FREE SOFTWARE INCLUDED

# MATHEMATICA® FOR MICROECONOMICS

## LEARNING BY EXAMPLE

JOHN ROBERT STINESPRING

AP

# MATHEMATICA® FOR MICROECONOMICS

## LEARNING BY EXAMPLE

**John Robert Stinespring**

School of Economics
Georgia Institute of Technology
Atlanta, Georgia 30332

ACADEMIC PRESS
A Division of Harcourt, Inc.

San Diego San Francisco New York Boston London Sydney Tokyo

# Preface

This book is intended for use as a supplemental tool for courses in advanced microeconomics and mathematical economics. Though the book assumes an understanding of basic calculus and linear algebra, no assumption has been made about the reader's familiarity with computer programming or mathematics software. Included are ample discussions of commands, syntax, and procedures for solving microeconomic problems and plotting the results with an emphasis on exploring common pitfalls that arise in the process. The book aims to accomplish the following:

- present cogent applications of the mathematics required to solve problems of microeconomics
- demonstrate the many uses of computational tools (in this case, *Mathematica*) to perform the mathematics
- provide discussion of the results obtained
- stimulate users to test their knowledge of the subjects presented by extending the programs and performing their own comparative statics and dynamics
- provide users with the tools to build their own *Mathematica* programs for solving economic problems

A book on computational economics such as this is important given the trend towards greater mathematical rigor in economics. Former requirements of only calculus and linear algebra are quickly being surpassed by the need for differential equations and dynamic optimization. This trend has left many students feeling they spend more time trying to master these important mathematical tools than doing actual economics. Professors also find themselves struggling with the trade-off between teaching core economic concepts and adequately covering the necessary mathematics requirements. Computational software packages such as *Mathematica* can help resolve this conflict. Using *Mathematica* allows professors to spend less time working through tedious calculations during lectures and more time discussing economic concepts. The economic programs contained in this book and CD-ROM walk users through solution procedures step-by-step. Including these programs in classroom lectures allows professors to perform multiple comparative statics and plot the results quickly thereby providing deeper, more comprehensive economic analysis. *Mathematica*'s intuitive interface and compatibility with other programs such as Microsoft's *Word* make producing presentation-style printouts of the procedures, solutions, and plots easy.

Students using *Mathematica* at home or in the lab will discover they have a virtual instructor. By typing only a few commands, economic and mathematical concepts are elucidated with solutions and plots. Whether solving particular problems first by hand and then testing them against the computer results or using the computer to create a general solution procedure, economic students who see their field becoming increasingly mathematical will find *Mathematica* to be an excellent teacher. By working through this book and learning the techniques herein, students will be on their way to acquiring a tool that has increasing importance.

The book is organized into four sections. The first section, *Mathematica* Overview, consists of an introductory chapter. The first half of the chapter provides an overview of basic notation, syntax and commands, covering algebraic operations, calculus, linear algebra, differential and difference equations. The latter half of the chapter explores *Mathematica*'s plotting tools. These tools are used to illustrate mathematical properties of common economic functions using two- and three-dimensional plots. The chapter closes with a discussion of *Mathematica*'s animation feature that allows users to view comparative statics dynamically.

The second section, Consumer Theory, explores common problems of consumer optimization. Chapter 2, "Consumer Choice and the Lagrangian Multiplier Method," begins the section by introducing the optimization workhorse of microeconomics. A two-commodity, single-constraint model is set up and input demands are found by differentiation and simultaneous equation solving. Comparative statics are performed and *Mathematica*'s linear algebra package is used to solve for second-order conditions with the bordered Hessian. The model is extended in two ways. First, piecewise budget constraints are used to examine two-part pricing problems. Second, multiple constraints are modeled and the labor–leisure trade-off and the optimal allocation of time in consumption are analyzed. Chapter 3, "Individual and Market Demand," builds on the previous chapter by using the Lagrangian multiplier method to create Marshallian and Hicksian demand curves and Engel curves and to calculate consumer surplus. Users employ *Mathematica*'s substitution and integration commands and learn about its implicit list structure while constructing expenditure functions and indirect utility. Chapter 4, "Compensating and Equivalent Variation," expands on the idea of consumer surplus by analyzing the welfare effects of price and quantity changes. *Mathematica*'s tools for limiting solution output to noncomplex values and its output selection procedures are introduced in the process. Chapter 5, "Pure Exchange," extends the basic consumer model to multiple consumers with endowments of goods. In this framework, market prices are endogenized and the optimal amount of trade between two consumers is derived. Pareto optimality is discussed and illustrated with Edgeworth box diagrams and contract curves. Chapter 6, "Intertemporal Trade," is divided into two parts. In the first part, a single-consumer, two-period model is developed and the optimal amount of lending and borrowing is derived. The model is expanded in the second part, to include two periods and two consumers. Consumers are given endowments in each period that they can trade intertemporally to maximize their intertemporal utility. Chapter 7, "Choice under Uncertainty and Imperfect Information," ends the section on consumer theory by moving away from the world of certainty to examine consumer decisions when multiple outcomes are possible with assigned probabilities. We consider a homeowner that faces a certain probability of his home being destroyed by fire. A lottery model and contingent commodity model are used to derive risk premiums, certainty equivalent income and optimal insurance coverage. Risk sharing, adverse selection, moral hazard and other issues of asymmetric information are addressed by extending the model to two consumers with different risk preferences and/or different probabilities of loss. Tools for plotting piecewise functions and solving inequalities are developed in the process.

The third section, Producer Theory, begins with Chapter 8, "Cost Minimization." This chapter uses the Constant Elasticity of Substitution production function and the Lagrangian multiplier method to derive input demands and the total cost function. In the process, methods of transforming variables and manipulating output are introduced. Plots of total cost functions under varying returns to scale technology are created along with their associated isocosts and isoquants. Chapter 9, "Short-Run and Long-Run Costs," explores the familiar envelope proposition: long-run cost curves are the envelopes of the short-run cost curves. To illustrate, plots of total, average and marginal costs are created for various returns to scale technology. Chapter 10, "Duality," explores the concept of duality in which a unique production function can be derived from a given cost function. The first section tests for the requirements of duality, namely that the cost function is (1) homogeneous of degree one, (2) monotonic, (3) concave, and (4) continuous. These tests require a re-visitation of *Mathematica*'s linear algebra procedures. The second section derives the underlying production function for various functions that have been proven to be proper cost functions. In the process, readers learn common algebraic tricks to derive production functions. Chapter 11, "Multiplant Production," introduces cost minimization under multiplant production. A total cost function is derived for a firm running two separate plants. The theory of multiplant production is generalized to examine complications such as cubic cost functions and fixed costs. In the process, *Mathematica*'s piecewise functions are again used. Having introduced tools for handling the optimization of multiple functions, the concepts are applied to monopolistic production in Chapter 12, "Profit Maximization." The chapter begins with an analysis of a monopolistic firm producing a single commodity but facing two different demands for it. The chapter ends with an examination of a firm that sells as a monopolist in one market and a perfect competitor in another. *Mathematica*'s animation techniques prove especially useful in illustrating comparative statics in these markets. Chapter 13, "Linear Programming," examines profit-maximization for multiproduct firms using fixed-proportions technology. *Mathematica*'s constrained optimization commands are used to solve both the *primal* and *dual* linear programming problems. Chapter 14, "Production and Trade," extends the general equilibrium models presented in Consumer Theory to multiple production and consumption among economic agents. Producer/consumers are given a fixed amount of labor that they allocate between production of two goods. Both output prices and input prices are endogenized in these models.

The fourth section, Economic Dynamics, utilizes the more advanced mathematical tools of differential and difference equations and the calculus of variations to study issues ranging from the dynamic adjustments of markets to the optimal amount of inventory held over time. Comparative dynamics are examined in each chapter to explore the dynamic properties of each problem. Chapter 15, "Market Dynamics," examines the dynamic properties of the most basic analytical tool, supply and demand. Concepts such as intertemporal equilibrium and convergence are discussed. Chapter 16, "Dynamic Optimization and the Calculus of Variations," studies dynamic firm optimization over a given finite time period. The examples used involve solving for the optimal level of inventory to hold over a specified period and optimal pricing for monopolists. The solutions are found by deriving Euler equations from *Mathematica*'s calculus of variations package.

*Mathematica* is best learned by experimentation. Readers are urged to explore the examples, try the end of chapter problems and experiment by changing functional forms and parameter values. Some readers may want to begin by simply executing the programs on the CD-ROM to see what *Mathematica* actually does. Executing the programs merely requires pressing [SHIFT] + [RETURN] at the end of each line.

The examples used in this book were created using *Mathematica version 4 for Windows*. Some differences may be noticed for those using *Macintosh* or earlier *Windows* versions. Most of the differences in version and platform will be transparent to the user.

John Robert Stinespring
Atlanta, Georgia

# Table of Contents

# *Mathematica* Overview

# *Chapter 1*

## Introduction

**Chapter Goals**

a. Discuss the book's notational conventions and typesetting.
b. Introduce *Mathematica*'s kernel and front end along with its palettes and character formatting.
c. Perform mathematics including algebraic operations, calculus, linear algebra, difference and differential equations.
d. Create two- and three-dimensional plots.
e. Animate graphics using `Do` loops.
f. Save output to read into other *Mathematica* notebooks.

**Notebooks**

Introduction.nb

*Mathematica* is a powerful software package that performs mathematical operations from simple calculations to solving complex high-order differential equations. It can create two- and three-dimensional plots of explicit and implicit functions, correspondences, points and figures of geometry. This chapter provides an overview of important *Mathematica* operations and commands for economics.

### Notational Conventions and Typesetting

To distinguish *Mathematica* input and output from the regular text, *Mathematica* input is preceded by >, appears in bold text and is left-justified, while *Mathematica* output appears below the input in Courier regular typeface and is centered.

**>input**

`output`

*Mathematica* commands, such as **`DSolve`**, appear in bold Courier font while economic definitions and *Mathematica* packages appear in bold regular text (Times New Roman).

Menu commands in this text are described using double arrows (⇒). For example, Format⇒Style⇒Input, set in Arial font, means go to the Format menu, then to the Style submenu, and then click on Input.

In addition to menu commands, keyboard commands and shortcuts are used throughout the text. Highlighted words in square brackets, such as [ESC], denote keyboard keys.

The backquote, `, is a special symbol used occasionally in *Mathematica* and should not be confused with an apostrophe.

Most *Mathematica* commands use an arrow, →, to specify options within the command. You may use -> (- followed by >) as an alternative, if you wish.

## The Kernel and Front End

The *kernel* is the computational engine of *Mathematica*. The user inputs instructions and the kernel responds with answers in the form of numbers, matrices, graphs and other appropriate displays. The interface between the user and the kernel, or *front end*, is called a **notebook** and has the file extension *.nb*. Input is typed into a notebook and then executed by pressing [SHIFT] + [ENTER]. (The "+" symbol is used to indicate that keys are pressed simultaneously.)

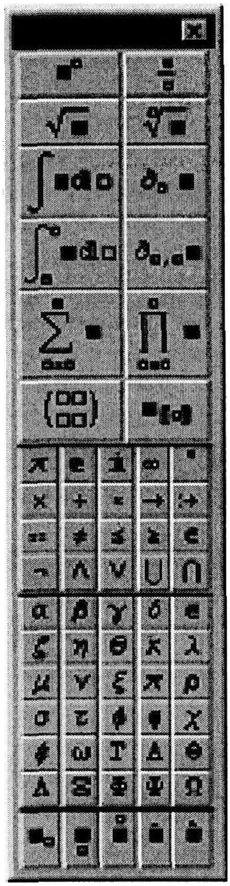
You will notice that the kernel assigns `In[1]` to the input and `Out[1]` to the output. These numbers represent the order in which the kernel evaluates instructions and does not always correspond to the physical position of the instruction within the notebook.

The notebooks discussed in this book and shown on the CD-ROM contain steps for solving constrained optimization problems. To perform these steps, *Mathematica* has many built-in commands contained in **packages**. Important packages for economics include tools for linear algebra and differential and integral calculus. Many of the packages used are preloaded with the interface while others must be "read" into the notebook as will be shown later.

## *Mathematica*'s Palettes

*Mathematica*'s menu interface has many palettes containing symbols and functions. Palettes are found under File⇒Palettes. Consider the palette on the following page. Clicking on the definite integral button (left, 4th from top) writes the following input to the screen

$$\int_{\square}^{\square} \square \, d\square$$



The boxes represent expressions, variables or parameters to be entered by the user. Users also have the option of creating matrices from the palette where the dots on the buttons indicate the dimension of the matrix to be created.

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$$

To fill in the matrix, click on the box and type a number. Additional palettes provide detailed automated operations for users to explore.

## Character Formatting

The *Basic Input* palette, found under File=>Palettes, has Greek letters, special fonts, matrices and operations. For example, clicking on the $\delta$ on the palette (4th column, 6th row from the bottom), writes the character to the notebook. Many users will prefer to use *Mathematica*'s keyboard shortcuts. The shortcut to input Greek symbols is to press [ESC], the corresponding letter on the keyboard and then [ESC] again. Thus $\delta$ is inputted by typing [ESC] d [ESC]. Subscripts are made by pressing [CTRL] + -. For example, $x_1$ is written by typing x [CTRL] + - 1. Superscripts are made by pressing [CTRL] + 6. For example, $x^2$ is written by typing x [CTRL] + 6 2. The following table lists some conventional characters used in economics. Note the shortcut for creating script characters, such as $\mathcal{L}$, is given by [ESC] scL [ESC].

| Key Sequence | Result |
|---|---|
| [ESC] a [ESC] | $\alpha$ |
| [ESC] b [ESC] | $\beta$ |
| [ESC] c [ESC] | $\chi$ |
| [ESC] sc L [ESC] | $\mathcal{L}$ |
| [ESC] r [ESC] | $\rho$ |
| [ESC] p [ESC] | $\pi$ |
| [ESC] d [ESC] | $\delta$ |
| [ESC] t [ESC] | $\tau$ |
| [ESC] q [ESC] | $\theta$ |

## Syntax and Basic Commands

Two points on *Mathematica*'s syntax should be mentioned at the start.

1) *Mathematica* is case sensitive. All *Mathematica* built-in functions begin with a capital letter, such as **`Integrate`** and others, such as **`FindRoot`**, use more than one capital letter.
2) Different brackets are used for different purposes. Square brackets are used for function arguments: **`Sin[x]`** *not* **`Sin(x)`**. Curly brackets are used to denote lists: {1,2,3,4}. Parentheses are used for grouping: (2+6)*4 means add 2+6 first, then multiply by 4.

The most basic computations in *Mathematica* are numerical. *Mathematica* understands both integers and floating-point numbers and has commands for common operations such as addition, **+**, subtraction, **-**, multiplication, *****, division, **/**, and finding factorials, **!**. Examples of each follow. Let us combine addition and multiplication in our first example and find the solution to 100 plus 32 times 4. In the examples that follow, we add fractions and find the factorial of 66.

**>100 + 32*4**

228

**>3/2 + 9/8**

$$\frac{21}{8}$$

**>66!**

```
5443449390774430640037292402478427526442930643887988745328
60126869671081148416000000000000000
```

Exponents can be written using the ^ sign or the keyboard superscript shortcut. Thus $(21.2)^4$ is written as

**>21.2^4**

201996.

or by typing 21.2 [CTRL] + 6 4

**>$21.2^4$**

201996.

Square roots use the common specification of **`Sqrt[expr]`** and exponential functions use **`Exp[expr]`**. The arguments of these functions are enclosed in square brackets.

**>Sqrt[7]**

$\sqrt{7}$

**>Exp[4]**

$$e^4$$

Sometimes the output returned will not be in the desired form. For instance, we might prefer to see the previous result in numerical terms. The `N[expr]` command gives an approximate numerical value as output.

**>N[Exp[4]]**

54.5982

As an alternative, we can simplify by using the % which returns the result of the previous calculation.

**>N[%]**

54.5982

This command can be doubled to refer to the next-to-last result, %%, or $k$ times to refer to the $k^{th}$ previous result. Because *Mathematica* input and output lines are numbered, it is easy to discern the order of previous commands.

To specify the number of digits appearing in output, use the command `N[expr,n]`, where $n$ refers to the number of digits. For instance, to get a precision of 20 digits, merely type

**>N[Exp[4], 20]**

54.598150033144239078

Often the user will want to assign a variable name to an equation or numeric value. In the following chapters, we will repeatedly use a particular utility function to derive demand curves, Engel curves, indirect utility functions and more. Rather than having to rewrite the functional form each time, we can assign it a name. Assignment in *Mathematica* requires `=` or `:=`. The former is an immediate assignment and is evaluated at the time of assignment. The latter is a delayed assignment that is evaluated only when the value is requested, in essence retyped. Output from the former is immediately printed to screen while output from the latter is suppressed. For example, the variable $n$ is defined as having a value of 10 as follows.

**>n := 10**

We can solve for 15 times 10 using

**>15*n**

150