Chapter 1

General-Equilibrium Modeling using GAMS and MPS/GE: Some Basics[1]

This chapter begins a tutorial on applied general-equilibrium modeling using the specific software of GAMS and MPS/GE. Before plunging into things, I want to let you know what I will not cover and what you need to know before continuing.

First, I will not provide a detailed tutorial on GAMS notation and syntax. For these you can consult the GAMS web site: www.gams.com. Click on documentation, and then on GAMS - A User's Guide. This will give you a lot of the basics you need to know. Unfortunately, this guide is badly out of date and focuses entirely on optimization problems, whereas applied GE modeling generally involves solving square systems of equations and inequalities. But the user's guide will give you the syntax and notation as I indicated. Try going through chapters 2 and 3 before continuing with this tutorial. Hopefully, sometime soon we will try to rewrite the user's guide.

Second, you will need to consult the GAMS web site for a copy of the software. I believe that a demonstration copy is currently provided for free, but this can change of course. Older versions of the software require the use of an external editor. You best bet for starting is to just use the DOS editor used under the DOS prompt. You could of could use a word processor and save your program each time as an ascii text file, but this is clumsy, awkward, and time consuming.

Again, consult the user's guide for how to actually run a program and find and view the output. This set of notes is limited, I am afraid, to actually formulating applied problems into code and it is beyond the scope of my time and patience to describe and teach that which logically comes first. The latter needs improvement over what is currently on the web site, but I will have to leave that to others.

James R. Markusen
Boulder, February 2002

[1] This material is the copyrighted intellectual property of James Markusen. Please feel free to use it for individual self-study. But for any use involving fees or changes, however indirect (e.g., course fees), written permission must be obtained from the author.

1. Introduction to applied general-equilibrium modeling

This is a set of notes to introduce you to applied general-equilibrium modeling and software used to analyze applied GE problems. First some general comments about general-equilibrium modeling.

There are many models which are portrayed by their authors' as "general equilibrium". The term assumes different meanings in different fields, so it is probably a good idea to begin with a definition of what this means. When we say general equilibrium, we are normally thinking of models which have the following characteristics.

(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.

General-equilibrium theory in economics is often quite abstract. A usual introductory formulation consists of a set of markets for goods and factors of production. Agents, which are typically labeled consumers and firms, optimize subject to the constraints they face such as technologies and budget constraints. These optimizations then lead to excess demand functions for each good and factor. Equilibrium is then obtaining by finding a set of prices such that all excess demands are zero. General-equilibrium theory is generally focused on abstract issues such as proving that a set of equilibrium prices and hence equilibrium itself exists.

While this is an important task, the theorists rarely bother with analyzing what those equilibrium prices are or how they are related to underlying features of the economy such as preferences, technologies and so forth. And it follows that the abstract theory is of little or no use in answering questions about how changes in policies such as taxes or tariffs influence the equilibrium. Some progress can be made in special theoretical models such as the Heckscher-Ohlin model of international trade. In this model, the direction of trade can be related to underlying technologies and factor endowments, and the effects of policies such as tariffs on welfare and the distribution of income among factor owners (the Stolper-Samuelson theorem) can be derived.

Yet even in the analytical Heckscher-Ohlin model, two problems persist. First, the results are "qualitative"; e.g., they give us the signs of comparative-statics derivatives or tell us that some elasticity is greater than one. But analytical results cannot be much more precise than that. Second, almost all results are only unambiguous in a version of the model in which there are two goods, two factors, two countries and consumers everywhere have identical and homogeneous preferences over goods. Three goods, three factors, three countries or two distinct consumer groups create problems that cause the elegant results of Heckscher-Ohlin to collapse.

Applied general-equilibrium modeling is the way around these difficulties, such that the concept of general-equilibrium actually becomes useful for analyzing real economies and real policies. Any number of good, factors, household types, and countries may be included. While the field started out with the assumptions of constant returns to scale and perfect competition in all production activities, we have learned how to incorporate scale economies and imperfect competition. We have learned how to include complex tax structures, public goods, externalities, and “rationing constraints” such as price controls or quotas that prevent markets from clearing.

Naturally, there is a price to be paid from the theorist’s point of view. We have to assume specific functional forms for preferences, production functions, and so forth. Many parameters of these functions can be drawn from published data or estimated with econometrics, but others remain educated guess work. This exercise draws criticism from both theorists and econometricians alike, but in the end applied GE modeling delivers answers to policy questions, however imprecise those answers might be.

What exactly is an applied GE model? It begins by the following theory: an economy and the equilibrium conditions for that economy are translated into a mathematical formulation. General equilibrium is then represented as the solution to a well-defined mathematical problem. More specifically, there are two general ways of formulating this mathematical problem. The first is to model the economy as an optimization or programming problem. This tend to be the first way a student of economics would approach the problem, since optimization and optimization techniques are a fundamental part of the theory of the consumer and the theory of the firm. Thus general equilibrium could be thought of as the solution to a big linear or non-linear programming problem, in which some objective function is maximized or minimized subject to a set of constraints.

It turns out that representing equilibrium as the solution to an optimization problem becomes awkward when there are several households or countries. What is it that should be optimized? There is no clear objective function to optimize. The second way of approaching the problem follows from formal theory. Individual optimizing behavior and decisions of consumers and firms are embedded in functions describing the agents’ choices in response to the values of variables facing them. So, for example, we use individual optimization to derive demand and supply functions that describe how consumers and firms will react to prices, taxes, and other variables.

Once we have done this, finding general-equilibrium is reduced to finding the solution to a square system of n equations in n unknowns. Individual behavior and optimization are embedded in those n equations. That is the approach we take here. An applied general-equilibrium model is a square system of n equations in n unknowns that is formulated in a fashion that permits a numerical solution by computational techniques, finding the actual values of the endogenous variables for given values of exogenous parameters. Endogenous variables include outputs, prices, trade volumes and so forth. Exogenous parameters include preferences, technologies, factor endowments and so forth.

As we will see shortly, the software we use permits a very important generalization of this notion of solving a square system of equations. For many economic problems, equilibrium may involve some goods not being produced or some possible trade links not being actively used. We really would like to formulate the general-equilibrium model as a system of weak inequalities, with each inequality associated with a particular non-negative variable such as a price or quantity. If a particular weak inequality holds as an equation, then the associated variable is strictly positive. If it holds as a strict inequality, then the associated variables is zero.

An example of this for a competitive model is the requirement that, in equilibrium, the profits from a given production activity must be non-positive. The associated variable to this inequality is the output level of that activity. In equilibrium, the weak inequality may hold as a strict equality, in which case there is positive output. If it holds as a strict inequality, (potential) profits from that production activity are negative, and no output is produced.

Thus we will formulate a general equilibrium model as a square system of weak inequalities, each with an associated non-negative variable. This is referred to as a complementarity problem in mathematics, and the associated variables are referred to as complementary variables.

Software other than that used here (GAMS and MPS/GE) generally do not allow the user to solve complementarity problems, greatly limiting model formulation and the range of comparative statics questions analyzed by the modeler.

2. Steps in Applied General-Equilibrium Modeling

Here are the “normal” 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.

Steps (3) and (4) are not strictly speaking necessary. The software can be used for pure simulation analysis, in which there initially is no data.

However, in learning the software, it is very valuable to start by writing down a micro-consistent data set and then transform that into code such that the solution to the model reproduces the initial data.

Let’s now turn to a concrete example of a simple general-equilibrium model.

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) $$X = X(L_x, K_x)$$

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

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

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

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

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

How do we find equilibrium, which in this case is a set of prices, and factor allocations to the two sectors? Many economists’ first reaction would be to formulate equilibrium as the solution to an optimization problem. 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. Suppose, for example, there are two different consumer types with different preferences and different factor endowments. What do we maximize? You could maximize the utility of one consumer subject to an arbitrary fixed level of the utility of the other consumer, exploiting the first theorem of welfare economics. But unless you are extraordinarily lucky, the solution will give each consumer an implied expenditure level which is not equal to the consumer’s income. Thus there is an inconsistency in the proposed solution.

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.

Similarly, we can solve for a cost function for the consumer, commonly called an expenditure function, which gives the minimum cost at given commodity prices of buying one unit of utility or welfare (W). These functions are given as follows.

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

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. This result, which in turn relies on the envelope theorem, states that the partial derivatives of these functions are quantities. In particular, we have:

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

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

$$\frac{\partial e}{\partial p_x} = cx_{px} = \text{Consumer's demand for X per unit of utility (similarly for Y)}$$

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

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. These three types are:

- 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. A good way to intuit the correct association is to think about the economics of what must be true if a particular weak inequality holds as a strict inequality.

If a zero profit conditions holds as a strict inequality in equilibrium, profits for that activity are negative, so that good will not be produced (strictly speaking that *activity* will not used; one good could be produced with several alternative activities). Thus 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 equation 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:

	Inequality		Complementary 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	$L^* \geq cx_wX + cy_wY$	p_l
(8)	Supply $\geq$Demand for K	$K^* \geq cx_rX + cy_rY$	p_k
(9)	Income balance	$I = p_lL^* + p_kK^*$	I

Now let’s turn to the issue of starting with a micro-consistent data set, a set of numbers which are in fact consistent with the above problem formulation. 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.

In what follows, 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. But the term SAM has been used in a rather different sense, so we will just refer to our rectangular matrix as a “MCM” – micro-consistency matrix.

In the present example, there are two types of columns in the rectangular MCM, corresponding to production 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 in the present example. Complementary variables are prices, so we have listed the price variables on the left to designate rows.

In the MCM, there are both positive and negative entries. 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. Here is our matrix of initial values.

	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 MCM 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).

With this interpretation, 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. X is capital intensive with capital having a share of 0.75 and labor a share of 0.25. Y is labor intensive with the opposite ordering of shares.

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) Note that 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.

Now we are ready to go. 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

This is the exact same model as M1_MPS.GMS but uses the MCP format.

                  Production Sectors          Consumers
   Markets   |    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 L
 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

MKT_X..         100 * X =G= 100 * W * PX**0.5 * PY**0.5 / PX;

MKT_Y..         100 * Y =G= 100 * W * PX**0.5 * PY**0.5 / PY;

MKT_W..         200 * W =E= CONS / PW;

MKT_L..         100 * LENDOW =G= 25 * X * PL**0.25 * PK**0.75 / PL +
                                 75 * Y * PL**0.75 * PK**0.25 / PL;

MKT_K..         100 =G= 75 * X * PL**0.25 * PK**0.75 / PK +
                        25 * Y * PL**0.75 * PK**0.25 / PK;

*       Income balance equations

I_CONS..        CONS =E= 100*LENDOW*PL + TX*100*X*PL**0.25*PK**0.75;

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.CONS /;


*       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. We will not go into a long explanation here, but rather just focus on the solution values. 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   -16.412
```

```
---- VAR PL                .         0.903     +INF       .
---- VAR PK                .         0.739     +INF       .
---- VAR CONS              .       213.359     +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       .
```

Here we see a relative shift to Y, the good using labor intensively, although X production also rises. The price of X rises relative to Y. The real price of capital, now the scarce factor, rises, and the real price of labor falls. Although labor has a 50% income share initially, doubling labor supply increases welfare by less than 50% (it increases W by 41.4%) due to diminishing returns from the presence of the fixed factor capital.

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,

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

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

3. 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. This higher-level language permits extremely efficient shortcuts for modelers, allowing us to concentrate on economics rather than coding.

There are several great features of MPS/GE. First, the program has routines for calibrating and writing all constant-returns CES and CET functions, up to three levels of nesting. All the modeler has to do is to specify the nesting structure, substitution elasticities in each nest and a representative point on the function, consisting of output quantities, input quantities and prices. This point and price vector uniquely determine the function, and MPS/GE then generates the cost function (or expenditure function). This is not that time and error saving in the simple simulation models of this book, but it is a wonderful feature for larger models.

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 as described in the previous appendix 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. I once refereed a paper in which the author claimed to have some weird numerical result. It turned out that the modeler had a tax, but forgot to put the tax revenue in the representative agent’s income balance equation. That cannot happen in MPS/GE. In short, MPS/GE automatically checks for and ensures many of the product-exhaustion and income-balance requirements discussed in the previous section.

In this appendix, I am going to give a short and superficial introduction to the MPS/GE subroutine of GAMS. I am going to use exactly the same problem as in the previous appendix, so that you can see the connection. First, 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

Declaration of 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 - An Introduction to the Basics

$ONTEXT

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


                  Production Sectors          Consumers
   Markets   |    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 (Hicksian 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 (discussed shortly).

```
$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, though 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.

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.

Exercises:

Do the same exercises as for the mcp version.

```
$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) = FD0(F,I) / SUM(FF, FD0(FF,I));
BETA(I) = C0(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, FD0(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.