User-Defined Functions in Matlab

Jake Blanchard
University of Wisconsin - Madison

Spring 2008

Matlab Functions

- Matlab permits us to create our own functions
- These are scripts that take in certain inputs and return a value or set of values
- We will need these as we use built-in functions for problem solving

Format of Function Declaration

function [output arguments] =function_name(input arguments)

User-Defined Functions

- Suppose we want to plot:

sin(3*x)+sin(3.1*x)

- Create user-defined function

```
function r=f(x)
  r=sin(3*x)+sin(3.1*x)
```

- Save as f.m

User-Defined Functions (cont)

- Now just call it:

```
x=0:0.1:50;
y=f(x);
plot(x,y)
```

The Matlab Path

- Matlab looks in the current path for functions (m-files)
- The path is shown near the top of the command window

Practice

- Create an m-file that calculates the function g(x)=cos(x)+cos(1.1*x)
- Use it to plot g(x) from x=0 to 100
- Note: previous function was

```
function r=f(x)
  r=sin(3*x)+sin(3.1*x)
```

- …and plot commands were

```
x=0:0.1:50;
y=f(x);
plot(x,y)
```

Practice

- Create an m-file that calculates the function $g(x, \delta)=\cos(x)+\cos((1+\delta)x)$ for a given value of δ
- Use it to plot $g(x, \delta)$ from $x=0$ to 100 for $\delta=0.1$ and 0.2

Flow Control

```
if x<10 then
    x=x+1
else
    x=x^2
end
```

Flow Control (cont)

```
for i=1:10
   z=z*i
end
```

Flow Control (cont)

```
A=0
sum=0
while A < 10,
    sum=sum+A;
    A=A+1;
end
```

Practice

- On the next slide is a Matlab function that calculates the sum of cubes of the first N integers
- Download **sumofcubes.m** and answer the following questions:
    - What is the result for N=20?
    - Modify the script to do the same calculation with a "while" loop.

Practice Script

```
function r=sumofcubes(N)
ans=0;
for i=1:N
    ans=ans+i^3;
end
r=ans;
```

Practice

- Now modify this script to add up the cubes of the even integers.
- Note that mod(i,2)=0 when i is an even number

Inline Functions

- One downside to Matlab functions in m-files is the proliferation of files resulting from having each function in it's own file
- For simple functions, this can be avoided with an inline function

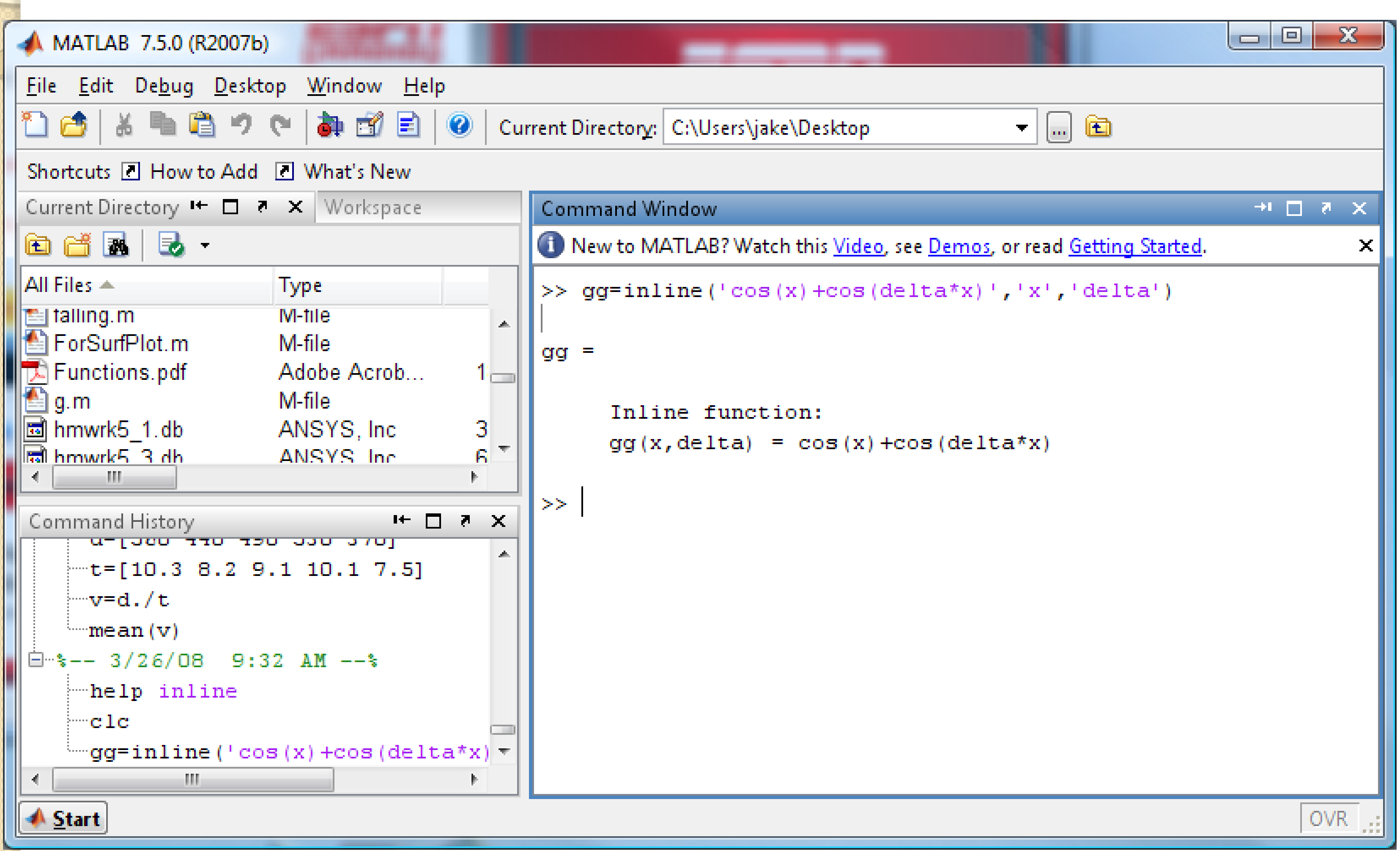
Example

```
g=inline('cos(x)+cos(1.1*x)')
x=0:0.01:100;
y=g(x);
plot(x,y)
```

Parameters

```
x=0:0.01:100;
gg=inline('cos(x)+cos(delta*x)','x','delta')
delta=1.05
y=gg(x,delta);
plot(x,y)
```

Command Window Shows Form of Function


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\jake\Desktop
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
Command History
Start
OVR

All Files	Type	
falling.m	M-file	
ForSurfPlot.m	M-file	
Functions.pdf	Adobe Acrob...	1
g.m	M-file	
hmwrk5_1.db	ANSYS, Inc	3
hmwrk5_3.db	ANSYS, Inc	6

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> gg=inline('cos(x)+cos(delta*x)','x','delta')

gg =

     Inline function:
     gg(x,delta) = cos(x)+cos(delta*x)

>> 
```

```
d=[360 440 490 530 570]
t=[10.3 8.2 9.1 10.1 7.5]
v=d./t
mean(v)
%-- 3/26/08  9:32 AM --%
help inline
clc
gg=inline('cos(x)+cos(delta*x)
```

An Alternative Form (Anonymous Functions)

```
x=0:0.01:100;
delta=1.05
gg=@(x, delta) cos(x)+cos(delta*x)
y=gg(x, delta);
plot(x,y)
```

Practice

- Consider the function

$$\boxed{f(x)=\exp(-a*x)*\sin(x)}$$

- Plot using an inline function
- Use 0<x<10 and a=0.25
- Note: syntax can be taken from:
- **gg=inline('cos(x)+cos(delta*x)','x','delta')**
- **gg=@(x, delta) cos(x)+cos(delta*x)**

Subfunctions

- Subfunctions allow us to put two functions in one file.
- The second function will not be available to other functions.
- I will use this to consolidate my files into one.

Example (save as example.m)

```
function example
clear all
r=sumofcubes(20);
fprintf('The sum of the first 20 cubes is %i\n',r)
%
function r=sumofcubes(N)
ans=0;
for i=1:N
    ans=ans+i^3;
end
r=ans;
```

Comments in Scripts

- Note that the % sign represents a comment
- Everything on a line after that will be ignored
- Comments can be on their own line or at end of a line of working code, eg.
- **y=gg(x); %function defined inline above**

An Example – with Numerics

- Suppose we're looking for a $100k, 30-year mortgage. What interest rate do I need to keep the payments below $700 per month?

$$\boxed{100000-700\left[\frac{(1+i)^{360}-1}{i(1+i)^{360}}\right]=0}$$

- Solve for *i*

Approach

- Create user-defined function
- Plot the function
- Find point where function is zero

Create the function

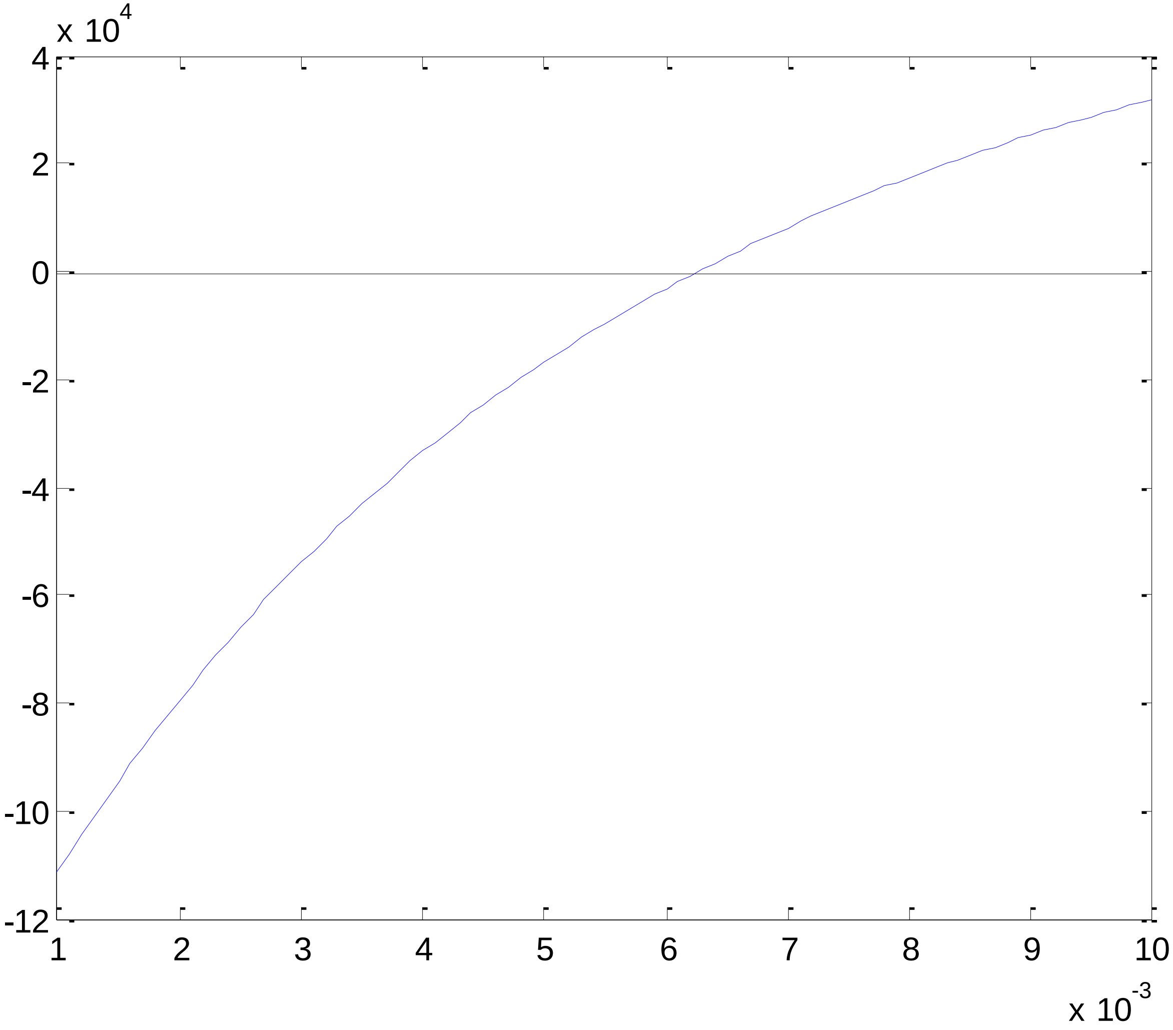
```
function s=f(i)
p=100000;
n=360;
a=700;
s=p-a*((1+i).^n-1)./(i.*(1+i).^n);
```

Plot the Function

- First save file as f.m
- Now enter the following:

```
i=0.001:0.0001:0.01;
y=f(i);
plot(i,y)
```

The Plot



Result

- Zero-crossing is around i=0.006
- Annual interest rate is 12*i, or about 7%
- Try more accurate solution

```
12*fzero('f',0.006)
```

- This gives about 7.5%

Debugging Scripts

Cells in Scripts

- Defining
- Documenting
- Incrementing Variables

Publishing Scripts

- Demo

Questions?