Formulation of a Simple Market Clearing

STEPS

1. Set definitions
2. Data entry
3. Variables specification
4. Equations specification
   a. declaration
   b. algebraic structure specification
5. Model statement
6. Solve statement
Formulation of a Simple Market Clearing

- Demand: \( P \geq P_d = 6 - 0.3*Q_d \)
- Supply: \( P \leq P_s = 1 + 0.2*Q_s \)
- Equilibrium: \( Q_s \geq Q_d \) and \( P, Q_s, Q_d \geq 0 \)

**POSITIVE VARIABLE**

- \( P \) Equilibrium price
- \( Q_d \) Quantity demanded
- \( Q_s \) Quantity supply

**EQUATION**

- DemandPrice Demand equation
- SupplyPrice Supply equation
- Qbalance Equilibrium equation

```
DemandPrice.. P =G= 6-0.3*Qd ;
SupplyPrice.. 1+0.2*Qs =G= P ;
Qbalance.. Qs =G= Qd ;
```

**MODEL** EQUIL /DemandPrice.Qd
SupplyPrice.Qs
Qbalance.P /

**OPTION** MCP = PATH ;
**SOLVE** EQUIL using MCP ;
Formulation of a Simple Market Clearing

IF

Qd > 0  then P = 6 - 0.3*Qd
Qs > 0  then P = 1 + 0.2Qs
P > 0  then Qs = Qd

Implies that Pd = Ps = P

Price ($)

S

P

D

Qd = Qs

Quantity
GAMS Solution

Solution

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>EQU DemandPrice</td>
<td>6.000</td>
<td>6.000</td>
</tr>
<tr>
<td>----</td>
<td>EQU SupplyPrice</td>
<td>-1.000</td>
<td>-1.000</td>
</tr>
<tr>
<td>----</td>
<td>EQU Qbalance</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>VAR P</td>
<td>.</td>
<td>3.000</td>
</tr>
<tr>
<td>----</td>
<td>VAR Qd</td>
<td>.</td>
<td>10.000</td>
</tr>
<tr>
<td>----</td>
<td>VAR Qs</td>
<td>.</td>
<td>10.000</td>
</tr>
</tbody>
</table>

---- VARIABLE P.L = 3.000 Equilibrium price
VARIABLE Qd.L = 10.000 Quantity demanded
VARIABLE Qs.L = 10.000 Quantity supply

At Equilibrium:
Pd = Ps = P => Pd = 6 – 0.3*10 = 3
Ps = 1 + 0.2*10 = 3
Qd = Qs = 10
**Variable Specification**

GAMS requires variables in each problem to be identified. In the example, we have variables $P$, $Q_d$, $Q_s$

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>POSITIVE VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$Q_d$</td>
</tr>
<tr>
<td>Equilibrium price</td>
<td>Quantity demanded</td>
</tr>
<tr>
<td>$Q_s$</td>
<td></td>
</tr>
<tr>
<td>Quantity supply</td>
<td></td>
</tr>
</tbody>
</table>

2 types of variables

**VARIABLE** unrestricted variables

**POSITIVE VARIABLE** restricted variables to be nonnegative

\[
\begin{align*}
    P & \geq 0 \\
    Q_d & \geq 0 \\
    Q_s & \geq 0 
\end{align*}
\]
Dissecting GAMS – Equation naming

- **Equation Specification** consists of two parts.

  1. **Naming equations:**
     
     GAMS requires the modeler name each equation, which is active in the model. In the example, the equations are named in the **EQUATION** line:

     ```
     EQUATION
     DemandPrice
     SupplyPrice
     Qbalance
     Demand equation
     Supply equation
     Equilibrium equation;
     ```

     The name for each equation can be anything up to 31 characters.
(2) Specifying algebraic structure:

After naming equations, the exact algebraic structure of equations must be specified by using .. notation.

\begin{align*}
\text{DemandPrice..} & \quad P = \geq 6 - 0.3*Qd \\
\text{SupplyPrice..} & \quad 1 + 0.2*Qs = \geq P \\
\text{Qbalance..} & \quad Qs = \geq Qd
\end{align*}

This algebraic form involves use of a special syntax to tell the exact form of the equation that may actually be an inequality.

- \(=E=\) indicates an equality constraint
- \(=L=\) indicates a less than or equal to constraint
- \(=G=\) indicates a greater than or equal to constraint
Model Specification

**MODEL** statement is used to identify models that will be solved. It involves 2 steps:

- give name of the model (e.g. EQUIL)
- specify equations that will be included in the model in slashes / /

**MCP** = Mixed Complementary Problem

**MCP** uses ‘.’ as complementary

```
MODEL EQUIL /DemandPrice.Qd
              SupplyPrice.Qs
              Qbalance.P  /

Omitting Qbalance equation
```
Dissecting GAMS

- **Solve Specification**

  `SOLVE` causes GAMS to use a solver to the model named (EQUIL) immediately after the `SOLVE` statement.

  ```
  SOLVE EQUIL using MCP ;
  ```

  **MCP = Mixed Complementary Problem**

  That model must already have been defined in a `MODEL` statement.

  ```
  MODEL EQUIL /DemandPrice.Qd
  SupplyPrice.Qs
  Qbalance .P / ;
  ```
GAMS requires to terminate each statement with a `;`.

**POSITIVE VARIABLE**

- \( P \) \hspace{1cm} \text{Equilibrium price}
- \( Q_d \) \hspace{1cm} \text{Quantity demanded}
- \( Q_s \) \hspace{1cm} \text{Quantity supply}

**EQUATION**

- \( \text{DemandPrice} \) \hspace{1cm} \text{Demand equation}
- \( \text{SupplyPrice} \) \hspace{1cm} \text{Supply equation}
- \( \text{Qbalance} \) \hspace{1cm} \text{Equilibrium equation}

\[
\begin{align*}
\text{DemandPrice..} & \quad P =G= 6-0.3*Q_d ; \\
\text{SupplyPrice..} & \quad 1+0.2*Q_s =G= P ; \\
\text{Qbalance..} & \quad Q_s =G= Q_d ;
\end{align*}
\]

**MODEL** EQUIL /DemandPrice.Qd, SupplyPrice.Qs, Qbalance.P /

**OPTION** MCP = PATH ;

**SOLVE** EQUIL using MCP ;

`;` is a very important part of the syntax. The omission often causes many syntax errors.
Dissecting GAMS – Finding errors

\[
\begin{align*}
\text{DemandPrice..} & \quad P & =G= & 6-0.3*Qd & ; \\
\text{SupplyPrice..} & \quad 1+0.2*Qs & =G= & P \\
\text{Qbalance..} & \quad Qs & =G= & Qd & ; \\
\end{align*}
\]

Error Messages

--- Starting compilation
--- SMALLMCP.GMS(14) 1 Mb 1 Error
*** Error 409 in C:\TASANA\685CGEPROJECT\SMALLMCP.GMS
Unrecognizable item - skip to find a new statement
looking for a ';' or a key word to get started again
--- SMALLMCP.GMS(22) 1 Mb 2 Errors
*** Error 257 in C:\TASANA\685CGEPROJECT\SMALLMCP.GMS
Solve statement not checked because of previous errors
--- SMALLMCP.GMS(24) 1 Mb 5 Errors
*** Error 141 in C:\TASANA\685CGEPROJECT\SMALLMCP.GMS
Symbol neither initialized nor assigned
   A wild shot: You may have spurious commas in the explanatory
text of a declaration. Check symbol reference list.
6. Run GAMSIDE (con’t)

--- Starting compilation => check if your file is ok
--- SMALLMCP.GMS(26) 1 Mb => (26) indicate line it is on
--- Starting execution => execute your file
--- SMALLMCP.GMS(20) 2 Mb => set up the problem
--- Generating model EQUIL => size of the problem
--- SMALLMCP.GMS(22) 2 Mb
--- 3 rows, 3 columns, and 6 non-zeroes.
--- SMALLMCP.GMS(22) 2 Mb
--- Executing PATH => start solver and gives a
--- SMALLMCP.GMS(22) 2 Mb name for which solver is used
--- Restarting execution => GAMS restarts
--- SMALLMCP.GMS(22) 0 Mb
--- Reading solution for model EQUIL => GAMS stops without errors
--- SMALLMCP.GMS(25) 2 Mb

*** Status: Normal completion => GAMS stops without errors
6. Run GAMSIDE (con’t)

- double click on lines in the process window to access output
- positioning of your access is determined by the color of the line
- blue lines => open *.LST file and jump to line in *.LST file
- black lines => open *.LST file and jump to a location of previous blue line
- red lines => jump to *.gms file (your program) where errors occur
### Solution

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>EQU DemandPrice</td>
<td>6.000</td>
<td>6.000</td>
<td>+INF</td>
</tr>
<tr>
<td>EQU SupplyPrice</td>
<td>-1.000</td>
<td>-1.000</td>
<td>+INF</td>
</tr>
<tr>
<td>EQU Qbalance</td>
<td>.</td>
<td>.</td>
<td>+INF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>VAR P</td>
<td>.</td>
<td>3.000</td>
<td>+INF</td>
</tr>
<tr>
<td>VAR Qd</td>
<td>.</td>
<td>10.000</td>
<td>+INF</td>
</tr>
<tr>
<td>VAR Qs</td>
<td>.</td>
<td>10.000</td>
<td>+INF</td>
</tr>
</tbody>
</table>

```plaintext
----- VARIABLE P.L = 3.000 Equilibrium price
VARIABLE Qd.L = 10.000 Quantity demanded
VARIABLE Qs.L = 10.000 Quantity supply
```

### At Equilibrium:

\[ P_d = P_s = P \Rightarrow P_d = 6 - 0.3 \times 10 = 3 \]
\[ P_s = 1 + 0.2 \times 10 = 3 \]

\[ Q_d = Q_s = 10 \]