

Chapter 1

General-Equilibrium Modeling using GAMS and MPS/GE: Some Basics

James R. Markusen

Thomas F. Rutherford

University of Colorado, Boulder

1. Introduction to applied general-equilibrium modeling
 - (1) Multiple interacting agents
 - (2) Individual behavior based on optimization
 - (3) Most agent interactions are mediated by markets and prices
 - (4) Equilibrium occurs when endogenous variables (e.g., prices) adjust such that
 - (i) agents, subject to the constraints they face, cannot do better by altering their behavior
 - (ii) markets (generally, not always) clear so, for example, supply equals demand in each market.

2. Steps in Applied General-Equilibrium Modeling

- (1) Specify dimensions of the model.
 - Numbers of goods and factors
 - Numbers of consumers
 - Numbers of countries
 - Numbers of active markets

- (2) Chose functional forms for production, transformation, and utility functions; specification of side constraints.
 - Includes choice of outputs and inputs for each activity
 - Includes specification of initially slack activities

- (3) Construct micro-consistent data set.
 - Data satisfies zero profits for all activities, or if profits are positive, assignment of revenues
 - Data satisfies market clearing for all markets

- (4) Calibration – parameters are chosen such that functional forms and data are consistent.
 - By “consistent” we mean that the data represent a solution to the model

- (5) Replication – run model to see if it reproduces the input data.
- (6) Counter-factual experiments.

Example M1: 2-good, 2-factor closed economy with fixed factor endowments, one representative consumer.

Take a very simply economy, two sectors (X and Y), two factors (L and K), and one representative consumer (utility function W).

L and K are in inelastic (fixed) supply, but can move freely between sectors. p_x , p_y , p_l , and p_k are the prices of X, Y, L and K, respectively.

I is consumer's income and p_w will be used later to denote the price of one unit of W.

These are the equations of the model.

$$(1) \quad X = X(L_x, K_x)$$

$$(2) \quad Y = Y(L_y, K_y)$$

$$(3) \quad L^* = L_x + L_y$$

$$(4) \quad K^* = K_x + K_y$$

$$(5) \quad W = W(X, Y)$$

$$(6) \quad I = p_l L^* + p_k K^* = p_x X + p_y Y$$

How do we find equilibrium, which in this case is a set of prices, and factor allocations to the two sectors.

Equilibrium could be solved for by a constrained optimization problem: Max (5) subject to the constraints (1), (2), (3), (4), and (6).

While this would work, the usefulness of this approach breaks down quickly as the model becomes more complicated.

The alternative approach is to convert the problem to a system of equations, and solve that system.

First, solve the underlying cost minimization problems for producers and consumers, so that individual optimizing behavior is embedded in the model.

In the present model, we want to solve for cost functions for each sector, which embody efficient and optimizing cost-minimizing behavior.

These give the minimum cost of producing a good at given factor prices.

Unit cost functions for X and Y $c_x = c_x(p_l, p_k), \quad c_y = c_y(p_l, p_k)$

Unit cost (expenditure) function for W $e = e(p_x, p_y)$

The next crucial step is provided by theory, Shepard's lemma in particular.

$$\frac{\partial cx}{\partial p_l} = cx_{pl} = \text{X producer's demand for labor per unit of output}$$

$$\frac{\partial cx}{\partial p_k} = cx_{pk} = \text{X producer's demand for capital per unit of output}$$

$$\frac{\partial e}{\partial p_x} = e_{px} = \text{Consumer's demand for X per unit of utility}$$

Now we are in a position to specify general equilibrium as the solution to a square system of 9 weak inequalities in 9 unknowns. These are as follows:

General-equilibrium formulated as a square system:

-
- | | | |
|----------------------------------|-------------------------|--------------------------|
| (1) Non-positive profits for X | $cx(p_l, p_k) \geq p_x$ | |
| (2) Non-positive profits for Y | $cy(p_l, p_k) \geq p_y$ | zero-profit inequalities |
| (3) Non-positive "profits" for W | $e(p_x, p_y) \geq p_w$ | |
-
- | | | |
|--------------------------------|----------------------------|--------------------------|
| (4) Supply \geq Demand for X | $X \geq e_{px}(p_x, p_y)W$ | |
| (5) Supply \geq Demand for Y | $Y \geq e_{py}(p_x, p_y)W$ | |
| (6) Supply \geq Demand for W | $W \geq I / p_w$ | market-clearing inequal. |
| (7) Supply \geq Demand for L | $L^* \geq cx_w X + cy_w Y$ | |
| (8) Supply \geq Demand for K | $K^* \geq cx_r X + cy_r Y$ | |
-
- | | | |
|--------------------|-------------------------|----------------|
| (9) Income balance | $I = p_l L^* + p_k K^*$ | income balance |
|--------------------|-------------------------|----------------|

These weak inequalities can be solved for the unknowns

X , Y , W , p_x , p_y , p_w , p_l , p_k , and I .

Note that these inequalities are of three types, and this is generally true of a very large class of general-equilibrium models.

- Zero-profit conditions, inequalities (1)-(3) in the above example.
- Market clearing conditions, inequalities (4)-(8) in the above example
- Income balance, equation (9) in the above example.

Formulating equilibrium as a complementarity problem requires that each inequality is associated with a particular variable.

If a zero profit condition holds as a strict inequality in equilibrium, profits for that activity are negative, that *activity* will not be used (one good could be produced with several alternative activities).

The complementary variable to a *zero-profit condition* is a *quantity*, the activity level.

If a market-clearing condition holds as a strict inequality, supply exceeds demand for that good or factor in equilibrium so its price must be zero.

Thus the complementary variable to a *market clearing inequality* is the *price* of that good or factor.

The complementary variable to an income balance equation is just the income of that agent.

The correct association of inequalities and unknowns in the square system is thus:

InequalityComplementary Variable

(1) Non-positive profits for X	$cx(p_l, p_k) \geq p_x$	X
(2) Non-positive profits for Y	$cy(p_l, p_k) \geq p_y$	Y
(3) Non-positive "profits" for W	$e(p_x, p_y) \geq p_w$	W
(4) Supply \geq Demand for X	$X \geq e_{px}(p_x, p_y)W$	$p_x,$
(5) Supply \geq Demand for Y	$Y \geq e_{py}(p_x, p_y)W$	p_y
(6) Supply \geq Demand for W	$W \geq I / p_w$	p_w
(7) Supply \geq Demand for L	$L^* \geq cx_w X + cy_w Y$	p_l
(8) Supply \geq Demand for K	$K^* \geq cx_r X + cy_r Y$	p_k
(9) Income balance	$I = p_l L^* + p_k K^*$	I

Now let's turn to the issue of starting with a micro-consistent data set. That is, let's start with a set of numbers that satisfy *zero profits*, *market clearing*, and *income balance*.

The above problems can be thought of as consisting of three production activities, X, Y, and W, and four markets, X, Y, L, and K.

We will represent the initial data for this economy by a rectangular matrix. This matrix is related to the concept of a "SAM" – social accounting matrix, which is discussed later.

There are two types of *columns* in the rectangular matrix, corresponding to *production activities(sectors)* and *consumers*. In the model outlined above, there are three production sectors (X, Y and W) and a single consumer (CONS).

Rows correspond to *markets*. Complementary variables are prices, so we have listed the price variables on the left to designate rows.

A positive entry signifies a receipt (sale) in a particular market. A negative entry signifies an expenditure (purchase) in a particular market.

Reading down a production column, we then observe a complete list of the transactions associated with that activity.

	Production Sectors			Consumers	
Markets	X	Y	W	CONS	Row sum
PX	100		-100		0
PY		100	-100		0
PW			200	-200	0
PL	-25	-75		100	0
PK	-75	-25		100	0
Column sum	0	0	0	0	

A rectangular SAM is balanced or “micro-consistent” when row and column sums are zeros.

Positive numbers represent the value of commodity flows into the economy (sales or factor supplies), while negative numbers represent the value of commodity flows out of the economy (factor demands or final demands).

A *row sum is zero* if the total amount of commodity flowing into the economy equals the total amount of commodity flowing out of the economy. This is *market clearance*, and one such condition applies for each commodity in the model.

Columns in this matrix correspond to production sectors or consumers. A production sector *column sum is zero* if the value of outputs equals the cost of inputs.

A consumer column is balanced if the sum of primary factor sales equals the value of final demands.

Zero column sums thus indicate *zero profits* or “*product exhaustion*” in an alternative terminology.

Finally, we emphasize that the numbers of the matrix are values, prices times quantities. The modeler is free as to how to interpret these as prices versus quantities.

A good practice is to choose units so that as many things initially are equal to one as possible.

Prices can be chosen as one, and “representative quantities” for activities can be chosen such that activity levels are also equal to one (e.g., activity X run at level one produces 100 units of good X).

In the case of taxes, both consumer and producer prices cannot equal one of course, a point we will return to in a later section.

Now we are in a position to adopt functional forms and write an actual GAMS program to solve this model.

First, we specify this general-equilibrium model as an MCP, writing out all the functions.

We use very simple Cobb-Douglas functions for the three activities. The share parameters for the functions are given in the data matrix above.

Goods in the utility function get equal shares of 0.5:

$$e(p_x, p_y) = p_x^{0.5} p_y^{0.5}$$

X is capital intensive with capital having a share of 0.75 and labor a share of 0.25.

$$c_x(p_l, p_k) = p_l^{0.25} p_k^{0.75}$$

Y is labor intensive with the opposite ordering of shares.

$$c_y(p_l, p_k) = p_l^{0.75} p_k^{0.25}$$

While we will not go into detail about GAMS syntax here, a few final points with respect to the actual program follows.

- (1) The opening line \$TITLE is not necessary, but used to the model in the listing (output) file.
- (2) \$ONTEXT.....\$OFFTEXT is a way of designating a block of comments, to be ignored by GAMS.

In this case, we put our data matrix inside this block, meaning it is not actually used in the computation.

- (3) A text line can also be preceded by a *. GAMS ignores any line beginning with a *.
- (4) We declare the parameter names, then assign them values (note where semi-colons do and do not go).
- (5) Next we declare positive variables and then equation names. We write out the equation names in the syntax shown [equation name], then the equation itself ending with a semi-colon.

Note the use of the reference quantities such as “100”, “75 ” etc. in the equations.

This will ensure that the activity levels will be $X = Y = W = 1$ in the initial solution to the model.

- (6) GAMS was written to use greater-than-or-equal-to syntax (=G=). Also note that we have avoided having variables in denominators, since if a variable (even temporarily during the execution of the algorithm) has a value of zero, this causes a divided by zero problem and *may* crash the solver.
- (7) Then the model is specified, and we chose a numeraire (recall from theory that only relative prices are determined). Here we choose utility as the numeraire, so that factor prices are then real values in terms of utility. The notation is PW.FX, “FX” for “fixed”.
- (8) Before the solve statement, we are going to help the solver by giving starting values for the variables. The syntax is, for example X.L, where the “L” stands for “level”. Default values are zero, and in non-linear problems it is very helpful and indeed sometimes necessary to help the solver with some initial guesses. We constructed this problem knowing the answer, so I give those values as .L values.
- (9) Finally, the solve statement.
After the first solve statement, we do two counterfactual experiments. The first sets a tax of 0.50 on the inputs to X production. Then we have a second solve statement. Finally,

we remove the tax and double the labor endowment of the economy.

“TX” is a parameter which sets the tax, and “LENDOW” is a multiplier on the initial labor endowment.

\$TITLE Model M1_MCP: Closed 2x2 Economy - An Introduction to
* the Basics

\$ONTEXT

Markets	Production Sectors			Consumers
	X	Y	W	CONS
PX	100		-100	
PY		100	-100	
PW			200	-200
PL	-25	-75		100
PK	-75	-25		100

\$OFFTEXT

PARAMETERS

TX Ad-valorem tax rate for X sector inputs
LENDOW Labor endowment multiplier;

TX = 0;
LENDOW = 1;

POSITIVE VARIABLES

X
Y
W
PX
PY
PW
PL
PK
CONS;

EQUATIONS

PRF_X Zero profit for sector X
PRF_Y Zero profit for sector Y

PRF_W Zero profit for sector W (Hicksian welfare index)

MKT_X Supply-demand balance for commodity X

MKT_Y Supply-demand balance for commodity Y

MKT_L Supply-demand balance for primary factor L

MKT_K Supply-demand balance for primary factor K

MKT_W Supply-demand balance for aggregate demand

I_CONS Income definition for CONS;

* Zero profit inequalities

PRF_X.. 100 * PL**0.25 * PK**0.75 * (1+TX) =G= 100*PX;

PRF_Y.. 100 * PL**0.75 * PK**0.25 =G= 100*PY;

PRF_W.. 200 * PX**0.5 * PY**0.5 =G= 200*PW;

* Market clearance inequalities

$$\text{MKT_X..} \quad 100 * X =G= 100 * W * \text{PX}^{**0.5} * \text{PY}^{**0.5} / \text{PX};$$

$$\text{MKT_Y..} \quad 100 * Y =G= 100 * W * \text{PX}^{**0.5} * \text{PY}^{**0.5} / \text{PY};$$

$$\text{MKT_W..} \quad 200 * W =E= \text{CONS} / \text{PW};$$

$$\begin{aligned} \text{MKT_L..} \quad 100 * \text{LENDOW} =G= & 25 * X * \text{PL}^{**0.25} * \text{PK}^{**0.75} / \text{PL} \\ & + 75 * Y * \text{PL}^{**0.75} * \text{PK}^{**0.25} / \text{PL}; \end{aligned}$$

$$\begin{aligned} \text{MKT_K..} \quad 100 =G= & 75 * X * \text{PL}^{**0.25} * \text{PK}^{**0.75} / \text{PK} \\ & + 25 * Y * \text{PL}^{**0.75} * \text{PK}^{**0.25} / \text{PK}; \end{aligned}$$

* Income balance equations

$$\begin{aligned} \text{I_CONS..} \quad \text{CONS} =E= & 100 * \text{LENDOW} * \text{PL} + 100 * \text{PK} + \\ & \text{TX} * 100 * X * \text{PL}^{**0.25} * \text{PK}^{**0.75}; \end{aligned}$$

```
MODEL ALGEBRAIC /PRF_X.X, PRF_Y.Y, PRF_W.W, MKT_X.PX,  
MKT_Y.PY, MKT_L.PL, MKT_K.PK, MKT_W.PW, I_CONS.CONST /;
```

```
*           Chose a numeraire (not necessary)
```

```
PW.FX = 1;
```

```
*           Set initial values of variables:
```

```
X.L=1; Y.L=1; W.L=1; PX.L=1; PY.L=1; PK.L=1; PL.L=1;  
CONS.L=200;
```

```
SOLVE ALGEBRAIC USING MCP;
```

```
*           Solve the counterfactuals:
```

```
TX = 0.5;  
LENDOW = 1;  
SOLVE ALGEBRAIC USING MCP;
```

```
TX = 0;
```

```
LENDOW = 2;  
SOLVE ALGEBRAIC USING MCP;
```

This file is saved in the relevant directory as M1_MCP.GMS, although you can use any name you want, it doesn't have to correspond to the model name.

At the DOS prompt, >, type GAMS M1_MCP. This command runs the model.

The output or listing file will automatically be written to the same directory, with name M1_MCP.LST.

In the listing file, we will see the following information, along with a statement that the solution is normal and optimal.

The list of 1.000 indicates we successfully reproduced our benchmark data as an equilibrium, since we chose units such that this would be the case.

		LOWER	LEVEL	UPPER	MARGINAL
----	EQU PRF_X	.	.	+INF	1.000
----	EQU PRF_Y	.	.	+INF	1.000
----	EQU PRF_W	.	.	+INF	1.000
----	EQU MKT_X	.	.	+INF	1.000
----	EQU MKT_Y	.	.	+INF	1.000
----	EQU MKT_L	-100.000	-100.000	+INF	1.000
----	EQU MKT_K	-100.000	-100.000	+INF	1.000
----	EQU MKT_W	.	.	.	1.000
----	EQU I_CONS	.	.	.	200.000

PRF_X Zero profit for sector X
 PRF_Y Zero profit for sector Y
 PRF_W Zero profit for sector W (Hicksian welfare index)
 MKT_X Supply-demand balance for commodity X
 MKT_Y Supply-demand balance for commodity Y
 MKT_L Supply-demand balance for primary factor L
 MKT_K Supply-demand balance for primary factor L
 MKT_W Supply-demand balance for aggregate demand
 I_CONS Income definition for CONS

	LOWER	LEVEL	UPPER	MARGINAL
----- VAR X	.	1.000	+INF	.
----- VAR Y	.	1.000	+INF	.
----- VAR W	.	1.000	+INF	.
----- VAR PX	.	1.000	+INF	.
----- VAR PY	.	1.000	+INF	.
----- VAR PW	1.000	1.000	1.000	EPS
----- VAR PL	.	1.000	+INF	.
----- VAR PK	.	1.000	+INF	.
----- VAR CONS	.	200.000	+INF	.

Now let's look at the results for our first counterfactual, in which we place a 50% tax on the inputs to X production.

***** counterfactual: tax on X inputs

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR X	.	0.845	+INF	.
---- VAR Y	.	1.147	+INF	.
---- VAR W	.	0.985	+INF	.
---- VAR PX	.	1.165	+INF	.
---- VAR PY	.	0.859	+INF	.
---- VAR PW	1.000	1.000	1.000	-1.602E-12
---- VAR PL	.	0.903	+INF	.
---- VAR PK	.	0.739	+INF	.
---- VAR CONS	.	196.946	+INF	.

We see that X production decreases, Y production increases, and welfare falls due to the distortionary nature of the tax, even though the tax revenue is redistributed back to the consumer. There is also a redistribution of income between factors. The relative price of capital, the factor used intensively in X falls, and the relative price of labor rises as resources are shifted to Y production.

In the second counterfactual, we remove the tax, and double the labor endowment of the economy.

*****counterfactual: double labor endowment (zero tax)

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR X	.	1.189	+INF	.
---- VAR Y	.	1.682	+INF	.
---- VAR W	.	1.414	+INF	.
---- VAR PX	.	1.189	+INF	.
---- VAR PY	.	0.841	+INF	.
---- VAR PW	1.000	1.000	1.000	6.617E-11
---- VAR PL	.	0.707	+INF	.
---- VAR PK	.	1.414	+INF	.
---- VAR CONS	.	282.843	+INF	.

Exercises:

- (1) explain the factor-price changes in the two counterfactuals relative to the benchmark.
- (2) change the capital endowment to 200, along with a labor endowment of 200. You should show that all quantities double from the base case, and all prices are unchanged.
- (3) Starting with the benchmark, lower the tax on X inputs to 0.25 and put a tax of 0.25 on Y inputs. What happens to welfare?
- (4) Starting with the benchmark, assume that only 50% of the tax revenue is redistributed back to the consumer.
- (5) change the X production functions to fixed coefficients. That is,

$$X = \min[L_x, K_x]; \quad \text{Hint: the cost function} \quad c_x = 0.25 * p_l + 0.75 * p_k$$

(A) reproduce the benchmark, (B) Run the counterfactuals and see what difference this makes.

. The MPS/GE subsystem of GAMS

GAMS now include a higher-level language, written by Rutherford, called MPS/GE, which stands for mathematical programming system for general equilibrium. MPS/GE uses the MCP solver in GAMS.

First, the program has routines for calibrating and writing all constant-returns CES and CET functions, up to three levels of nesting.

Second, and closely related, the form of the data required to specify a CES/CET function is exactly the data modelers have, so there is a swift and easy move from an accounting matrix to the calibration of the model.

Third, a lot of market-clearing and income-balance equations are written automatically by MPS/GE so the modeler doesn't have to worry about doing so.

Fourth, and closely related, a lot of errors that can occur when a modeler writes out his or her equations cannot occur in MPS/GE. If there is a tax or markup, for example, the revenues must be assigned to some agent and will be allocated automatically to that agent by the income-balance properties of the coding.

A few key words.

SECTOR (ACTIVITY)

Production activities that convert commodity inputs into commodity outputs. The variable associated with a sector is the activity level.

COMMODITY (MARKETS)

A good or factor. The variable associated with a commodity is its price, not its quantity.

CONSUMERS

Individuals who supply factors and receive tax revenues, markups, and pay subsidies. In imperfectly competitive models, firm owners can be designated as consumers. A government that receives tax revenue and buys public goods is also designated as a consumer. The variable associated with a consumer is income from all sources.

AUXILIARY

Additional variables, such as markup formulae or taxes with endogenous values which are functions of other variables such as prices and quantities. Please note the spelling of auxiliary: mistakes cause MPS/GE to crash, and you won't know why.

CONSTRAINT

An equation that is typically used to set the value of an auxiliary variable. In these appendix programs, constraint equations will be used to set the values of markups, which are auxiliary variables.

Here is what an MPS/GE program, embedded in a GAMS file, looks like, where the model name is M1_MCP.

GAMS statements such as declaring sets, parameters, parameter values, etc.

```
**** now control is passed to the MPS/GE subsystem ****
```

```
$ONTEXT [this tells the GAMS compiler to ignore what  
follows, but the MPS/GE compiler will recognize  
the model statement that follows and will begin  
to pay attention]
```

```
$MODEL: M1_MCP
```

Declare sectors, commodities, consumers, auxiliary variables

Production Blocks

Demand Blocks

Constraint equations

\$OFFTEXT [control is passed back to GAMS]

**** now we are back in GAMS ****

\$SYSINCLUDE MPSGESET M1_MCP

GAMS statements such as setting starting values of variables, other parameter values, etc.

\$INCLUDE M1_MCP.GEN

SOLVE M1_MCP USING MCP;

GAMS statements processing output

Below, we formulate exactly the same problem introduced above using MPS/GE. We present the file M1_MCP.GMS and then discuss its details.

\$TITLE Model M1_MPS: Closed 2x2 Economy

\$ONTEXT

This is the exact same model as M1_MCP.GMS but uses the MPS/GE format.

Markets	Production Sectors			Consumers
	X	Y	W	CONS
PX	100		-100	
PY		100	-100	
PW			200	-200
PL	-25	-75		100
PK	-75	-25		100

\$OFFTEXT

PARAMETERS

TX Ad-valorem tax rate for X sector inputs
 LENDOW Labor endowment multiplier;

```
TX = 0;  
LENDOW = 1;
```

```
$ONTEXT
```

```
$MODEL:M1_MPS
```

```
$SECTORS:
```

```
  X      ! Activity level for sector X  
  Y      ! Activity level for sector Y  
  W      ! Activity level for sector W (welfare index)
```

```
$COMMODITIES:
```

```
  PX      ! Price index for commodity X  
  PY      ! Price index for commodity Y  
  PL      ! Price index for primary factor L  
  PK      ! Price index for primary factor K  
  PW      ! Price index for welfare (expenditure function)
```

```
$CONSUMERS:
```

```
  CONS      ! Income level for consumer CONS
```

\$PROD:X s:1
O:PX Q:100
I:PL Q:25 A:CONS T:TX
I:PK Q:75 A:CONS T:TX

\$PROD:Y s:1
O:PY Q:100
I:PL Q:75
I:PK Q:25

\$PROD:W s:1
O:PW Q:200
I:PX Q:100
I:PY Q:100

\$DEMAND:CONS
D:PW Q:200
E:PL Q:(100*LENDOW)
E:PK Q:100

\$OFFTEXT

```
$SYSINCLUDE mpsgeset M1_MPS
```

```
PW.FX = 1;
```

```
$INCLUDE M1_MPS.GEN  
SOLVE M1_MPS USING MCP;
```

```
*           Solve the counterfactuals
```

```
TX = 0.5;  
LENDOW = 1;
```

```
$INCLUDE M1_MPS.GEN  
SOLVE M1_MPS USING MCP;
```

```
TX = 0;  
LENDOW = 2;
```

```
$INCLUDE M1_MPS.GEN  
SOLVE M1_MPS USING MCP;
```

Now some more details.

(1) Production blocks

The terminology here is a bit confusing, since MPS/GE takes the information in a production block and generates a cost function, not a production function.

But the variable complementary with a production block (cost function) is an activity level. Let's take an example from the above program, adding the price field.

```
$PROD:Y   S:1
          O:PY       Q:100     P:1
          I:PL       Q: 75     P:1
          I:PK       Q: 25     P:1
```

First line

Name of activity (Y), value of substitution (here s:1) and transformation elasticities if there are several outputs. Default elasticity of substitution is 0 (not 1!).

First column

Names of commodity outputs (O:) and inputs (I:).

Second column

Reference commodity quantities (Q:) – used for calibration. Default = 1 if none specified.

Third column

Reference commodity prices (P) – used for calibration. Default = 1 if none specified, which is why they are omitted in the program above.

MPS/GE then takes this information to construct a cost function and, as a feature of CES functions, it is globally defined by a single reference point.

Think of putting an isoquant labeled 100 units of output, with elasticity of substitution 1, through input points $L = 70$, $K = 30$, with slope $PL/PK = 1$. That is what MPS/GE does for you.

In this simple case, it constructs the cost function:

$$100*(PL^{.75})*(PK^{.25}) = G = 100*PY;$$

The saving from using MPS/GE might not seem like a big deal, but believe me with many inputs, different prices for all inputs, and an elasticity of substitution of 3.5, it is a huge saving indeed.

One example of the treatment of taxes (others will follow later, including those with endogenous rates) is in the production block for X.

```
$PROD:X  S:1
          O:PX  Q:100
          I:PL  Q:25   A:CONS  T:TX
          I:PK  Q:75   A:CONS  T:TX
```

The “A” field means “assign” the revenue from tax TX to the agent CONS. Read it as the statement “assign to agent CONS the revenue from tax rate TX on inputs L and K”.

A utility function is also represented by a production block; that is, utility is a good which is produced from commodity inputs (including possibly factor inputs such as leisure).

Here is the utility function (W for welfare), in which utility (good PW) is produced from inputs of X and Y. MPS/GE constructs the underlying expenditure (cost) function.

```
$PROD:W  S:1
          O:PW   Q:200
          I:PX   Q:100
          I:PY   Q:100
```

A consumer's income constraint is also represented by a "block" in this case called a demand block.

In what follows, the consumer demands the utility good PW (the "D" field), and receives income from endowments (the "E" fields) of labor and capital.

MPS/GE automatically handles tax revenue or subsidy payments in the background, adding or subtracting them to the consumer's endowment income.

\$DEMAND : CONS

D : PW Q : 200

E : PL Q : (100 * LENDOW)

E : PK Q : 100

MPS/GE also automatically looks after the market clearing conditions without the modeler having to worry about specifying these additional equations.

4. Example of how the Q and P fields are used to construct the underlying cost function.

All constant returns CES and CET functions can be completely characterized by a single point consisting of (1) input quantities (2) output quantities (3) input and output prices, and (4) the elasticity of substitution or transformation (there may be several levels of substitution elasticities).

MPS/GE constructs the underlying cost function from such a single observation. It is particularly important to specify the reference prices correctly. Consider the two production blocks:

```

$PROD: X      S:1
O:PX   Q:100  P:1
I:PL   Q: 25  P:1
I:PK   Q: 75  P:1

```

```

$PROD: X      S:1
O:PX   Q:100  P:1
I:PL   Q: 25  P:2
I:PK   Q: 75  P:0.667

```

These are both Cobb-Douglas production functions that can produce 100 X from inputs of 25 labor and 75 capital, and in each case the value of the inputs equals the value of the output.

But the isoquant has a different slope through that input combination in the two cases: the marginal rate of substitution in the first case is one, but 3 in the second case. Thus these are not the same technologies.

Later, with various taxes, it is important to divide values into price and quantity components when all prices cannot be normalized to one. Consider the following two production blocks.

```

$PROD: X      S:1
O:PX  Q:100  P:1
I:PL  Q: 25  P:1
I:PK  Q: 75  P:1

```

```

$PROD: X      S:1
O:PX  Q:100  P:1
I:PL  Q: 50  P:0.5
I:PK  Q: 75  P:1

```

In both cases, the values of inputs and output are the same, and would appear the same in the data matrix. But these are not the same functions, the first being a clearly more efficient technology than the second.

While both will have the same share parameters on L and K (0.25 and 0.75 respectively), the first technology will have a higher multiplicative “efficiency” parameter scaling up the output from given inputs (or scaling down the cost of output at given factor prices). MPS/GE will automatically calculate this scaling parameter.

More will be said about this when we get to an example with taxes in the benchmark data shortly.

Exercises:

Do the same exercises as for the mcp version.

5. Vector syntax for GAMS and MPS/GE.

GAMS has some features that streamline the formulation of the computer code, features that are immensely useful in large dimension problems in particular. We cannot go through all the features of GAMS syntax here, students must rely on the GAMS manual itself. But we can show you what these features are and how they are used.

First, actual numbers need not appear in MCP program equations or in MPS/GE production and demand blocks.. These can be specified somewhere else in the program or indeed read from external files.

Second, set notation can be used when convenient, as in cases where there are many goods, factors, countries or consumers. Combining these two features, data can be specified in an array or table, and read into the computation program in a straightforward way.

In what follows, we rewrite our basic model from above using these features. We present first the MPS/GE version and then the MCP version for completeness.

The program begins with the GAMS key word “TABLE”. Now our micro-consistency matrix is actually going to be used directly in the program to load the values of variables

and parameters. Then we use the GAMS key word “SET”. We use two sets here, the set of goods (X, Y) and the set of factors (L, K). “ALIAS” is a GAMS key word that allows two different designators for a given set.

We then declare parameters, and after that parameter assignments are used to extract the data from the TABLE.

Following these assignments of data to parameters, we use a “DISPLAY” statement, which will write out the values of the parameters in the listing file, so we can do a quick check that the assignments are correct.

Again, you should consult the GAMS manual for a much more complete description of these operations and definitions (for example the use of “” when referring to a specific element of a table or set; e.g., BENCH(I, “W”)).

The MPS/GE block follows. Note the absence of any numbers. Consider the production block:

```
$PROD:Z(I)  S:1
          O:PC(I)          Q:Z0(I)
          I:PF(F)         Q:FD0(F,I)   A:CONS   T:T(I)
```

Two production (cost) functions for two goods using two factors with different factor intensities and different tax rates across sectors are specified in this extremely parsimonious way.

Following the MPS/GE version, we present the MCP version of the basic model. The share parameters for the Cobb-Douglas functions are calculated from the data (ALPHA, BETA).

Note also the use of GAMS key words SUM (sum of) and PROD (production of).

Note also the use of the alternative set designators, in CMKET(I), for example.

If we have an equation for market $I = X$, for example, GAMS does not permit us to sum over all values of I on the right-hand side. But we can sum over the alias J, which is what we do.

In the counterfactuals, note again the use of quotation marks to designate a particular member of a set.

\$TITLE Model M1_VEC: Closed 2x2 Economy - An Introduction to the Basics

*This is the exact same model as M1_MPS.GMS but uses vector syntax.

TABLE BENCH(*,*) Benchmark financial flows (e.g. an input-output table)

	X	Y	W	CONS
X	100		-100	
Y		100	-100	
W			200	-200
L	-40	-60		100
K	-60	-40		100;

SET I Produced goods /X, Y/,
 F Factors of production /L, K/;

```
ALIAS (I, J), (F, FF);
```

```
PARAMETER
```

```

Z0(I)           Benchmark sectoral output,
FD0(F, I)       Benchmark factor demands,
C0(I)           Benchmark consumption demand,
E(F)            Factor endowments,
T(I)            Sectoral ad-valorem tax rate,
W0              Benchmark total consumption;
```

* Extract data from the original format into model-specific arrays:

```

Z0(I)           = BENCH(I, I);
FD0(F, I)       = -BENCH(F, I);
C0(I)           = -BENCH(I, "W");
W0              = SUM(I, C0(I));
E(F)            = BENCH(F, "CONS");
T(I)            = 0;
```

```
DISPLAY Z0, FD0, C0, E, T;
```

\$ONTEXT

\$MODEL:M1_VEC

\$SECTORS:

 Z(I) ! Commodity production index
 W ! Welfare index

\$COMMODITIES:

 PW ! Utility price index
 PC(I) ! Commodity price index
 PF(F) ! Factor price index

\$CONSUMERS:

 CONS ! Representative consumer

\$PROD:Z(I) s:1

 O:PC(I) Q:Z0(I)
 I:PF(F) Q:FD0(F,I) A:CONS T:T(I)

```
$PROD:W  s:1
          O:PW          Q:W0
          I:PC(I)      Q:C0(I)
```

```
$DEMAND:CONS
          D:PW          Q:W0
          E:PF(F)      Q:E(F)
```

```
$OFFTEXT
```

```
$SYSINCLUDE mpsgeset M1_VEC
```

```
PW.FX = 1;
```

```
$INCLUDE M1_VEC.GEN
```

```
SOLVE M1_VEC USING MCP;
```

* Solve the counterfactuals:

T("X") = 0.5;

\$INCLUDE M1_VEC.GEN
SOLVE M1_VEC USING MCP;

E("L") = 2 * E("L");
T(I) = 0;

\$INCLUDE M1_VEC.GEN
SOLVE M1_VEC USING MCP;

* Present the MCP version for the sake of
 * completeness:

PARAMETER ALPHA(F,I) Factor input benchmark value share
 BETA(I) Consumption value share;

ALPHA(F,I) = FDO(F,I) / SUM(FF, FDO(FF,I));
 BETA(I) = CO(I) / W0;

EQUATIONS

PROFIT(I) Zero profit condition
 CMKT(I) Commodity market clearance
 FMKT(F) Factor market clearance
 PRF_W Zero profit for aggregate consumption
 MKT_W Market clearance for aggregate consumption
 I_CONS Income = factor earnings plus taxes;

PROFIT(I) .. (1+T(I)) * PROD(F, PF(F)**ALPHA(F,I)) =E=
 PC(I);

PRF_W .. PROD(I, PC(I)**BETA(I)) =E= PW;

```

CMKT(I) ..      Z0(I) * Z(I) =E= C0(I) * W *
                 PROD(J, PC(J)**BETA(J))/PC(I);

MKT_W ..      W0 * W =E= CONS / PW;

FMKT(F) ..     E(F) =E= SUM(I, FDO(F,I) * Z(I)
                 * PROD(FF, PF(FF)**ALPHA(FF,I))) / PF(F);

I_CONS ..     CONS =E= SUM(F, PF(F) * E(F)) +
                 SUM(I, T(I) * Z0(I) *
                 Z(I) * PROD(F, PF(F)**ALPHA(F,I)) );

MODEL ALGEBRAIC /PROFIT.Z, PRF_W.W, CMKT.PC, FMKT.PF,
MKT_W.PW, I_CONS.CONS/;

E("L") = 100;
T(I) = 0;

SOLVE ALGEBRAIC USING MCP;

E("L") = 100;
T("X") = 0.5;

```

SOLVE ALGEBRAIC USING MCP;

$E("L") = 2 * E("L");$

$T(I) = 0;$

SOLVE ALGEBRAIC USING MCP;

Exercises:

Do the same exercises as for the mcp version.