High Performance Computing with GAMS

F. Fiand, M. Bussieck
September 7th, 2017
GAMS Software GmbH
Outline

• Introduction
• Model Annotation
• Distributed Model Generation
• Outlook
Introduction
Algebraic Modeling Language
Facilitates to formulate mathematical optimization problems similar to algebraic notation
→ Simplified model building:

*Model is executable algebraic description of optimization problem.*
Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building:

Model is executable algebraic description of optimization problem.

\[
\sum_{p \in P: pr, p} \text{POWER}_{t, r, p} + \sum_{r2 \in R: net_r2, r} (\text{FLOW}_{t, r2, r}) - \sum_{r2: net_r, r2} \text{FLOW}_{t, r, r2} + \sum_{s \in S: rs_r, s} (\text{STORAGE\_OUTFLOW}_{t, r, s} - \text{STORAGE\_INFLOW}_{t, r, s}) \geq \text{demand}_{t, r} \quad \forall t \in T, r \in R
\]

\text{eq\_power\_balance}(t, r) ..
\begin{align*}
\text{sum}(r, p), & \quad \text{POWER}(t, r, p) \\
+ \text{sum}(r2, r), & \quad \text{FLOW}(t, \text{net})) - \text{sum}(r2, r2), \quad \text{FLOW}(t, \text{net}) \\
+ \text{sum}(s, r, s), & \quad \text{STORAGE\_OUTFLOW}(t, r, s) - \text{STORAGE\_INFLOW}(t, r, s) = \text{demand}(t, r); \\
\end{align*}
Algebraic Modeling Language
Facilitates to formulate mathematical optimization problems similar to algebraic notation
→ Simplified model building

Declarative elements

• Similar to mathematical notation
• Easy to learn - few basic language elements: sets, parameters, variables, equations, models
• Model is executable (algebraic) description of the problem
Algebraic Modeling Language

Facilitates to formulate mathematical optimization problems similar to algebraic notation

→ Simplified model building

**Declarative elements**

- Similar to mathematical notation
- Easy to learn - few basic language elements: sets, parameters, variables, equations, models
- Model is executable (algebraic) description of the problem

**Procedural elements**

- Control Flow Statements (e.g. loops, for, if, ...),
- Build complex problem algorithms within GAMS
- Simplified interaction with other systems
  - Data exchange
  - GAMS process control
### GAMS – System Overview

- **Broad Range of 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>
GAMS – System Overview

- Broad Range of 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>

- GAMS is widespread in the ESM community: [http://www.energyplan.eu/othertools/](http://www.energyplan.eu/othertools/)
Goal: Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models
Goal: Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models
## Limitations of “standard” Soft- & Hardware

<table>
<thead>
<tr>
<th>#t</th>
<th>#r</th>
<th>#blocks</th>
<th>#rows (E6)</th>
<th>#cols (E6)</th>
<th>#NZ (E6)</th>
<th>~Mem (GB)</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>730</td>
<td>10</td>
<td>10</td>
<td>0.7</td>
<td>0.8</td>
<td>2.8</td>
<td>2.0</td>
<td>00:01:22</td>
</tr>
<tr>
<td>730</td>
<td>10</td>
<td>500</td>
<td>35.0</td>
<td>38.7</td>
<td>142.8</td>
<td>95.7</td>
<td>01:09:36</td>
</tr>
<tr>
<td>730</td>
<td>10</td>
<td>2,500</td>
<td>175.3</td>
<td>193.5</td>
<td>713.9</td>
<td>478.8</td>
<td>09:32:55</td>
</tr>
<tr>
<td>730</td>
<td>10</td>
<td>4,000</td>
<td>280.5</td>
<td>309.6</td>
<td>1,142.2</td>
<td>767.1</td>
<td>19:22:55</td>
</tr>
<tr>
<td>730</td>
<td>10</td>
<td>7,500</td>
<td>526.1</td>
<td>580.5</td>
<td>2,141.2</td>
<td>~1,436.4</td>
<td>-</td>
</tr>
<tr>
<td>8,760</td>
<td>10</td>
<td>10</td>
<td>8.4</td>
<td>9.3</td>
<td>34.3</td>
<td>18.2</td>
<td>00:28:57</td>
</tr>
<tr>
<td>8,760</td>
<td>10</td>
<td>50</td>
<td>42.1</td>
<td>46.4</td>
<td>171.6</td>
<td>90.4</td>
<td>02:26:25</td>
</tr>
</tbody>
</table>

Test runs were made on JURECA @ JSC
- 2x Intel Xeon E5-2680 v3 (Haswell), 2 x 12 cores @ 2.5GHz
- “fat” node with 1,024 GB Memory
- GAMS 24.8.5 / CPLEX 12.7.1.0
- Barrier Algorithm, Crossover disabled, 24 threads
Model Annotation
Original problem with “random” matrix structure

\[
\begin{align*}
\text{min/max} & \quad c \\
A & \quad b \\
\ast & \quad \leq \quad x & \quad \leq & \quad \ast
\end{align*}
\]

PIPS exploits matrix block structure

\[
\begin{align*}
\text{min} & \quad c_0 & \quad c_{s1} & \quad c_{sn} \\
A^{=} & \quad = & \quad = & \quad b \\
b^{\text{low}} & \quad \leq \quad A^{=} & \quad \leq \quad A^{\pm} & \quad \leq \quad b^{\text{upp}} \\
b_{s1}^{\text{low}} & \quad \leq \quad T_{s1}^{=} & \quad \leq \quad T_{s1}^{\pm} & \quad \leq \quad b_{s1}^{\text{upp}} \\
b_{sn}^{\text{low}} & \quad \leq \quad T_{sn}^{=} & \quad \leq \quad T_{sn}^{\pm} & \quad \leq \quad b_{sn}^{\text{upp}} \\
b_c^{\text{low}} & \quad \leq \quad C^{=} & \quad \leq \quad C^{\pm} & \quad \leq \quad b_c^{\text{upp}} \\
x_0^{\text{low}} & \quad \leq \quad x_0 & \quad \leq \quad x_0^{\text{upp}} \\
x_{sn}^{\text{low}} & \quad \leq \quad x_{sn} & \quad \leq \quad x_{sn}^{\text{upp}}
\end{align*}
\]
Original problem with “random” matrix structure

\[ \text{min/max} \quad C \quad \leq \quad b \quad \leq \quad \star \]

Permutation reveals block structure

\[ \text{min/max} \quad C' \quad \leq \quad b' \quad \leq \quad \star \]

Model annotation

PIPS exploits matrix block structure

\[ \text{min} \quad c_0 \quad c_{s1} \quad c_{sn} \quad = \quad b \]
\[ b_{low} \quad \leq \quad A = \quad A_{\leq} \quad = \quad b_{low} \quad \leq \quad b_{upp} \]
\[ b_{s1_{low}} \quad \leq \quad T_{s1} = \quad W_{s1} = \quad = \quad b_{s1_{low}} \quad \leq \quad b_{s1_{upp}} \]
\[ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \]
\[ b_{sn_{low}} \quad \leq \quad T_{sn} = \quad W_{sn} = \quad = \quad b_{sn_{low}} \quad \leq \quad b_{sn_{upp}} \]
\[ b_{c_{low}} \quad \leq \quad C = \quad C_{s1} = \quad C_{sn} = \quad = \quad b_{c_{low}} \quad \leq \quad b_{c_{upp}} \]
\[ x_{0_{low}} \quad \leq \quad x_{0} \quad \leq \quad \cdots \quad \leq \quad x_{sn_{low}} \quad \leq \quad x_{sn_{upp}} \]
Model Annotation

Original problem with “random” matrix structure

\[
\begin{align*}
\text{min/max} & \quad c \\
A & \quad \leq \quad x \\
\end{align*}
\]

Model generation

PIPS exploits matrix block structure

\[
\begin{align*}
\text{min} & \quad c_0 & \quad c_{s1} & \quad \ldots & \quad c_{sn} \\
A & \quad = & & \leq & \quad b \\
A^l & \quad \leq & \quad T_{s1}^l & \quad W_{s1}^l & \quad \leq & \quad b_{s1}^l \\
A^u & \quad \leq & \quad T_{s1}^u & \quad W_{s1}^u & \quad \leq & \quad b_{s1}^u \\
\vdots & & & \vdots & & \vdots \\
A^l & \quad \leq & \quad T_{sn}^l & \quad W_{sn}^l & \quad \leq & \quad b_{sn}^l \\
A^u & \quad \leq & \quad T_{sn}^u & \quad W_{sn}^u & \quad \leq & \quad b_{sn}^u \\
C & \quad \leq & \quad C_{s1}^l & \quad \ldots & \quad C_{sn}^l & \quad \leq & \quad b_C^l \\
C & \quad \leq & \quad C_{s1}^u & \quad \ldots & \quad C_{sn}^u & \quad \leq & \quad b_C^u \\
X_{0}^l & \quad \leq & \quad X_{0} & \quad \leq & \quad X_{0}^u \\
X_{sn}^l & \quad \leq & \quad X_{sn} & \quad \leq & \quad X_{sn}^u \\
\end{align*}
\]

Permutation reveals block structure

\[
\begin{align*}
\text{min/max} & \quad c' \\
A' & \quad \leq \quad x' \\
\end{align*}
\]
To which block do variable $X$ and equation $E$ belong?
Model Annotation cont.

Model Annotation by .Stage

- The .stage attribute is available for variables/equations in GAMS

<table>
<thead>
<tr>
<th>Matrix structure required by PIPS API</th>
</tr>
</thead>
<tbody>
<tr>
<td>stage 0</td>
</tr>
<tr>
<td>min</td>
</tr>
<tr>
<td>b_{\text{low}} \leq T_{s1} \leq \ldots \leq T_{sn} \leq b_{\text{up}}</td>
</tr>
<tr>
<td>b_{s1\text{low}} \leq W_{s1} \leq \ldots \leq W_{sn} \leq b_{s1\text{up}}</td>
</tr>
<tr>
<td>b_{\text{low}} \leq C \leq \ldots \leq C_{sn} \leq b_{\text{up}}</td>
</tr>
</tbody>
</table>

- Annotation

(stage 0)

- (stage 1)

- (stage n)

- (stage n+1)
Model Annotation by .Stage

- The .stage attribute is available for variables/equations in GAMS

**Exemplary Annotation for SIMPLE model (regional decomposition)**

```plaintext
Set
   rr 'regions'          
   p                   
   tt 'time steps'    
   e 'emissions'      
   t(tt) 'subset of active time steps' 
   rp(rr,p) 'region to plant mapping' 
   net(rr,rr) 'transmission links' 

Alias(rr,rrl,rr2,rr3);

* Master variables and equation
FLOW.stage(t,net(rrl,rr2)) = 0;
LINK_ADD_CAP.stage(net(rrl,rr2)) = 0;

* Block variables and equations
POWER.stage(t,rp(rr3,p)) = ord(rr);
EMISSION.stage(rr3,e) = ord(rr);

* Linking Equation
eq_power_balance.stage(t,r) = ord(rr);
eq_emission_region.stage(rr3,e) = ord(rr);
eq_emission_cap.stage(e) = card(rr)+1;
```

**Matrix structure required by PIPS API**

- Stage 0
  - min
  - \( c_0 \) \( c_{s1} \) \( c_{sn} \)
  - \( A^= \) \( A^{s} \)
  - \( b_{\text{low}} \) \( \leq \) \( b_{\text{upp}} \)

- Stage 1
  - \( T_{s1} \) \( W_{s1} \)
  - \( b_{s1} \) \( \leq \) \( b_{s1}^{upp} \)

- Stage n
  - \( T_{sn} \) \( W_{sn} \)
  - \( b_{sn} \) \( \leq \) \( b_{sn}^{upp} \)

- Stage n+1
  - \( C \) \( C_{s1} \) \( C_{sn} \)
  - \( b_{c} \) \( \leq \) \( b_{c}^{upp} \)

- \( x_{0} \) \( \leq \) \( x_{0}^{upp} \)
- \( x_{sn} \) \( \leq \) \( x_{sn}^{upp} \)
• How to annotate Model depends on how the model should be “decomposed” (by region, time,...)
Model Annotation cont.

- How to annotate Model depends on how the model should be “decomposed” (by region, time,...)

Plots show three different annotations of identical model
Model Annotation cont.

- How to annotate Model depends on how the model should be “decomposed” (by region, time,...)

Plots show three different annotations of identical model

- How important are blocks of equal size?
Distributed Model Generation
Distributed Model Generation

- “Usual Model”: model generation time << solver time
- For LARGE-scale models the model generation may become significant:
  - due to time consumption
  - due to memory consumption
  - due to hard coded limitations of model size (# non-zeroes < ~2.1e9)

→ Distributed “block-wise” model setup in PIPS-IPM
→ Model annotation determines block membership of all variables and constraints
→ Distributed GAMS processes can generate the separate blocks (model needs to be prepared accordingly!)
Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):

\[
\begin{align*}
\text{min} & & c_0 & & c_{s_1} & & c_{s_n} \\
& & A_1 & & A_2 & & = & & b \\
& & A_3 & & A_4 & & \leq & & b_{upp} \\
& & T_{s_1}^= & & W_{s_1}^= & & = & & b_{s_1} \\
& & T_{s_1}^\leq & & W_{s_1}^\leq & & \leq & & b_{s_1}^{upp} \\
& & \vdots & & \vdots & & \vdots & & \vdots \\
& & T_{s_n}^= & & W_{s_n}^= & & = & & b_{s_n} \\
& & T_{s_n}^\leq & & W_{s_n}^\leq & & \leq & & b_{s_n}^{upp} \\
& & C^= & & C_{s_1}^= & & \ldots & & C_{s_n}^= & & = & & b_C \\
& & C^\leq & & C_{s_1}^\leq & & \ldots & & C_{s_n}^\leq & & \leq & & b_C^{upp} \\
& & x_0^\leq & & x_0 & & \ldots & & x_0 & & \leq & & x_0^{upp} \\
& & x_{sn}^\leq & & x_{sn} & & \ldots & & x_{sn} & & \leq & & x_{sn}^{upp}
\end{align*}
\]
Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):

![Parallel generation of n+1 blocks & Distribution to MPI processes as needed by PIPS-IPM](image)

<table>
<thead>
<tr>
<th>min</th>
<th>(c_0)</th>
<th>(c_{s1})</th>
<th>(c_{sn})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b^{low})</td>
<td>(A^+)</td>
<td>(A^-)</td>
<td>(b)</td>
</tr>
<tr>
<td>(b^{upp})</td>
<td>(b^{low})</td>
<td>(b^{upp})</td>
<td></td>
</tr>
<tr>
<td>(b_{s1}^{low})</td>
<td>(T_{s1} =)</td>
<td>(W_{s1} =)</td>
<td>(b_{s1})</td>
</tr>
<tr>
<td>(b_{s1}^{upp})</td>
<td>(T_{s1} \leq)</td>
<td>(W_{s1} \leq)</td>
<td></td>
</tr>
<tr>
<td>(b_{sn}^{low})</td>
<td>(T_{sn} =)</td>
<td>(W_{sn} =)</td>
<td>(b_{sn})</td>
</tr>
<tr>
<td>(b_{sn}^{upp})</td>
<td>(T_{sn} \leq)</td>
<td>(W_{sn} \leq)</td>
<td></td>
</tr>
<tr>
<td>(b_c^{low})</td>
<td>(C =)</td>
<td>(C_{s1} =)</td>
<td>(b_c)</td>
</tr>
<tr>
<td>(b_c^{upp})</td>
<td>(C \leq)</td>
<td>(C_{s1} \leq)</td>
<td></td>
</tr>
<tr>
<td>(x_0^{low})</td>
<td>(x_0 =)</td>
<td>(x_{sn} =)</td>
<td>(x_0^{upp})</td>
</tr>
<tr>
<td>(x_{sn}^{low})</td>
<td>(x_{sn} \leq)</td>
<td>(x_{sn} \leq)</td>
<td>(x_{sn}^{upp})</td>
</tr>
</tbody>
</table>
Consider LP with block-diagonal structure, linking constraints, and linking variables (the kind of problem we want to solve):

<table>
<thead>
<tr>
<th>min</th>
<th>$c_0$</th>
<th>$c_{s1}$</th>
<th>$c_{sn}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_{low}$ ≤ $A^{-}$ &amp; $A^\leq$ ≤ $b_{low}$ ≤ $b_{s1}$ ≤ $b_{sn}$ ≤ $b_{upper}$</td>
<td>$b_{low}$ ≤ $T_{s1} = T_{s1}^\leq W_{s1} = W_{s1}^\leq$ ≤ $b_{s1}$ ≤ $b_{sn}$ ≤ $b_{upper}$</td>
<td>$b_{low}$ ≤ $C^{-}$ &amp; $C^\leq$ ≤ $b_{low}$ ≤ $b_{c}$ ≤ $b_{upper}$</td>
<td>$x_0_{low}$ ≤ $X_0$ ≤ $x_0_{upper}$</td>
</tr>
</tbody>
</table>

$\rightarrow$ Time to generate $n+1$ blocks in parallel $\ll$ time to generate monolithic model
Outlook
Outlook

• Model generation and solution are currently separated
  – Integrate those steps into one user friendly process
  – Better user control of GAMS/PIPS
    • options (algorithmic, limits, tolerances)

• Analyze IO bottlenecks (generation/solving)

• Annotation can be adapted for other Decomposition approaches (e.g. CPLEX Benders)

• GAMS-MPI/Embedded Code:
  – Implementation of Benders Decomposition in GAMS for ESM using the GAMS embedded code facility with Python package mpi4py to work with MPI (see talk of L. Westermann, WC-02)

Project BEAM-ME

A PROJECT BY

GAMS
H L R I S
JÜLICH

Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag