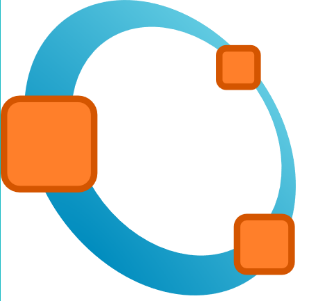
**Octave Tutorial[[1]](#footnote-1) - Intro and Ch 11**

*Background, Basic Calculations, Vector Operations, and Onramp*

**Background:**

Octave is a free computing program that is designed to [mirror](https://en.wikibooks.org/wiki/MATLAB_Programming/Differences_between_Octave_and_MATLAB) MATLAB.

MATLAB stands for “Matrix Laboratory,” since all values are stored in matrices.

Octave and MATLAB solve problems numerically, not algebraically. But what does this mean?

Solving Numerically (Such as Octave and MATLAB)

* Using really good decimal approximations
* Think of finding area under a curve using 1000 rectangles.
* This is how a TI-84 evaluates a derivative or definite integral.

Solving Algebraically (such as Mathematica and Maple)

* In this case the program would actually know ***algebra***.
* Think about finding area under a curve by actually finding an expression for the antiderivative and then plugging in values.
* This is how a TI-Nspire evaluates a derivative or definite integral.
* Note: CAS stands for Computer ***Algebra*** System

While Octave and MATLAB are both originally designed to solve problems numerically, they each have extra add-on packages that enable them to perform calculations algebraically. The one for Octave is called “Symbolic.” We will not be focusing on this for now.

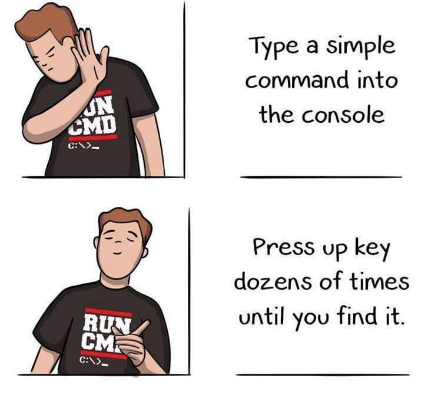
Octave is “[A high-level interactive language for numerical computations](https://octave.org/octave.pdf).”

* “***High-level***” – Computers work all in 0s and 1s. But you don’t have to program that way. You can type in high level commands like “polyder” (polynomial derivative) and the program will translate that all the way down to 0s and 1s for you.
* “***Interactive***” – Sometimes in programing (such as C++), the user must write a whole program before telling the computer to start even the first step. But in Octave, you can first define x = 3; ENTER. Then y = 4; ENTER. Then z = x + y; ENTER. Then type z ^ 2 and get the answer. There is a lot of back-and-forth.
* “***Numerical computations***” – See above.

The name “Octave” does not come from music. Rather, the designers who created Octave named it in honor of one of their former professors, who was really fast with calculations.[[2]](#footnote-2)

**Basic Calculating:**

To get familiar with basic expression syntax, try the following. On a test, be ready to replicate or explain any of these procedures.

1. 5 + 7 🡨 Note: spaces are just used to make your code readable. Use them as you wish.
2. sqrt(25)
3. sqrt(-1)
4. sqrt(5 + 3i)
5. log10(10000)
6. log(10000)
7. log(e ^ 4) 🡨 What do these results tell you about the “log” function?
8. log(e ^ 5) 🡨 Do this by pressing the “up” arrow and editing the previous input.
9. sin(90)
10. sin(pi/2) 🡨 What do these results tell you about Octave?
11. sind(90) 🡨 What does the “d” stand for?
12. Guess the command for (answer below)
13. To repeat , but with more accuracy, try:
    1. Type: format long <ENTER>
    2. Type the command again (Use the ↑ arrow twice!)
    3. To switch back, type: format short <ENTER>

**Basic Vectors and Matrices:**

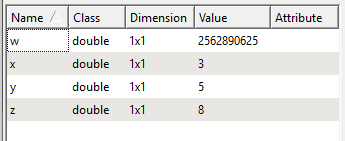
Try this:

x = 3

y = 5

z = x + y

Each time you do this, Octave prints out the results. But what if you want to assign a variable but NOT print the results? Use a semicolon. Try this:



w = (x \* y) ^ z; 🡨 use the semicolon

If you want to see the value of w, just type: w <ENTER>

To see a list of all the variables you defined, look on the left:

(if you do not see this, go to Window 🡪 Show Workspace

Now, clear all the variables, type: clear <ENTER>

Octave actually encodes all variables values in matrices. Did you see the “Dimension” column your workspace? All those variables were really 1x1 matrices. Now, let’s create a 2x3 matrix:

y = [3 5 7 ; 10 11 12] (then look in your workspace)

and we wish to square each element. Try typing:

y ^ 2

This should return an error. The reason is that you just now asked the computer to do:  
   
which does not work in matrix multiplication. For component-wise arithmetic between matrices, use a period.

y .^ 2

Try some of these and learn what they do:

1. linspace(0, 20, 3)
2. linspace(0, 20, 21)
3. linspace(0, 20, 1001) Can you figure out what these three inputs do?[[3]](#footnote-3)

Another way to create a vector of evenly spaced values:

1. 10 : 2 : 50
2. 1 : 0.01 : 2 This is one time when choosing to use spaces helps a lot, visually.
3. 0 : 0.1 : 2\*pi Does it go all the way up to 6.28?

Indexing Vectors

1. Create x = 10 : 3 : 28 and then look at it.
2. x(1)
3. x(2) What is this doing?
4. x(1 : 3)
5. x(5 : -1 : 1)

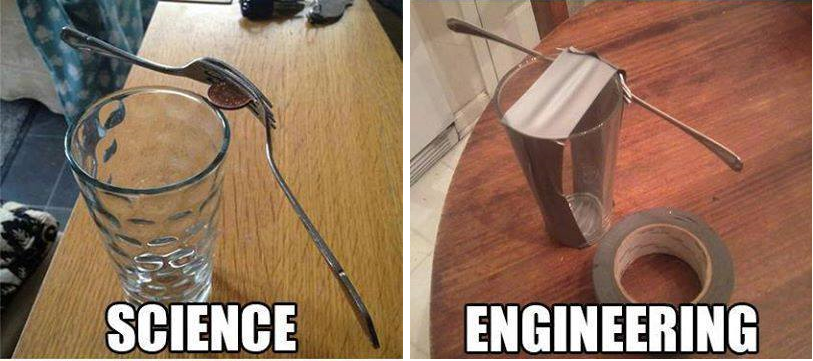
Indexing Matrices

1. Create y = [1:10; 11:20; 21:30] and look at it.
2. Try y(1)
3. Try y(2) . Then y(3) What are these giving?
4. y(2 , 5)
5. What command will show the entry in row 3, column 1?

To specify an entire row or column, use a colon ( : )

1. y( : , 8)
2. y( 2 , : ) In my head, I (Malan) think “row 2, columns-ALL” [[4]](#footnote-4)

**Side Note:**

****Throughout Algebra and Calculus, you are encouraged and often required to give *EXACT* answers. They are perfect; elegant; pure.

means, “The true exact value of the square root of two. We could never write it down accurately with decimals. So here it is in its most perfect form, expressed as an idea.” Wow.

But in practice, when you actually need build a bridge and measure dimensions for the beams, a value like this…

>> format long

>> sqrt(2)

ans = 1.41421356373095

…is plenty good enough. Octave (along with MATLAB) stores values and evaluates expressions *numerically*, rather than *algebraically*. This may seem annoying/weak at first, but in practice it serves just fine and may even be more useful. If you shot laser at the moon and the 14th digit of your degree angle of measurement were rounded (and so was not the true perfect angle), you would miss your target by less than 0.000067090mm [[5]](#footnote-5).

**Chapter 11 Skills**

Along with the skills and concepts above, you must be ready to demonstrate the following Chapter-11-specific skills on the unit test.

11.1 – Vectors in the Plane

* Magnitude of a vector  
  - Create a vector x = [3, 6]  
  - Use norm() to find its magnitude norm(x)  
  - Challenge: can you combine those into 1 step?
* Linear Combination  
  - Create two vectors x = [3, 6]; y = [7, 7]; 🡨 one line of code! ☺  
  - Use +, –, and \* to combine z = 3\*x + 9\*y
* Make a unit vector  
  - Create a vector x = [9, 2]  
  - Divide by it by its magnitude Try it! You should get 0.97619 0.21693
* Create vector from angle and magnitude Example: Plane flies SW at 300 k/hr  
   Symbols:   
   Can you calculate in octave syntax?

11.2 – Space Coordinates and Vectors in Space

* Determine if two vectors are parallel a = [3, 4, 5]  
   b = [9, 3, 2]  
   Do component-wise division a ./ b  
   If all quotients are the same, then vectors were //.
* Are three points collinear? Encode the points as vectors x, y, and z  
   v1 = y – x v2 = z – x   
   Test whether v1 and v2 are parallel.

11.3 – The Dot Product of Two Vectors

* Basic dot Use a and b from above  
   dot(a,b)
* vector projection  
  in algebraic notation Use a and b from above 🡪 dot(a,b)/dot(b,b)\*b  
   This is the projection of \_\_\_\_ on to \_\_\_\_.

11.4 – The Cross Product of Two Vectors in Space

* Cross Use a and b from above: cross(a,b)
* Magnitude of cross product norm(cross(a,b))
* Volume of parallelepiped formed by a, b, c Use a and b from above, and c = [1, 2, 3]  
   together = [a; b; c]  
   abs(det(together))  
   Challenge:  
   🡪 Can you also do this with a triple scalar product?

11.5 – Lines and Planes in Space

* Form plane spanned by 3 points For any three ***points*** P, Q, R (encoded as vectors)  
   PQ = Q-P PR = R-P  
   n = cross(PQ, PR)  
   Then use n by hand to get plane
* Acute angle (in degrees) between planes Get normal for each: n1 and n2  
   Angle = acosd(dot(n1, n2) / (norm(n1)\*norm(n2)))

11.6 – Surfaces in Space

* Draw a Sphere Try sphere(20) and sphere(80). What’s the input do?
* Draw ellipse Try ellipse(0, 0, 0, 10, 2, 5)  
   Then type axis equal.  
   Try changing the 0, 0, 0. What does that do?

11.7 – Cylindrical and Spherical Coordinates

* None

**MATLAB Onramp**

MATLAB is the professional-grade software off of which Octave is patterned. In an effort to promote their product, Mathworks offers a free intro course called Onramp. You can sign up for Onramp and complete assigned lessons.

Fall 2021 – Onramp is great, but I am not sure how much I will assign. But if you want to go ahead and use the tutorials, go for it! Then if I require it for homework, you will already be done 😊.

Getting Started on Onramp

1. Go to <https://matlabacademy.mathworks.com/>
2. On the top right, click “Sign In”
3. Look for “No account?” Click on “Create one!”  
   If a “Select a Web Site” popup appears, pick 中国 English. I believe this is to make it run on a more local server. (faster)
4. Create your account. For ‘How will you use MathWorks software?” select “Student use”
5. Once you have an account, go back to <https://matlabacademy.mathworks.com/>. Look for MATLAB Onramp and click Launch

Onramp Comments/Tips:

* Logging on can take a long time, but once you’re on one lesson page, it should move from one page to the next fairly quickly.
* It should work fine with or without a VPN.
* Some Mac users found better luck with Safari.
* If you find any challenges and discover good tricks, please let me know.

1. Many of the examples and explanations found here come from the [official Octave documentation](https://octave.org/octave.pdf) and another great tutorial written by [Dr. P.J.G. Long](http://www-h.eng.cam.ac.uk/help/programs/octave/tutorial/). Both are available in PDF form on Canvas. [↑](#footnote-ref-1)
2. Appropriate reaction: “Wow, cool! If I ever write a programming language, I’ll name it after my Advanced Calculus teacher!” [↑](#footnote-ref-2)
3. To see the documentation, click on “Documentation” tab at the bottom of Octave  and use the index option to search for linspace. [↑](#footnote-ref-3)
4. And then I still have to pause and do hand motions in the air to visualize where those entries are in the matrix. ☺ [↑](#footnote-ref-4)
5. 2\*384400e6\*sind(0.000000000000005) [↑](#footnote-ref-5)