
COMPUTATIONAL ECONOMICS

Revised Edition

Introduction

David A. Kendrick
Ruben P. Mercado
Hans M. Amman

FOR

Alejandro

Ann

Colin

Eveline

Liselotte

DRAFT

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 that you modify as you experiment with them-and this is the approach we take in this book.

Each chapter presents an economic model. First we discuss the economics and mathematics of the model and then analyze its computational form. 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 studying economic growth 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 the economic modeling.

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

The book is intended for use by advanced undergraduates and professional economists and even, as a first exposure to computational economics, for graduate students. We expect the coming years to see the development of undergraduate courses with a focus on economic modeling along the lines we have outlined in these pages. Moreover, we envisage the development of a two-course sequence in computational economics in graduate programs. The introductory course would have a broad economic-modeling focus with an approach similar to that used in some of the chapters in this book. The second course would focus on algorithms and numerical methods.

Part of our motivation for writing this book is spelled out in a couple of paragraphs that are taken from a paper the three of us wrote entitled *Computational Economics: Help for the Underestimated Undergraduate*. These comments-although written for that paper-apply equally well to the present work.

The ubiquitous personal computer has filtered deeply through the lives of college undergraduates; however undergraduate education in economics has so far failed to take full advantage of this sweeping change. We are underestimating the learning ability and insufficiently challenging a whole generation of undergraduate students in economics. Our thesis is that computational economics offers a way to improve this situation and to bring new life into the teaching of economics in colleges and universities.

With its early focus on algorithms, computational economics seemed well-suited for a relatively small group of graduate students and unlikely to have much impact on undergraduates. However, that is changing as we are discovering that computational economics provides an opportunity for some students to move away from too much use of the lecture-exam paradigm and more use of a laboratory-paper paradigm in teaching undergraduate economics. This opens the door for more creative activity on the part of the students by giving them models developed by previous generations and challenging them to modify those models. The modifications can be altering the models to make them applicable to the student's interest or finding weaknesses in the model that can be strengthened by changes in the structure of the model. In the process the students become much more involved in their own education.

The organization of the chapters in the book reflects primarily the outlines of the courses at the University of Texas, which are designed to allow students to find an area of computational economics of particular interest and to pursue that area. Since some of them are interested in microeconomics, others in macroeconomics, and others in 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 Alan Manne, Manfred Gilli, and other reviewers for comments on earlier drafts of this book that helped us to improve it substantially. We also want 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 the book. In addition, we would like to thank Peter Dougherty of the Princeton University Press for his encouragement

of the development of this book over a period of many years. 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 that added to the quality of several of the chapters and who helped to create and maintain the web sites, viz. Pichit Akkrathit, Joe Breedlove, Michael Evanchik, Shyam Gouri-Suresh, Miwa Hattori, Carter Hemphill, Kyle Hood, Seung-Rae Kim, Kevin Kline, Paul Maksymonko, Juan Carlos Navarro and Huber Salas.

David Kendrick, Ruben Mercado and Hans Amman

Preface to the Revised Edition

In this revised edition we have ordered the chapter based on topic. Furthermore, we have added additional chapters on optimization, automata and a number of other topics.

The example codes were updated to the latest versions of the various software platforms. Where possible we have created examples of the various chapters using software free available on the internet, like Octave, LibreOffice, Maxima and GAMS (free student edition).

David Kendrick, Ruben Mercado and Hans Amman

1 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 such as **MATLAB** or Mathematica and illustrate 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, and 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, and so on, 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 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 economics was converted to mathematics and then to computational models in those areas.

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

2 Software Systems

Students who begin studying computational economics frequently ask the question, *What programming language should I learn?* The answer we give 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 familiar with each of these software systems in the midst of solving the kind of economic models that are naturally developed in each of them. Then later one can dig deeper into one or more of these systems and gain some level of mastery while writing a short midterm 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 for representing 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 with relatively simple models represented in Excel (spreadsheet paradigm) as a way of beginning with a software paradigm that is well known and accessible to almost everybody, since this software system is available on most PCs. Excel is useful for solving small models that do not involve simultaneous systems of equations; however, is not well designed for vector-matrix operations. For this type of operation we use **MATLAB** later in the book. However, Excel has a nonlinear optimization solver that 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.

Early in the book we also introduce Access (relational database paradigm), which, like Excel, is a very accessible software system. Access is well suited for developing 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 for dealing 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 be used as indices, and specifying scalars, parameters, variables, and equations parsimoniously. With GAMS we solve models of transportation, financial planning, general equilibrium, macroeconomics, and global warming.

The vector-matrix paradigm is introduced with **MATLAB**. This software system is useful for dealing with models or problems involving intensive use of vector and matrix operations, cell arrays, and data structures, as well as 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 for solving symbolic algebra and calculus problems, and we use it to represent partial equilibrium and game theoretic problems.

Finally, by means of macroeconomic applications in a Special Topics Section in Part III, 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 departments and students of economics already have many software systems available on their computers and hopefully can also acquire most of the rest of those used in this book. We have provided the input files for the economic models used in this book on our web site at

https://github.com/hansamman/Computational_Economics_Lectures

The web site also contains pointers to software sources, supporting books, and user guides. In an effort to keep student costs down, we have endeavored to keep most of our models small enough that they can be solved with the student versions of the software systems.

With the exception of Duali, all the software systems that we use are commercial products. In contrast, the Duali software is academic software that 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 in an early stage of development and must be used with caution.

3 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 them.

A number of the models presented in the book are solved with linear programming or nonlinear optimization methods based on gradient and/or Newton methods. Thus we provide an introduction to these methods in a separate appendix at the end of the book. Other methods are introduced directly in individual chapters. Neural nets, Monte Carlo methods, and genetic algorithms are applied, respectively, to a stock price prediction problem, a portfolio selection problem, and an evolutionary game and 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 for solving optimal control models with forward-looking variables, is applied to a prototype macro model developed by Taylor (?).

4 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 we explained before, the first class each week is on the economic theory and mathematical model of the subject for the week; the second is on the computational methods used to solve the model, and the third is not in a lecture room but rather in a computer laboratory where students are asked to solve the base model and then to modify (and solve) it 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 paper a few pages long that describes their own experiments with the model during the week and the results they obtained. The weekly teaching cycle is reflected in this book with some suggested experiments listed at the end of each chapter. However, students are encouraged to strike out on their own—a process that enhances both enjoyment and learning.

Since the emphasis in these computational economics courses is on creativity, there is both a midterm paper and a final paper. Students are asked in the midterm 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. However, until now most books in the field have focused on graduate level instruction, whereas we are hoping to be helpful to both undergraduates and graduate students. For an approach using the GAMS software exclusively and concentrating on linear and nonlinear programming methods, see Thompson and Thore (?). For approaches using numerical methods, see Judd (?), who uses several computer languages, or Miranda and Fackler (?), who use **MATLAB**. Varian (?, ?) presents a variety of models in Mathematica. For a web site that supports a course on applied macroeconomics using computational methods taught by Professor 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 books that focus on numerical methods in macroeconomics with some applications

in **MATLAB**, see Marimon and Scott (?) and Adda and Cooper (?). For a book with a collection of articles that consider a variety of numerical methods for solving macroeconomic models see Hughes Hallett and McAdam (?). For a handbook with a collection of articles about computational economics see Amman, Kendrick, and Rust (?). You are also 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. Given the array of materials that are becoming available for teaching computational economics, we are hopeful that courses in this field will become a part of the core curriculum in both undergraduate and graduate education in economics, as was the case earlier with mathematical economics and econometrics. Moreover, we hope our book will motivate and help instructors in those areas to offer courses in computational economics. We are aware that such courses have been offered in recent years at Stanford, Yale, Maryland, Ohio State, Bern, Harvard, and Texas and believe that some of them will migrate toward the core as courses are added at other universities.