The GFDL In-House MATLAB Course

Session 2

Remik Ziemlinski
Phone: +1 - 609 - 452 - 6500 ext. 6977
Fax: +1 - 609 - 987 - 5063
Email: 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/mc2]()

2. Start matlab and load course material.

```
/home/$USER> cp –r ~rsz/mc2 ~/mc2
/home/$USER> cd mc2
/home/$USER/mc2> matlab &
>> mc2
```

Table of Contents

I. Customizing Your Environment II

1st file to be executed on startup is

`/home/$USER/matlab/startup.m`

Use it to:

- Include new search paths to data and M-files.
- Change environment variable defaults.
- Perform routine initialization.

Example: Display Format

MATLAB displays all numbers in fixed point format with 5 digits. Add a variant of the `format` command to startup.m

```
>> a = [1 .2 100000];
```

Output display for **a** with

```
>> format, a
% 1.0e+05 * 0.0000 0.0000 1.0000
>> format short g, a
% 1     0.2    1e+05
>> format long g, a
% 1   0.2    100000
```

Example: Loading Toolboxes

- Create the search path for custom files, like the NetCDF toolbox.

```
vi ~/matlab/startup.m
```

Append the following lines:

```
addpath('/home/rsz/matlab');
ncstartup;
```

II. File I/O – ASCII

- ASCII output examples with `writea.m`
  ```
  >> write2da
  >> write3da
  ```

- ASCII input
  Read N dimension files following format similar to `writea.m`
  ```
  >> write2da == reada('2d.txt')
  >> write3da == reada('3d.txt')
  ```

See also `dlmread, csvread, textread`

File I/O – Binary

- MATLAB read/write
```
>> write3d
>> read3d('m')
```

- Fortran 3D
  Read results from `write3d.f90`
```
>> read3d('f')
```

- Sanity check
```
>> read3d('m') == read3d('f')
```

See also **`fseek, ftell`**

File I/O – NetCDF

- You must use a toolbox from USGS
  (see previous slide on how to load it at startup)

- Read a variable

```
>> edit readncvar
>> tref = readncvar(which('AM2.nc'), 't_ref');
```

- Write a variable

```
>> edit writencvar
>> writencvar
```

File I/O – NetCDF cont.

Example script that reads all the variables of a file into the workspace.

```
>> edit readncx;
>> readncx
```

File I/O – MAT-Files

Workspace data can be saved quickly with MATLAB binary format.

```
% save all workspace variables
>> savemat1
>> dir
>> load foo
```

Selectively save workspace variables by listing them.

```
>> savemat2
>> load foo
```

File I/O – DODS

- You can retrieve datasets through a DODS function `loaddods*`
- The data variables can be loaded directly into your workspace

```
>> edit dodsx

>> dodsx
```

```
*addpath('/home/rsz/matlab/'); dodsstartup;
```

III. Advanced Graphics – Colormaps

- 1 colormap per figure window
- Some defaults available:
  `jet, gray, bone, hot, cool, pink, copper, flag`

```
>> colormaps
```

- Max. 256 color entries for default colormaps,
  e.g. **`colormap(jet(256))`**

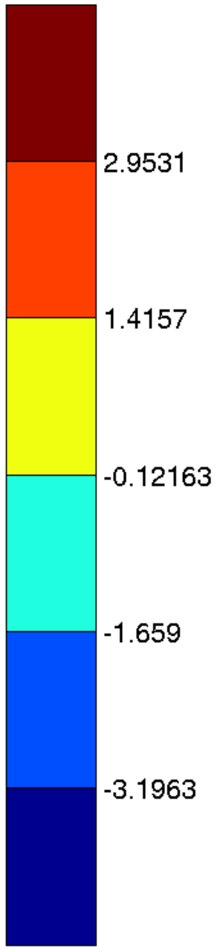
- True colors possible (>256) – each plotted value is assigned RGB triple – only practical for specialized "PR" analysis plots

```
% generate some random RGB triples for custom colormap
>> truecolor
```

- Colormaps can be graphically edited too!

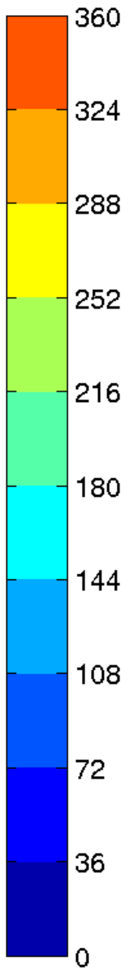
```
>> coloredit
```

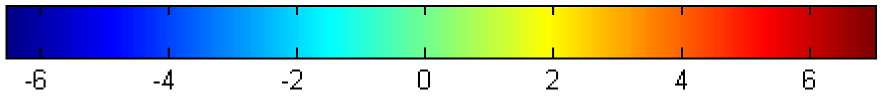
Colorbars



2 Colorbar orientations:
vertical (`'vert'`) or
horizontal (`'horiz'`)

3 flavors:
Created by `colorbarf.m`
Created by `contourcmap.m`
Created by `colorbar.m`



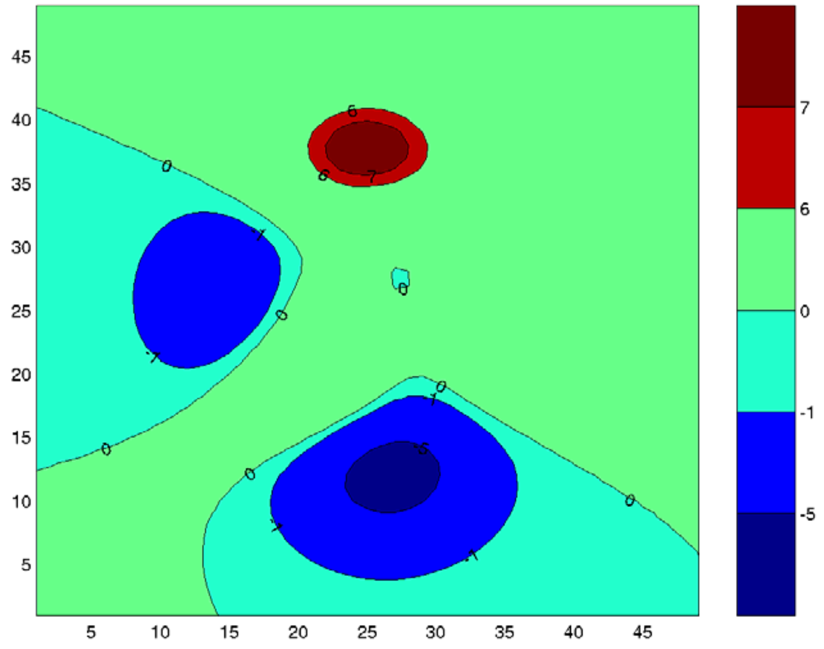


Colorbars Applied

Raytheon

Create a contour plot for specific levels:

```
>> edit ctr
>> ctr
```

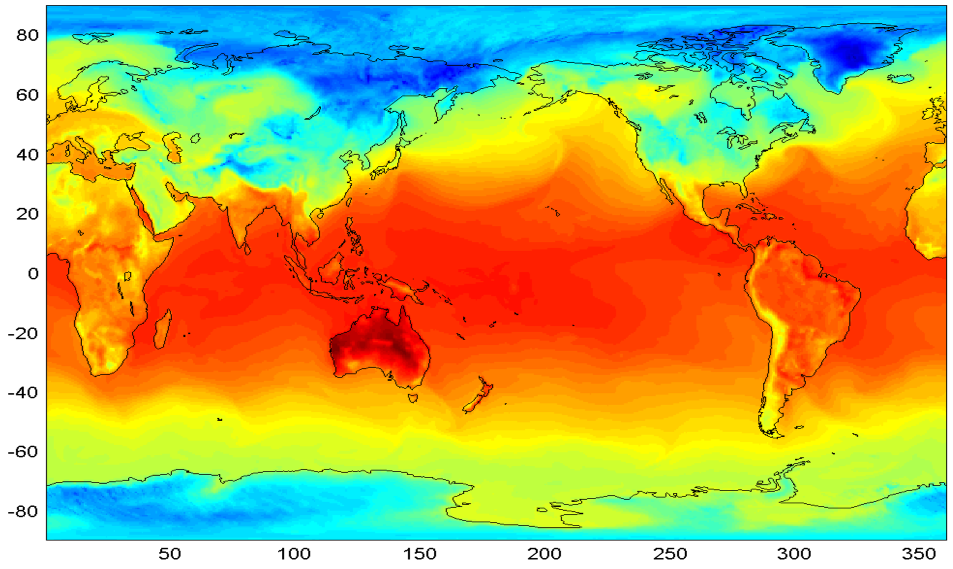


See plotting functions **`surf, pcolor, mesh`**

Maps Revisited – Cartesian

Surface plotting with cartesian coastline and default `jet` colormap (64 colors):

```
>> edit surfcart
>> surfcart
```



Maps Revisited – Projected

3 ways to project plots

1. Use projected plot functions like **contourfm**

```
% traditional contour fill
>> edit azi
% contour lines labeled
>> edit cml
% try to zoom in, then reposition the legend
```

2. **project** graphic object onto map

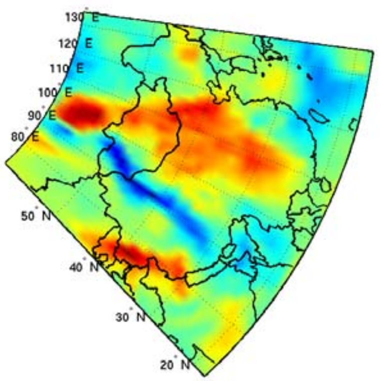
3. Apply a texture (2D image data)

```
% create a texture & smooths its colors
>> edit tms
```

See also **textm, linem**

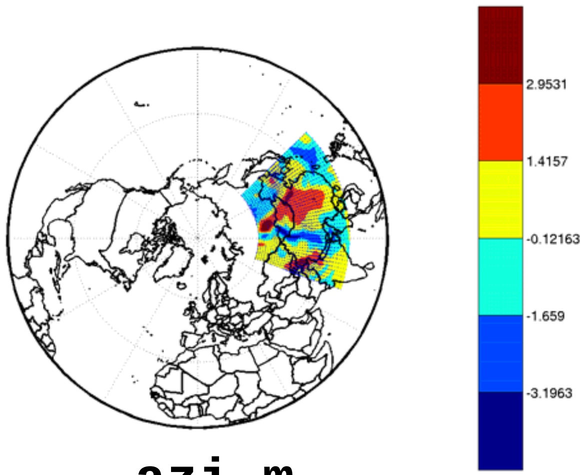
Maps Revisited – Projected Gallery

Raytheon

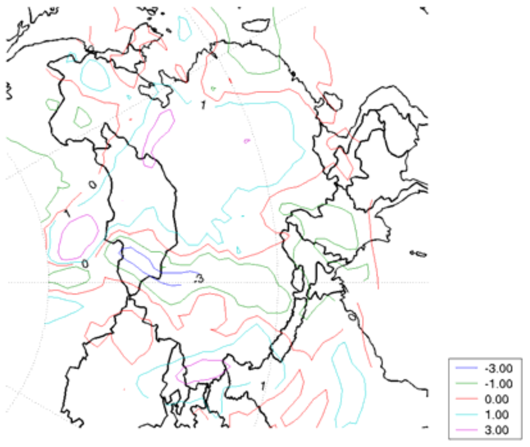


tms.m

cml.m



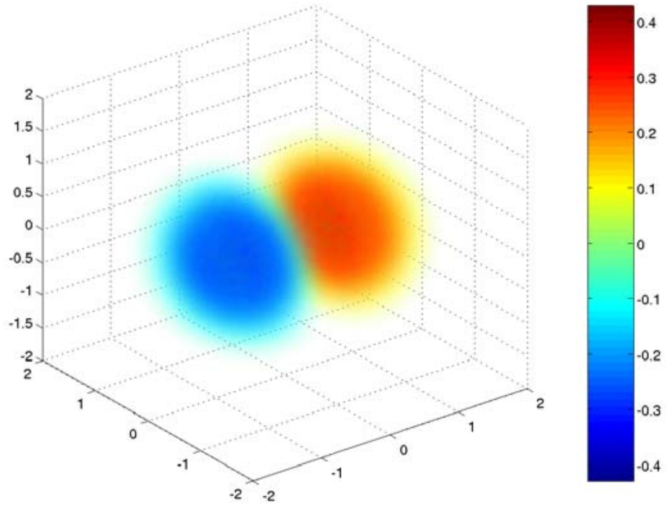
azi.m

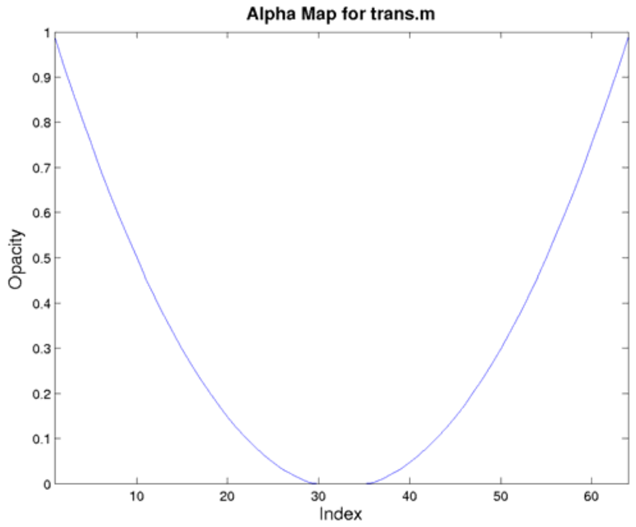


3D Plots

Create pseudo-volumes by changing the transparency map (*alpha* channel).

```
>> edit trans
>> trans
```





See also **`isosurface, alphamap, stream3`**

IV. Advanced Programming – Execution

- Text mode
  `matlab –nosplash –nodisplay`

- Batch mode
  `matlab [–nosplash] [–nodisplay] –r script`

- Be sure `script.m` has a way to call the **`exit`** command when done or upon error.

- To do graphics in text/batch mode, open all your figure windows with `figure('visible','off')`

- Alias these in your `.cshrc` file

Execution cont.

- You can make shell calls and capture any output (stdout, stderr, etc.)

```
>> out = evalc('!hpcsrpt –r');
>> out
```

Runtime Optimization

- If a script/function is slow, don’t assume the cause, but actually profile it!

```
% profile a sample script
>> profile on
>> colormaps

% plot the results and view the details
>> profile plot
>> profile viewer

% create HTML of results
>> profile report
```

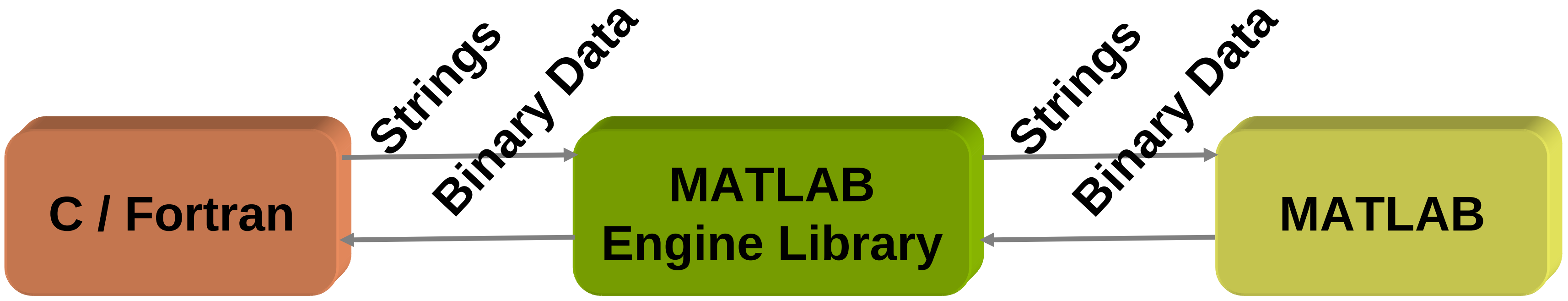
Error Handling

- Critical errors may be caught or created.
- Use error handling to gracefully exit a script, i.e. clear workspace memory & close file handles.

```
>> edit err
>> err
```

C / Fortran Interfaces

- MATLAB has shared libraries to access its engine from C / Fortran. Just include the header files and link.
- Expressions are passed to the MATLAB engine as strings.
- Data is transferred in binary.

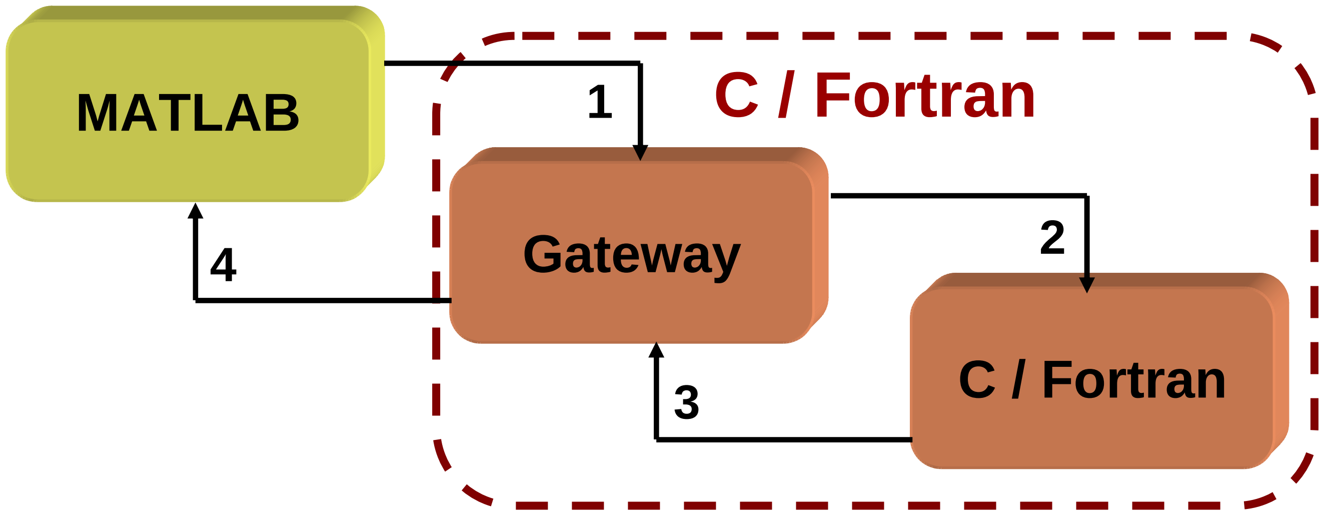


See also **helpdesk** Contents | MATLAB | External Interfaces / API

MEX Programs

Interface existing C / Fortran code so you can use them directly from a MATLAB command prompt or script like any other function.

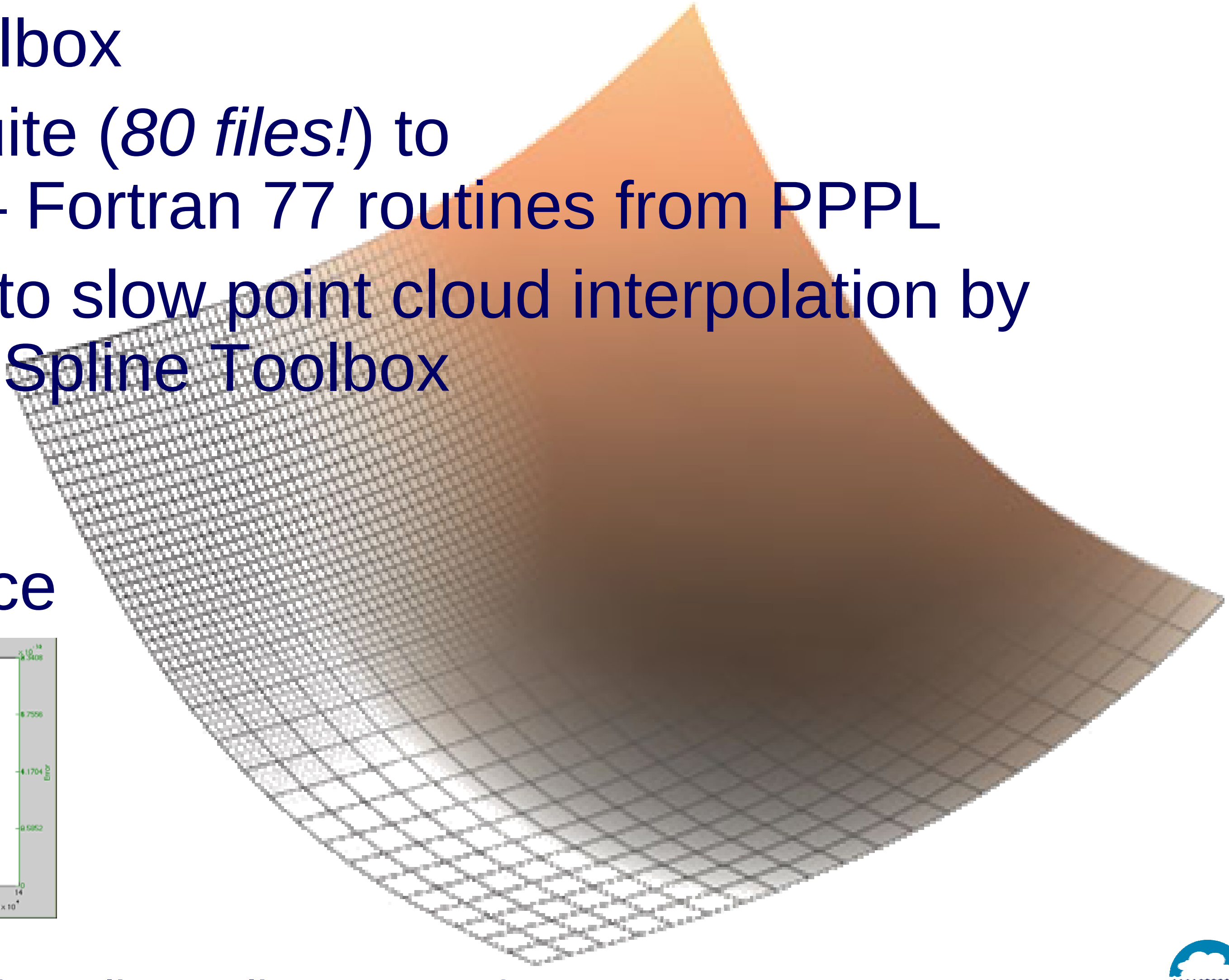
Function Call Sequence



See also **helpdesk** Contents | MATLAB | External Interfaces / API

MEX Programs – Case Study

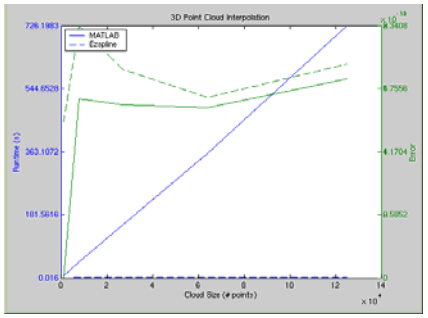
Raytheon



EZspline Toolbox

- gateway suite (*80 files!*) to PSPLINE – Fortran 77 routines from PPPL
- alternative to slow point cloud interpolation by MATLAB's Spline Toolbox

Performance



See also http://ezsplinetoolbox.sourceforge.net

V. Topics to Explore

Raytheon

- Mathematical functions in MATLAB for: Linear algebra, Polynomials, Statistics, Differential equations, Sparse matrices
- Parallel MATLAB
  MATLABMPI
  MPI Toolbox
- Creating GUIs with `guide`
- `demos` provided with MATLAB

Questions

- Do we have enough [toolbox] licenses?
- Where can I get support?

  Use GFDL URL help

See also http://www.custardpie.com/grad4.jpg