

SUB-FUNCTIONS

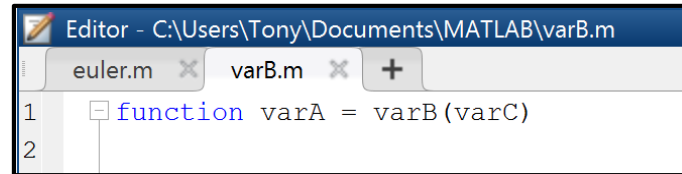
Presented by Tony Vo

Slides by Tony Vo



RECALL: HOW TO CREATE A FUNCTION FILE

- Start a new m-file (script file)
 - Declare it as a function with a function header
- The function header declaration
`function outputs = function_name(inputs)`



- Function name and file name must be the same
 - The `varB` function must be saved as `varB.m`

- An m-file can contain multiple functions
 - This is achieved using multiple function header declarations in the file
 - The parent function is the one with the same name as the filename
 - The other functions are known as sub-functions

- Sub-functions can only be seen and called by the parent function
 - This helps with information hiding
 - Will not overwrite built-in functions
 - Don't need to create separate function files for related/complementary functions

SIMPLE SUBFUNCTION EXAMPLE

- `EnergyDifference()` calculates the difference between the gravitational potential energy (mgh) and kinetic energy ($0.5mv^2$)

```
function ED = EnergyDifference(m, h, v)
% Calculates difference between Potential energy and Kinetic Energy of an object
ED = PotentialEnergy(m,h) - KineticEnergy(m, v);
end

% Potential Energy Subfunction
% This is private to EnergyDifference.m    You cannot see or call this function elsewhere
function E = PotentialEnergy(m, h)
g = 9.81;
E = m * g * h;
end

% KineticEnergy Subfunction
function E = KineticEnergy(m, v)
E = 0.5 * m * v^2;
end
```

SUB-FUNCTION OVERWRITE

- In many engineering calculations, such as electrical power calculations, values are squared before taking their mean
 - If we write a function that subtracts a vector by its squared mean, it makes sense to use $x - \text{mean}(x)$
 - We can overwrite MATLAB's built-in `mean()` function temporarily using our own `mean()` sub-function

```
function x_minus_mean = SubtractMean(x)
% Subtracts mean (DC signal) from x
x_minus_mean = x - mean(x); % <---- using the mean subfunction
end
```

```
function res = mean(x)
% My own mean() subfunction
% This will overwrite MATLAB's built-in mean()
% within this file ONLY
res = sum(x).^2 / length(x);
end
```

NESTED FUNCTIONS

- Functions can be nested inside other functions
 - A nested function can access all the variables in its parent function
 - Only the parent function can see and call its nested function
 - This promotes information hiding and reuse

```
function f = parabolaGenerator(a, b, c)
% Generates a parabola function and returns its handle
% a,b,c = coefficients of the equation a*x^2 + b*x + c
% f is a function handle pointing to the parabola

    % Nested function to create parabola
    function y = p(x)
        y = a*x.^2 + b*x + c;
    end
f = @p;    % Returns handle to parabola function
end
```

USE A SUB-FUNCTION INSTEAD?

```
function f = parabolaGenerator2(a, b, c)
% Generates a parabola function and returns its handle
% a,b,c = coefficients of the equation a*x^2 + b*x + c
% f is a function handle pointing to the parabola

% Will return an invalid function handle as
% a, b and c are undefined in the p(x) subfunction

f = @p;    % Returns handle to parabola function
end

% Subfunction function to create parabola
function y = p(x)
    y = a*x.^2 + b*x + c;
end
```

```
>> f = parabolaGenerator2(2, 4, 8)
f =
    @p
```

```
>> x=-4:0.1:4;
>> plot(x,f(x))
```

Undefined function or variable 'a'.

Error in parabolaGenerator2>p (line 14)
y = a*x.^2 + b*x + c;

- Sub-functions and nested functions
- These advanced functions can be used to temporarily overwrite MATLAB built-in functions and hide information
- Can a parent function call upon a function nested inside one of its sub-functions (i.e. sub-function of a sub-function)?