Preface

One of the best ways to learn computational economics is to do computational economics. One of the best ways to do computational economics is to begin with existing models and modify them as you experiment with them. This is the approach used in this book.

In each chapter an economic model is presented. First the economics and mathematics of the model is discussed and then the computational form of the model is analyzed. This process enables one to learn the economics and the mathematics of the problem area as well as the computational methods that are used in that area. For example, in the economic growth area we make use of a Ramsey type model. The economics of growth theory are first discussed along with the equations that model this process. Then the software representation of the model is presented so that the reader can see how the model can be solved on a computer. The student can then modify the model in order to analyze its sensitivity to various parameters and functional specifications. In the process of experimenting with the model one can gain an improved understanding of both the software and of the economic modeling.

This book grew out of undergraduate and graduate level courses on computational economics taught by us at the University of Texas, ITESM (Mexico), ISEG (Argentina) and the University of Amsterdam. Also, a number of teaching assistants and undergraduate students participated in the development of chapters, notably Daniel Gaynor and Genevieve Solomon.

This book is intended to provide an intensive hands-on training to advanced undergraduates and also to master's students and professional economists. In addition Ph.D. students could profit from the book as a broad introduction to computational modeling techniques and software systems before they move on to more advanced and specialized work in the field of Computational Economics.[1]

The organization of the chapters in the book reflects primarily the outline of the courses at the University of Texas. The aim is to let the students find an area of computational economics that interest them and to pursue that area. Since some of the students are interested in microeconomics, others in macroeconomics and others in

---

[1] Thus we hope that a two course sequence will evolve for graduate level computational economics. The first course might focus more on modeling issues and use a number of chapters from this book and the second focus more on algorithms.

finance an effort is made to give a quick and broad exposure to models across a range of fields early in the semester. Then the range is covered again later in the semester in greater depth. The book is structured to follow this pattern. In Part I there is a “once over lightly” treatment of computational economics examples from a number of fields. This is then repeated in greater depth and complexity in Part II. Part III covers an advanced area that is of special interest to the authors, namely the solution of macroeconomic models with stochastic control methods.

We would like to thank Provost Sheldon Ekland-Olson and Dean Brian Roberts of the University of Texas for funding which was used to support preparation of some of the materials in this book. Also, thanks are due to a number of undergraduate and graduate students who took the computational economics courses at the University of Texas and contributed ideas and models which added to the quality of several of the chapters and who helped to create and maintain the web sites, viz. Pichit Akrathit, Joe Breedlove, Michael Evanchik, Miwa Hattori, Carter Hemphill, Kyle Hood, Seung-Rae Kim, Kevin Kline, Paul Maksymonko and Huber Salas.

Introduction

One can think of learning computational economics by following one of three different routes - via computational methods, via mathematical methods or via economic areas. The computational methods route would focus on the use of a particular computer software system like MATLAB or Mathematica and teach the students the capabilities of those languages with examples from economics. The mathematical route would focus on algorithms to solve various classes of mathematical models such as linear or nonlinear programming models, differential or difference equations, dynamic programming models and provide examples of the use of each kind of model in economics. The economic areas approach would focus on microeconomics, macroeconomics, finance, game theory, environmental economics etc. and teach the students how to formulate and solve economic models in each of these areas. For this book we have chosen the last of these three approaches.

Thus this is a book about *computational economics*, but also about *economic modeling*. As a student approaches a new area of interest we want to help him or her first think through the economics of the subject. Then we develop this economics into a mathematical model. Finally we specify the mathematical model as a computational model in a particular software system. We believe that this process can be greatly facilitated by encouraging the students to follow Professor Paul Samuelson's advice and "stand on the shoulders" of those who have gone before. This is done by beginning from subject areas and problems that other economists have studied and learning how the economics was converted to mathematics and then to computational models in those areas.

Therefore this book is organized around economic topics rather than around mathematical or computational topics. However, we did not put all the microeconomics in the first section, then the macroeconomics etc. Rather the book is divided into two rounds of relatively simple models and then more complex models as was discussed above in the Preface.

Software Systems

Students who begin studying computational economics frequently ask the question, "What programming language should I learn?" [2] The answer given in this book is to first become *acquainted* with a number of high-level languages such as GAMS, Mathematica, MATLAB and Duali as well as the Solver in Excel and the Access database software. Moreover, it is useful to become acquainted with each of these software systems in the midst of solving the kind of economic models that are naturally developed in each of these systems. Then later one can dig deeper into one or more of the software systems and gain some level of mastery of it while writing a short mid-term paper, a term paper or doing research. At a still later stage, students who find that they have a continuing interest in computational economics would be well advised to progress to lower level languages such as Visual Basic, Fortran, C, C++, C# or Java.

There are different types of software paradigms, each of them more or less suitable to represent specific types of models. In this book, we present a selected set of high-level software systems, each corresponding to a specific paradigm.

We start the book with relatively simple models represented in Excel ("spreadsheet paradigm") and Access ("relational database paradigm") as a way of beginning with software paradigms that are either well known and/or very accessible to almost everybody, since both of these software systems are available on most PC's. Excel is useful to solve small models that do not involve simultaneous systems of equations; however, is not well designed for vector-matrix operations. For this type of operations we will use MATLAB later in the book. However, Excel has a nonlinear optimization solver which can handle constrained optimization problems and is very handy to set up and solve interesting models such as a Ramsey type model of economic growth and a small neural net. Access is a software system well suited to develop relatively simple relational databases and its use is illustrated with a prototype U.S. database.

The "set driven" paradigm is introduced with GAMS. This software system, particularly well suited to deal with medium and large size models involving from tens to hundreds of variables and equations, allows us to specify problems in an organized and compact way, defining sets to used as indexes, and specifying scalars, parameters, variables and equations in a parsimonious way. We solve with GAMS models of

---

[2] For a discussion of some of the software systems used in economics see Amman and Kendrick (1999b).

transportation, general equilibrium, financial planning, macroeconomics and global warming.

The “vector matrix” paradigm is introduced with MATLAB. This software system is useful to deal with models or problems involving intensive use of vector and matrix operations, cell arrays and data structures, and also to deal with problems of recursive structure requiring intensive use of “loops”. We use MATLAB to solve problems of portfolio optimization, genetic algorithms, agent-based models and dynamic programming.

The “symbolic math” paradigm is introduced with Mathematica. This software system is particularly powerful to solve symbolic algebra and calculus problems, and we use it represent game theoretic problems.

Finally, in a Special Topics Section in Part III of the book, and by means of macroeconomic applications we introduce Duali, a “dialog box driven” software designed to solve stochastic control and dynamic policy analysis problems. The basic code of this software is written in C, and contains a variety of simple and complex quadratic linear dynamic programming algorithms.

Most economics departments already have many software systems available on their computers and hopefully will also have the ability to acquire most of the rest of those used in this book. We have provided in our Web site at

http://www.eco.utexas.edu/compeco

the input files for the economic models that are used in this book. Also, this web site contains pointers to supporting books and user guides and instructions on how to use the various software systems on local computers. These instructions are a most important part of any course on computational economics and will need to be modified for each location. Also, in an effort to keep student cost down, we have endeavored to keep most of the models used in this book small enough that they can be solved with the student versions of the software systems.

With the exception of Duali, all of the software systems we use are commercial products. In contrast, the Duali software is academic software which is under development by two of us (Kendrick and Amman) and has no support staff or help desk. It is designed to greatly reduce the learning curve for developing dynamic deterministic and stochastic optimization models and is a most useful starting point into economic research in these areas. However, it is early in its stages of development and must be used with caution.

Numerical Methods

In this book we present not only a variety of models and software paradigms, but also introductions to diverse numerical methods needed to solve them. As with the software systems, we think that is useful to become acquainted with each of those numerical methods in the midst of solving the kind of economic models that are naturally involved with each of these methods.

A number of the models presented in the book are solved with linear programming methods or nonlinear optimization methods based on gradient and/or Newton methods. Thus we provide an introduction to these methods in appendices at the end of the book. Other methods are introduced directly in particular chapters. Neural nets are applied to a stock price prediction problem, Monte Carlo methods and genetic algorithms are applied to a portfolio selection problem. Quadratic linear dynamic programming is illustrated with a simple macroeconomic policy analysis application. Finally, the Fair and Taylor iterative method to solve rational expectations models, together with the Amman and Kendrick method to solve optimal control models with forward looking variables is applied to a prototype macro model developed by Taylor.

Teaching Methods

A description of the teaching methods used in the computational economics courses at the University of Texas will help the reader to understand the way in which the materials in this book have been developed. One aspect of these courses is that they have a weekly cycle. As was described above, the first class each week is on the economic theory and mathematical model of the subject for the week. The second class is on the computational methods used to solve the model. The third class of the week is not in a lecture room but rather in a computer laboratory where the students are ask to solve the base model and then to modify (and solve) the model several times in order to study its structure and operation. One week after the computer laboratory class the students are asked to turn in a short paper a few pages in length that describes their own experiments with the model during the week and the results obtained. The weekly teaching cycle is reflected in this book with some suggested experiments listed at the end of each chapter. However, the students are encouraged to strike out on their own – a process which enhances both enjoyment and learning.

Since the emphasis in these computational economics courses is on creativity, there is both a mid-term paper and a final paper. The students are asked in the mid-term paper to modify one of the models from the course or to select an existing model from the GAMS library or another similar source and then to make minor improvements in the model. In the final paper they are asked to carry this process forward and make major modifications to an existing model or to create a model of their own.

Several alternative approaches to the one used in this book are available for the study of computational economics. For an approach using the GAMS software exclusively and focusing on linear and nonlinear programming methods see Thompson and Thore (1992). For approaches using numerical methods see Judd (1998) who uses several computer languages or Miranda and Fackler (2002) who use MATLAB. Varian (1993a) and (1996) presents a variety of models in Mathematica. For a website that supports a course on applied macroeconomics using computational methods taught by Prof. Harris Dellas at the University of Bern that is somewhat similar to the approach taken in this book see

http://www.vwi.unibe.ch/amakro/Lectures/computer/

For a book that focuses on numerical methods in macroeconomics with some applications in MATLAB see Marimon and Scott (1999). For a book with a collection of articles that consider a variety of numerical methods to solve macroeconomic models see Hughes Hallett and McAdam (1999). For a handbook with a collection of articles about computational economics see Amman, Kendrick and Rust (1996). Also, you are encouraged to browse the Internet site of the Society for Computational Economics at

http://comp-econ.org

where you will find information about meetings, journals and book series.