SOME USEFUL AND NOT SO USEFUL EQUATIONS

Order of Operations	x = 1/(2+1)=0.333 vs x=1/2+1=1.5
Format	format long g for 15 significant digits
Variable Types	double = 8 bytes , char = 2 bytes
Bits to Bytes	8 bits = 1 byte
Base10 to Base2	bin2dec(), dec2bin()
Double to ASCII	char(), double()
Double to Char	num2str(), str2num()
Inputting Vectors	x = [1 2 3 4]
Inputting Matrices	x = [1 2; 3 4]
Transpose	x = x’
start:increment:end	x = 1:2:10 yields [1 3 5 7 9]
linspace(start,end,numpts)	x = linspace(1,10,5) yields [1 3.25 5.5 7.75 10]
Length Command	length(x) will compute the length of a vector
Matrices of Zeros and Ones	zeros(row,col), ones(row,col)
Dot Operator	u = [1 2 3]; v = [1 2 3]; u.*v = [1 4 9]
Matrix multiplication	u = [1 2 3]; v = [1;2;3]; u*v = 14
Reference Elements of an Array	u = [3 8 9]; u(3) is 9
Reference Elements of a Matrix	A = [3 8 ;9 10]; A(2,1) is 9
Reference Entire Row	A = [3 8 ;9 10]; A(:,1) is the first column
Evaluating Functions	x = linspace(-pi,pi,100); y = sin(x) + 5;
Function Headers	function [out1,out2] = name_of_function(in1,in2)
If/Else/End	if *statement* *execute this block of code if true* else *execute this block of code if false* end
Logical Operators	==, >=, <=, >, <, ∼=
Summation with a For loop	I = 0; for idx = 1:2:10 I = I + idx; end
Summation with a While Loop	I = 0; idx = 1; while idx <= 10 I = I+idx; idx = idx + 2; end
Plotting	x = linspace(-pi,pi,100); y=sin(x); plot(x,y)
Meshing	x = linspace(-pi,pi,100);y = linspace(-pi,pi,100); [xg,yg] = meshgrid(x,y);zg = sin(xg.*yg); mesh(xg,yg,zg)
Parts of the Computers	CPU,Motherboard,GPU,RAM,HDD,I/O
Absolute Error	$\vert actual - computed\vert$
Percent Error	$(AbsoluteError)/actual$
Base 10 to Binary	$15_{10} = 8 + 4 + 2 + 1 = 2^3 + 2^2 + 2^1 + 2^0 = 1111_2$
Binary to Base10	$110_2 = 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = 4 + 2 = 6_{10}$
Capital Letters to ASCII	$A = 65_{10}, B = 66_{10}, etc$
Lower Letters to ASCII	$a = 97_{10}, b = 98_{10}, etc$
Examples of ASCII	$cat = [99_{10}, 97_{10}, 116_{10}] = 1100011_2, 1100001_2, 1110100_2$

Taylor Series	$f(t_{i+1}) \approx f(t_i) + f'(t_i)\Delta t + f''(t_i)\frac{\Delta t^2}{2!} + ... + f^{(N)}(t_i)\frac{\Delta t^N}{N!}$
Bi-Section Method	Define upper(xU) and lower(xL) bounds $x_{new} = (xU + xL)/2$ $if\ f(x_{new}) > 0, x_U = x_{new}\ else\ x_L = x_{new}$
Newton-Raphson	$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
Newton-Raphson (Optimization)	$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$
First-Order Forward Differencing	$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{\Delta x}$
First-Order Backward Differencing	$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{\Delta x}$
First-Order Center Differencing	$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x}$
First-Order Second Derivative	$f''(x_i) \approx \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{\Delta x^2}$
Secant Method	Combine Newton-Raphson with First-Order Derivative
Reimmann Sum	$x(t) \approx x_0 + \sum\limits_{i=1}^{N} v(t_i)\Delta t$
Trapezoidal Rule	$I \approx \sum\limits_{i=1}^{N} \frac{1}{2}(f(t_i) + f(t_i + \Delta t))\Delta t$
Euler's Method	$f_2 \approx f_1 + \dot{f}_1\Delta t$
Runge-Kutta-4	$y_{k+1} = y_k + \phi\Delta t \quad \phi = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ $k_1 = f(t_k, y_k) \quad k_2 = f(t_k + \Delta t/2$ $y_k + k_1\Delta t/2) \quad k_3 = f(t_k + \Delta t/2, y_k + k_2\Delta t/2)$ $k_4 = f(t_k + \Delta t, y_k + k_3\Delta t) \quad \dot{y} = f(t, y)$
Determinant	if A = [a b;c d], det(A) = a*d-b*c
Inverse	$[A\|I]$ perform rref to yield $[I\|A^{-1}]$
Solve for Eigenvalues	det(sI-A) = 0
Solve for Eigenvectors	$(sI - A)\vec{v} = 0$
Eigenvalue Decomposition	$A = V\Lambda V^{-1}$
Solution to ODEs	$\vec{x}(t) = a_1\vec{v}_1e^{\lambda_1 t} + \ldots + a_N\vec{v}_Ne^{\lambda_N t} = \sum\limits_{n=1}^{N} a_n\vec{v}_ne^{\lambda_n t}$
Alternate Solution to ODEs	$\vec{x}(t) = e^{At}\vec{x}_0$
Heat Equation	$\frac{d^2T}{dx^2} - h'(T - T_a) = 0$
FDM (Heat Equation)	$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_a$
2-D Heat Equation	$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$
FDM (2-D Heat Eq)	$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} - 4T_{i,j} + T_{i,j-1} = 0$
Time Dependent Heat Eq	$k\frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$
FDM (Heat Eq Time)	$T_i^{l+1} = T_i^l + \lambda(T_{i+1}^l - 2T_i^l + T_{i-1}^l)$
Crank-Nicolson	$-\lambda T_{i-1}^{l+1} + 2(1+\lambda)T_i^{l+1} - \lambda T_{i+1}^{l+1} = \lambda T_{i-1}^l + 2(1-\lambda)T_i^l + \lambda T_{i+1}^l$
λ	$\frac{k\Delta t}{\Delta x^2}$
Stiffness for a Bar Element	$\left[K_{(e)}\right] = k\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
Stiffness for a Bar	$[K] = \sum\limits_{e=1}^{n} \left[K_{(e)}\right]$
Bar Equation	$[K]\vec{d} = \vec{F}$
FEA Heated Rods	$\frac{1}{x_2 - x_1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix} = \begin{Bmatrix} -T_1' \\ T_2' \end{Bmatrix} + \begin{Bmatrix} \int\limits_{x_1}^{x_2} f(x)N_1dx \\ \int\limits_{x_1}^{x_2} f(x)N_2dx \end{Bmatrix}$