AN INTRODUCTION TO GEMPACK

T. Hertel, R. Keeney and C. Ludena

September 2005

Why GEMPACK?

In AGEC 620 we will use the GEMPACK software suite to compute solutions to the set of economic models covered. Given the widespread use of GAMS elsewhere in the Ph.D. curriculum, the first question to ask is: Why GEMPACK instead of GAMS? In fact, when this course was initiated, we utilized GAMS. However, the outcome was sub-optimal because GAMS is a general purpose algebraic modeling package, oriented primarily towards optimization problems. It is extremely powerful and excellent for computing solutions to a wide range of problems, including those considered in this course. However, when it came time to *analyze* the resulting solutions, we found it to be somewhat limiting. GEMPACK, on the other hand, is a special purpose software package built to facilitate the solution and analysis of partial and general equilibrium problems. It is not an optimization package, and is therefore not as versatile as GAMS with respect to the scope of problems that can be properly addressed. But for purposes of quantitative analysis of equilibrium problems associated with markets and competition (the focus of this course), it is ideal, due both to its analytical decomposition facility and its software design.

Analytical Decomposition

The virtues of GEMPACK for our purposes derive from the fact that it has evolved in direct response to the needs of partial and general equilibrium modelers. For example, most of the pencil and paper analysis we will do this semester involves the analysis of *percent changes* in endogenous variables associated with the farm and food system (e.g., the supply of farm output) in response to *percent changes* in exogenous variables (e.g., wage rates). Key parameters are expressed in terms of elasticities. These are typically what we estimate econometrically, and it is therefore most natural to present the computational model in this form as well.

Let us illustrate this point with the case of a Cobb-Douglas profit function for a farm firm, which gives rise to the following output supply equation, where q is output supply, p is supply price and β_j is the elasticity of profits with respect to input price j:

$$q = \left(\frac{\pi}{p}\right)\left(1 - \sum_j \beta_j\right)$$

Now suppose that a major policy reform takes place so that this farm is confronted with a set of price changes, including change in both input and output prices. We can plug these into the profit function to see how optimum profits change and then use this equation to compute the new level of supply. However, this won't tell us how much each of the individual price changes contributed to the supply change. Does the wage change dominate, or is it the supply price change? To answer these questions, it is convenient to log-differentiate the supply equation, so that we obtain a percentage change representation of the supply response to individual price changes:

$$q = \left(\frac{\pi}{p}\right)\left(1 - \sum_j \beta_j\right)$$

$$\ln q = \ln \pi - \ln p + \ln\left(1 - \sum_j \beta_j\right)$$

$$\tilde{q} = \tilde{\pi} - \tilde{p}$$

$$\tilde{q} = \left(-\sum_j \beta_j\right)\tilde{p} + \sum_j \beta_j \tilde{p}_j + \sum_k \gamma_k \tilde{z}_k$$

By using the percentage change representation as the basis for the model implementation, we can readily identify the contributions of each individual price change to the total change in supply. In particular, by right-clicking on this equation in the AnalyseGE software suite, GEMPACK will reveal the values of each of the right-hand side terms for any given model solution. Thus, you can immediately see the relative impacts on the endogenous variables of an exogenous shock to wages or an increase in the price of fertilizer. This greatly facilitates the economic analysis!!

Of course there is a problem in solving the model via the linearized equation above; this is only a *local approximation* to the true, underlying non-linear model. This point is illustrated in a class example whereby we consider a one-input, Cobb-Douglas production function. Initially 100 units of labor are used to produce 100 units of agricultural output, with an output elasticity of labor equal to 0.4359. When the labor output is doubled, to 200 units, the linearized form of the production function predicts a new output level of 143.59 (i.e. an increase of 43.59%). However, the true increase is just 135.275, due to the curvature in the production function. Clearly for large shocks, the results can be misleading by simply solving the non-linear model using the linearized equations alone.

To overcome this problem, GEMPACK follows a strategy commonly employed in solving non-linear problems where an initial equilibrium is known. As is the case in “equations of motion” for a system, GEMPACK adds equations that update the variables in the system as they change. In this way, the software is able to track the curvature of the concave production function as the variable input (i.e. labor) increases. To do so, it breaks the labor shock into many small steps, updating the prices and quantities (and potentially the elasticities) after each step. GEMPACK makes available the "Euler" and "Gragg" solution methods for doing this. It is usual to solve the model three times, each time with a finer grid, using the change in solution values to *extrapolate* to the true solution. This approach also gives the user an extrapolation accuracy report which tells how many digits of accuracy can be anticipated for any given variable. If this is insufficient, the user can break the problem up into more “sub-intervals”. In short, GEMPACK uses the linearized representation of the model, along with updated equations, to obtain a *non-linear* solution.

Returning to the linearized supply equation above, we note that, once the model is solved, the true (non-linear) change in output supply will not equal the simple sum of the right hand side terms. This is because the solution is a local approximation to the non-linear supply equation. Fortunately, in most policy analysis involving small shocks, the linearization error is relatively small (a doubling of prices or quantities would be an exception), so the decomposition of effects offered by this equation are useful for interpretation and analysis. (Of course the final results for the endogenous variables are always those obtained from the non-linear simulation.) In closing, it should also be noted that there are more sophisticated decompositions – for example with respect to economy-wide welfare – that are also facilitated by the linearization approach to non-linear modeling.

Software Design

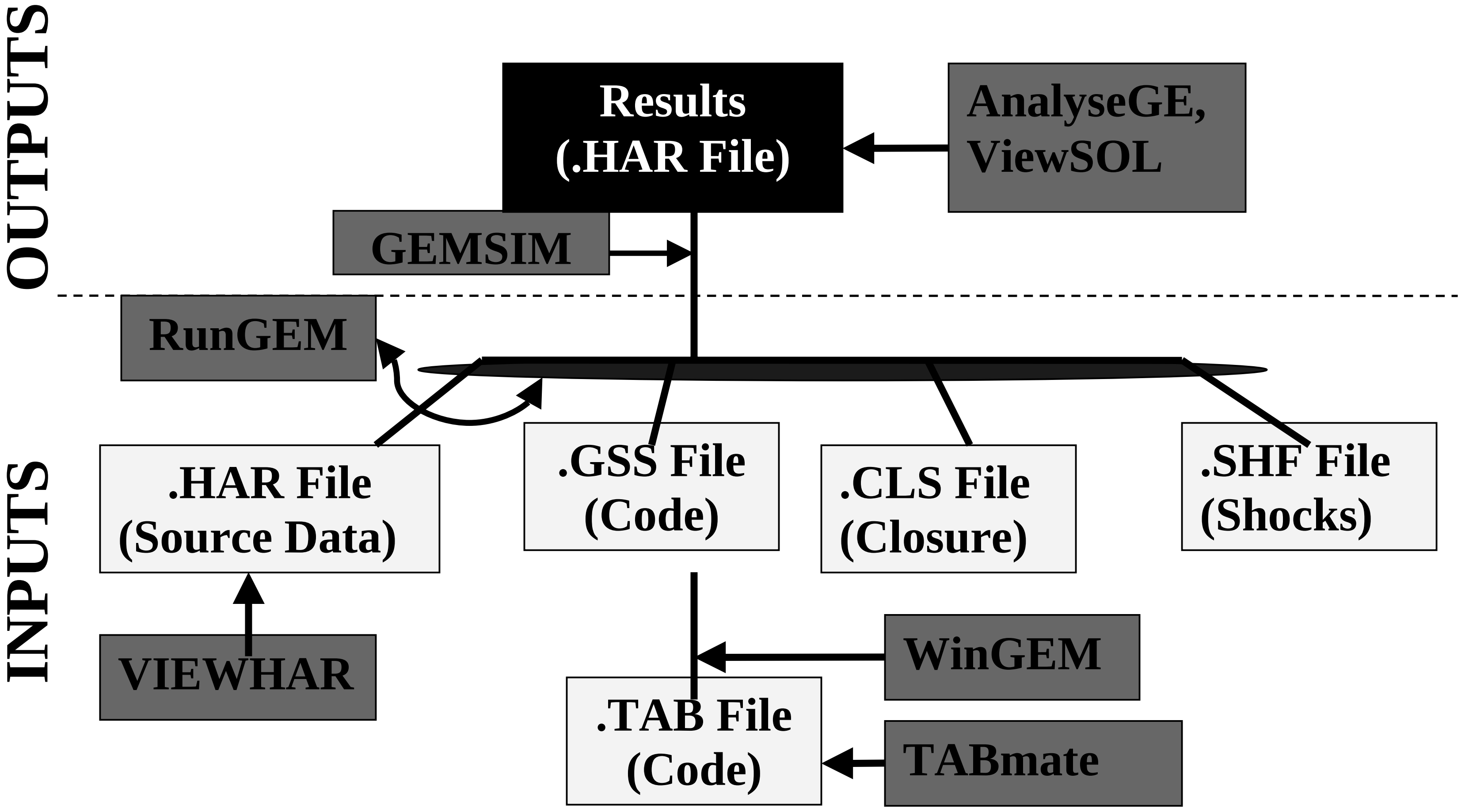
The other big advantage of GEMPACK is the structure of the software itself and its system of file management, as well as the user interface. This overall structure is summarized in Figure 1. While it may seem cumbersome at first, the different file systems and step-by-step procedure used in constructing a policy simulation is quite efficient and very indicative of the way in which high quality analysis is conducted. The first step typically involves data collection. The underlying data are typically assumed to be indicative of an initial equilibrium. In a separate step, behavioral responses of the economic agents are derived based on economic theory and restricted in a way that is analytically or computationally tractable through the choice of exogenous parameters, model closure, and functional forms. Other steps to be taken are dictated by the research questions and generally involve inducing a change in one or more of the exogenous variables in the system describing the equilibrium (i.e., a shock to the system).

Figure 1 provides a stylized overview of the GEMPACK suite as it is used in AGEC 620. At the lowest level we have a *.TAB file (all files are depicted in yellow) which contain the behavioral representation of the economic unit we assume is in equilibrium. As with GAMS, this is written in an algebraic language readable by the software. The program TABmate (all programs in blue) is used to edit and debug this file, thereby ensuring computational, although not necessarily economic logic! Directly above this we have the WinGEM program that processes our *.TAB file into a *.GSS file. The latter is in machine language that is readable by the computer. At the next level we have four files which represent the key inputs into any computational analysis. The *.GSS file reflects the theory behind the economic model. The *.HAR file contains the initial equilibrium data (which can be edited and viewed by the program, ViewHAR – where HAR stands for “header array file” and it is a tidy way to store large data sets). The *.CLS file is the “closure” file for the model. Since we are likely to have more variables than equations, we must declare some variables exogenous. The partitioning of variables between the set of endogenous and exogenous variables is termed the “closure” of the model. Finally, we have the *.SHF file listing the exogenous variables we want to shock and by how much they should be perturbed. In large scale quantitative analyses, it is critical to read these shocks from a file to ensure replication and the accuracy of the analysis.

The blue oval encompassing these four key files and connected to the RunGEM interface indicates that all four of these files are managed by RunGEM. The .GSS file is the model structure that RunGEM will use GEMSIM to process, the .HAR files are where RunGEM will load data for initializing the solution, and the .CLS and .SHF files are loaded into RunGEM to define the experiment to be run. RunGEM is also where the modeler specifies the solution method and initiates the model solution (by clicking the solve button). In the background, RunGEM processes the four inputs it has been given and prepares a set of instructions for GEMSIM which actually solves the system of linear equations and passes the output back to RunGEM for display. At this point, we can view results in the .HAR format either in the RunGEM interface directly or through the program ViewSOL (for multiple solutions) and AnalyseGE for interactive solutions with the equations of the model code.

In conducting repeated simulations, it is easy to forget which assumptions you have used in any given simulation as well as where you have saved the files. RunGEM gives you a structure for managing these aspects of your analysis, forcing you to revisit all the key steps before resolving the model.

Computational Structure using GEMPACK



OUTPUTS

Results
(.HAR File)

AnalyseGE,
ViewSOL

GEMSIM

RunGEM

INPUTS

.HAR File
(Source Data)

VIEWHAR

.GSS File
(Code)

.CLS File
(Closure)

.SHF File
(Shocks)

WinGEM

.TAB File
(Code)

TABmate