1 A Quick Introduction to CGE Modeling

When a student begins to learn general equilibrium modeling with GAMS/MPSGE, it is easy to lose sight of underlying methodology because a number of steps are automated by the software. In order to teach students how to properly use these tools, I often find it helpful for the students to begin by implementing a few models “from scratch”. After having done this, it is easier for the novice modeller to understand how MPSGE works.

The purpose of this document is to provide an overview of some of the characteristics of CGE models and the steps involved in their construction and application. Following this summary, I will present some demand function results which will prove useful for writing down the equations which define an economic equilibrium model. Finally, I will conclude by working through the implementation of a couple of elementary general equilibrium models, both as systems of equations and using the MPSGE representation. The purpose of this document is to provide an introduction to the basic concepts rather than to introduce MPSGE modeling, per se.

1.1 Characteristics of a CGE Model

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 I say general equilibrium, I am normally thinking of models which have the following characteristics:
1. Multiple interacting agents.

2. Individual behaviour based on optimization. (Two types of decision problems are prominent: budget-constrained utility maximization by households and technology-constraint profit maximization by firms.)

3. Most agent interactions are mediated by markets and prices.

4. Typically disaggregate, with many agents and markets.

5. Limited data in comparison with the number of behavioral and technological parameters in the model. CGE data are often limited to a representative (benchmark) equilibrium observation and some independent estimates of elasticities of demand or supply. CGE models are typically constructed so as to “replicate” the transactions in the benchmark equilibrium.

6. Equilibrium allocations which typically cannot be characterized as the solution to a single (planner’s) optimization problem.

7. Formulation has an implicit or explicit focus on policy analysis. General equilibrium models can be used to perform positive economic analysis, attempting to explain aspects of market performance, etc., but even when models are intended for this purpose, it is customary for a CGE model to provide insight into the effects of changes in government instruments of some sort.

There are a number of canonical general equilibrium models, differing with respect to the degree of disaggregation in goods and markets, the policy issues which can be addressed, and the degree of difficulty in collecting data and analyzing equilibrium outcomes. Perhaps the most common of these is a static open-economy model of taxation and trade.

1.2 Steps in Building a CGE Model

There is, to my knowledge, no recipe book for CGE modelling, but if there were such a book it would probably include many of the following chapters:
1. **General design** Based on the policy issues of interest and considerations of the availability of data, the basic dimensions of the analysis should be defined: regions, households, firms, governments.

2. **Individual choice design** Specify a choice problem for each agent in the model (household, firm, government). These problems could be quite simple (e.g., the government collects revenue and purchases public goods and services; or it could be quite complex, as in a consumer or produce choice, allocating revenue across a range of alternative goods or inputs.

3. **Demand and supply functions** Solve for the behavioural equations for each of the agents in the model. This typically involves closed-form constrained optimization. A few examples of these derivations will be presented below.

4. **Develop a system of notation** This step in the model design process requires careful attention to the correspondences between markets and prices. Select indices for underlying sets and names for the prices of goods and factors. It is often convenient to employ more than one price for a given commodity, if that good is subject to taxation, or if there is uncertainty as to the direction of trade.

5. **Specify Parameters** This step confronts behavioural equations with the reference equilibrium values. In the calibration process, observed choices by agents in the reference case are combined with the behavioural equations and other exogenous parameters to infer parameters of the individual choice problems.

6. **Program the Model** Write out a system of equations which reads the input data, assigns parameter values and solves a system of equations representing the models’ equilibrium conditions. The solution of systems of nonlinear equations is a task which can be performed using a variety of software tools. The relative ease of one or another approach depends to a great extent on individual experience.

7. **Replicate the benchmark** If preference and technology parameters have been calibrated from a given equilibrium dataset, then if all exogenous parameters are assigned to their benchmark levels, the model should return the benchmark equilibrium.
8. **Test the model** A new model should be subject to some “shake-down tests” to verify that it works as intended. These tests can include calculations of simple policy reforms, such as tax reforms, but they should also include diagnostic calculations, such as marginal excess burden calculations which provide insight into the relative inefficiency of alternative tax instruments.

1.3 **Modeling Language Issues**

One approach to setting up your model would be do conventional scientific programming with C, Pascal or Fortran. This approach is the most difficult if you program your own solution algorithm. Considerable savings can be realized by using callable numerical libraries (such as Numerical Recipes, NAG or IMSL). A somewhat easier strategy would be to code up the equilibrium conditions in a spreadsheet such as (Excel or Quattro) which now provide internal routines for solving systems of nonlinear equations. Both of these methods are less than ideal because of the level of abstraction required to translate the economic equilibrium conditions into a machine-readable format.

It is far easier to work with a model which is presented to the computer in algebraic terms, based the mathematical representation of the equations as they are documented in the model write-up. This is possible if the model is programming in a higher-level language. These languages include a number of econometric packages (such as SAS or GAUSS) as well as conventional mathematical programming languages (such as MATLAB, MATHEMATIC, AMPL or GAMS).

MPSGE is based on GAMS and permits use of GAMS matrix language for data preparation. In addition, MPSGE provides a compact, non-algebraic representation of the model’s core equation.

Advantages: compact model representation, resulting in fewer opportunities for mistakes. Provides a structured framework for model representation: three classes of equations in an MPSGE model are: market clearance (one equation for each price in the model), income balance (one for each household and government agent), and zero excess profit (one equation for each representative firm which operating under constant returns to scale)\(^1\)

Disadvantages: non-transparent for the uniformed reader. Algebraic modeling, while tedious, has a distinct advantage of transparency.

\(^1\)It is conceptually convenient to consider decreasing returns to scale as a special case of constant returns, namely a situation in which there is a specific factor of production.
1.4 Model Application

There is no single way to apply a model to a policy issue. The following steps provide a very general set of guidelines:

1. **Translate policy into the model instruments.** This step involves some conceptualization – you need to think about how to represent the policy instruments within the context of your model. In some cases, this may require modification of your base model in order to add tax parameters or other policy instruments which better capture the relevant issues.

2. **Guess at the policy result.** It can be very helpful before running your model to sketch out a few diagrams illustrating how you expect the model to behave. This will require that you reason through the economic intuition underlying the model and begin to develop your understanding of the relevant aspects of the formulation and representation of the policies.

3. **Run the simulation and compare results.** This seems quite simple, but careful programming is needed.

4. **Compare the model results with your earlier guess.** Analyze the experimental outcome and explain why the results may differ from your initial assessment. This diagnostics may require some additional experiments with the model in order to develop a clear understanding of the critical issues.

5. **Evaluate the outcome and write up your key findings.** It is important to do your writing as soon as you have the results in hand. This provides a very important test of your understanding – if you cannot write a clear and simple explanation of what the model is telling you, then you probably need to do more work with the model.

6. **Develop sensitivity analyses.** Once you have the central assessment completed, you can then return to the model calculations and run a number of additional scenarios testing sensitivity of your findings to the various assumptions underlying the analysis.

7. **Write up the model.** Any CGE paper requires a clear explanation of how the model is formulated. It is important to do this writing as soon after you have completed the calculations as possible in order that all the details of the analysis remain at your fingertips.
2 Some Common Functional Forms

2.1 Consumer Utility and Uncompensated Demand Functions

All of these problems assume a budget constraint of the form \( \sum_i p_i c_i = M \). Budget shares at the demand point \( p_i c_i / M \) are denoted \( s_i \). Assume that utility parameters \( \alpha_i \) are shares, i.e. \( \alpha_i \geq 0 \forall i \) and \( \sum_i \alpha_i = 1 \). \( \bar{p}_i \) and \( \bar{c}_i \) denote a benchmark price and quantity, and \( \theta_i = \bar{p}_i \bar{c}_i / \bar{M} \) where \( \bar{M} \) is the value of expenditure at the benchmark point.

In each of these utility functions I present two forms of the demand function, one based on utility function parameters \( \alpha_i \), and another based on the reference price-quantity point, the “calibrated demand function”. The calibrated function is useful in model specification because it avoids the need for intermediate calculation of the share vector, \( \alpha \).

2.1.1 Leontief Preferences

Utility function: \( U = \min_i \left[ \frac{c_i}{\alpha_i} \right] \)

Indirect utility function: \( V = \frac{M}{(\sum_i \alpha_i p_i)} \)

Expenditure function: \( E = U \sum_i \alpha_i p_i \)

Conventional demand function: \( c_i = \alpha_i V \)

Calibration: \( \alpha_i = \bar{c}_i / \sum_j \bar{c}_j \)

Calibrated demand function: \( c_i = \bar{c}_i (\sum_i \theta_i \bar{p}_i / \sum_i \theta_i p_i) (\bar{M} / \bar{M}) \)

2.1.2 Cobb-Douglass Preferences

Utility function: \( U = \prod_i c_i^{\alpha_i} \)

Indirect utility function: \( V = M \prod_i \left( \frac{\alpha_i}{\bar{p}_i} \right)^{\alpha_i} \)

Expenditure function: \( E = U \left( \frac{\bar{p}_i}{\bar{p}_i} \right)^{\alpha_i} \)

Conventional demand function: \( c_i = \alpha_i M / p_i \)

Calibration: \( \alpha_i = \theta_i \)

Calibrated demand function: \( c_i = \bar{c}_i (\bar{p}_i / p_i) (\bar{M} / \bar{M}) \)
2.1.3 Constant Elasticity of Substitution (C.E.S.) Preferences

Utility function: \( U = \left( \sum \alpha_i^\sigma c_i^{(\sigma-1)/\sigma} \right)^{\sigma/(\sigma-1)} \)

Indirect utility function: \( V = M \left( \sum \alpha_i p_i^{1-\sigma} \right)^{1/(\sigma-1)} \)

Expenditure function: \( E = U \left( \sum \alpha_i p_i^{1-\sigma} \right)^{1/(1-\sigma)} \)

Price index: \( P_U = \left( \sum \alpha_i p_i^{1-\sigma} \right)^{1/(1-\sigma)} \)

Conventional demand function: \( c_i = \alpha_i M (\frac{P_i}{P_U})^{\sigma} \)

Calibration: \( \alpha_i = \tau_i \left( \sum \alpha_i p_i^{1-\sigma} \right) P_i^{\sigma} / M \)

Calibrated price index: \( \tilde{P}_U = \left( \sum \alpha_i \left( \frac{P_i}{\tilde{P}_U} \right)^{1-\sigma} \right)^{1/(1-\sigma)} \)

Calibrated demand function: \( c_i = \tau_i (M/M) \left( \tilde{P}_U p_i / p_i \right)^{\sigma} \)

2.1.4 Linear Expenditure System (L.E.S.):

Utility function: \( U = \prod_i (c_i - \beta_i)^{\alpha_i} \)

Indirect utility function: \( V = (M - \sum_j p_j \beta_j) \prod_i \left( \frac{\alpha_i}{\beta_i} \right)^{\alpha_i} \)

Expenditure function: \( E = \sum_j p_j \beta_j + U \left( \frac{\alpha_i}{\beta_i} \right)^{\alpha_i} \)

Conventional demand function: \( c_i = \beta_i + \alpha_i (M - \sum_j p_j \beta_j) / p_i \)

2.2 Technology, Cost and Compensated Demand Functions

All of these problems assume a cost-minimizing representation of producer behavior where the underlying technology exhibits constant returns to scale. Unit demand for the ith input is \( x_i \), and the producer price of the ith input is \( w_i \). I use non-negative production function parameters \( \alpha_i \) which do not necessarily sum to unity. Where necessary, I use a output scaling parameter \( \phi \).

In the calibrated form of the demand functions, let \( \tilde{p}_i \) and \( \tilde{x}_i \) denote a benchmark price and quantity, and \( \theta_i = \tilde{p}_i \tilde{x}_i / \sum_j \tilde{p}_j \tilde{x}_j \) represent the benchmark cost share of the ith input. Let \( \tilde{Q} \) denote benchmark output.

As in the previous section, here I present two forms of the demand function, one based on the native function parameters \( \alpha \), and another based on the reference price-quantity point, the
“calibrated demand function”. The calibrated function is useful in model specification because it avoids the need for intermediate calculation of the parameter vector, $\alpha$.

2.2.1 Leontief Technology

Production function: $Q = \min_i \left[ \frac{c_i}{\alpha_i} \right]$

Cost function: $C = Q \sum_i \alpha_i w_i$

Unit cost function: $c = \sum_i \alpha_i w_i$

Conventional demand function: $x_i = \alpha_i Q$

Calibrated demand function: $x_i = \pi_i (Q/Q)\tilde{c}$

2.2.2 Cobb Douglas Technology

Production function: $Q = \phi \prod_j x_j^{\alpha_j}$

Cost function: $C = (Q/\phi) \prod_j \left( \frac{w_j}{\alpha_j} \right)^{\alpha_j}$

Unit cost function: $c = (1/\phi) \prod_j \left( \frac{w_j}{\alpha_j} \right)^{\alpha_j}$

Conventional demand function: $x_i = (Q/\phi) (\alpha_j/\omega_j) \prod_j \left( \frac{w_j}{\alpha_j} \right)^{\alpha_j}$

Demand function under zero profit: $x_i = Q^{\alpha_j/\omega_j}$

Calibrated unit cost function: $\tilde{c} = \prod_j \left( \frac{w_j}{\alpha_j} \right)^{\alpha_j}$

Calibrated demand function: $x_i = \pi_i (Q/Q)(\tilde{c})/\omega_i$

2.2.3 Constant Elasticity of Substitution (C.E.S.) Technology

Production function: $Q = \phi \left( \sum_j \alpha_j \frac{1}{\sigma} x_j^{\sigma - 1/\sigma} \right)^{\sigma/(\sigma - 1)}$

Cost function: $C = (Q/\phi) \left( \sum_j \alpha_j w_j^{1-\sigma} \right)^{1/(1-\sigma)}$

Unit cost function: $c = (1/\phi) \left( \sum_j \alpha_j w_j^{1-\sigma} \right)^{1/(1-\sigma)}$

Conventional demand function: $x_i = \alpha_i w_i^{-\sigma} (Q/\phi) \left( \sum_j \alpha_j w_j^{1-\sigma} \right)^{\sigma/(1-\sigma)}$
Demand function under zero profit: 

\[ x_i = \alpha_i \phi^{\sigma - 1} (p/w_i)^\sigma Q \]

Calibrated unit cost function: 

\[ \tilde{c} = \left[ \sum_j \theta_j (w_j/w_j)^{1-\sigma} \right]^{1/(1-\sigma)} \]

Calibrated demand function: 

\[ x_i = \pi_i (Q/Q) (\tilde{c} \bar{w}_i/w_i)^\sigma \]