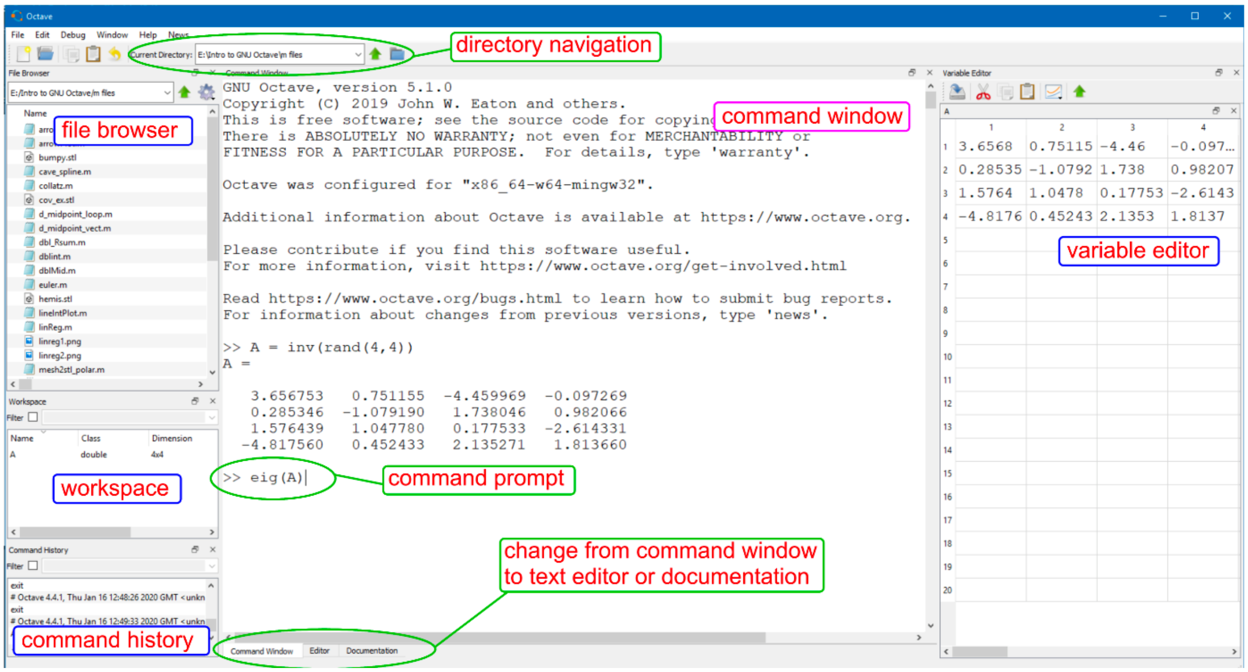
Tutorial 1: Quick Octave

Contents

- Octave is a fully functioning programming language, but it is not a general-purpose programming language (like C++, Java, or Python). Octave is numeric, not symbolic; it is not a computer algebra system (like Maple, Mathematica, or Sage).
- However, Octave is ideally suited to all types of numeric calculations and simulations. Matrices are the basic variable type, and the software is optimized for vectorized operations.
- Navigating the GUI:



- A simple calculation: $\frac{6}{2} + 3(7-4)^2$

```
octave:2> 6/2 + 3 * (7-4)^2
ans = 30
```

- Define a row vector

```
octave:3> u = [1 -4 6]
u =
    1   -4    6
```

- Define a column vector

```
octave:5> u = [1; -4; 6]
u =
    1
   -4
    6
```

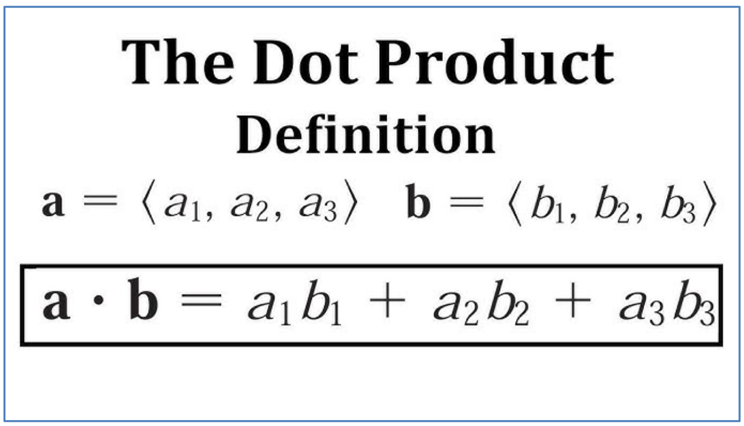
- Vector operations:
  - Define the following column vectors, if they are not already defined

```
octave:7> u = [1; -4; 6]
octave:8> v = [2; 1; -1]
```

  - Compute $2v + 3u$

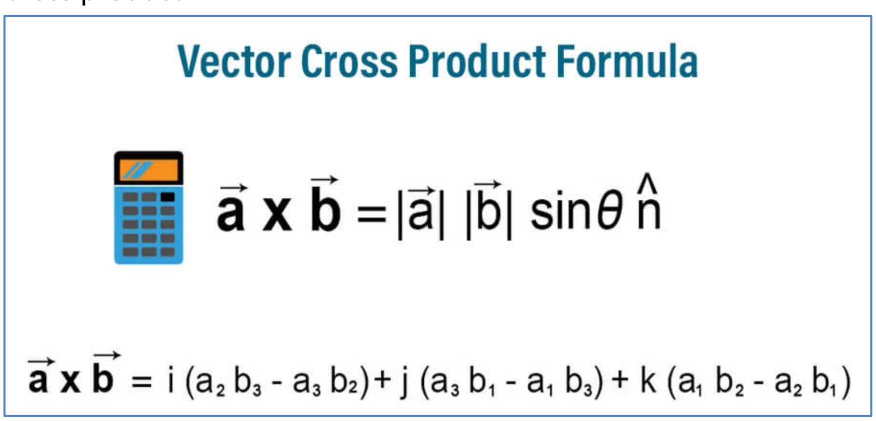
```
octave:9> 2*v + 3*u
ans =
    7
  -10
   16
```

  - Dot product of u and v




```
octave:10> dot(u, v)
ans = -8
```

  - Cross product




```
octave:11> cross(u, v)
ans =
   -2
   13
    9
```

- Is $cross(u, v)$ the same as $u * v$?
    - What about $u .* v$?

Matrix operations

- Define these matrices

$$\text{Let } A = \begin{bmatrix} 1 & 2 & -3 \\ 2 & 4 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -2 & -4 & 6 \\ 1 & -1 & 0 & 0 \end{bmatrix}.$$

- Compute $A \times B$

```
octave:14> A = [1, 2, -3; 2, 4, 0; 1, 1, 1]
octave:14> B = [1, 2, 3, 4; 0, -2, -4, 6; 1, -1, 0, 0]
octave:15> A * B
```

- Try $B * A$ and $A + B$
- To transpose a matrix, use the single quote

```
octave:19> B' * A
```

- Computing the determinant of matrix

```
octave:20> det(A)
ans = 6
```

- Computing the inverse of a matrix

```
octave:21> inv(A)
ans =
   0.6667  -0.8333   2.0000
  -0.3333   0.6667  -1.0000
  -0.3333   0.1667        0
```

- Matrix slicing

```
>> B = [1, 2, 3, 4; 0, -2, -4, 6; 1, -1, 0, 0]
>> B(3, 4)
>> B(3, 2)
>> B(1, :)
>> B(:, 2)
>> B(2:3, 3:4)
```

Plotting

- Plotting requires two sets of data: x-values and y-values.
- Plot the graph of the function $\sin(x)$ on the interval $[0, 2\pi]$

```
octave:28> x = linspace(0, 2*pi, 50);
octave:29> y = sin(x);
octave:30> plot(x, y);
```

- We can change the size of the line

```
octave:31> plot(x, y, 'linewidth', 2)
```

- To adjust the x-axis and y-axis locations and the font size

```
octave:33> set(gca, 'XAxisLocation', 'origin', 'YAxisLocation', 'origin')
octave:38> set(gca, 'fontsize', 14)
```

- We can control the limits of the plot using $axis([Xmin, Xmax, Ymin, Ymax])$ function

```
octave:37> axis([0, 2*pi, -1, 1])
```

- Improve the graph

```
octave:42> grid on
octave:43> set(gca, 'fontsize', 14)
octave:44> xlabel('x')
octave:45> ylabel('y')
octave:47> title('Sine graph')
octave:48> legend('y=sin(x)');
```

- We can plot the data as scatter plot

```
octave:50> x = [1, 2, 3, 4]
octave:51> y = [1, 2, 5, 4]
octave:52> plot(x, y, 'o')
```

- We can graph a line plot to the same plot using $hold\ on$, assume we want to graph $(x, 1.2x)$

```
octave:58> hold on
octave:60> plot(x, 1.2 *x, 'linewidth', 12)
```

- We can do the same thing in one $plot$ function

```
octave:8> x = [1, 2, 3, 4];
octave:9> y1 = [1, 2, 5, 4];
octave:10> y2 = 1.2 * x;
octave:11> plot(x, y1, 'o', x, y2, 'linewidth', 3);
octave:12> axis([0, 5, 0, 6]);
octave:13> grid on
octave:14> legend('data points', 'regression line')
octave:15> set(gca, 'fontsize', 14)
```

- Plot options

PLOT OPTIONS

MARKER	'+'	crosshair	COLOR	'k'	black
	'o'	circle		'r'	red
	'*'	star		'g'	green
	'.'	point		'b'	blue
	's'	square		'm'	magenta
	'^'	triangle		'c'	cyan
SIZE	'linewidth', n (n is some positive value)				
	'markersize', n				
LINE STYLE	'-'	solid line (default)			
	'--'	dashed line			
	':'	dotted line			

PLOT LABELS

HORIZONTAL AXIS LABEL	xlabel('axis name');
VERTICAL AXIS LABEL	ylabel('axis name');
LEGEND	legend('curve 1', 'curve 2', ...) ;
TITLE	title ('plot title ');

Single variable calculus

- You can define the limit $\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ as follows

```
>> f = @(n) (1 + 1./ n) .^ n ; % anonymous function
```

- We can solve the limit by trying values from 1 to 1,000,000,000

```
octave:23> f = @(n) (1+1./n).^n
octave:24> k = [0: 1: 9];
octave:25> n = 10.^k
octave:26> format long
octave:28> f(n)
```

- Plotting limacon: $r = 1 - 2\sin(\theta)$

```
octave:56> theta = linspace(0, 2*pi, 100);
octave:57> r = 1 - 2*sin(theta);
octave:58> x = r .* cos(theta);
octave:59> y = r .* sin(theta);
octave:60> plot(x, y);
octave:61> comet(x, y);
octave:63> polar(theta, r);
```

- You can plot a 3D figure with animation:

```
octave:52> t = 0: pi/20: 5*pi;
octave:54> comet3(cos(t), sin(t), t, 0.05);
```

Symbolic package

- An open-source library for performing symbolic mathematics that relies on python.
- To load the package

```
octave:65> setenv("PYTHON", "./venv/bin/python");
octave:66> pkg load symbolic
octave:67> syms x
```

- Example: Let $f(x) = x^3 + 3x^2 - 10x$
  - Evaluate $f\left(\frac{1}{2}\right)$
  - Factor the expression and find all real zeros

```
octave:68> f = x^3 + 3*x^2 - 10*x
octave:69> subs(f, 1/2)
octave:70> factor(f)
octave:71> solve(f == 0, x)
octave:72> subs(f, -5)
```

- Example: let $f(x) = x^2 \sin(x)$
  - Find $f`(x)$
  - Find $\int f(x)\, dx$
  - Find $\int_0^{\pi/4} f(x)\, dx$

```
octave:80> f = x^2 * sin(x)
octave:81> dffx = diff(f, x)
octave:82> int(dffx, x)
octave:83> int(dffx, x, 0, pi/4)
```

Taylor and Maclaurin Series

- **Taylor series**: a representation of a function as an infinite sum of terms, where each term is calculated from the function's derivatives at a specific point a.
  - It is used to approximate a function near that point.
- The Taylor series of a function $f(x)$ centered at a is given by:

$$\boxed{6}\quad f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

$$= f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

  - $f^{(n)}(a)$: the n-th derivative of f at a
  - $n!$: factorial of n
  - $x - a$: the distance from the center point a

- **Maclaurin series**: a special case of the Taylor series where the expansion is centered at $a = 0$

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n = f(0) + \frac{f'(0)}{1!} x + \frac{f''(0)}{2!} x^2 + \cdots \qquad (7)$$

- Compute Taylor series over the function e^x centered around $a = 5$ of order 6

```
octave:3> setenv("PYTHON", "./venv/bin/python")
octave:4> pkg load symbolic
octave:5> syms x
octave:6> f = exp(x)
octave:7> t = taylor(f, x, 5, 'Order', 6)
```

- Use Octave to approximate $\cos(x)$ using Maclaurin series. Plot $\cos(x)$ and the series.

```
% Maclaurin Series approximation for cos(x)
clc; clear;

% Define the x range
x = linspace(-2*pi, 2*pi, 1000);

y_true = cos(x);

% Degree of the Maclaurin polynomial
N = 5;

% Initialize approximation
y_approx = zeros(size(x));

% Maclaurin series terms up to N
for n = 0:N
    term = ((-1)^n) * (x.^(2*n)) / factorial(2*n);
    y_approx += term;
end

plot(x, y_true, 'b', 'LineWidth', 2);
hold on;
plot(x, y_approx, 'r--', 'LineWidth', 2);
legend('cos(x)', 'Maclaurin approx');
title('Maclaurin Series Approximation of cos(x)');
xlabel('x');
ylabel('y');
grid on;
```

TASK

- Implement the Maclaurin series to approximate $\sin(x)$ and plot the approximation and $\sin(x)$
- Use $comet3$ function in Octave to plot $x = t * \sin(t)$, $y = t * \cos(t)$, $z = t$, where $t = [0, 20]$, assuming t is increasing by 0.1