THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

ENG 101
User-Defined Functions

Dr. Emil Jovanov
Electrical and Computer Engineering
University of Alabama in Huntsville
emil.jovanov@uah.edu
http://www.ece.uah.edu/~jovanov

MATLAB functions

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

- function [y1,...,yN] = myfun(x1,...,xM)
- declares a function named myfun that accepts inputs
  x1,...,xM
  and returns outputs
  y1,...,yN
- This declaration statement must be the first executable line of the function. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores.

MATLAB functions

- In a function file which contains only function definitions.
- The name of the file should match the name of the first function in the file.
- Functions must be at the end of the file. Script files cannot have the same name as a function in the file.
- can include multiple local functions or nested functions
- For readability, use the end keyword to indicate the end of each function in a file. The end keyword is required when:
  - Any function in the file contains a nested function.
  - The function is a local function within a function file, and any local function in the file uses the end keyword.
  - The function is a local function within a script file.

Example

```
function C=FtoC(F)
%FtoC converts degrees F to degrees C
C=5*(F-32)./9;
```

Function definition line.

Assignment to output argument.

Function example

Define a function in a file named stat.m that returns the mean and standard deviation of an input vector.

```
function [m,s] = stat(x)
        n = length(x);
        m = sum(x)/n;
        s = sqrt(sum((x-m).^2/n));
end
```

Call the function from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];
[ave,stdev] = stat(values)
ave =
  47.3400
stdev =
  29.4124
```

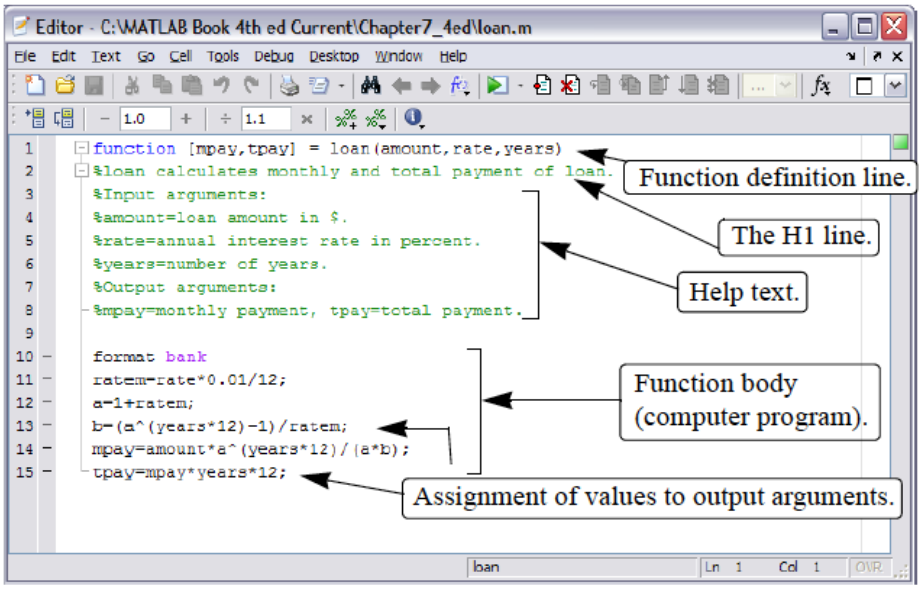

Matlab functions

```
% Compute the value of the integrand at 2*pi/3.
x = 2*pi/3;
y = myIntegrand(x)

% Compute the area under the curve from 0 to pi.
xmin = 0;
xmax = pi;
f = @myIntegrand;
a = integral(f,xmin,xmax)

function y = myIntegrand(x)
y = sin(x).^3;
end
```

Loan function

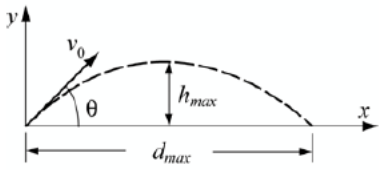


Help

- Help line
- Document your code

Sample Problem 7-6: Motion of a projectile

Create a function file that calculates the trajectory of a projectile. The inputs to the function are the initial velocity and the angle at which the projectile is fired. The outputs from the function are the maximum height and distance. In addition, the function generates a plot of the trajectory. Use the function to calculate the trajectory of a projectile that is fired at a velocity of 230 m/s at an angle of 39°.



The motion of a projectile can be analyzed by considering the horizontal and vertical components. The initial velocity v_0 can be resolved into horizontal and vertical components

$$v_{0x} = v_0\cos(\theta) \quad \text{and} \quad v_{0y} = v_0\sin(\theta)$$

In the vertical direction the velocity and position of the projectile are given by:

$$v_y = v_{0y} - gt \quad \text{and} \quad y = v_{0y}t - \frac{1}{2}gt^2$$

The time it takes the projectile to reach the highest point ($v_y = 0$) and the corresponding height are given by:

$$t_{hmax} = \frac{v_{0y}}{g} \quad \text{and} \quad h_{max} = \frac{v_{0y}^2}{2g}$$

The total flying time is twice the time it takes the projectile to reach the highest point, $t_{tot} = 2t_{hmax}$. In the horizontal direction the velocity is constant, and the position of the projectile is given by:

$$x = v_{0x}t$$

```
function [hmax,dmax]=trajectory(v0,theta)
% trajectory calculates the max height and distance of a
projectile, and makes a plot of the trajectory.
% Input arguments are:
% v0: initial velocity in (m/s).
% theta: angle in degrees.
% Output arguments are:
% hmax: maximum height in (m).
% dmax: maximum distance in (m).
% The function creates also a plot of the trajectory.
g=9.81;
v0x=v0*cos(theta*pi/180);
v0y=v0*sin(theta*pi/180);
thmax=v0y/g;
hmax=v0y^2/(2*g);
ttot=2*thmax;
dmax=v0x*ttot;
% Creating a trajectory plot
tplot=linspace(0,ttot,200);
x=v0x*tplot;
y=v0y*tplot-0.5*g*tplot.^2;
plot(x,y)
xlabel('DISTANCE (m)')
ylabel('HEIGHT (m)')
title('PROJECTILE''S TRAJECTORY')
```

Function definition line.

Creating a time vector with 200 elements.

Calculating the x and y coordinates of the projectile at each time.

Note the element-by-element multiplication.

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Function example

Write a user-defined MATLAB function for the following math function:

$$y(x) = -0.2x^4 + e^{-0.5x}x^3 + 7x^2$$

The input to the function is x and the output is y.
Write the function such that x can be a vector
(use element-by-element operations).

(a) Use the function to calculate y (–2.5), and y (3).

(b) Use the function to make a plot of the function for .

feval

The feval (short for "function evaluate") command evaluates the value of a function for a given value (or values) of the function's argument (or arguments). The format of the command is:

```
variable = feval('function name', argument value)
```

The value that is determined by feval can be assigned to a variable, or if the command is typed without an assignment, MATLAB displays the value of the function.

- The function name is typed as string.
- The function can be a built-in or a user-defined function.
- If there is more than one input argument, the arguments are separated with commas.
- If there is more than one output argument, the variables on the left-hand side of the assignment operator are typed inside brackets and separated with commas.

feval

feval Execute the specified function.

feval(F,x1,...,xn) evaluates the function specified by a function handle or function name, F, at the given arguments, x1,...,xn.

For example, if F = @foo,

feval(F,9.64) is the same as foo(9.64).

If a function handle is bound to an overloaded function, then the data type of the arguments x1 through xn, determines which function is executed.

feval is usually used inside functions which take function handles or function strings as arguments. Examples include FZERO and EZPLOT.

[y1,..,yn] = feval(F,x1,...,xn) returns multiple output arguments.