

USING *MATHEMATICA* AS A TEACHING TOOL IN THE UNDERGRADUATE ECONOMICS CURRICULUM

Gary Hodgin*

Abstract

In recent years, the quantitative skills required of economics students have increased significantly. Students often experience difficulty in making the transition from their mathematics courses to the use of mathematics in their economics courses. Symbolic computation programs are mathematical tools that can potentially smooth this transition. This paper illustrates a few ways in which one symbolic computation program, Mathematica, has been used in the undergraduate economics curriculum.

I. Introduction

The undergraduate curriculum in economics has become increasingly quantitative. An economics major typically includes college algebra, statistics, and some form of calculus in its degree requirements.¹ In addition, mathematical economics and econometrics are now standard courses in many undergraduate programs. Moreover, this quantitative emphasis generally extends to other courses across the economics curriculum, particularly the intermediate theory and managerial economics courses. For example, approximately 90 percent of the intermediate microeconomics instructors included in von Allmen and Brower's survey (1998, 279) use calculus in their courses. Although one might question whether too much emphasis is placed on quantitative skills, there appears to be consensus among economics professors that mathematics is an important ingredient in undergraduate economic education.

Over the past three years, I have used a symbolic computation program entitled *Mathematica* in two of my intermediate microeconomics and three of my managerial economics courses. My observations in these courses suggests that some students experience difficulty in making the transition from their courses in mathematics to their use of mathematics in economics. By smoothing this transition, symbolic computation programs can assist students in learning economics. The objective of this note is to demonstrate a few of *Mathematica's* many features and capabilities that I have found useful.

The remainder of this note consists of three sections. The first provides a brief description of symbolic computation programs with a focus upon those features that are most likely to be of interest to instructors of intermediate microeconomics and managerial economics. Because of my limited knowledge of other symbolic computation programs, I will refer to those features in *Mathematica*. The second section discusses three ways in which instructors might use a symbolic computation program in their courses. This section includes examples of how I have used *Mathematica* in my classes. The final section concludes with a brief discussion of a few factors potential users should consider before integrating a symbolic computation program into their courses.

* Associate Professor of Economics, Belmont University, Nashville, TN 37212. Email: hodging@mail.belmont.edu I would like to express appreciation to Sam Dennis for introducing me to *Mathematica* and to an anonymous referee for helpful comments on a previous version of this paper.

II. Symbolic Computation Programs

Symbolic computation programs are computer programs that possess the capability to perform numerical, symbolical, and graphical analysis. Five of the more popular programs listed in the appendix with web addresses, telephone numbers, and prices. Each of these programs combines built-in features and programming capabilities in much the same way as the econometrics software packages. *Mathematica* consists of a system interface and a notebook interface. The notebook interface is actually a front-end that allows one to use the many built-in functions and packages without having to learn the more difficult system language. Thus, a programming novice, such as myself, can effectively use the program without much difficulty.

Although the use of symbolic computation programs has become commonplace in science and mathematics, their use in economics seems to be largely confined to research (Varian 1993, 1996) and graduate courses in mathematical economics (Huang and Crooke 1997). However, judging from the literature their use is becoming more common in the undergraduate curriculum (Boyd 1998; Walbert and Ostrosky 1997). Undoubtedly, the present emphasis on technology in the classroom and the greater availability of computers for classroom use have played a role in their expanded use.

As a tool for learning, the capability of symbolic computation programs to combine graphical, numerical, and symbolical capabilities offers advantages in the economics classroom. For example, a commonly stated objective in many economics courses is to teach students to "think like an economist." In microeconomics, this means being able to analyze an event, such as a hypothetical change in the minimum wage, in an appropriate economic model of wage determination that describes the effects of the event on those variables under consideration. At a minimum, students must be able to translate the event from words into diagrams. In some cases, however, diagrams alone are not enough. When a model involves several variables, algebra and elementary calculus are better for tracking the relationships. Students learn to apply economic theory by diligently engaging in this practice of model building. Symbolic computation programs are excellent tools for assisting in this aspect of learning because they allow students to concentrate more on understanding the economic principles behind their models and less on the computational details of their models.

III. Uses of *Mathematica*

Mathematica is extremely flexible and offers instructors several application options.² First, it can be used as a sophisticated graphic-numeric-symbolic calculator. The software can create customized examples for lectures, handouts, problem sets, and exams. Because of its vast library of built-in functions, one can quickly learn to use *Mathematica* for these purposes. For example, one can easily experiment and generate a cubic function whose corresponding average and marginal functions make sense from an economic perspective. Similarly, a specific type of utility function is easily defined and graphically illustrated with a corresponding set of indifference curves. The primary benefit in using *Mathematica* in this way is the reduction in time required to create good examples for classroom use.

Second, *Mathematica* notebooks can generate in-class presentations or documents for display on the Web. Notebooks are ideal for presenting material on overhead transparencies and computer presentation systems. Notebooks consist of cells, which can contain text, mathematics, graphics (with color and animation options), and sound. These cells can be collapsed and expanded with a simple click of the mouse. Thus, one can progress through a multi-stepped procedure in a systematic manner or move back and forth among the various steps by expanding and collapsing cells. In addition, notebooks can be saved in HTML format and displayed on the Web. Crooke, Froeb, and Tschantz (1998) provide an excellent example of *Mathematica's* interactive Web capabilities with applications in economics.

Third, instructors can fully integrate *Mathematica* into their courses. Clearly, this is the most costly and time-consuming way to use *Mathematica* because students must purchase the software and learn basic *Mathematica* programming. At prices ranging from approximately \$75 to \$140, the cost of a symbolic computation program is significant. Moreover, the additional time required for instructors to assist students in learning basic programming is substantial and would most likely come at the expense of course content. Primarily for these reasons, I have confined my use to that of creating course materials and in-class presentations. Nevertheless, as the availability of computers and students' familiarity with symbolic computation programs spread, these concerns may diminish to a point where the use of symbolic computation software will become as commonplace in the undergraduate curriculum as that of econometric software.

IV. Examples

The following five examples are similar to exercises developed for my intermediate microeconomics and managerial economics classes over the past three years. As each example is presented, bear in mind that *Mathematica* is extremely flexible, and there is usually more than one way to achieve the same result. In these examples, the bold statements preceded by "*In[.] :=*" represent input statements and the statements preceded by "*Out[.] :=*" represent output statements. These expressions are generated internally and not by the user. They are included here so that the examples will be more resemblant of the output actually generated in *Mathematica*.

Example 1

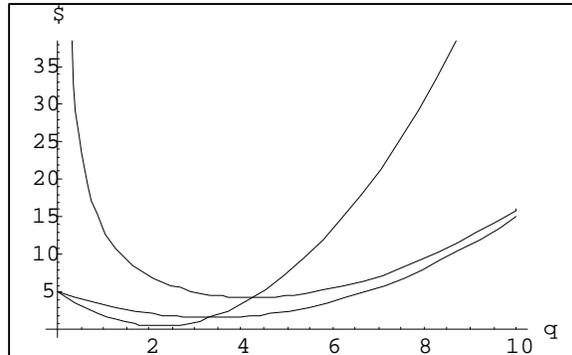
In this example, *Mathematica* graphically illustrates the unit cost functions corresponding to a specific short-run total cost function and then finds the rate of output at which average variable cost and marginal cost are equal. The first set of input statements specifies the total cost and total variable cost functions and defines the corresponding unit cost functions. The semicolon at the end of each input line tells *Mathematica* to refrain from printing the functions.

```
In[1] := tc = 10 + 5q - 2q^2 + .3q^3;
      tvc = 5q - 2q^2 + .3q^3;
      atc = tc/q;
      avc = tvc/q;
      mc = D[tvc, q];
```

The "Plot" and "Solve" commands are now used to plot the unit cost curves and solve for the rate of output at which average variable cost and marginal are equal. The plot range is specified as 0 through 10 units and the axes are labeled as "q" and "\$".

```
In[2]:= Plot[{atc, avc, mc},{q, 0, 10}, AxesLabel -> {"q","$"}]
        Solve[mc == avc, q]
```

Figure 1: Unit Cost Curves



```
Out[2]= -Graphics-
```

```
Out[3]= {{q -> 0}, {q -> 3.3333}}
```

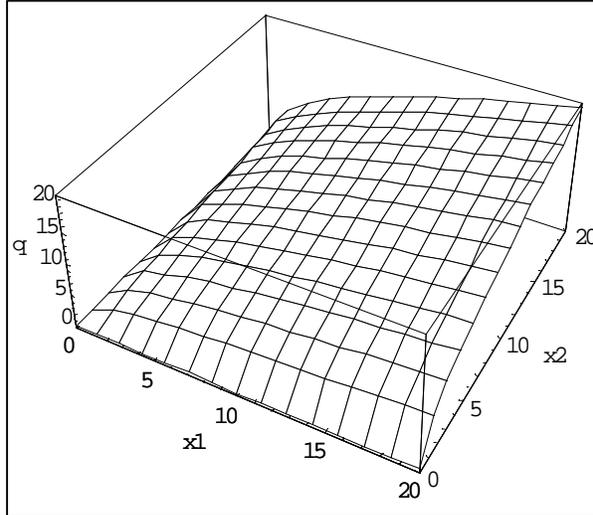
Mathematica found two solutions for the rate of output at which average variable cost and marginal cost are equal, the trivial solution of 0 and economically useful 3.33 units.³

Example 2

This example involves specifying a Cobb-Douglas production function, $q = \sqrt{x_1 x_2}$, and constructing a three-dimensional diagram of the production function along with five isoquants corresponding to the production function. The shading feature has been disabled.

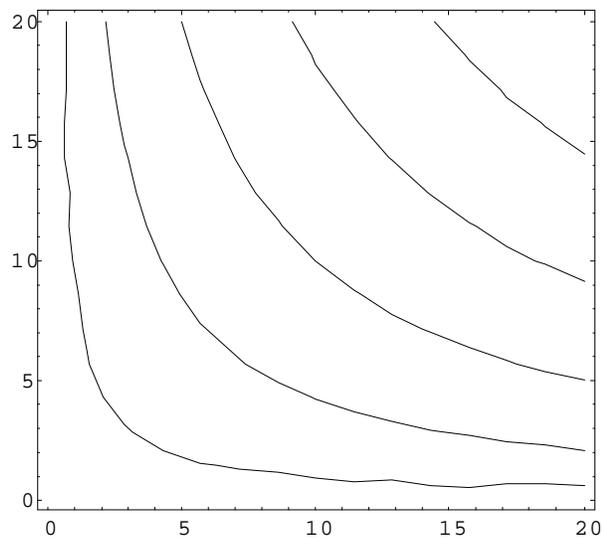
```
In[1]:= f[x1_, x2_] = x1^.5 * x2^.5;
        Plot3D[f[x1, x2], {x1, 0, 20}, {x2, 0, 20}, AxesLabel -> {"x1", "x2", "q"},
        Shading -> False]
        ContourPlot[f[x1, x2], {x1, 0, 20}, {x2, 0, 20}, ContourShading -> False,
        Contours -> 5]
```

Figure 2: Production Function



`Out[1]= - SurfaceGraphics -`

Figure 3: Isoquants



`Out[2]= - ContourGraphics -`

A nice feature of generating production functions and corresponding isoquants in this manner is that one can demonstrate how specific changes in technology affect the shape of the isoquants by using specific numerical values.

Example 3

In this example, *Mathematica* is used to graphically illustrate revenue and cost functions and solve for the profit-maximizing rate of output and price. First, an inverse demand function is specified with its

corresponding total and marginal revenue functions. A total cost function is then specified with its corresponding marginal cost function.

```
In[1]:= p = 200 - 7.5q
        tr = p*q
        mr = D[tr, q]
        tc = 400 + 10q + 4q^2 + .5q^3
        mc = D[tc, q]
        profit = tr - tc
```

Out[1]= $200 - 7.5q$

Out[2]= $(200 - 7.5q)q$

Out[3]= $200 - 15q$

Out[4]= $400 + 10q + 4q^2 + 0.5q^3$

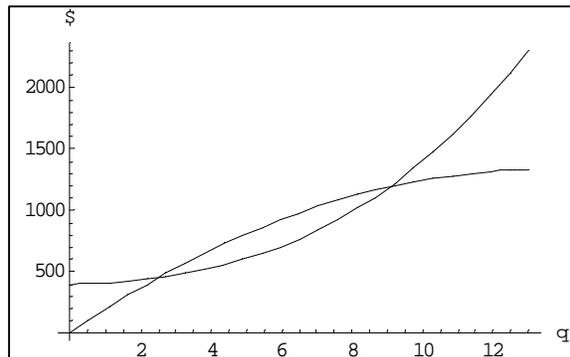
Out[5]= $10 + 8q + 1.5q^2$

Out[6]= $-400 - 10q + (200 - 7.5q)q - 4q^2 - 0.5q^3$

Three "Plot" commands are now used to generate separate diagrams for the total revenue and total cost functions; the price, marginal revenue (dashed), and marginal cost functions; and the profit function. The plot range is specified as 0 through 13 units.

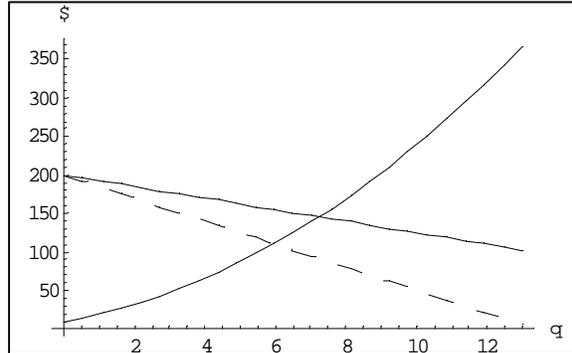
```
In[7]:= Plot[{tr, tc}, {q, 0, 13}, AxesLabel -> {"q", "$"}]
        Plot[{p, mr, mc}, {q, 0, 13}, AxesLabel -> {"q", "$"},
        PlotStyle -> {{}, Dashing[.05], {}]}]
        Plot[profit, {q, 0, 13}, AxesLabel -> {"q", "$"}]
```

Figure 4: Total Revenue and Total Cost



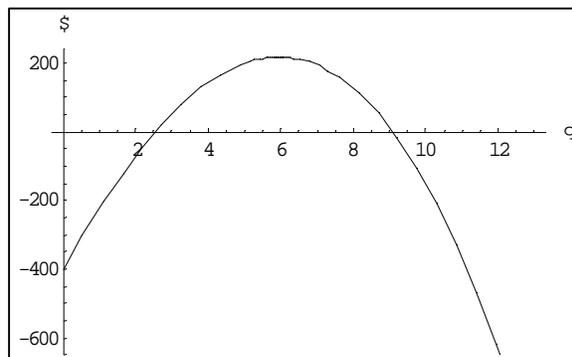
Out[7]= - Graphics -

Figure 5: Price, Marginal Revenue, and Marginal Cost



Out[8]= - Graphics -

Figure 6: Profit Function



Out[9]= - Graphics -

Lastly, *Mathematica* solves for the profit-maximizing rate of output and substitutes this solution back into the price and profit equations to obtain the profit-maximizing price and profits.

In[10]:= **Solve[*mr* == *mc*, *q*]**

Out[10]= **{{*q* -> -21.2845}, {*q* -> 5.95113}}**

In[11]:= ***p* /. *q* -> 5.95**

profit /. *q* -> 5.95

Out[11]= 155.375

Out[12]= 218.049

Thus, the profit-maximizing rate of output is 5.95 units; price is \$155.38, and profits are \$218.05.

Example 4

In this example, the Lagrangian multiplier method is used to solve the first order conditions for finding maximum output subject to a cost constraint. The production function is assumed to be

$f(x_1, x_2) = .5 \log(x_1) + .5 \log(x_2)$ and the cost constraint is assumed to be

$g(x_1, x_2) = 2x_1 + x_2 - 60$. The critical values of input x_1 , input x_2 and $lambda$, the Lagrangian multiplier, are found by setting the derivative of the Lagrangian function with respect to each variable equal of zero and solving the three equations simultaneously.

```
In[1]:= f[x1_, x2_] := .5 * Log[x1] + .5 * Log[x2];
      g[x1_, x2_] := 2*x1 + x2 - 60;
      L[x1_, x2_, lambda_] := f[x1, x2] + lambda*g[x1, x2];
      foc1 = D[L[x1, x2, lambda], x1]
      foc2 = D[L[x1, x2, lambda], x2]
      foc3 = D[L[x1, x2, lambda], lambda]
      c = Solve[{foc1 == 0, foc2 == 0, foc3 == 0}, {x1, x2, lambda}]
      maxvalue = L[x1, x2, lambda] /. ->c[[1]]
```

$$Out[1]= 2 * lambda + \frac{0.5}{x1}$$

$$Out[2]= lambda + \frac{0.5}{x2}$$

$$Out[3]= -60 + 2(x1) + x2$$

$$Out[4]= \{\{lambda \rightarrow -0.1667, x1 \rightarrow 15.0, x2 \rightarrow 30.0\}\}$$

$$Out[5]= 3.05462$$

The critical values are given in *Out[4]* and the maximum rate of output that satisfies the constraint is given in *Out[5]*. Although not a part of this exercise, it is relatively easy solve problems such as this one using the standard graphical method of isoquants and isocost lines.

Example 5

This last example is identical to the previous example 4 except the critical values are solved symbolically rather than numerically. The production function is

$f(x_1, x_2) = a_1 \log(x_1) + a_2 \log(x_2)$ and the constraint is $g(x_1, x_2) = p_1 x_1 + p_2 x_2 - c$, where

p_1 is the price per unit of input x_1 , p_2 is the price per unit of input x_2 , and c is total cost.

"SetAttributes" commands *Mathematica* to treat a_1, a_2, p_1, p_2 , and c as constants.

```

In[1]:= Clear[f, g, x1, x2, L, C, lambda];
SetAttributes[{a1, a2, p1, p2, C}, Constant];
f[x1_, x2_] := a1 * Log[x1] + a2 * Log[x2];
g[x1_, x2_] := p1 * x1 + p2 * x2 - C;
L[x1_, x2_, lambda_] := f[x1, x2] + lambda * g[x1, x2];
foc1 = D[L[x1, x2, lambda], x1];
foc2 = D[L[x1, x2, lambda], x2];
foc3 = D[L[x1, x2, lambda], lambda];
Solve [{foc1 == 0, foc2 == 0, foc3 == 0}, {x1, x2, lambda}]

```

$$Out[1]= \left\{ \left\{ \lambda \rightarrow \frac{-a_1 - a_2}{C}, x_1 \rightarrow \frac{(a_1)C}{(a_1 + a_2)p_1}, x_2 \rightarrow \frac{(a_2)C}{(a_1 + a_2)p_2} \right\} \right\}$$

The solution in *Out[1]* can be verified by comparing these critical values to those found in the previous example. A major advantage of symbolic computation programs is their ability to solve problems symbolically as well as numerically.

V. Summary

Over the past decade, technology has had a pronounced effect on the way economic educators accomplish their mission. Used appropriately, symbolic computation programs can assist in accomplishing that mission. Although I have never formally surveyed students on their opinions of *Mathematica*, their causal comments have been generally favorable.⁴ Specifically, some were impressed by its graphical capabilities and its ability to reduce computational difficulties. Nevertheless, there are a few factors that should be carefully considered before integrating such technology into the classroom.

First, there are practical questions surrounding such integration. One must decide on which software is most appropriate and on what level it will be used. Obviously, a decision to fully integrate the software into a course is the most costly and time-consuming. Such factors as the adequacy of existing computer facilities, the costs of site licenses for the software or student versions of the software, the size of the class, the students' experience with computer software, and the learning objectives of the course, are important considerations. Some potential problems can be minimized if decisions are coordinated with other economics and mathematics faculty, but these practical concerns are ultimately the responsibility of the individual instructor.

Second, there is an issue of pedagogy surrounding the implementation of new technology into the classroom. New technology often influences "what" one does and not simply "how" one does it. Moreover, it often does so in abstruse ways. In this case, symbolic computation programs should be used

as mathematical tools to assist students in learning economics, not one of limiting "what" economics students learn to that which can be taught using mathematics.

Notes

1. In a 1995 survey of the American Association of Collegiate Schools of Business membership, von Allmen and Brower (1998, 278) found that 62.8 percent of economics departments required either calculus or applied calculus in their curriculum.
2. See Wolfram (1996) for a comprehensive description of *Mathematica*.
3. For classroom use, a shortcoming of *Mathematica*'s graphics is that one cannot directly apply text labels to functions, such as the unit cost functions in this example. However, one can insert them with a compatible picture editor.
4. I am not sure how well this generally favorable opinion would hold if students were required to interact directly with *Mathematica*.

Appendix

Table A1: Software Vendors

Product	Company	Web Address	Telephone	Price^a
<i>Maple</i>	Waterloo Maple, Inc.	http://www.Maplesoft.com	800-267-6583	\$129.95
<i>Mathcad</i>	Mathsoft, Inc.	http://www.mathsoft.com	800-628-4223	\$129.95
<i>Mathematica</i>	Wolfram Research, Inc.	http://www.wolfram.com	800-441-6284	\$139.95
<i>MATLAB</i>	The MathWorks, Inc.	http://www.Mathworks.com	508-647-7000	\$99.00
<i>Scientific Notebook</i>	Mackichan Software, Inc.	http://www.Mackichan.com	206-780-2799	\$74.95

^aWith the exception of *Scientific Notebook*, all prices are for student or academic versions as of July, 1999. There is only one version of *Scientific Notebook*. All offer site licenses except Mackichan Software, which offers quantity discounts. Most, if not all, of these vendors provide time-locked versions of their products for review.

References

- Boyd, David W. 1998. "On the use of symbolic computation in undergraduate microeconomics instruction." *Journal of Economic Education* 29 (3): 227-46.
- Crooke, Phillip, Luke Froeb, and Steven Tschantz. 1998. "Pedagogy using Mathematica through the Web." <http://mba.vanderbilt.edu/luke.froeb/mathserv>.
- Huang, Cliff J. and Phillip Crooke. 1997. *Mathematics and Mathematica for economics*. Malden, MA: Blackwell Publishers, Inc.
- Varian, Hal, ed. 1993. *Economic and financial modeling with Mathematica*. New York: Springer-Verlag, Inc.
- . 1996. *Computational economics and finance: Modeling and analysis with*

Mathematica. New York: Springer-Verlag, Inc.

Von Allmen, Peter and George Brower. 1998. "Calculus and the teaching of intermediate microeconomics: Results from a survey." *Journal of Economic Education* 29 (3): 277-84.

Walbert, Mark S. and Anthony L. Ostrosky. 1997. "Using Mathcad to teach undergraduate mathematical economics." *Journal of Economic Education* 28 (4): 304-15.

Wolfram, Stephen. 1996. *The Mathematica book*. Cambridge, UK: Cambridge University Press.