Using Matlab's Symbolic Math Toolbox: A Tutorial

AGEC 642 - 2024

I. Matlab as a tool, not a crutch

Matlab's Symbolic Math Toolbox (or any other symbolic algebra program) can be used to find analytical solutions to problems. It can be an invaluable tool, helping to save hours of frustrating pencil time and avoiding careless mistakes. But it can also be a crutch. It is often the case that valuable intuition arises in the process of solving a problem. Sometimes computers will jump over these intermediate steps or simplify a solution in a fashion that robs the work of any intuition. I have reviewed papers for journals in which algebra was clearly done by a computer and the result was the author had little understanding of what was actually being done in the process. Do not let this happen to you. *Use Matlab to help you understand; do not let it keep you from understanding.*

Furthermore, I recommend that you use an AI Large Language Model (e.g., ChatGPT which I use) to help you develop code. These can be extremely valuable tools for coding, helping with both syntax and organization. However, it is important that you understand what the code does after you request it, otherwise you will be unable to identify errors in your code, especially errors that give results you do not intend but errors that do not generate an error message.

II. The big picture: An example of some Matlab code

Just to give you an idea of the type of thing that Matlab can do for you. Here is an example of simple code to solve a consumer utility maximization problem and obtain the Marshallian demand function using Roy's identity **(do not try this yet)**

```
% Initialize variables
syms a b c x y z l m px py pz
% Specify utility function
u = a*log(x)+b*log(y) + c*log(z)
% Lagrangian and FOCs
L = u -l*(m-px*x-py*y-pz*z)
dLdx = diff(L,x)
dLdy = diff(L,y)
dLdz = diff(L,z)
dLdl = diff(L,l)
% Solve the system for x, y and z
[xstar ystar zstar lstar ] = solve(dLdx, dLdy, dLdz, dLdl, x, y, z, l)
% Indirect utility function and Roy's Identity
V = a*log(xstar)+b*log(ystar) + c*log(zstar)
dVdm = diff(V,m)
dVdpx = diff(V,px)
xM = -dVdpx/dVdm
```

III. Getting Started

1. Start Matlab. We will begin by using the command-line interface. A more sophisticated programming approach will be discussed later. You know you are in the command-line interface if you see a >> prompt.

Starting at this point, you should try all of these commands by typing them in. You should type; do not copy and paste, because you will notice more when you're actually typing.

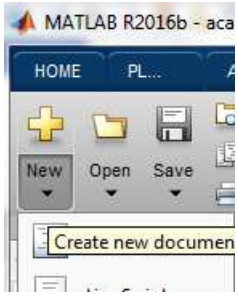
2. Any variable to be used on the right-hand side of an equation must be introduced with a syms command, e.g.,
   >> syms a b c x
3. Unlike variables on the right-hand side, if a function or variable is introduced on the left-hand side of an equation, it does **not** need to be specified first, e.g.,
   >> f=a+b*x+c*x^2
   This creates a new variable, f, that is a function of a, b, c, and x.
4. Note that if you put a semicolon (;) at the end of a line, it does not print the output to the screen. Compare the output of the previous line with
   >> f=a+b*x+c*x^2;
5. Using the up and down arrows on your keyboard, you can access previous command lines and then repeat or edit them as needed.
6. Note that Matlab is case sensitive. For example, if you simply enter the command
   >> f
   it will reproduce the equation. If you type F, on the other hand, you will get an error message. Commands are similarly case sensitive.

IV. Very Basic Commands

7. Using your preferred AI LLM, get the code to solve the quadratic equation:
   $a \times x - d \times x^2 = m$.
   Implement this command using the command prompt.
   Stop and we will discuss what each line of the code does.

V. Scripts (a.k.a. M-files)

8. Up to this point, we have been implementing our code using the command-line interface. You can also, and usually preferably, use script files that contain a sequence of commands. These used to be called M-files. A script has the advantage that you can print it, and easily share and reproduce your results. No matter what programming environment you are working in, when you are working on anything other than a very short calculation, <u>always</u> write a program so that you have a record of what you did and it is easily reproduced.
9. To create a Matlab script, you can click on the "New" icon and select "Script"



or just type ctrl-n. This will create an empty file in which you can type commands. Type a sequence of commands, such as

```
syms a b c x
solve(a+b*x+c*x^2==3,x)
```

Save this file under a name such as MatlabTutorial.m.

Caution: For some reason, Matlab is very particular about names for script files. You must start the file's name with a letter and make sure there are no spaces in the name.
Note: the solve command could have been written instead

```
solve(a+b*x+c*x^2-3,x)
```

10. Cells within an M file can be extremely convenient in Matlab. A cell is a piece of a larger program that can be run separately. Each cell starts with the symbols "%%" For example:

```
%% This cell solves a differential equation
```

A cell can be run by pressing ctrl-enter (Windows).
A full program can be run with F5

11. Place the code that you generated above within a cell in your M file and run it.
12. We will walk through the process of debugging.

VI. Excercises

Using Matlab and aided by an AI LLM, complete the following exercises.

13. Take the derivative of the function f(x) = a*x – ln(x*2).
14. Find the value of x where df(x)/dx = 0.
15. Find the solution to the problem

$$\max_x u(x) \qquad s.t\ \ px \le m$$

where $u(x) = x^{\alpha}$ with $0<\alpha\le 1$.

16. Solve the constrained optimization using the Lagrangian to find the solution to the problem

$$L(x,\lambda) = u(x) + \lambda(m - px)$$

revealing the optimal values of x and λ.

17. Find the solution to the indefinite integral

$$\int f(x)\,dx \quad \text{where } f(x) = 1/x^3$$

18. Find the solution to the definite integral

$$\int_1^{100} f(x)\,dx \quad \text{where } f(x) = 1/x^3$$

19. Solve the differential equation

$$\dot{x}_t = rx_t$$

20. Solve the differential equation
    $\dot{x}_t = rx_t$
21. Solve the differential equation
    $\dot{x}_t = rx_t$ with $x(t=0) = x_0$
22. Solve the differential equation
    $\dot{x}_t = a \cdot x_t + b \cdot t$ with $x(t=0) = x_0$
23. Create a phase diagram for the system of equations
    $\dot{x}_{1t} = 8 \cdot x_{1t} + .2 \cdot x_{2t} - 7$
    $\dot{x}_{2t} = x_{2t} - 3$
24. Create a phase diagram for the system of equations
    $\dot{x}_{1t} = a \cdot x_{1t} + b \cdot x_{2t} - 7$
    $\dot{x}_{2t} = x_{2t} - c$
25. Find the Eigen, λ_1 and λ_2 of the 2×2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
    And displaying them to the screen in a way that is visually appealing.
26. Get latex code for λ_1 and λ_2.