Modeling Demand

Short Course on CGE Modeling, United Nations ESCAP

John Gilbert

Professor
Department of Economics and Finance
Jon M. Huntsman School of Business
Utah State University
jgilbert@usu.edu

September 24-26, 2014
In CGE models, final demand is derived from the utility maximization problem of a representative household (or in some cases households).

This is a very familiar problem, but working through it carefully will provide us with a number of insights into how CGE-type models are constructed in practice.

In this session we will review the basic consumer problem, and work through an implementation the model in GAMS.

Finally, we will explore consumer behavior using counter-factual simulation.
Session Outline

1. The utility maximization problem
2. Building the model in GAMS
   - Setting up the model
   - Calibration
   - Simulation and testing
Consider a consumer that has preferences satisfying the axioms of consumer choice, and where their preferences can be summarized by the utility function \( U = U(c_1, c_2) \), where \( c_i \) is consumption of the \( i \)th good.

The usual properties apply to the utility function.

The consumer choice problem can be viewed as choosing \( c_1 \) and \( c_2 \) such that the consumer maximizes \( U = U(c_1, c_2) \) subject to the budget constraint \( Y = p_1 c_1 + p_2 c_2 \), where \( Y \) is money income.

We assume an internal solution for simplicity.
At an optimal solution, the consumer will spend all income, and the marginal utility per dollar spent on each good must equal the marginal utility of income.

Another way of interpreting the optimal choice is that the money value of the utility generated by the last unit of each good purchased must equal its price.

Solving explicitly for the optimal consumption bundles yields the Marshallian demand functions.
A common (and simple) way of describing the preferences is with a Cobb-Douglas function:

\[ U = \alpha c_1^\beta c_2^{1-\beta} \]

where:

- \( \alpha \) is an arbitrary scale factor (why is it arbitrary?)
- \( 0 < \beta < 1 \) is a share parameter, reflecting the consumption share of good 1.
Now let us consider exactly how the problem can be expressed in the GAMS language, using Cobb-Douglas utility.

Our first task is to create a set which will index the goods:

```
SET I Goods /1,2/;
```

The keyword is followed by an arbitrary name for the set, I, an optional description, then the elements of the set enclosed in forward slashes and separated by commas. The names used for set elements are also arbitrary. The command is completed with a semicolon.
Next, we need to define labels for all of the parameters and exogenous variables in the model. We are also going to define labels for the initial values of our endogenous variables:

```plaintext
PARAMETERS
ALPHA       Shift parameter in utility
BETA(I)     Share parameters in utility
Y           Income
P(I)        Prices
UO          Initial utility level
CO(I)       Initial consumption levels;
```

The keyword PARAMETERS is followed by a list of parameter names, indexed as appropriate, and their descriptions.
Our next task is to assign names for the variables:

VARIABLES
U Utility level
C(I) Consumption levels;

The keyword VARIABLES is followed by a list of variable names, indexed as appropriate, and their descriptions.
We enter names for equations in the model in the same way that we enter names for parameters and variables:

EQUATIONS
UTILITY Utility function
DEMAND(I) Demand functions;

The keyword is EQUATIONS, followed by a list of names, which are also indexed, with optional descriptions. We define the structure of the equations in terms of the variables and parameters:

UTILITY..U=E=ALPHA*PROD(I, C(I)**BETA(I));
DEMAND(I)..C(I)=E=BETA(I)*Y/P(I);

These are the GAMS equivalents of the optimality conditions.
With the labels created, we can assign some suitable values to the parameters:

\[ P(I) = 1; \]
\[ Y = 100; \]
\[ CO(I) = 50; \]
\[ UO = Y; \]
\[ BETA(I) = \frac{P(I) \times CO(I)}{Y}; \]
\[ ALPHA = \frac{UO}{PROD(I, CO(I)^{**BETA(I)})}; \]

The process is called **calibration** – which is essentially solving the model for the parameter values consistent with the desired solution.
Since the values of the variables are determined by the model, we cannot assign values to them as such. But, we can give GAMS starting values for the variables. We use the initial values we assigned/calculated:

\[
\begin{align*}
U.L &= U_0; \\
C.L(I) &= C_0(I);
\end{align*}
\]

We can also assign logical bounds on the values the variables can take:

\[
C.LO(I) = 0;
\]
The last stage is to tell GAMS which of our equations constitute the model, in this case all of them, and then run a test solution:

MODEL UMAX /ALL/;
SOLVE UMAX USING NLP MAXIMIZING U;

The first statement says that the model that we will call UMAX consists of all the equations (i.e., it is equivalent to MODEL UMAX /UTILITY, DEMAND/).
We have now built our first simple numerical simulation model. Once we are satisfied that the model is correctly replicating the solution, we can examine the effects of changes in the economic situation by altering the values of parameters and/or exogenous variables and executing another SOLVE command.

Some things to try:
- Changing prices and/or income in various combinations.
- Altering the model parameters.
- Experimenting with different utility function forms.
- Solving the corresponding expenditure minimization problem.
- Constructing alternative measures of welfare changes.
Most of this material is covered in intermediate microeconomics texts such as Varian (2009).

Dixit (1990) is a good introduction to optimization.

This GAMS example is developed more fully in Gilbert and Tower (2013), chapter 3.