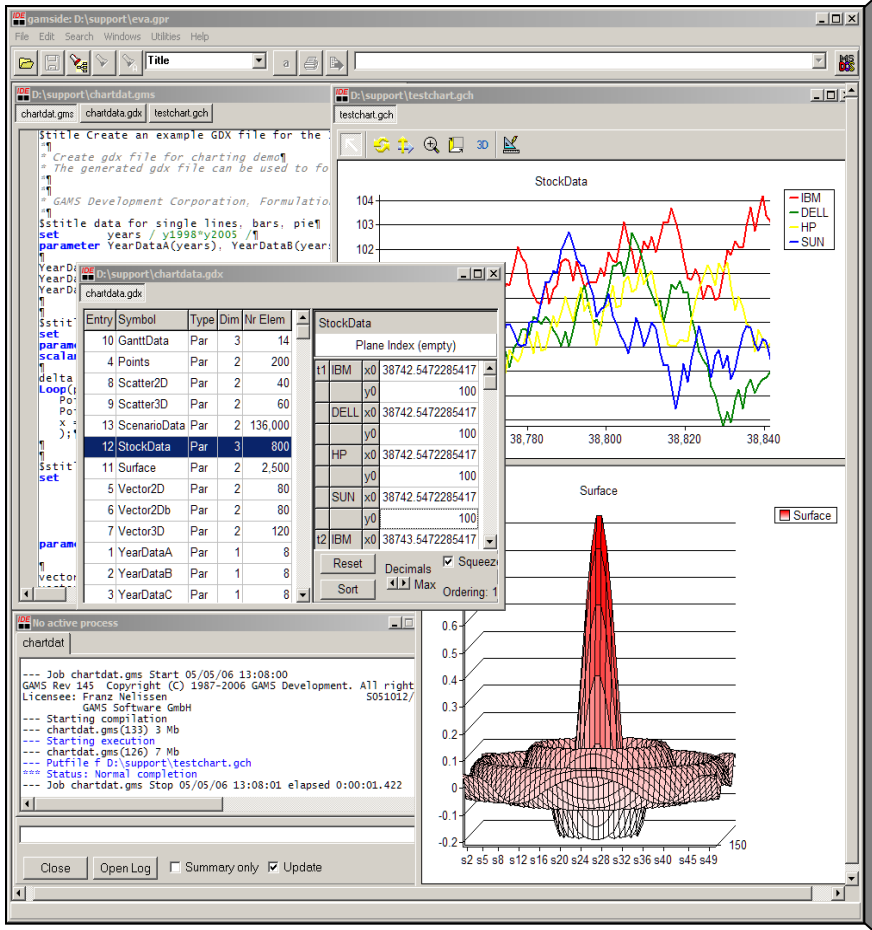
- Facilitates to formulate mathematical optimization problems similar to algebraic notation

  ➔ Simplified model building

- Provides links to appropriate state-of-the-art external algorithms

  ➔ Efficient solution process

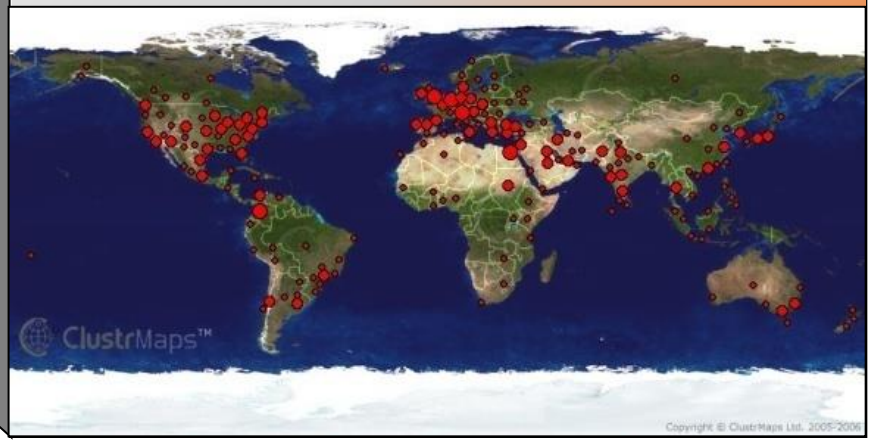
GAMS at a Glance



General **A**lgebraic **M**odeling **S**ystem

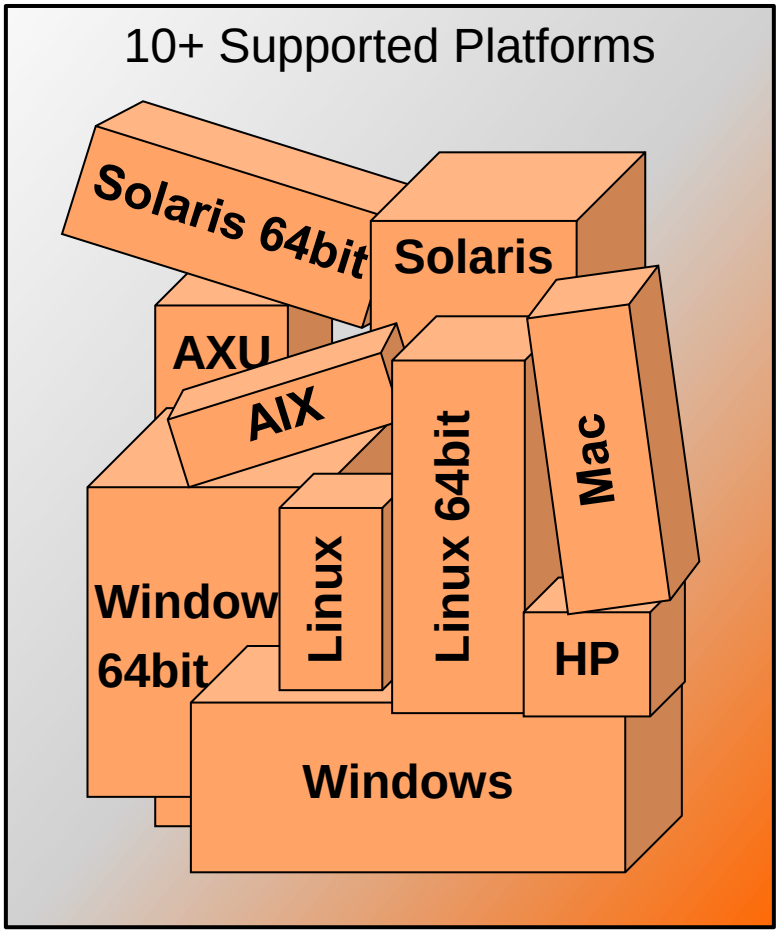
- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corp.
- GAMS Software GmbH

- Broad academic & commercial user community and network



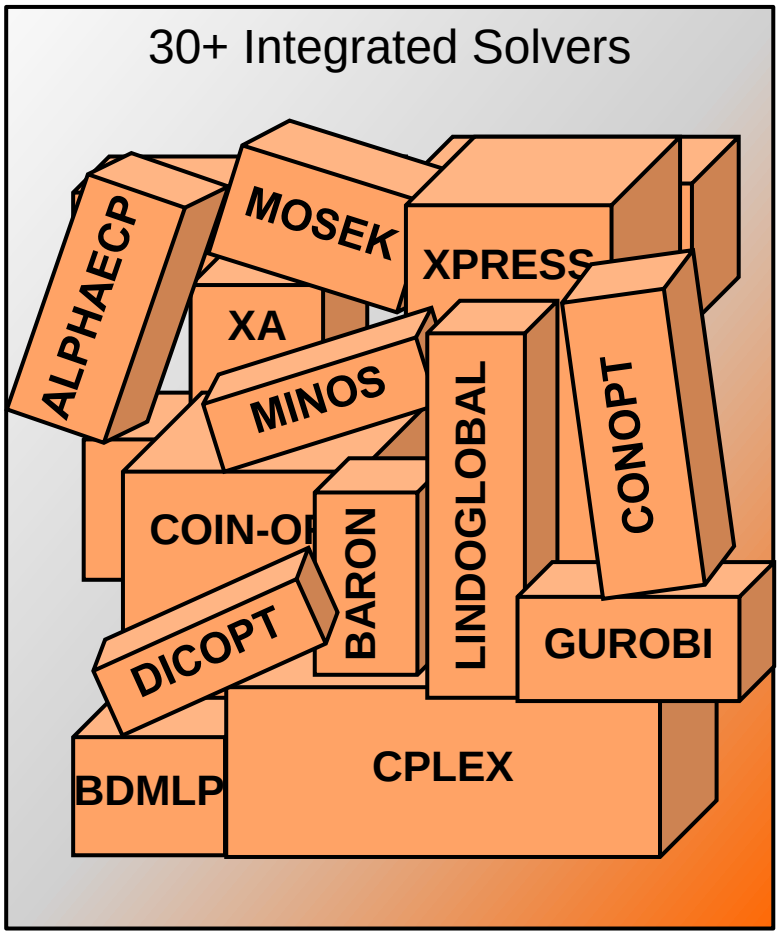
GAMS' Fundamental concepts

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



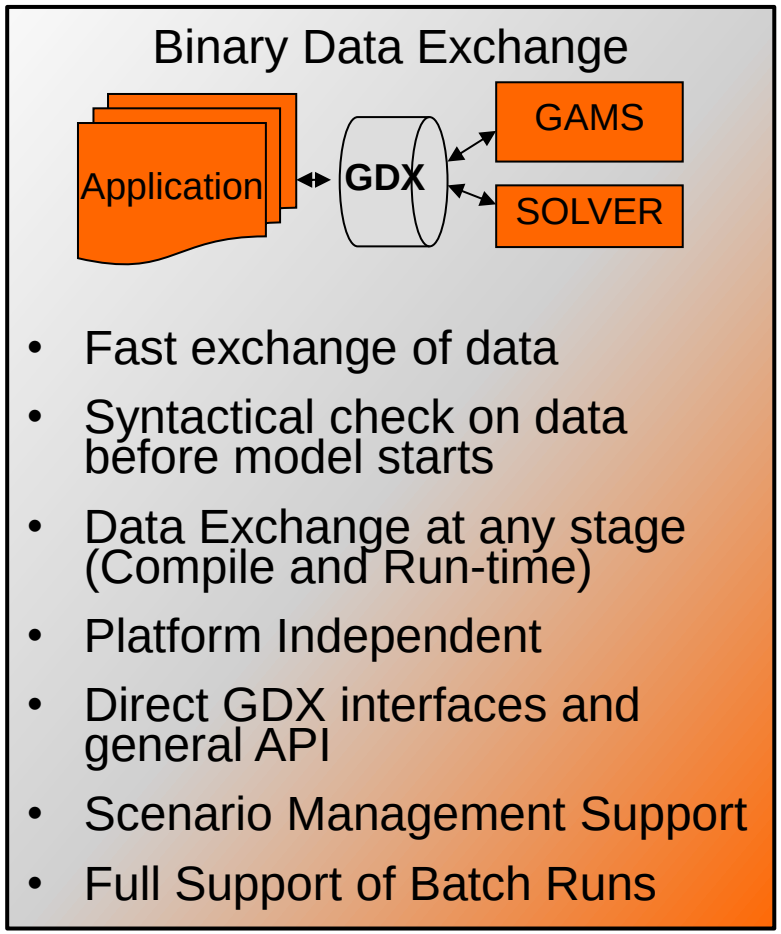
GAMS' Fundamental concepts

- 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



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…

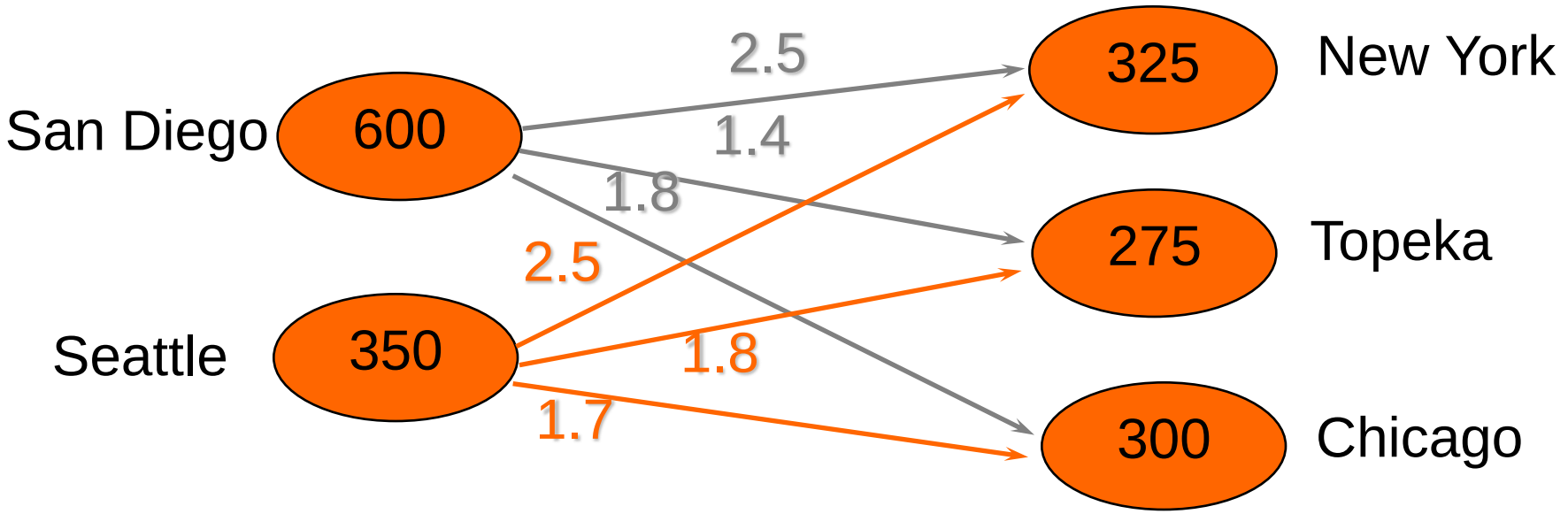
- **loops**
- **if-then-else**
- **…**

Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

A Transportation Model

A Transportation Model



Minimize	Transportation cost
subject to	Demand satisfaction at markets
	Supply constraints

Model Formulation

Indices: i (Canning plants)

j (Markets)

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

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

$\min \quad \sum_i \sum_j c_{ij} \cdot x_{ij}$ (Minimize total transportation cost)

subject to

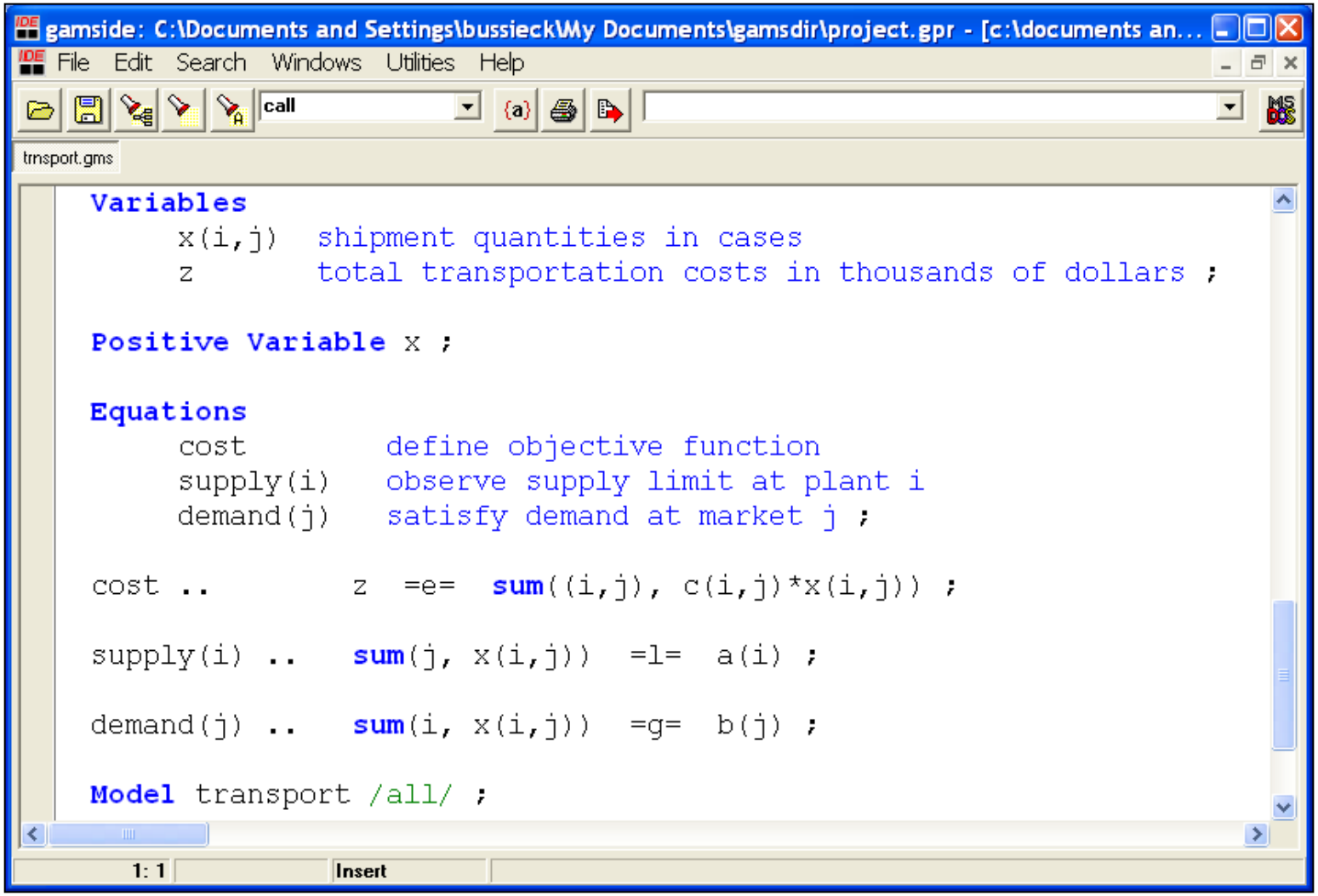
$\sum_j x_{ij} \leq sup_i \quad \forall\, i$ (Shipments from each plant $\leq$ supply capacity)

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

$x_{ij} \geq 0 \qquad \forall i, j$

$i, j \in \mathbb{N}$

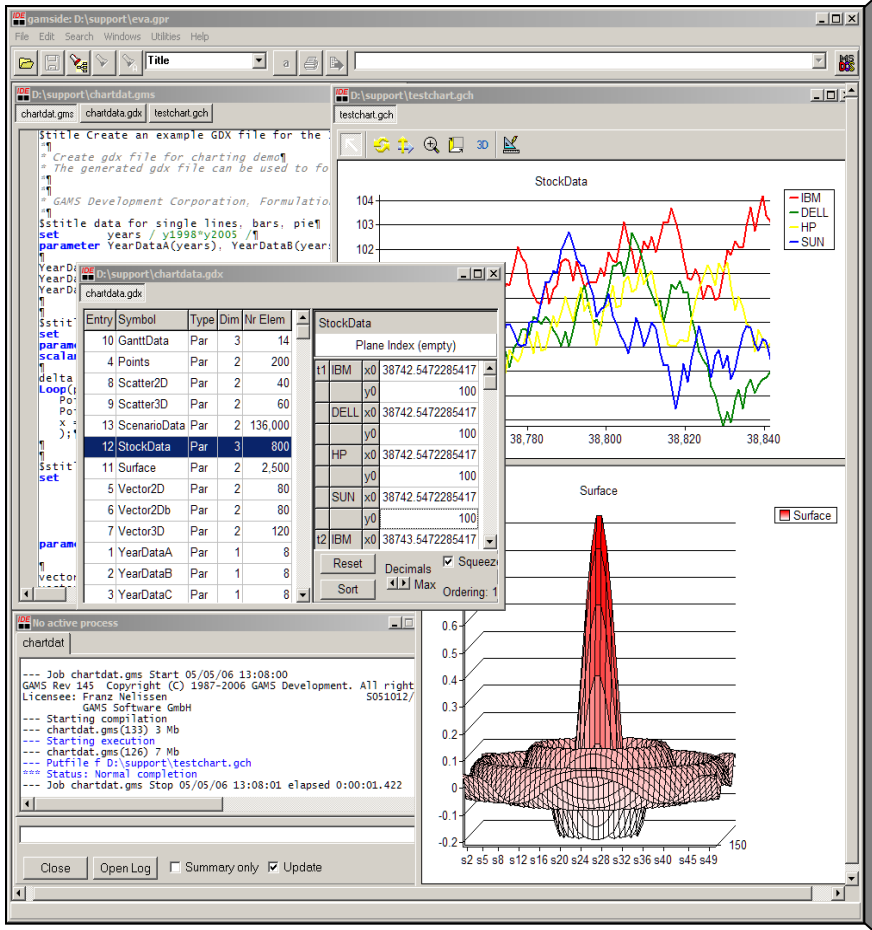
GAMS Algebra



Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

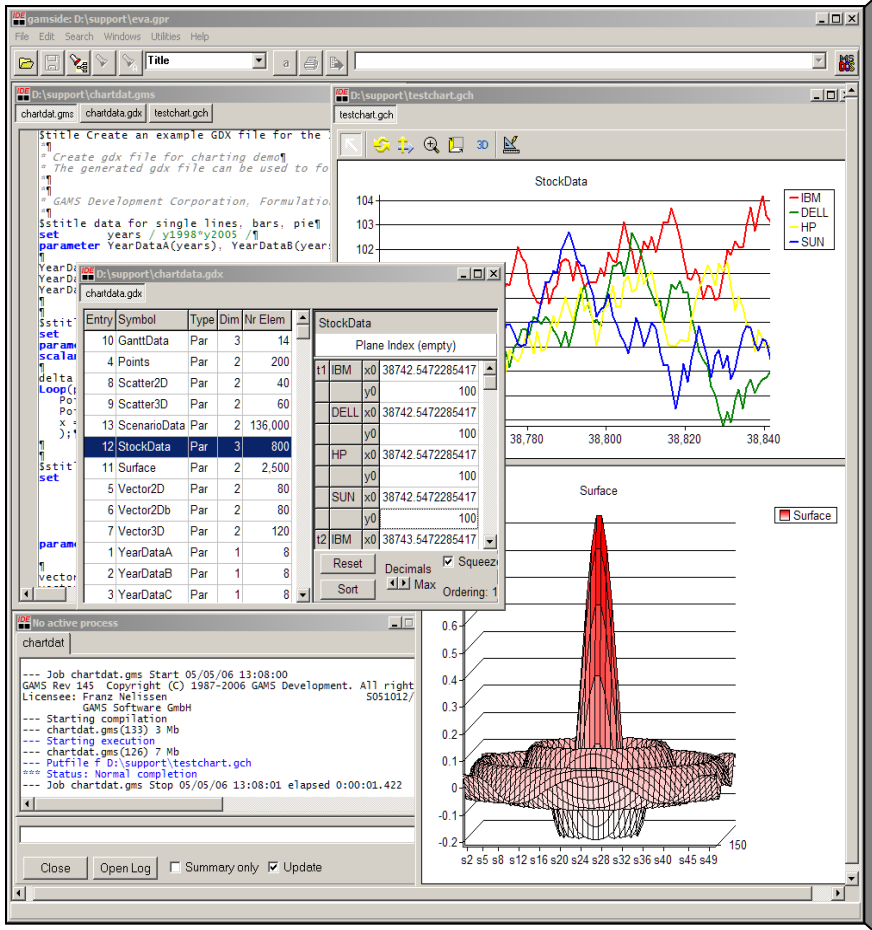
GAMS at a Glance



The GAMS/BASE Module

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

GAMS at a Glance



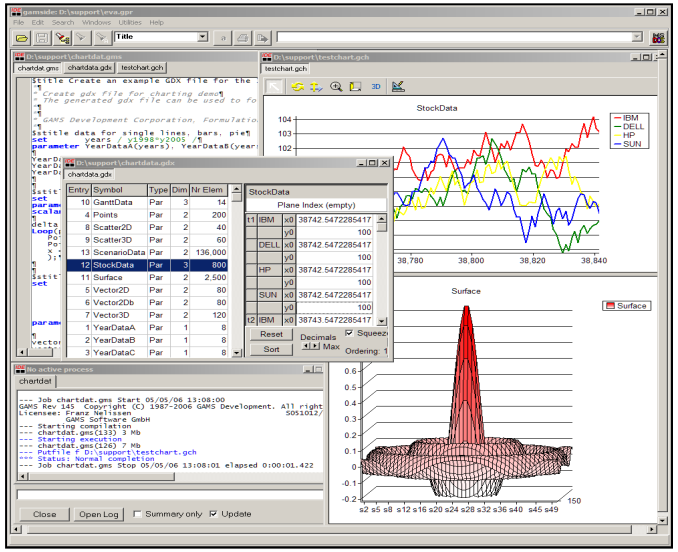
The GAMS/BASE Module

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

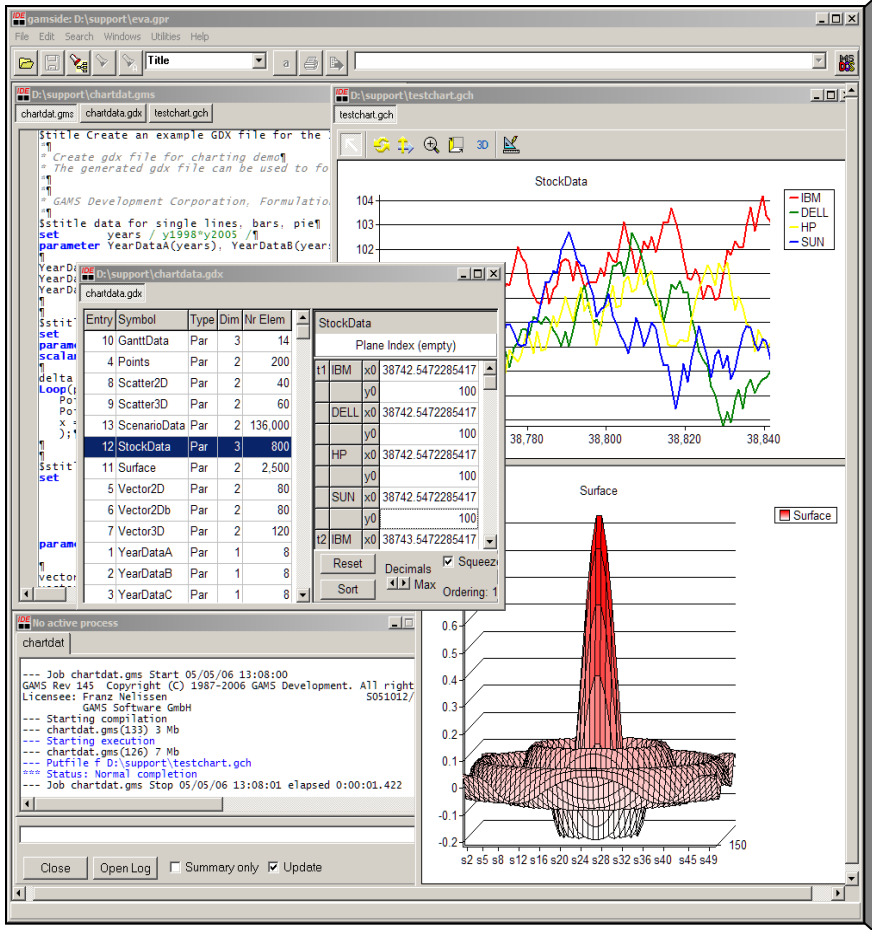
Integrated Development Environment

- 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

Hands-On



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

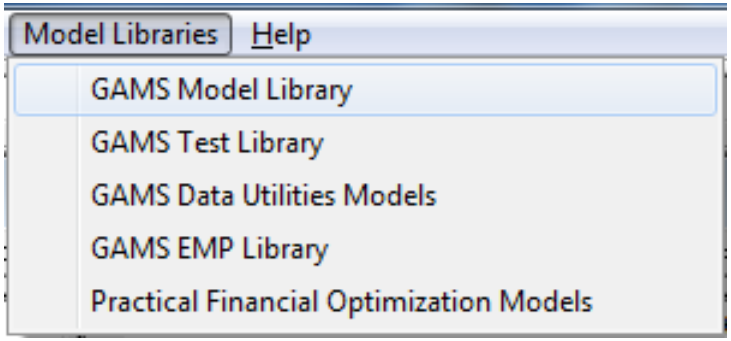
- **Wikis**
  - Support Wiki http://support.gams-software.com
  - Interfaces Wiki http://interfaces.gams-software.com

Documentation

- **Groups**
  - User Group http://www.gams.com/maillist/gams_l.htm
  - Google Group http://groups.google.de/group/gamsworld
- **Newsletter**
  - McCarl‘s News http://www.gams.com/maillist/newsletter.htm
  - Release List
- **Search all GAMS Websites** 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

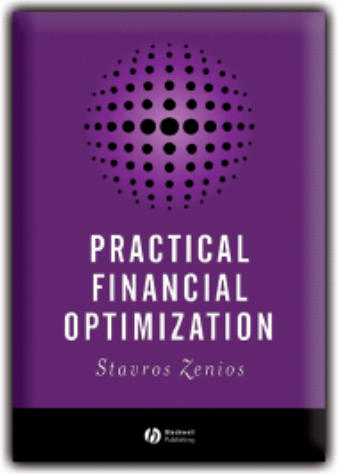


- **GAMS Test Library**
  - Transparent and reproducible Quality Assurance Tests
  - Tests for
    - Solver correctness
    - Special functions
    - GAMS utilities

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



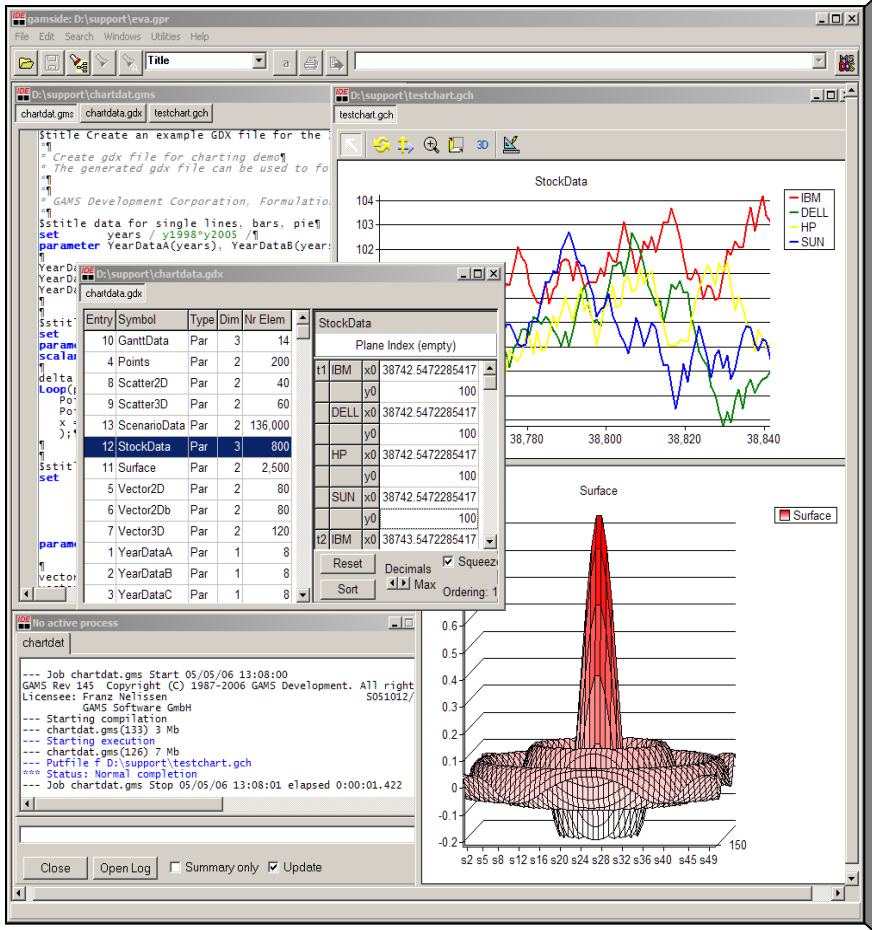
- **Practical Financial Optimization Models**

  Models of the book

  "PRACTICAL FINANCIAL OPTIMIZATION – A Library of GAMS Models"

  by Consiglio, Nielsen and Zenios

GAMS at a Glance

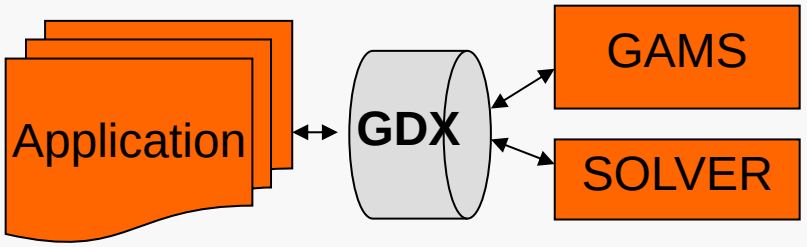


The GAMS/BASE Module

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

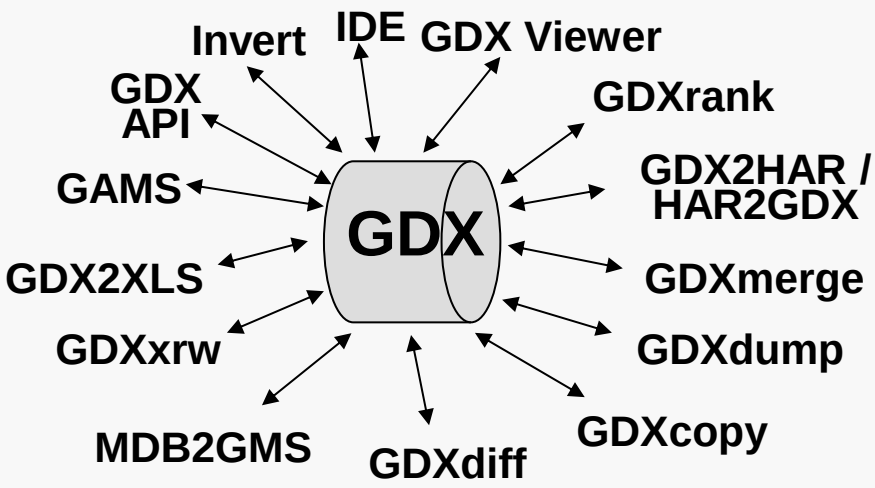
Gams Data eXchange

Binary Data Exchange

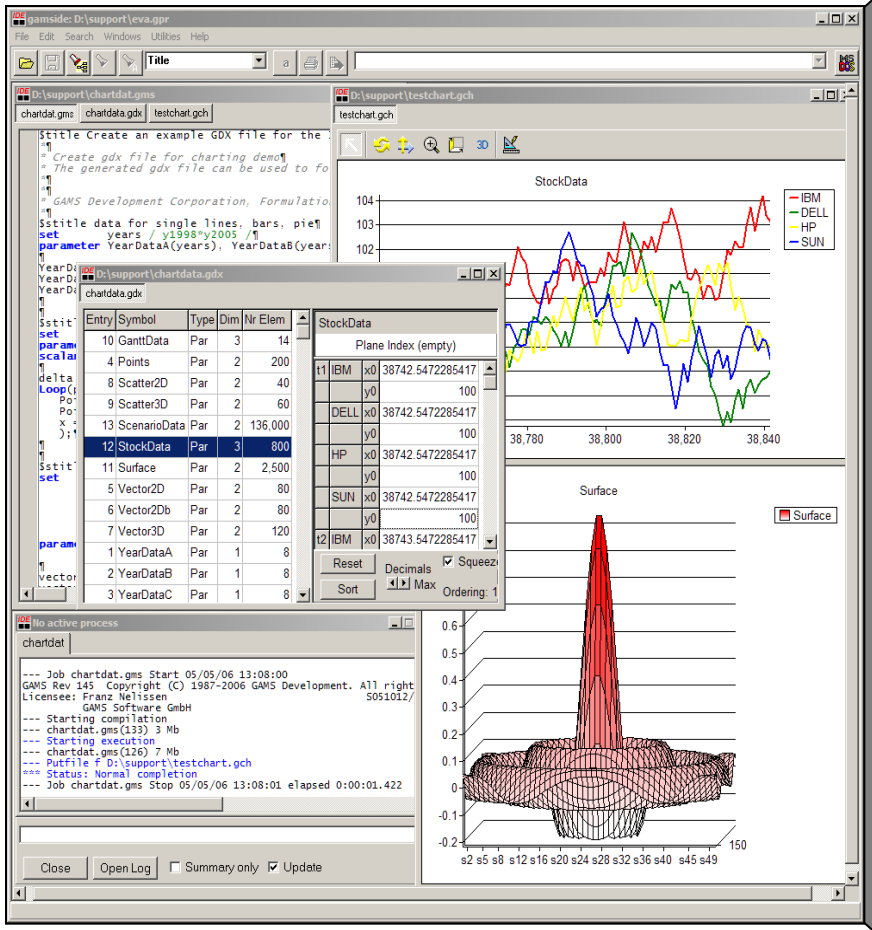


- Fast exchange of data
- Syntactical check on data before model starts
- Data Exchange at any stage (Compile and Run-time)
- Platform Independent
- Direct GDX interfaces and general API
- Scenario Management Support
- Full Support of Batch Runs

GDX Tools



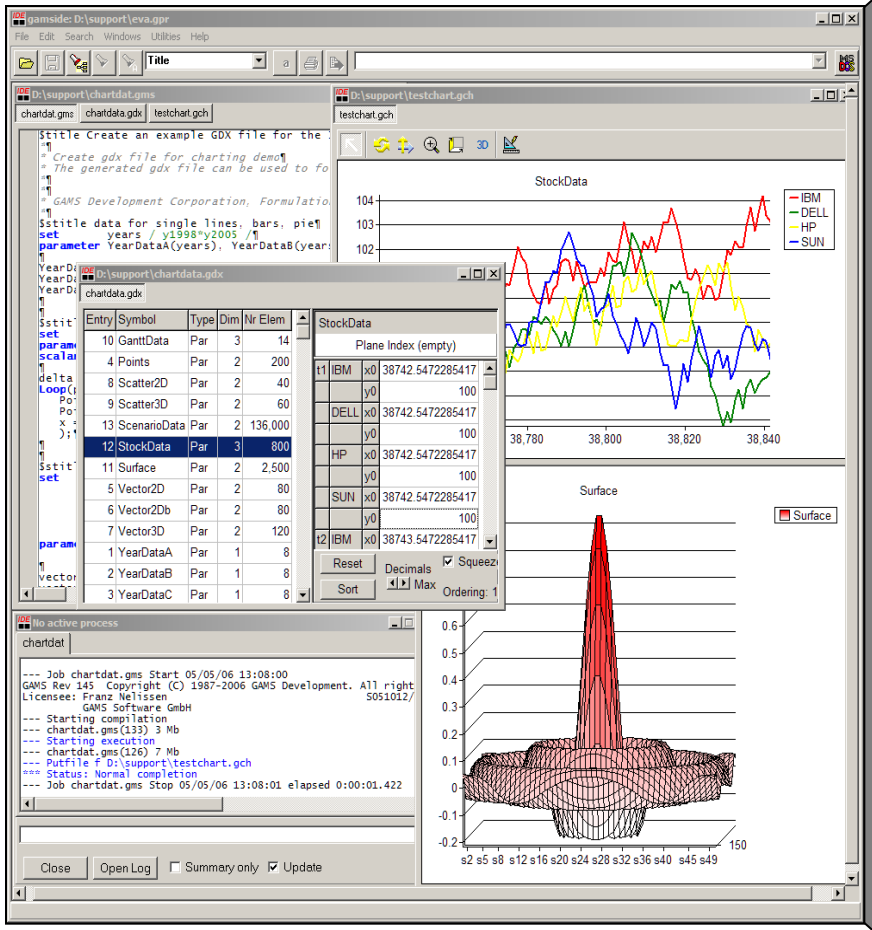
GAMS at a Glance



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

GAMS/Kestrel

- Remote Solver Execution on NEOS Servers
- From within your usual GAMS modeling environment
- Receiving results that can be processed as with any local solver


```
Model transport /all/;

Option lp=kestrel;
transport.optfile=1;

$onecho > kestrel.opt
kestrel_solver xpress
neos_server www.neos-server.org
socket_timeout 10
$offecho

Solve transport using lp minimizing z;
```

```
--- Executing KESTREL: elapsed 0:00:00.005
Connecting to: http://www.neos-server.org:3332

NEOS job#=956988, pass=LXBsrGJe
Check the following URL for progress report :
http://www.neos-server.org/neos/cgi-bin/nph-neos-solver.cgi?admin=results&jobnumber=956988&pass=LXBsrGJe

FICO-Xpress      24.1.2 r40979 Released Jun 16, 2013 LEG x86_64/Linux

Xpress Optimizer 24.01
Xpress Optimizer 64-bit v24.01.04 (Hyper capacity)
```

Coin-OR

An initiative to spur the development of open-source software for the OR community

http://www.coin-or.org/

- A repository of currently ~50 open-source projects
  - Solvers
  - Interfaces
  - Tools
- An active OR community
  - Mailing lists
  - Wikis

The Coin-OR / GAMSLinks Project

https://projects.coin-or.org/GAMSlinks

Goals

- Easy access to COIN-OR solvers via GAMS
- Broadening the audience of COIN-OR
- Broadening the audience of GAMS
- Help developers to connect their solvers to GAMS
- Provide access to GAMS benchmarking and quality assurance tools

The Coin-OR / GAMSLinks Project

GAMS interfaces to open-source Solvers

- COIN-OR Linear Programming (**CLP**) and Branch and Cut (**CBC**)
  - LP and MIP solver from J. Forrest
- COIN-OR Open Solver Interface **(OSI)**
  - Bare bone LP/MIP solver links to Cplex, Gurobi, Mosek, Soplex and Xpress
- Interior Point Optimizer (**IPOPT**)
  - Large scale NLP solver from A. Wächter

The Coin-OR / GAMSLinks Project

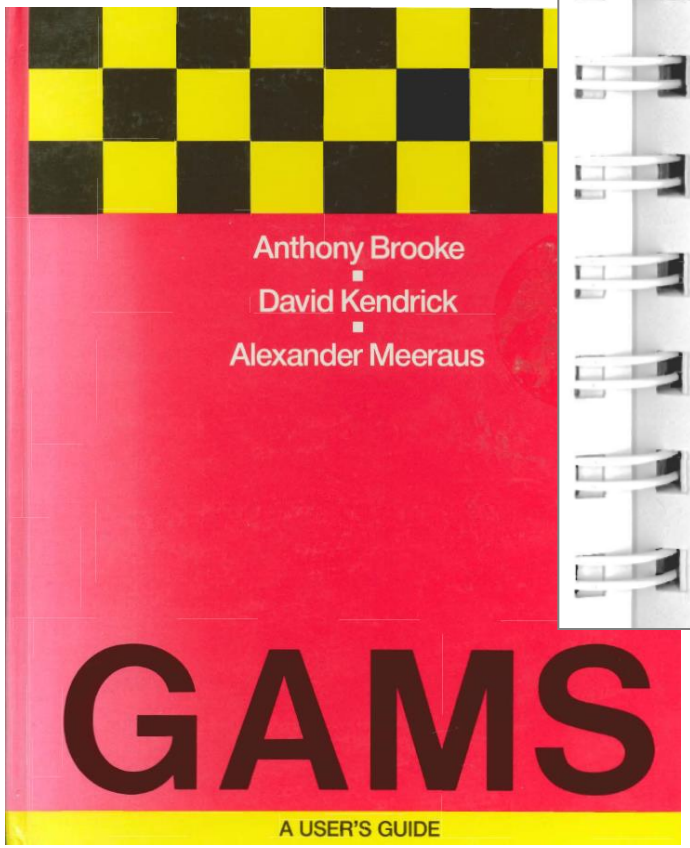
GAMS interfaces to open-source Solvers

- Solving Constraint Integer Programs (**SCIP**)
  - MIP/MINLP solver developed at Zuse Institute Berlin, TU Darmstadt and FAU Erlangen/Nürnberg
- Basic Open-source Nonlinear Mixed Integer programming (**BONMIN**)
  - Branch and Cut based MINLP solver from P. Bonami et.al.
- Convex Over and Under Envelopes for Nonlinear Estimation (**COUENNE**)
  - Branch and Bound MINLP solver from P. Belotti

Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

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

Name	*Number of Rows*	*Number of Columns*	*Number of Nonzeroes*	*Generation Time*[a]	*Solution Time*[a]	*Iterations*	*Solver*
DINAMICO	318	425	4156	3.0	30.1	628	MINOS
SARF	532	542	3949	37.7	115.8	2775	MINOS
FERTD[b]	458	2968	7252	11.4	28.3	1368	ZOOM
CAMCGE[c]	243	280	1356	0.8	7.0	189	MINOS
GANGES[d]	274	357	1405	1.8	7.3	187	MINOS
YEMCEM[e]	168	258	953	0.9	7.6	600	ZOOM
EGYPT[f]	281	618	3168	4.0	25.3	1551	ZOOM

[a] Measured in minutes.
[b] The problem is too big for MINOS. ZOOM was used instead.
[c] A nonlinear problem. 63% of the non-zeroes are nonlinear.
[d] A nonlinear problem. 58% of the non-zeroes are nonlinear.
[e] A mixed binary problem, with 55 binary variables (solved with a relative termination criterion of 10%).
[f] A linear problem, solved using XMP which is contained within ZOOM.

GAMS Users Guide (1988)

… and now

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

* 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

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

Basic Set Declaration

```
Set i       / i1, i2, i3                /;
Set j       / j1 * j3                   /;
Set k       / k3 * k1                   /;
Set l(i,j) / i1.j1, i1.j2, i1.j3
             i2.j1, i2.j2, i2.j3
             i3.j1, i3.j2, i3.j3 /;
Set m(i,j) / (i1*i3).(j1*j3)            /;
Set n(i,j) / #i.#j                      /;
```

Matching Operator

```
Set i / t1.s3,t2.s4,t3.s5 /;
```
(Product Operator)

can be written as

```
Set i / t1*t6:s3*s5 /;
```
(Matching Operator)

- Example “Count Tuples”:

```
Sets      h /h1*h24/, d /d1*d365/, t /t1*t8760/
          dh(d,h)     /#d.#h/
          tdh(t,d,h) /#t:#dh/;
```

➔ `t1.d1.h1, t2.d1.h2, ..., t25.d2.h1, t26.d2.h2, ...`

Hands-On

Matching Operator in Option Statements

```
Set i /i1*i2/, j /j3*j5/
    k /k1*k5/, cnt /c1*c100/
    ijk(i,j,k), x(I,j,k,cnt);

Option ijk(i:j,k), x(ijk:cnt);
```

➔ ijk: i1.j3.k1, i1.j3.k2, …,
i2.j4.k1, i2.j4.k2, …,
i2.j4.k5

➔ x: i1.j3.k1.c1, i1.j3.k2.c2, …,
i2.j4.k1.c5, i2.j4.k2.c6, …,
i2.j4.k5.c10

Enhanced Data Statements

- Allow initial values for equations and variables
- Follow the syntax for list and table data statement for parameters by adding an additional dimension to specify the specific data attribute

```
Variable table x(i,j) initial values

                          l       m
  seattle.  new-york     50
  seattle.  Chicago     300
  seattle.  topeka              0.36
  san-diego.new-york    275
  san-diego.Topeka      275
  san-diego.chicago           0.009   ;
```

The GAMS Macro Facility

- Basic Definition
  - `$macro name macro body`
  - `$macro name(arg1,…) macro body with tokens arg1,…`
- Multi-Argument Example

```
$macro ratio(a,b) a/b
z = ratio(x1,x2);
```

➔ `z = x1/x2;`

- Macros within Macros

```
$macro product(a,b) a*b
$macro addup(i,x,z) sum(i,product(x(i),z))
z = addup(j,a1,x1);
```

➔ `z = sum(j,a1(j)*x1);`

Hands-On?

The GAMS Macro Facility (contd.)

- Careful expansion (&)

```
$macro f(i)   sum(j, x(i,j))
$macro equ(q) equation equ_&q; equ_&q.. q =e= 0;
equ(f(i))
```

➔ `equation equ_f(i); equ_f(i).. sum(j, x(i,j)) =e= 0;`

- Removing outer set of quotes (&&)

```
$macro d(q) display &&q;
d('"here it is", i, k')
```

➔ `display "here it is", i, k;`

```
$macro dd(q) &&q)
z=dd('sum(j,a1(j)');
```

➔ `z=sum(j,a1(j));`

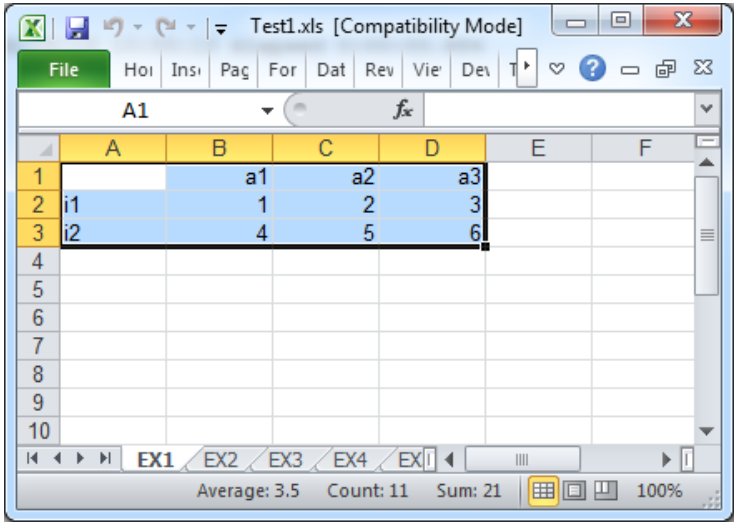
Outline

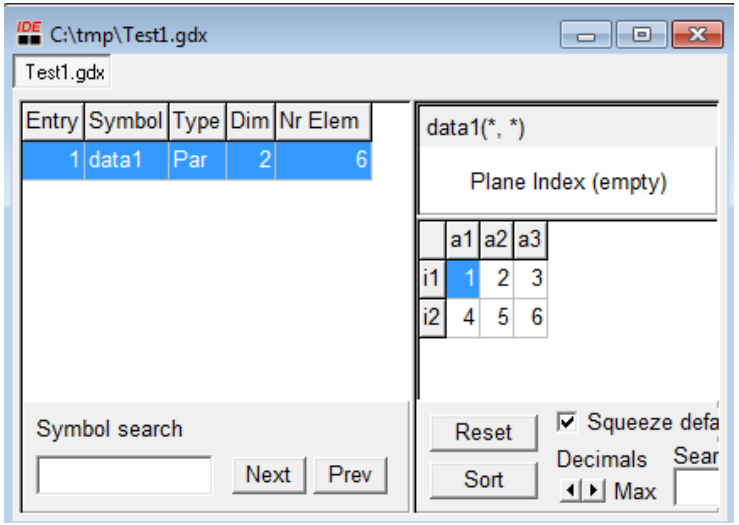
- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

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

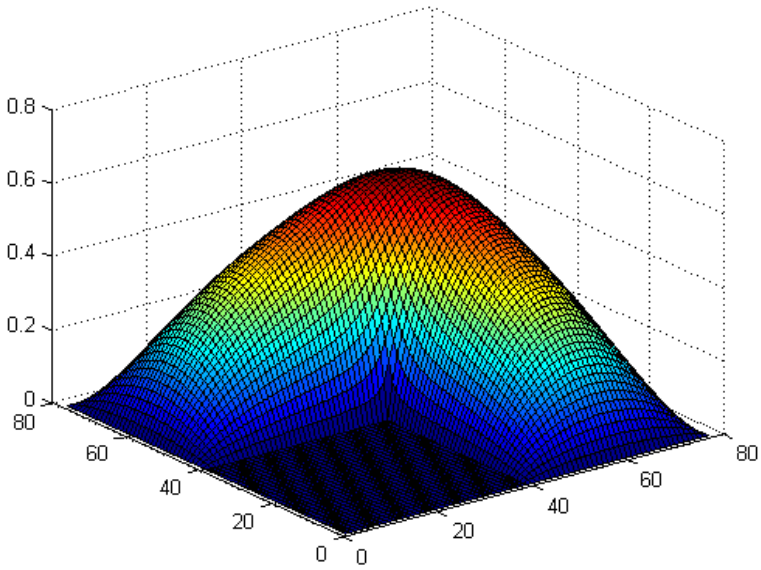
Hands-On





GDXMRW

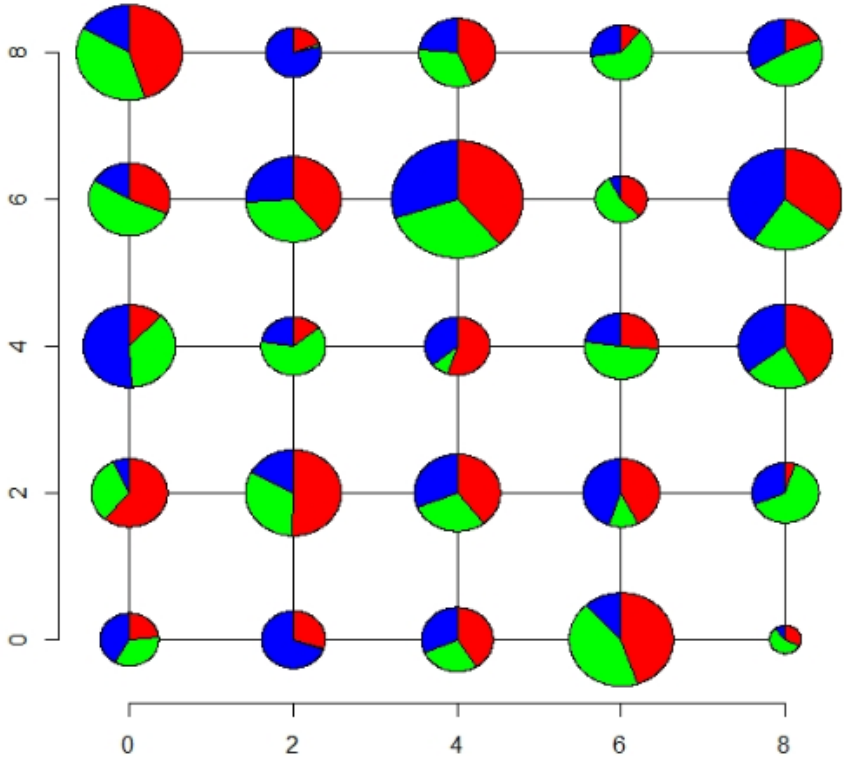
- Import/export data between GAMS and MATLAB
- Call GAMS models from MATLAB
- Get results back in MATLAB
- Gives MATLAB users the ability to use all the optimization capabilities of GAMS



- Allows visualization of GAMS models directly within MATLAB
- More Information: http://support.gams.com/doku.php?id=matlab_and_gams:interfacing_optimization_and_visualization_software_via_the_gdxmrw_utilities

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



Source: http://blog.modelworks.ch

- More information: http://support.gams.com/doku.php?id=gdxrrw:interfacing_gams_and_r

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;
```

Hands-On

Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

Solvelink Option

```
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

Hands-On

Solving Scenarios

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

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

Setting	Solve time (secs)
Solvelink=%Solvelink.ChainScript%	52.221
Solvelink=%Solvelink.CallModule%	37.366
Solvelink=%Solvelink.LoadLibrary%	03.252

Gather-Update-Solve-Scatter (GUSS)

Setting	Solve time (secs)
Solvelink=%Solvelink.ChainScript%	52.221
Solvelink=%Solvelink.CallModule%	37.366
Solvelink=%Solvelink.LoadLibrary%	03.252
GUSS	01.046

- 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

GUSS

- Dynamic model – rolling horizon

- Example:
  - Combined Heat and Power Planning with Heat Storage. All data known apriori but heat storage level
  - Can’t use GUSS
  - Implement GUSS in programming language
    - Identify some parameters as “modifiable” parameters
    - Implement rolling horizon in programming language

Tracing Solve Process

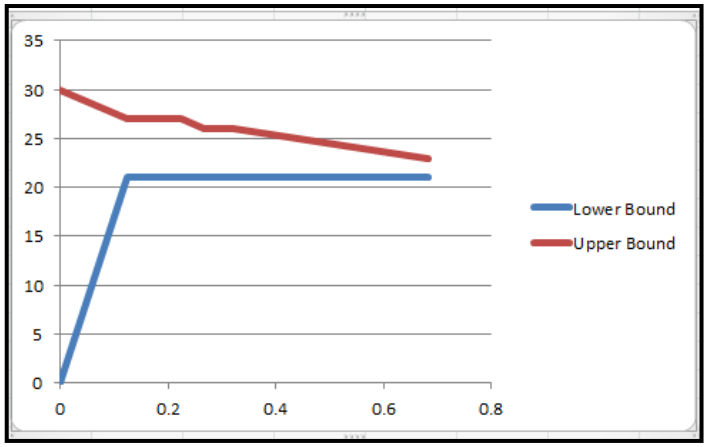
Solver Options (e.g. Cplex, Gurobi, Xpress):

- **`MipTrace`**
  - Writes a file that records the best integer and best bound values every *`miptracenode`* nodes and at *`miptracetime`*-second intervals
- **`MipTraceNode`**
  - Specifies the node interval between entries to the *`MipTrace`* file [Default: 100]
- **`MipTraceTime`**
  - Specifies the time interval, in seconds, between entries to the *`MipTrace`* file [Default: 5]

Tracing Solve Process – cont.

Generates a Trace file during solve:

```
* miptrace file gurobi.trc: ID = Gurobi
* fields are lineNum, seriesID, node, seconds, bestFound, bestBound
1, S, 0, 0, -0, 30
2, N, 100, 0.113, 21, 27
3, N, 200, 0.169, 21, 27
4, N, 300, 0.212, 21, 27
5, N, 400, 0.255, 21, 26
6, N, 500, 0.31, 21, 26
7, E, 683, 0.668, 21, 23
* miptrace file gurobi.trc closed
```



- Common format among all solvers that support this option
- Available with: ANTIGONE, BONMIN, CBC, CPLEX, COUENNE, GloMIQO, Gurobi, SBB, SCIP, Sulum, Xpress (Partly using different option names)

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

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

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

```
$FuncLibIn <IntLibName> <ExtLibName>
```

- Declare functions similar to sets, parameters, ..., :

```
Function <IntFuncName> /<IntLibName>.<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:

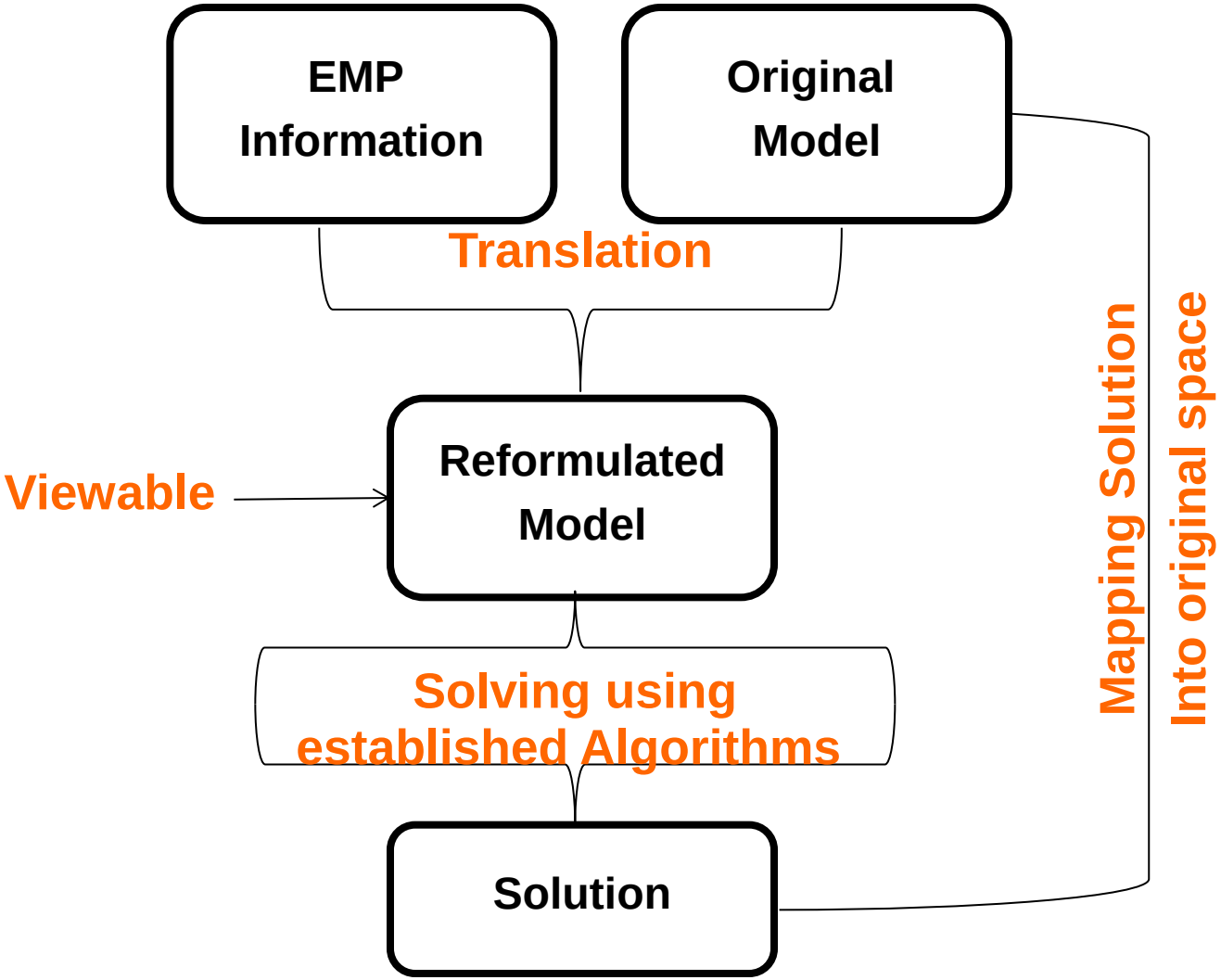
- overload existing GAMS notation right away !
- attempt to build new solvers right away !

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



Simple Example: Newsboy (NB) Problem

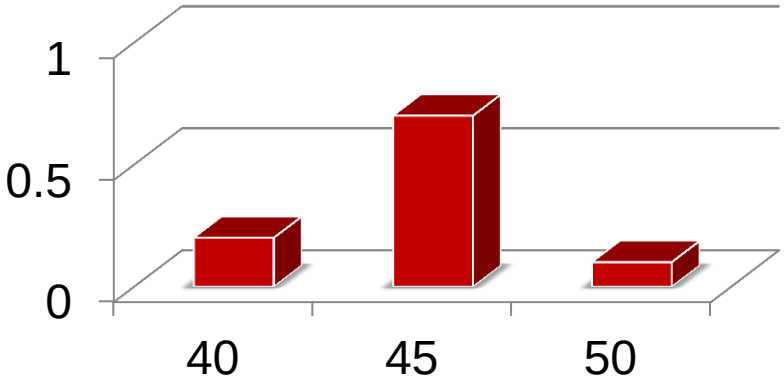
- A newsboy faces an uncertain demand for newspapers
- He can buy newspapers for fixed costs per unit
- He can sell newspapers for a fixed price
- For hold units he has to pay a disposal fee
- He has to satisfy his customers demand or has to pay a penalty

- Decisions to make:
  - How much newspaper should he buy “here and now” (without knowing the outcome of the uncertain demand)?
    → *First-stage decision*
  - How many customers are lost after the outcome becomes known?
    → *Second-stage* or *recourse decision*
  - Recourse decisions can be seen as
    - penalties for bad first-stage decisions
    - variables to keep the problem feasible

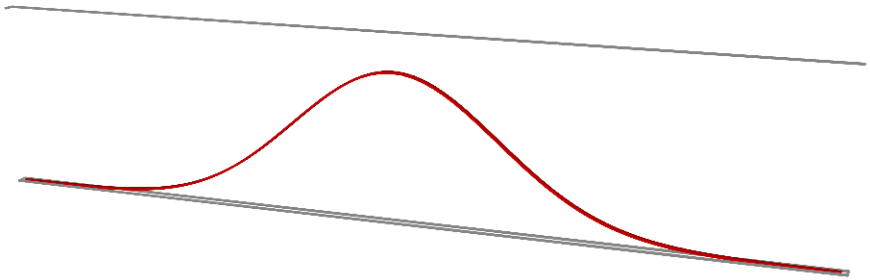
Hands-On
➔ nbsimple.gms

Random Variables

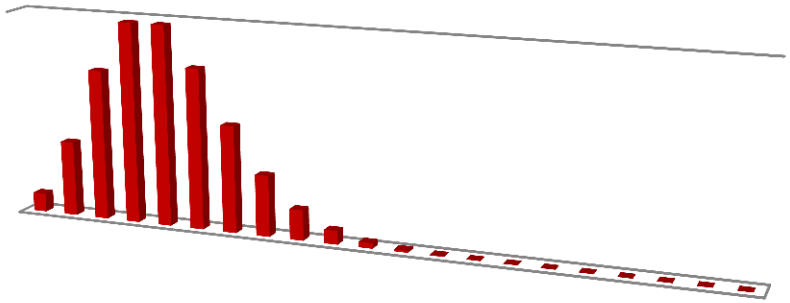
Discrete Distribution



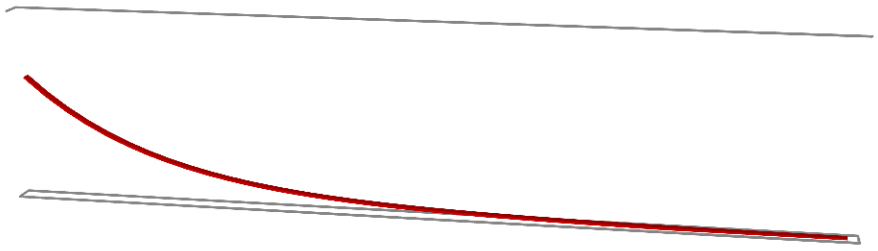
Normal Distribution



Poisson Distribution



Exponential Distribution



Random Variables (RV) [`randVar`]

- Defines both discrete and parametric random variables:

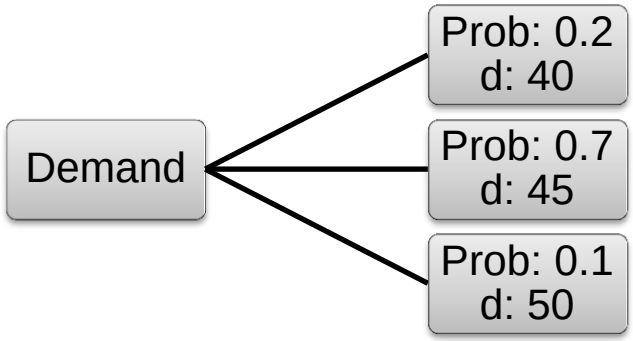
```
randVar rv discrete prob val {prob val}
```

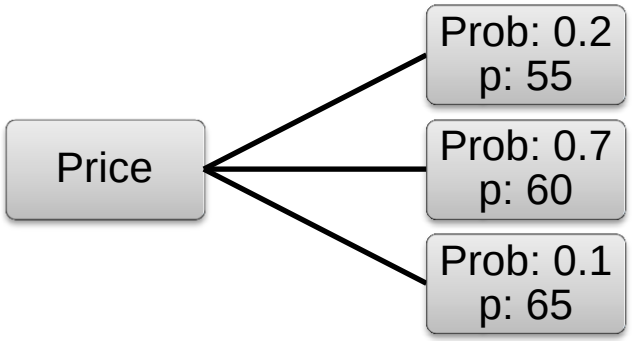
- The distribution of discrete random variables is defined by pairs of the probability `prob` of an outcome and the corresponding realization `val`

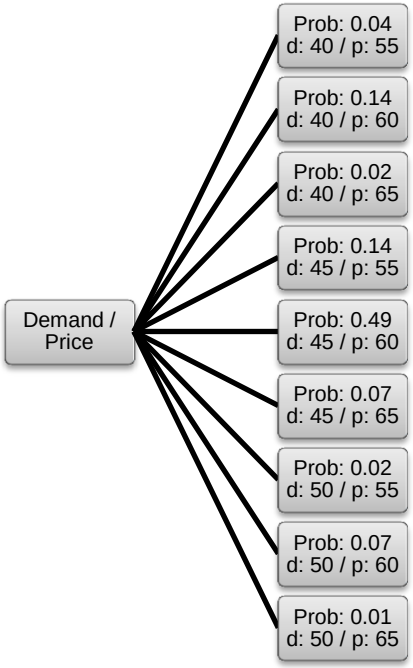
```
randVar rv distr par {par}
```

- The name of the parametric distribution is defined by `distr`, `par` defines a parameter of the distribution

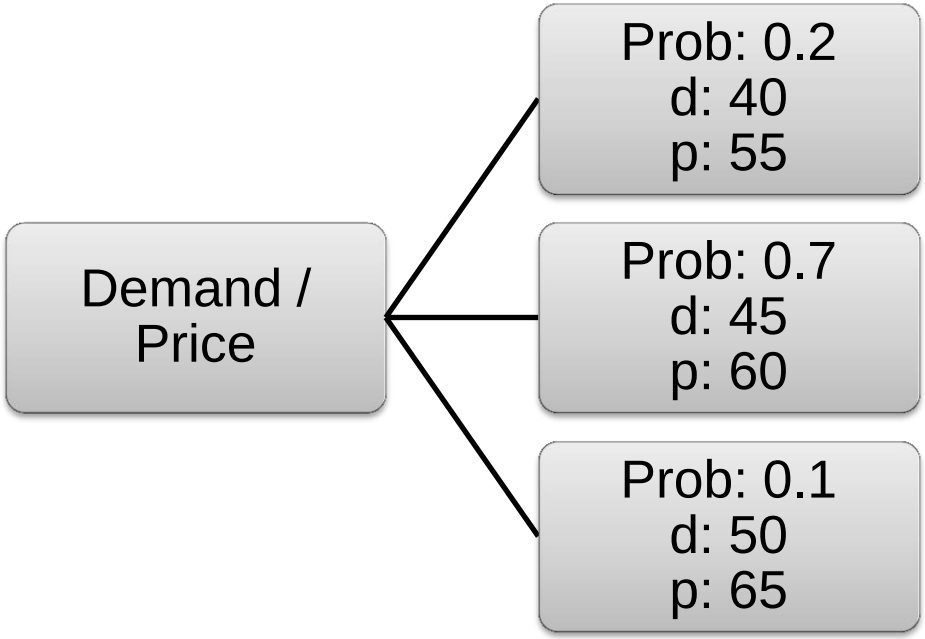
Joint Random Variables







vs.



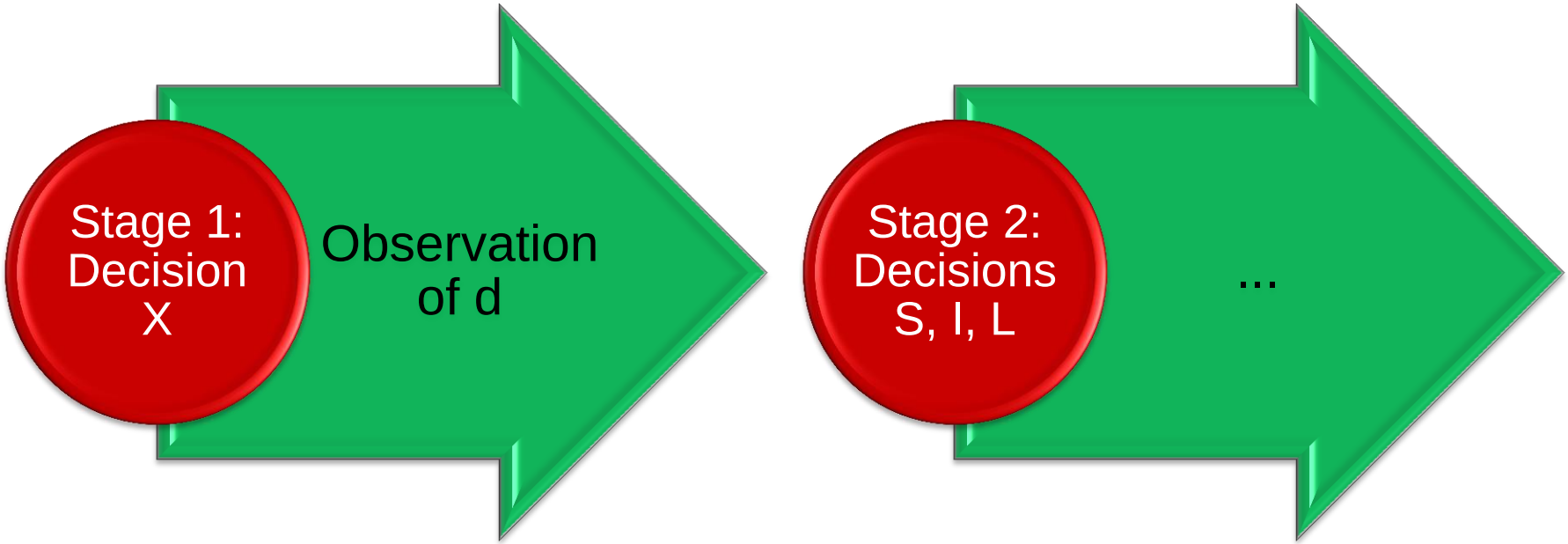
Joint RVs [`jRandVar`]

- Defines discrete random variables and their joint distribution:

```
jRandVar rv rv {rv} prob val val {val}
                        {prob val val {val}}
```

- At least two discrete random variables `rv` are defined and the outcome of those is coupled
- The probability of the outcomes is defined by `prob` and the corresponding realization for each random variable by `val`

Stages



Stages [stage]

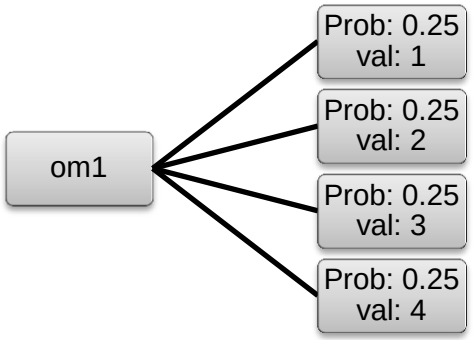
- Defines the stage of random variables (`rv`), equations (`equ`) and variables (`var`):

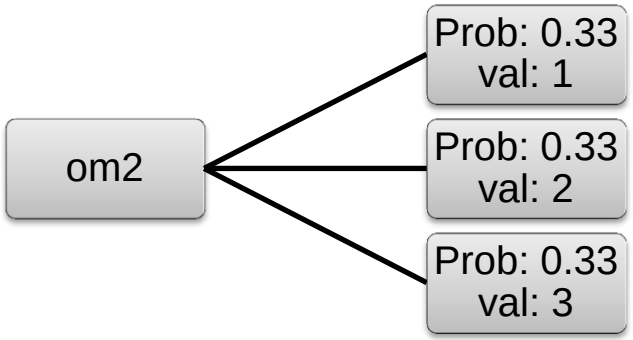
```
stage stageNo rv | equ | var {rv | equ | var}
```

- `StageNo` defines the stage number
- The default `StageNo` for the objective variable and objective equation is the highest stage mentioned
- The default `StageNo` for all the other random variables, equations and variables not mentioned is 1

Chance Constraints

```
OBJ..  Z  =e= X1 + X2;
E1..   om1*X1 +   X2 =g=  7;
E2..   om2*X1 + 3*X2 =g= 12;
Model sc / all /;
solve sc min z use lp;
```






```
chance E1 0.6
chance E2 0.6
```

Chance Constraints

3 out of 4 must be true

[0.75 ≥ 0.6]

- `1*X1 + X2 =g= 7;`
- `2*X1 + X2 =g= 7;`
- `3*X1 + X2 =g= 7;`
- `4*X1 + X2 =g= 7;`

2 out of 3 must be true

[0.66 ≥ 0.6]

- `1*X1 + 3*X2 =g= 12;`
- `2*X1 + 3*X2 =g= 12;`
- `3*X1 + 3*X2 =g= 12;`

Chance Constraints [chance]

- Defines individual or joint chance constraints (CC):

```
chance equ {equ} [holds] minRatio [weight|varName]
```

- Individual CC: A single constraint `equ` has to hold for a certain ratio ($0 \leq$ `minRatio` ≤ 1) of the possible outcomes
- Joint CC: A set of constraints `equ` has to hold for a certain ratio ($0 \leq$ `minRatio` ≤ 1) of the possible outcomes
- If `weight` is defined, the violation of a CC gets penalized in the objective (weight violationRatio)
- If `varName` is defined the violation get multiplied by this existing variable

Outline

- GAMS
  - GAMS at a Glance
  - Simple Example
  - GAMS/Base
- Features you might not know about
  - Syntax
  - Data Import/Export
  - Advanced Use of GAMS Solver Links
  - Extending the GAMS Syntax
  - Other Tools

Matrix Utilities

- INVERT
  - Calculates the inverse of a matrix
- CHOLESKY
  - Computes the Cholesky factors of a symmetric positive-definite matrix
- EIGENVALUE
  - Computes the eigenvalues of a symmetric matrix
- EIGENVECTOR
  - Computes the eigenvalues and eigenvectors of a symmetric matrix

Check for GAMS Updates