

# The GFDL In-House MATLAB Course

## Session 1

Remik Ziemlinski  
Phone: +1 - 609 - 452 - 6500 ext. 6977  
Fax: +1 - 609 - 987 - 5063  
Email: [Remik.Ziemlinski@noaa.gov](mailto:Remik.Ziemlinski@noaa.gov)

National Oceanic and Atmospheric Administration  
Geophysical Fluid Dynamics Laboratory  
Princeton, NJ 08542  
<http://www.gfdl.noaa.gov>



**Raytheon**



# Course Material

1. Point web browser to instructor slides at [www/~rsz/mc1](http://www/~rsz/mc1)
2. Copy course material and start MATLAB.

```
/home/$USER> cp -r ~rsz/mc1 ~/mc1
```

```
/home/$USER> cd mc1
```

```
/home/$USER/mc1> matlab &
```

```
>> mc1
```

# Assumptions About Audience

- Already know how to program
- Familiar with an analysis package, i.e. Ferret, Grads, IDL
- Heard or saw what MATLAB can do, but not sure how to get started
- Seasoned user looking to learn a few more tricks

# Goals

- Learn to use MATLAB for rapid model prototyping and analysis.
- Reference these examples & build upon them.
- Learn to use the Help system extensively.

# Table of Contents

- I. Overview
  - 1. Architecture
  - 2. Licenses
- II. Environment Basics
  - 1. GUI
  - 2. Help System
- III. Programming Basics
  - 1. Datatypes & Operators
  - 2. Scripts & Functions
  - 3. File Search Path
  - 4. Built-in Editor & Debugger
- IV. Intro. to MATLAB Graphics
  - 1. Point & Line Plots
  - 2. Maps
  - 3. Labeling
  - 4. Exporting
- V. Customizing Your Environment I

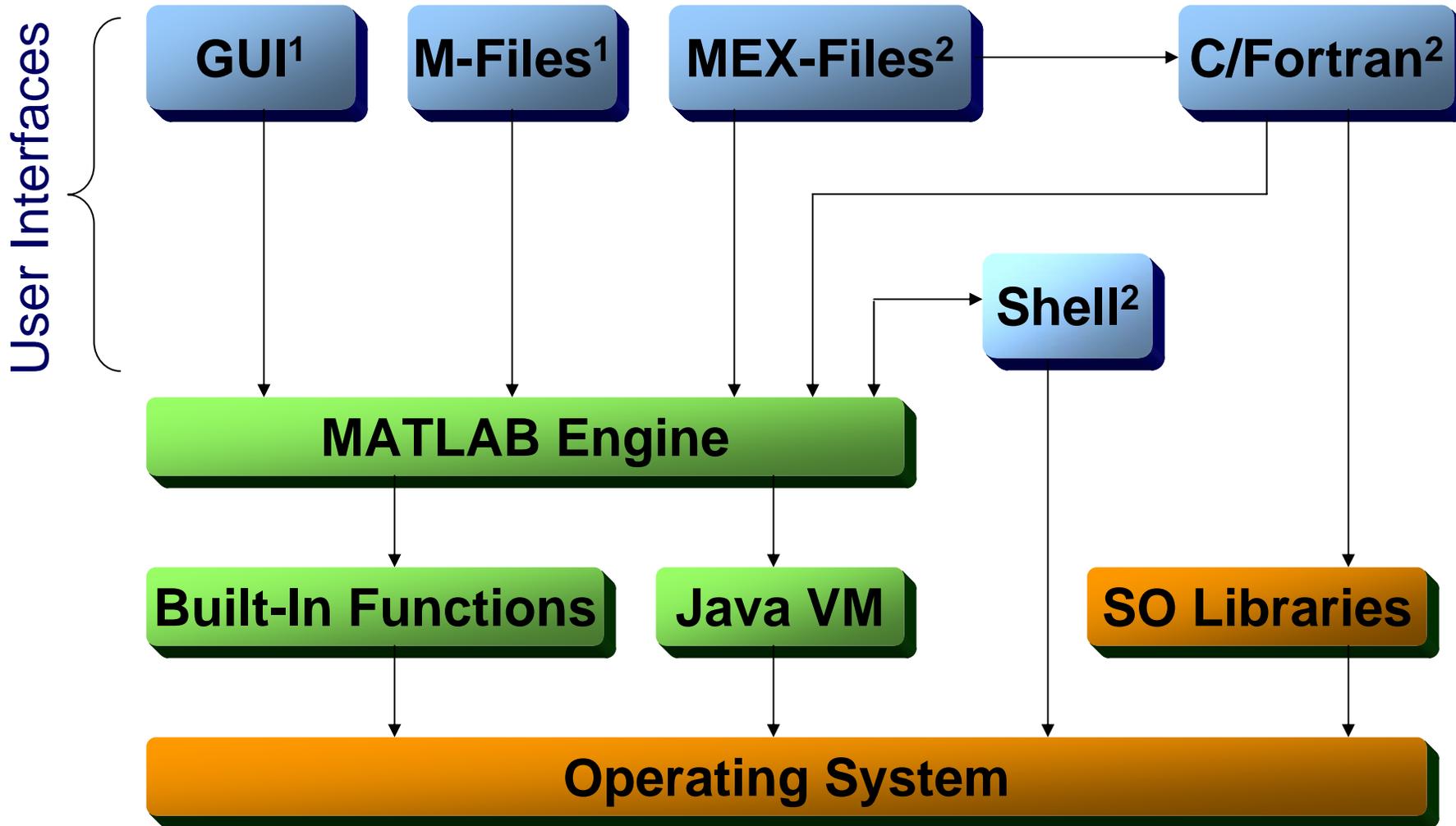
# I. Overview

- MATLAB: “Matrix Lab”, originally written in Fortran
- Like UNIX, there’s more than 1 way to do something
- Pros
  - Large library of functions for linear algebra, statistics, DE, graphics, ...
  - Both scriptable and has GUI
  - Extendable with C or Fortran
  - Large community and help system
- Cons
  - Slow loop execution in scripts



# Architecture

Raytheon



Topics covered are indicated with super-scripted session #



# Licenses

- Limited number (18 for Linux+HPCS at GFDL)
- 1 License used per User-Machine pair
- Check availability in shell window:  
`/home/rsz> lmgfdl`

```
Users of MATLAB: (Total of 18 licenses available)
  rsz anc2      start Wed 3/3 10:34
Users of MAP_Toolbox: (Total of 2 licenses available)
  rsz anc2      start Wed 3/3 10:39
Users of Optimization_Toolbox: (Total of 1 license available)
Users of Signal_Toolbox: (Total of 3 licenses available)
Users of Spline_Toolbox: (Total of 1 license available)
Users of Statistics_Toolbox: (Total of 1 license available)
  rsz anc2      start Tue 3/2 11:03
```

# II. Environment Basics - GUI

The screenshot displays the MATLAB GUI with three main panels:

- Workspace:** A table showing current variables in the workspace.
- Command Window:** A text area for entering and executing MATLAB commands.
- Command History:** A list of previously executed commands.

Name	Size	Bytes	Class
ans	1x51	102	char array
data2	1x134217728	107374182	double array

```
>> isunix
ans =
     1

>> ispc
ans =
     0

>> beep
>> beep
>> beep
>> lookfor cfd
cfd not found.
>> ver

-----
MATLAB Version 6.5.0.180913a (R13)
MATLAB License Number: 10562
Operating System: Linux 2.4.18-xfs-1.1 #2 SMP Wed Feb 12 05:45:46 EST 2003 i686
Java VM Version: Java 1.3.1 with Blackdown Java-Linux Team Java HotSpot(TM) Server VM
-----
MATLAB                               Version 6.5           (R13)
Mapping Toolbox                       Version 1.3           (R13)
Optimization Toolbox                 Version 2.2           (R13)
Signal Processing Toolbox             Version 6.0           (R13)
Spline Toolbox                       Version 3.1.1         (R13)
Statistics Toolbox                   Version 4.0           (R13)
>> pwd

ans =

/home/rsz/projects/seminars/matlab/apr2004/examples

>>
```

Command History:

```
help lookfor
help apropos
apropos
help
help matlab/general
help save
doc
helpbrowser
computer
is unix
isunix
ispc
beep
lookfor cfd
ver
pwd
```

**Variables**

**Command Prompt**

**Command History**

# GUI cont.

Try some commands:

```
>> ver
```

```
>> pwd
```

Double-click a command in History window

Double-click “ans” variable in Workspace window

# Help System

- Command Prompt Help

lookfor acts like UNIX `apropos`

```
>>lookfor eigen
```

help [command | topic] displays full help text

```
>> help svd
```

```
>> help
```

```
>> help matlab/ops
```

- GUI Help

Online help with graphics, index and search

```
>> helpdesk
```

same as

```
>> helpbrowser
```

# III. Programming Basics

## Highlights

- Interpreted code, can be compiled\*
- Global variables supported
- Fully featured general-purpose language w/o pointers
- Syntax similar to C & Fortran
- Case sensitive
- Relies on a search path, known as PATH.

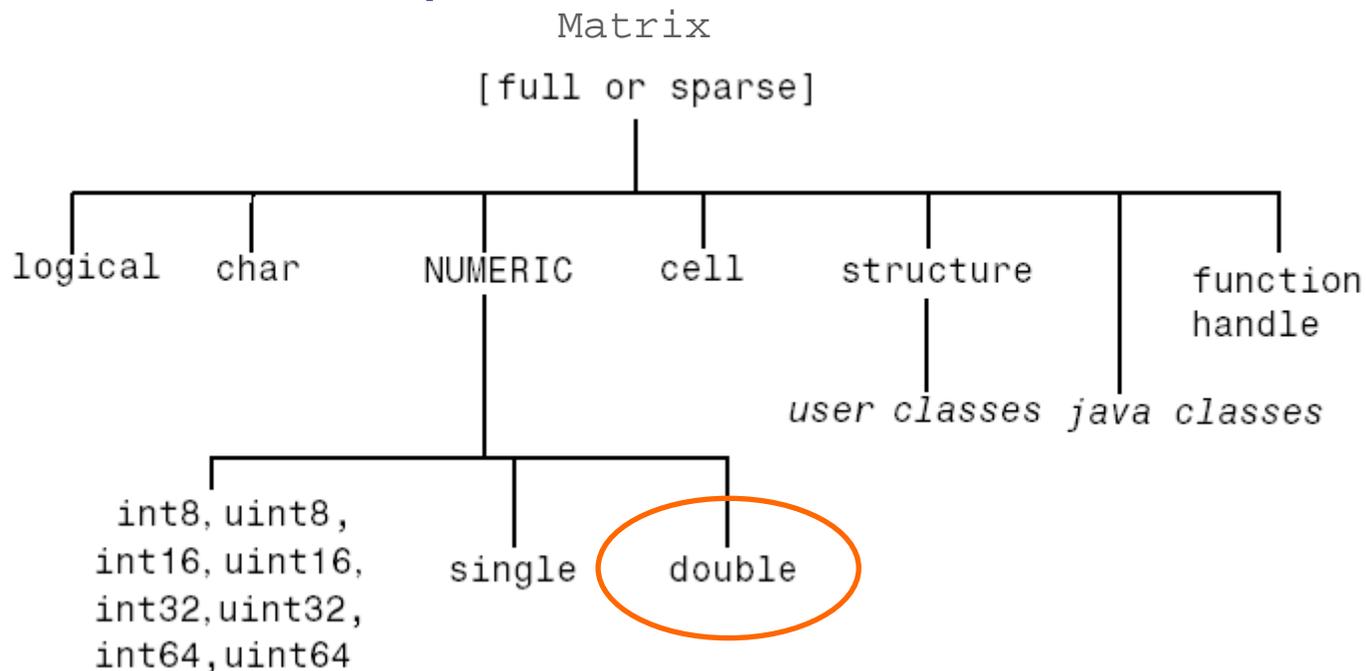
\*Can compile into MEX,  
Compile into C code with Compiler Toolbox (\$\$\$)



# Datatypes

Everything is a matrix!

All computations use double, but data can be casted to save space.



# Datatypes – Examples

```
% keywords for IEEE numbers
```

```
>> Inf < -Inf
```

```
% complex number, ';' suppresses display of answer
```

```
>> n = 2 + 3i;
```

```
>> real(n), imag(n)
```

```
>> n + NaN
```

```
% string concatenation, '...' multi-line operator
```

```
>> s1 = 'lab'; ['mat', ...  
              s1]
```

```
% array of strings
```

```
>> s = {'time', 'lat', 'qc'};
```

```
>> s{3}
```

# Datatypes – Examples cont.

```
% simplest matrix, acts like NULL pointer
>> empty = [];
>> isempty( empty )

% identity matrix
>> id1 = [1 0; 0 1];
>> id2 = eye(3);
>> id1 == id2           % produces error

% modify column 3, then do find & replace
>> id2(:,3) = 2
>> id2( find(id2 > 1) ) = -1

% make 3D, then remove singleton dimension
>> id3(1, :, :) = id1;
>> squeeze( id3 )
```

# Scripts & Functions (M-files)

Scripts execute in scope of workspace.  
File name prefix is arbitrary.

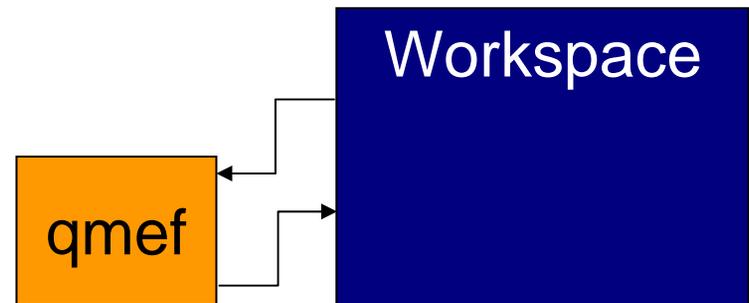
```
>> qme  
>> edit qme
```



Functions have private scope.

Must have same name as file prefix, otherwise MATLAB won't find it (in this case qmef.m).

```
>> clear all;  
>> [c, u] = qmef;  
>> c, u  
>> edit qmef  
>> help qmef  
>> lookfor qme  
>> which qmef
```



# Scripts & Functions cont.

Functions have 2 forms, inline or as a file.

- Inline

```
% auto-parsed by MATLAB to find arguments
```

```
>> f = inline('sin(theta) + x');
```

```
>> f(pi/2, 0)
```

```
% same as inline('sin(theta) + x', 'theta', 'x');
```

Written in a M-file f.m

```
>> edit f
```

```
function [y] = f(theta, x)
```

```
    y = sin(theta) + x;
```

```
return
```

# File Search Path

MATLAB will only see & execute files found in the PATH variable (files include M-files, data files, ...).

Modify it in 2 ways:

## 1. Environment Variable

```
set MATLABPATH = ~/pp/matlab #.cshrc
```

## 2. Inside MATLAB

```
>> addpath( '/home/rsz/pp/matlab' );
```

Use genpath for recursive path creation.



# Built-In Editor & Debugger

- Supports syntax highlighting & smart indenting

```
>> edit qmef
```

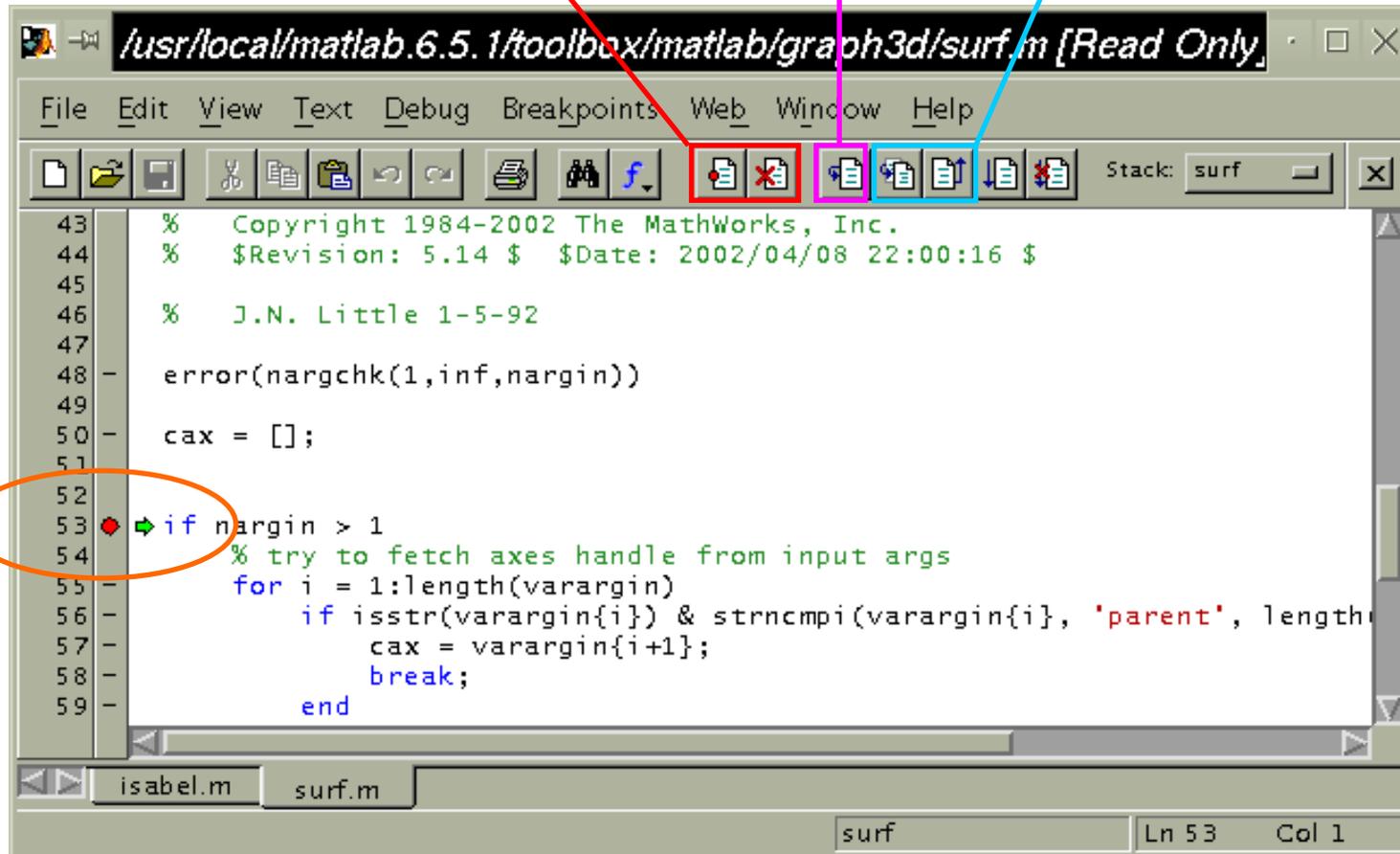
- Debugging features
  1. Stepping line-by-line
  2. Diving in/out of functions & scripts
  3. Breakpoints

# Built-In Editor & Debugger cont.

Breakpoints

Stepping

Diving



# IV. Intro. to MATLAB Graphics

- All graphics are displayed in a figure
- Properties can be changed with get & set or with the GUI

```
>> figure
```

```
>> get( gcf )
```

```
% what color is the background? make it white
```

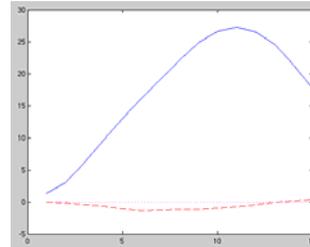
```
>> set(gcf, 'Color', [1 1 1])
```

```
% now with the GUI (Edit | Figure Properties)
```

# Point & Line Plots

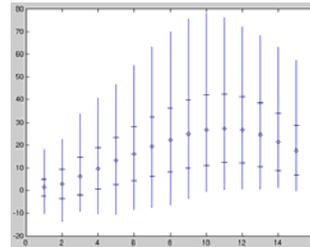
- Example of different line colors & styles

```
>> edit lines  
>> lines
```



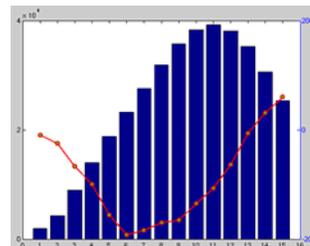
- Markers & lines combined in an error bar

```
>> edit errbar  
>> errbar
```



- Dual y-axis line & bar chart

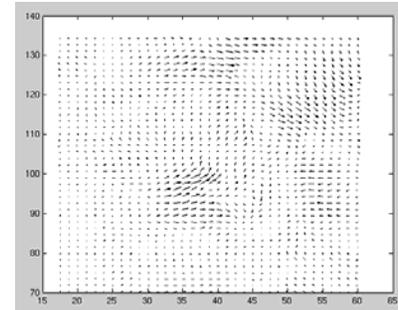
```
>> edit baryy  
>> baryy
```



# Point & Line Plots cont.

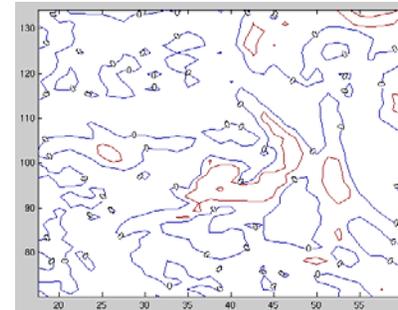
- Vector plot

```
>> edit q  
>> q
```



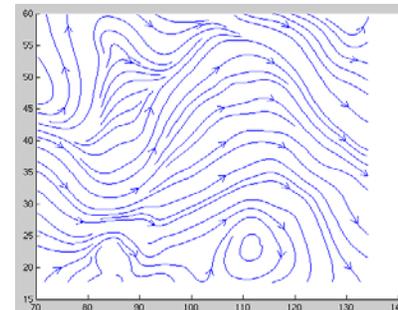
- Contour plot with labels

```
>> edit c  
>> c
```



- Streamlines with arrows

```
>> edit s  
>> s
```



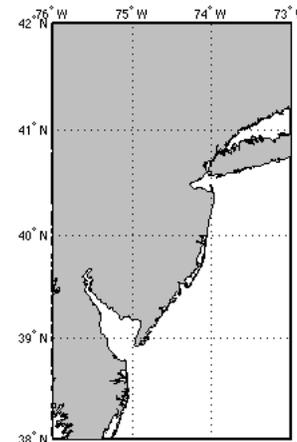
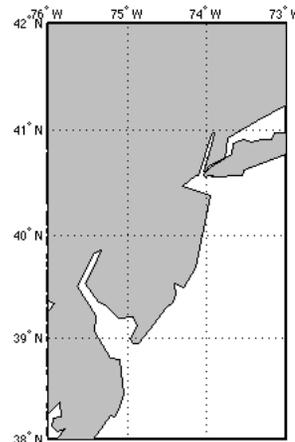
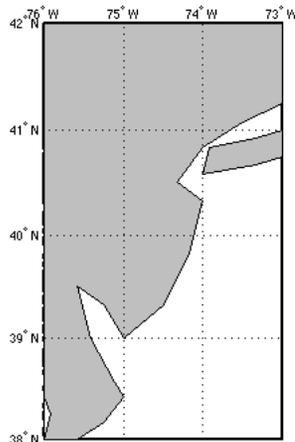
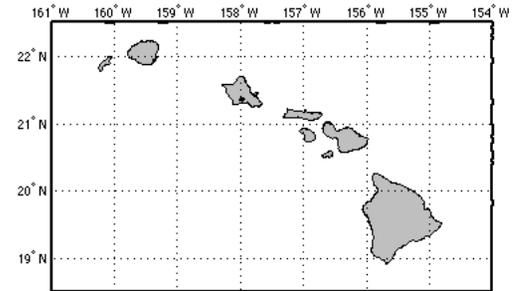
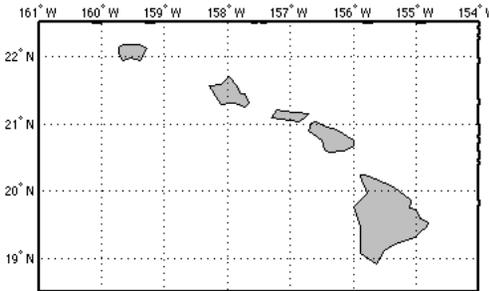
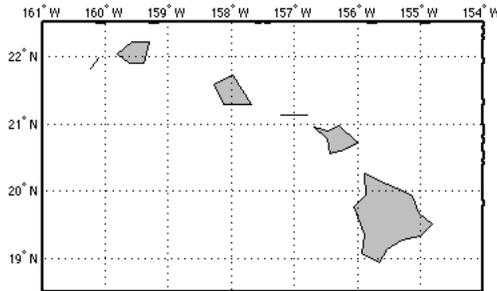
# Maps

3 (border) resolutions provided with MATLAB

“coast”

“worldlo”

“worldhi”



see [hwmmaps.m](http://hwmmaps.m)

# Maps – Plot Types

## 1. Cartesian (simpler code)

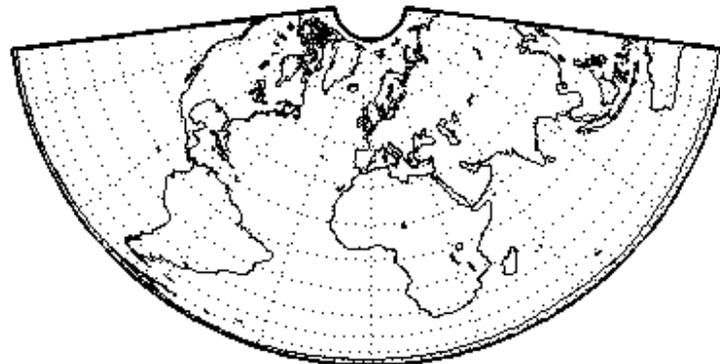
```
>> edit cart
```

```
use load coast2; % or worldlo2, worldhi2*
```



## 2. Projected

```
>> edit proj
```

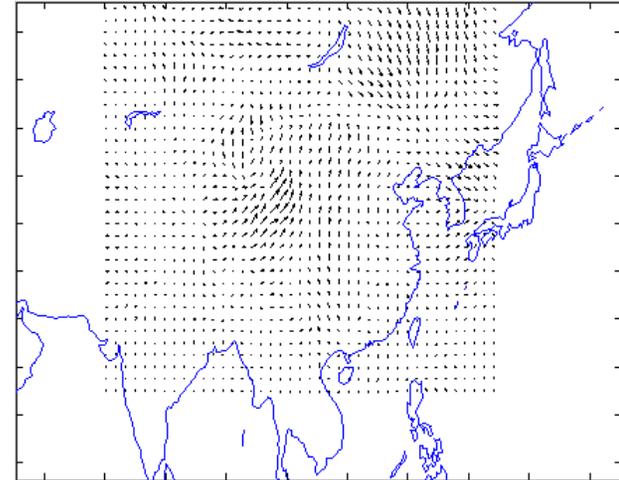


# Maps – Cartesian Examples

Raytheon

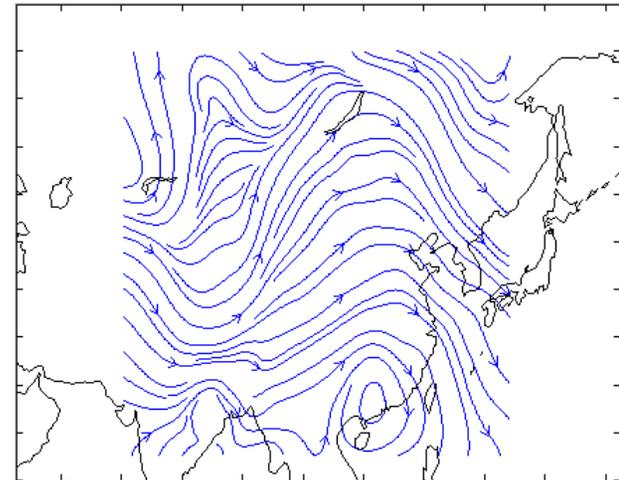
## Vector plot

```
>> load quiverm; load coast2;  
>> quiver(x,y,u,v,'k'); hold on;  
>> plot(long,lat);
```



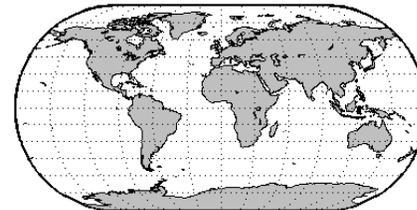
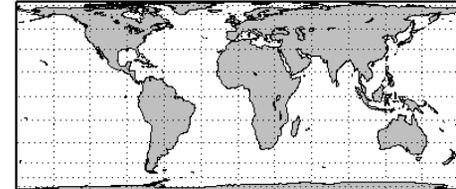
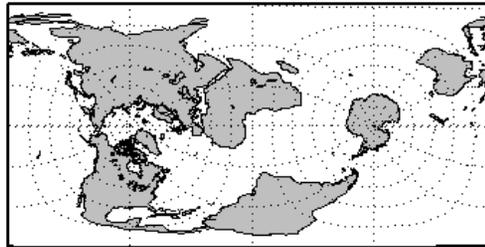
## Streamline plot

```
>> load sm; load coast2;  
>> plot(long,lat, 'k'); hold on;  
>> [li,ar] = streamslice(x,y,u,v);  
>> streamline( [li,ar] );
```



# Maps – Projected

- Types: Azimuthal 13, Cylindrical 45, Conic 9

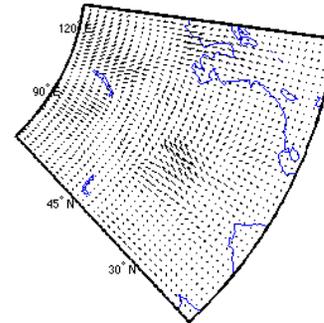


- Projected plot functions have familiar names, but with “m” appended. Examples: `plotm`, `quiverm`, `contourm`

# Maps – Projected Examples

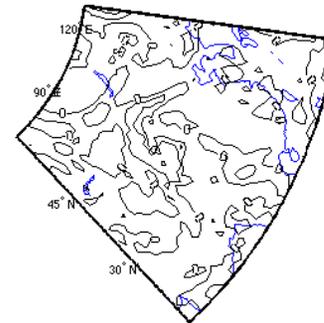
- Vector plot

```
>> edit qm
```



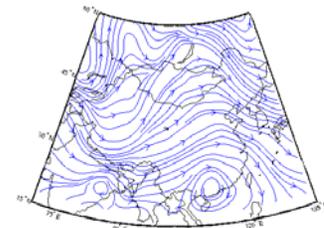
- Contour plot

```
>> edit cm
```



- Streamline plot

```
>> edit sm
```



see also contour3m , clegendm , clabelm , quiver3m

# Labeling

- TeX is supported\*

```
>> edit tex
```

- Create labels with GUI or text

```
>> edit label
```

```
% adjust positions with GUI
```

```
% find x,y with ginput followed by ENTER
```

- Change X,Y,Z ticks through axes properties

```
>> edit ticks
```

\* `addpath('/home/rsz/matlab/tools');`

# Exporting

- Image Files

## PNG, TIFF

```
print -dpng 'out.png';  
print -dtiff 'out.tif';
```

## EPS

```
% EPS color, dpi, preview, filename  
print -depsc2 -r72 [-opengl] -tiff out
```

- Printing

```
% print current figure  
print  
% print a specific figure using it's handle  
h = gcf; print( h )
```

# V. Customizing Your Environment I

- `/home/$USER/matlab` is searched by MATLAB at startup
- If this directory doesn't exist, make it!
- Then insert your configuration files, like the following slide demonstrates

# Setting a Default Printer

Each print causes MATLAB to look for **printopt.m**

 A printer cannot be set within MATLAB itself.

1. `cp ~rsz/matlab/printopt.m ~/matlab`
2. Modify line 47:  
`defprinter = 'double104';`
3. Enable interactive mode on line 45  
(`true` Or `false`)

# Questions

