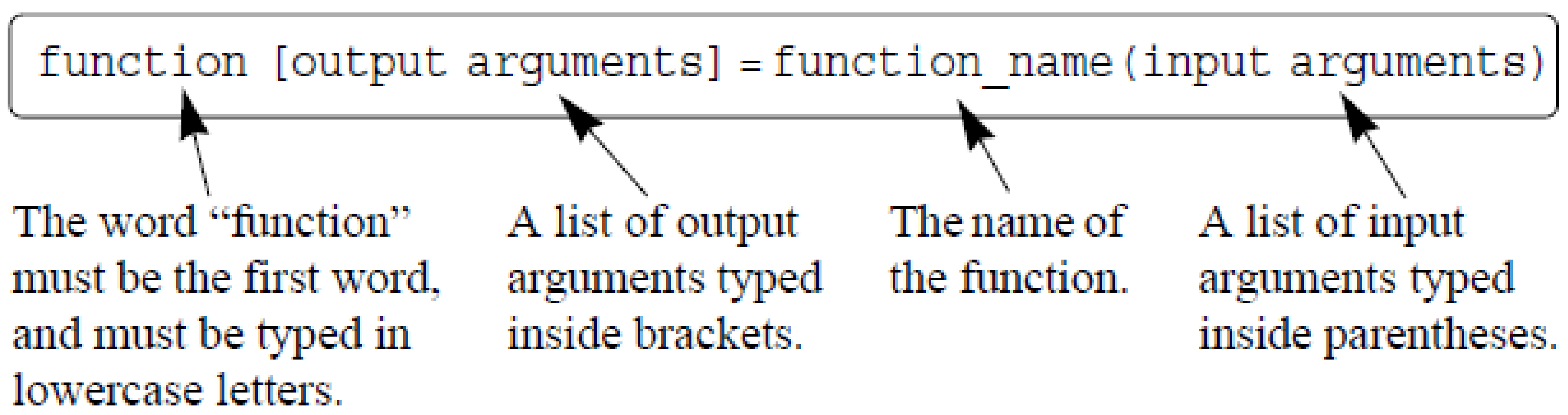
Thursday Sept 22 : What you should be doing:

- Finish assignment 4. Due Friday at noon
- Complete/redo Assignment 3 by Tuesday 3:30
- Last Check Your Understanding: Chapter 10 next Tuesday
- Stay tuned for info on Assignments 5 & 6 (last ones!)
- covering Chapter 10 plotting
- Download the files for today from my web page: chpc.utah.edu/atmos_5020/

Why use functions?

- Use same code in more than one place in program without rewriting code
- Reuse code for different programs
- Make debugging easier by putting all of one kind of functionality in one place
- Can develop one line anonymous functions inside program, but recommend not doing so
- Matlab uses many function functions, passing a function into another function, but again complicated and not necessary to do so

Form of function definition line is



```
function [output arguments] = function_name(input arguments)
```

The word “function” must be the first word, and must be typed in lowercase letters.

A list of output arguments typed inside brackets.

The name of the function.

A list of input arguments typed inside parentheses.

Function name

- Made up of letters, digits, underscores
- Cannot have spaces
- Follows same rules as variable names

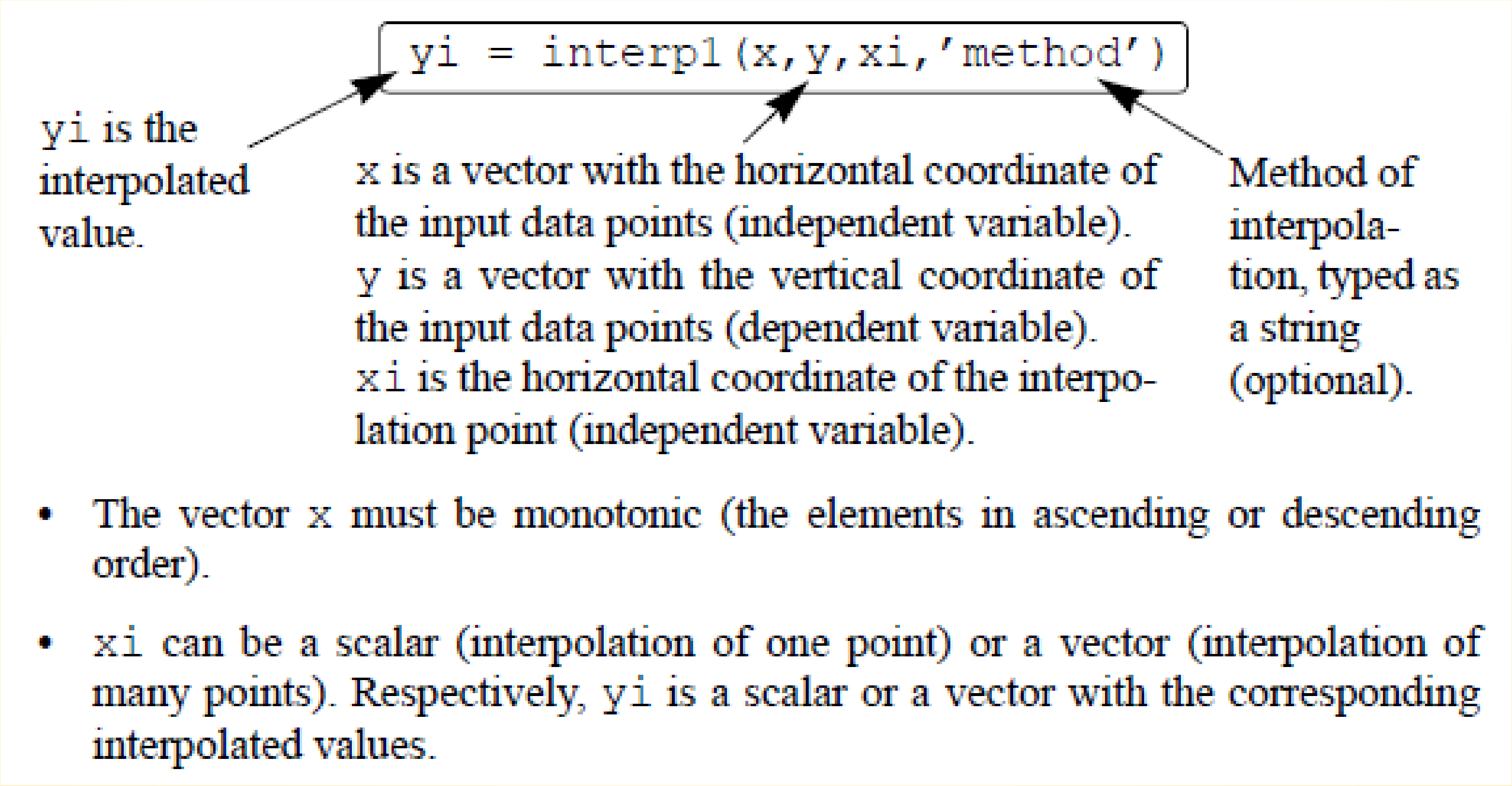
***Avoid making function names that are same as names of built-in functions

Interpolation is estimating values between data points.

One-dimensional interpolation:

linear interpolation is estimating value between two data points by connecting points with a straight line and then using value on line as estimated value "one"

MATLAB function `interp1()` does one-dimensional interpolation



```
yi = interp1(x,y,xi,'method')
```

`yi` is the interpolated value.

`x` is a vector with the horizontal coordinate of the input data points (independent variable). `y` is a vector with the vertical coordinate of the input data points (dependent variable). `xi` is the horizontal coordinate of the interpolation point (independent variable).

Method of interpolation, typed as a string (optional).

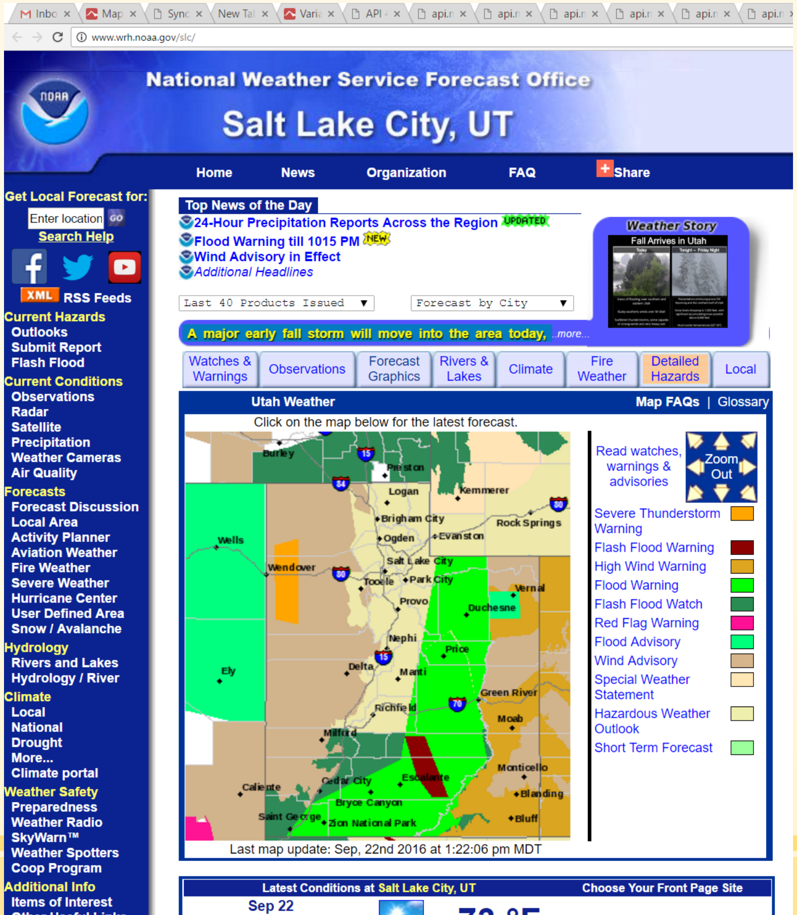
- The vector `x` must be monotonic (the elements in ascending or descending order).
- `xi` can be a scalar (interpolation of one point) or a vector (interpolation of many points). Respectively, `yi` is a scalar or a vector with the corresponding interpolated values.

What are we trying to do???

- Understand environmental information
- Input → Process → **Visualize** → Understand
- Environmental fields rely heavily on graphical depictions of information

NWS visualizations

http://www.wrh.noaa.gov/slc/



National Weather Service Forecast Office

Salt Lake City, UT

Home | News | Organization | FAQ | Share

Get Local Forecast for:

Enter location GO

Search Help

XML RSS Feeds

Current Hazards
- Outlooks
- Submit Report
- Flash Flood

Current Conditions
- Observations
- Radar
- Satellite
- Precipitation
- Weather Cameras
- Air Quality

Forecasts
- Forecast Discussion
- Local Area
- Activity Planner
- Aviation Weather
- Fire Weather
- Severe Weather
- Hurricane Center
- User Defined Area
- Snow / Avalanche

Hydrology
- Rivers and Lakes
- Hydrology / River

Climate
- Local
- National
- Drought
- More...
- Climate portal

Weather Safety
- Preparedness
- Weather Radio
- SkyWarn™
- Weather Spotters
- Coop Program

Additional Info
- Items of Interest

Top News of the Day

- 24-Hour Precipitation Reports Across the Region UPDATED
- Flood Warning till 1015 PM NEW
- Wind Advisory in Effect
- *Additional Headlines*

Last 40 Products Issued ▼ | Forecast by City ▼

Weather Story

Fall Arrives in Utah

A major early fall storm will move into the area today, ...more...

Watches & Warnings | Observations | Forecast Graphics | Rivers & Lakes | Climate | Fire Weather | Detailed Hazards | Local

Utah Weather | Map FAQs | Glossary

Click on the map below for the latest forecast.

Read watches, warnings & advisories

Zoom Out

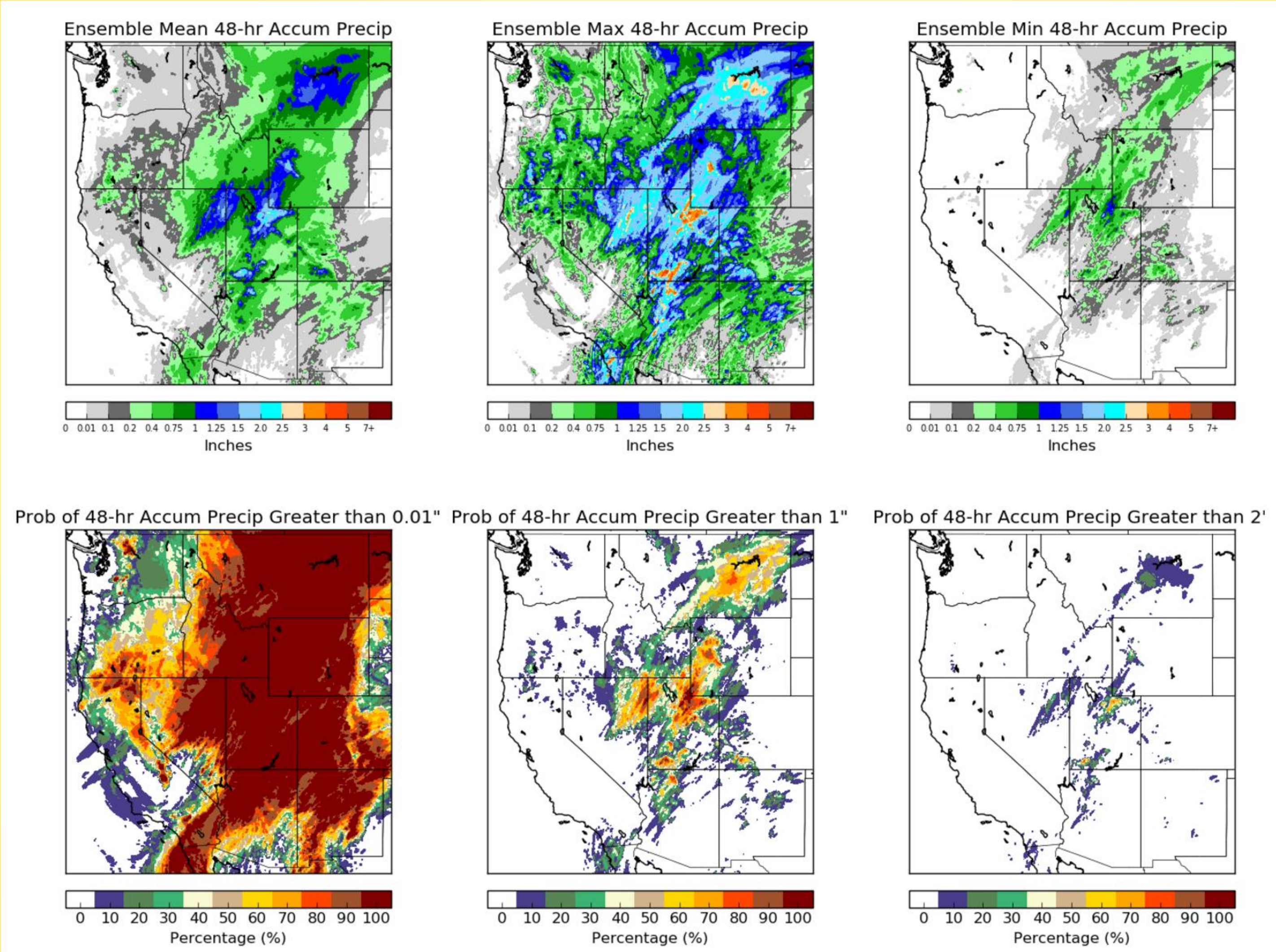
Severe Thunderstorm Warning	🟧
Flash Flood Warning	🟥
High Wind Warning	🟧
Flood Warning	🟩
Flash Flood Watch	🟩
Red Flag Warning	🟥
Flood Advisory	🟩
Wind Advisory	🟫
Special Weather Statement	🟨
Hazardous Weather Outlook	🟨
Short Term Forecast	🟩

Last map update: Sep, 22nd 2016 at 1:22:06 pm MDT

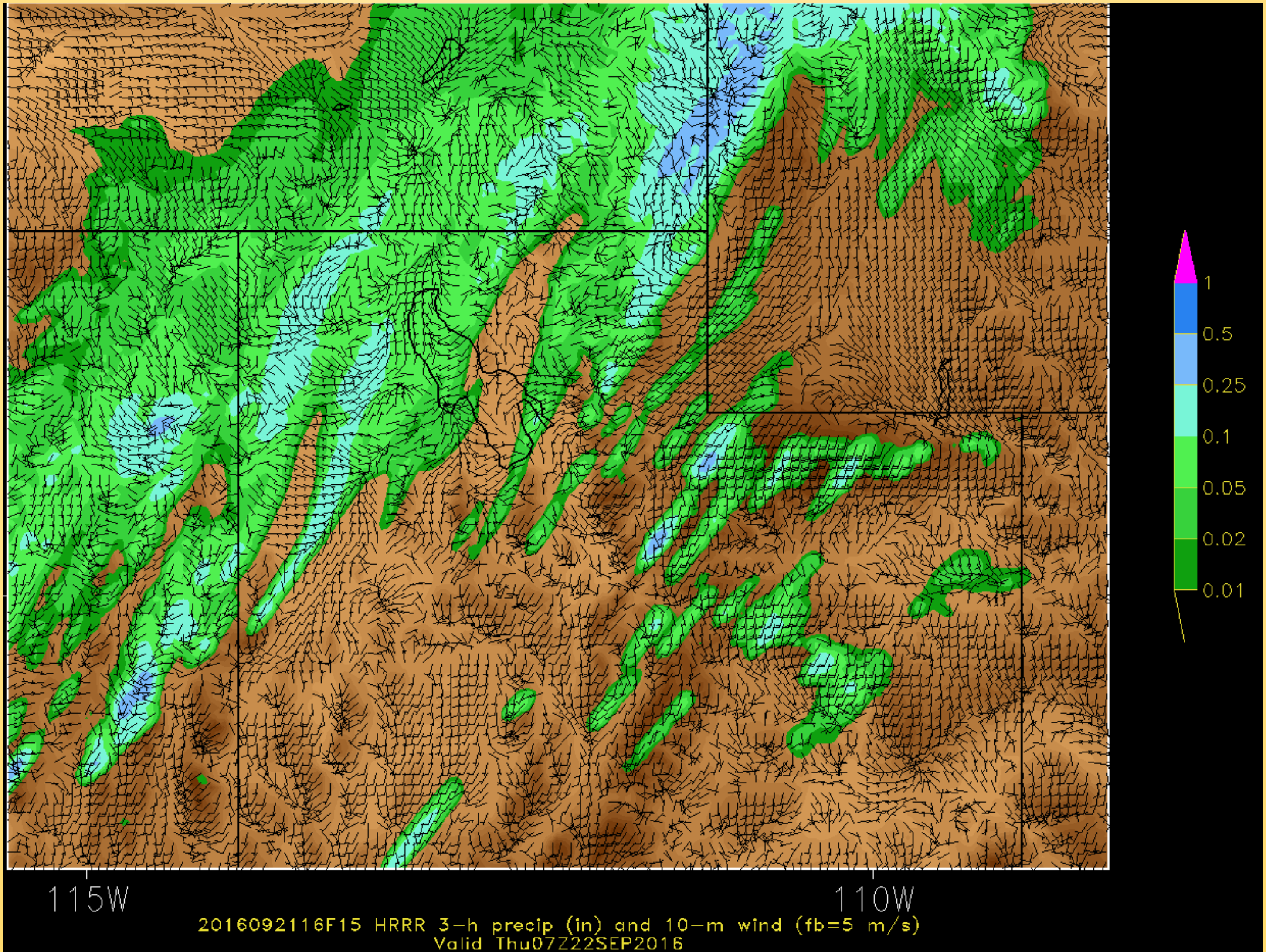
Latest Conditions at Salt Lake City, UT | Choose Your Front Page Site

Sep 22

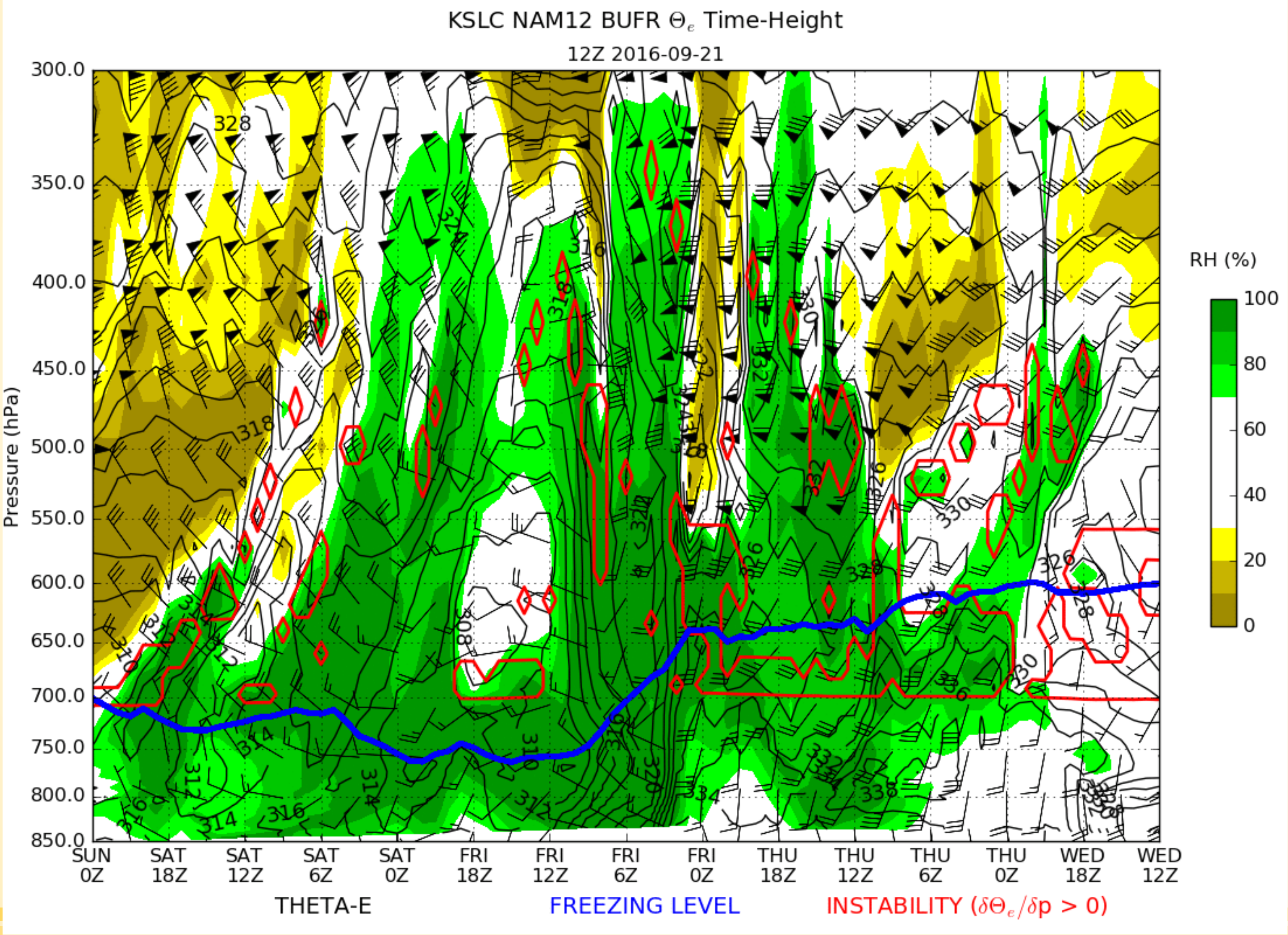
http://weather.utah.edu/index.php?t=ncarens&r=WE&d=PP


Ensemble Mean 48-hr Accum Precip
Ensemble Max 48-hr Accum Precip
Ensemble Min 48-hr Accum Precip
0 0.01 0.1 0.2 0.4 0.75 1 1.25 1.5 2.0 2.5 3 4 5 7+
Inches
Prob of 48-hr Accum Precip Greater than 0.01"
Prob of 48-hr Accum Precip Greater than 1"
Prob of 48-hr Accum Precip Greater than 2'
0 10 20 30 40 50 60 70 80 90 100
Percentage (%)

http://weather.utah.edu/index.php?runcode=2015112822&t=hrrr&d=SF&r=NU



http://weather.utah.edu/index.php?runcode=2016092112&t=nam218&d=ET&r=KSLC

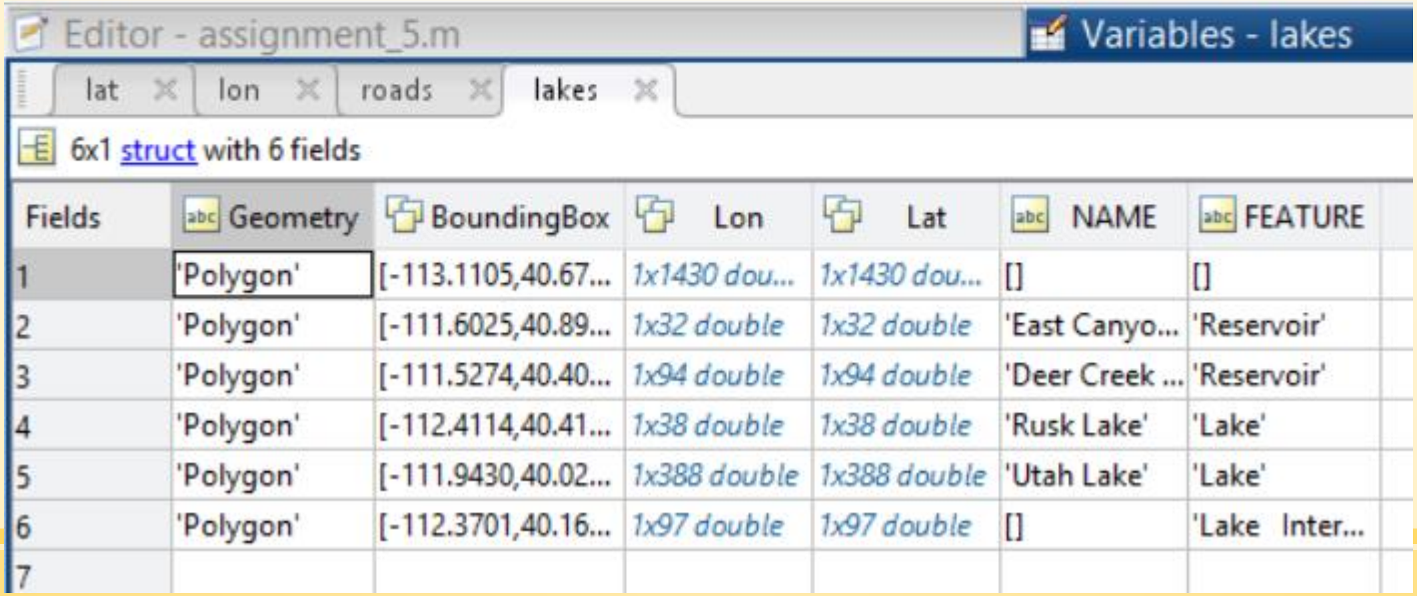


Types of spatial environmental data

- vector data: points, lines, polygons
- grids/raster data: 2 dim matrix organized into rows and columns where each cell contains a value representing information, such as temperature

GIS shapefiles

- Shapefiles describe vector features: points, lines, and polygons
  - Shapefile shape format (.shp)
  - Shapefile shape index format (.shx)
  - Shapefile attribute format (.dbf)



6x1 struct with 6 fields

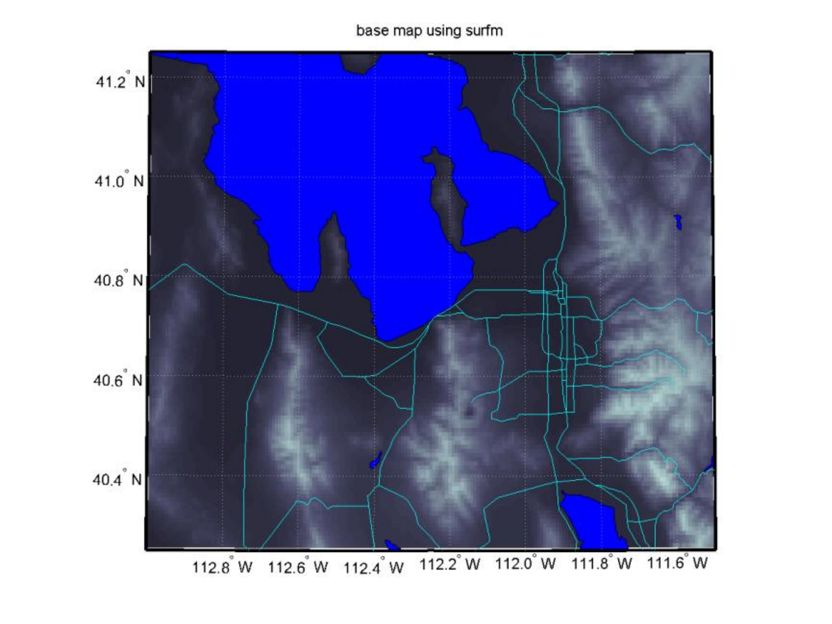
Fields	Geometry	BoundingBox	Lon	Lat	NAME	FEATURE
1	'Polygon'	[-113.1105,40.67...	1x1430 dou...	1x1430 dou...	[]	[]
2	'Polygon'	[-111.6025,40.89...	1x32 double	1x32 double	'East Canyo...	'Reservoir'
3	'Polygon'	[-111.5274,40.40...	1x94 double	1x94 double	'Deer Creek ...	'Reservoir'
4	'Polygon'	[-112.4114,40.41...	1x38 double	1x38 double	'Rusk Lake'	'Lake'
5	'Polygon'	[-111.9430,40.02...	1x388 double	1x388 double	'Utah Lake'	'Lake'
6	'Polygon'	[-112.3701,40.16...	1x97 double	1x97 double	[]	'Lake Inter...
7						

Netcdf

- NetCDF (Network Common Data Form): software self-describing, machine-independent data formats to support creation, access, and sharing of array-oriented scientific data
- http://www.unidata.ucar.edu/software/netcdf/

High resolution elevation data surrounding Salt Lake radar in netcdf format

- Obtain lat’s (588), lon’s (804) and elevations (804x588)
- Matlab has lat’s as rows and lon’s as columns, so need to take transpose

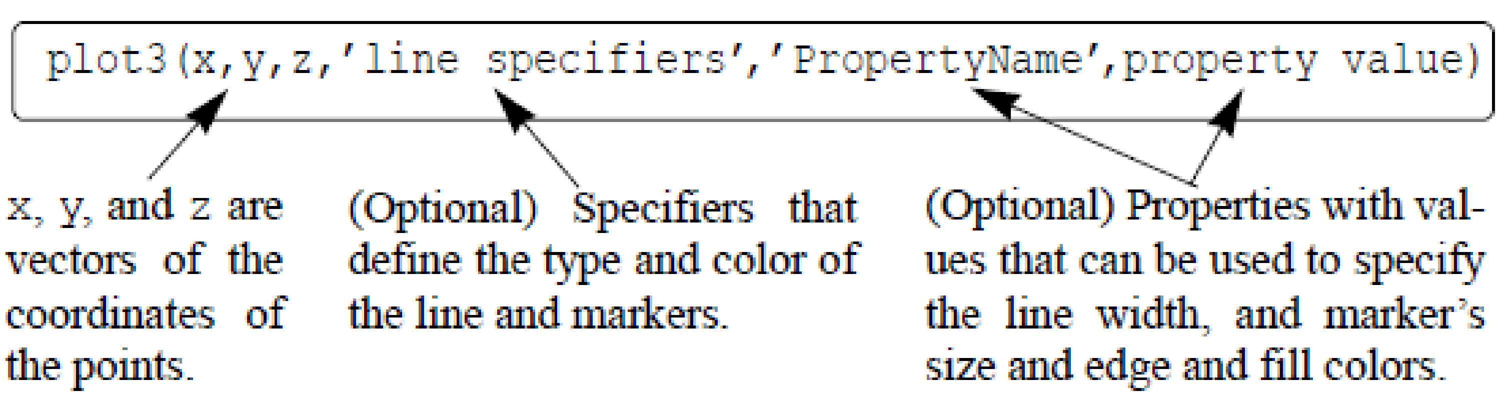


What we often consider as 2-d plots are technically 3-d (location + value)

Three-dimensional (3-D) plots useful for:

- Scalar or vector function of two-independent variables (lat/lon; elev/lon or elev/lat)
- Scalar or vector data measurements in 3-D space (google earth- lat/lon/elev)
- Movement over time (sequences of "2d" images or elv/time time-height sections, etc.)

A *three-dimensional line plot* is a plot obtained by connecting points in 3-D space. MATLAB command is



```
plot3(x,y,z,'line specifiers','PropertyName',property value)
```

x, y, and z are vectors of the coordinates of the points.

(Optional) Specifiers that define the type and color of the line and markers.

(Optional) Properties with values that can be used to specify the line width, and marker's size and edge and fill colors.

- x, y, and z must be same sized **<u>vectors</u>**
- Remaining arguments are same as in 2-D plots

If the spatial coordinates of a set of points are each functions of the same independent variable, the coordinates form a set of *parametric equations*.

- Often independent variable is time (t) and the set shows how a particle moves through space over time

EXAMPLE

spatial coordinates varying with time as

$$\begin{aligned} x &= \sqrt{t}\sin(2t) \\ y &= \sqrt{t}\cos(2t) \\ z &= 0.5t \end{aligned}$$

Make a line plot for $0 \leq t \leq 6\pi$

```
t=0:.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
Figure(1)
plot3(x,y,z,’r’,’linewidth’,1);
grid on
xlabel(‘x’); ylabel(‘y’);zlabel(‘z’)
```

10.1 Line Plots

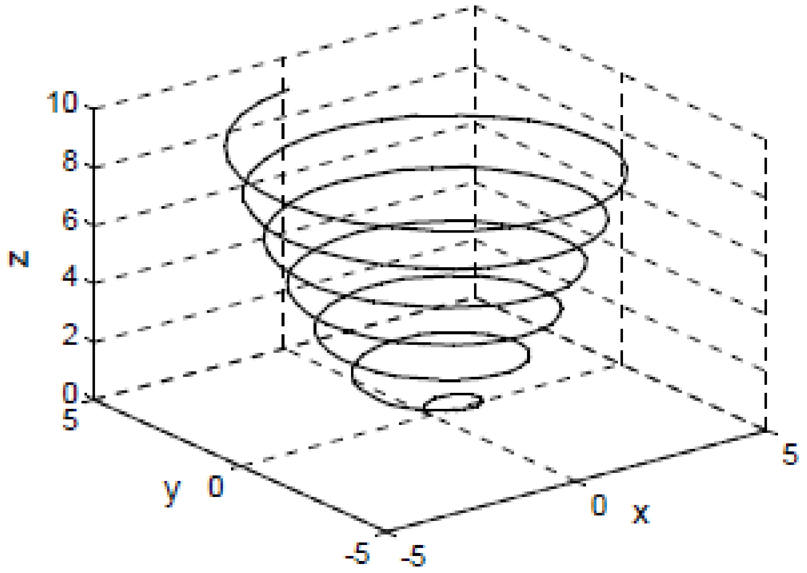


Figure 10-1: A plot of the function $x = \sqrt{t}\sin(2t)$, $y = \sqrt{t}\cos(2t)$, $z = 0.5t$ for $0 \le t \le 6\pi$.

Mesh and surface plots are 3-D plots used to graph functions of the form $a = f(x,y)$

- *x* and *y* are independent variables (lat/lon), *a* is a dependent variable
- A *mesh plot* connects values of *a* with lines to form the outline of a surface
- A *surface plot* connects lines in a mesh plot with planes to show a solid representation of the surface

Three steps to making mesh or surface plot

1. Create grid in the x-y plane that contains points you're interested in
2. Calculate the value of *a* at every point of the grid
3. Make the plot

Creating a grid in the *x y* plane (Cartesian coordinates):

The grid is the set of points on which you want to evaluate *z*. For example

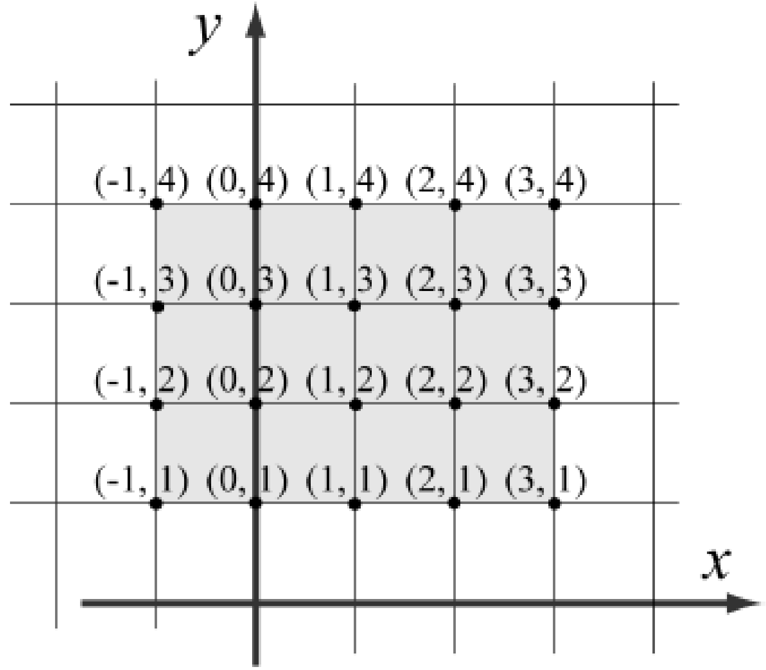


Figure 10-2: A grid in the *x*-*y* plane for the domain $-1 \le x \le 3$ and $1 \le y \le 4$ with spacing of 1.

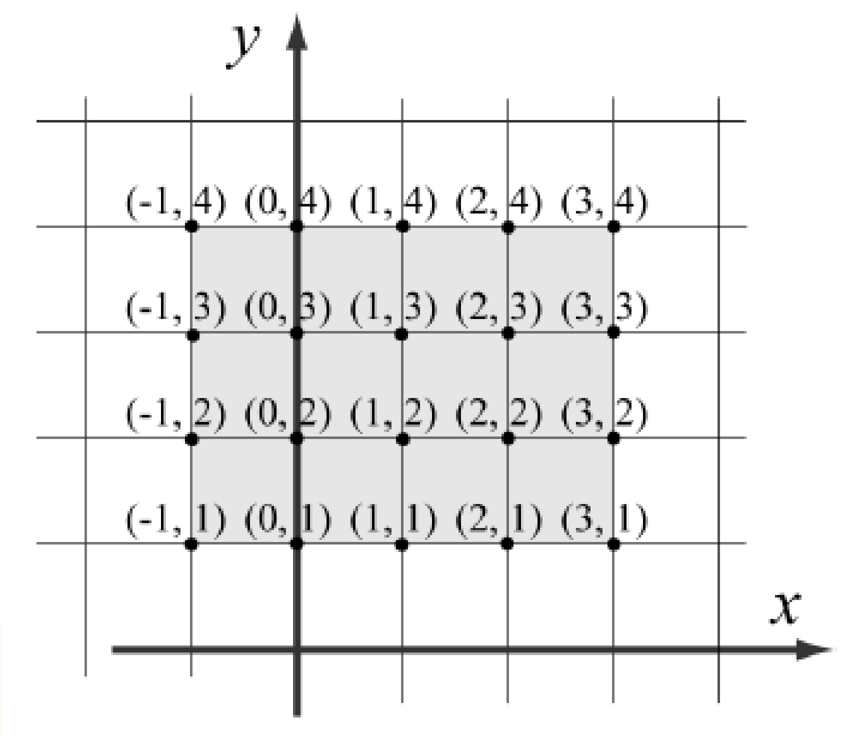
Can define the grid by using two matrices, X and Y

- X has x-coordinates of all grid points
- Y has y-coordinates of all grid points

For grid shown

$$X = \begin{bmatrix} -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{bmatrix}$$

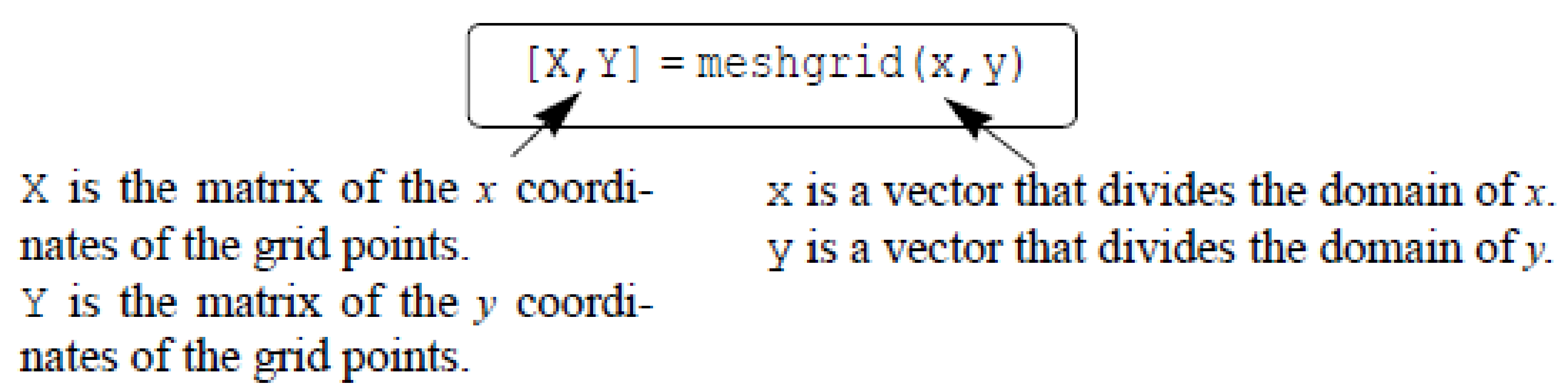
$$\mathrm{Y} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



Note that

- X is made of identical rows because each row of grid has the same x-coordinates
- Y is made of identical columns because each column of grid has same y-coordinates

To make matrices, use MATLAB command

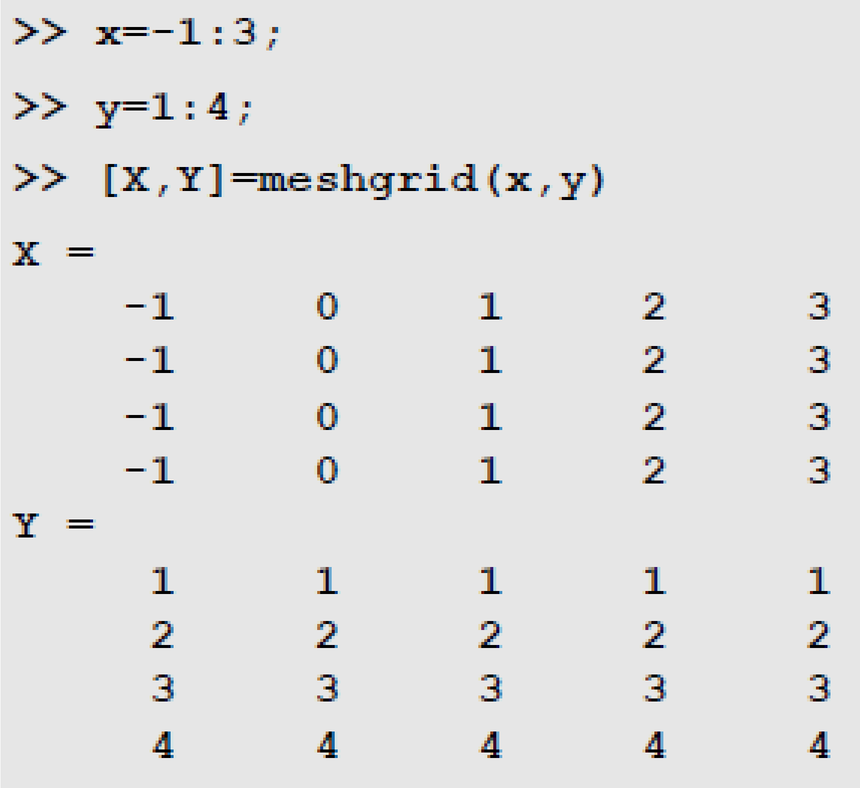


```
[X,Y] = meshgrid(x,y)
```

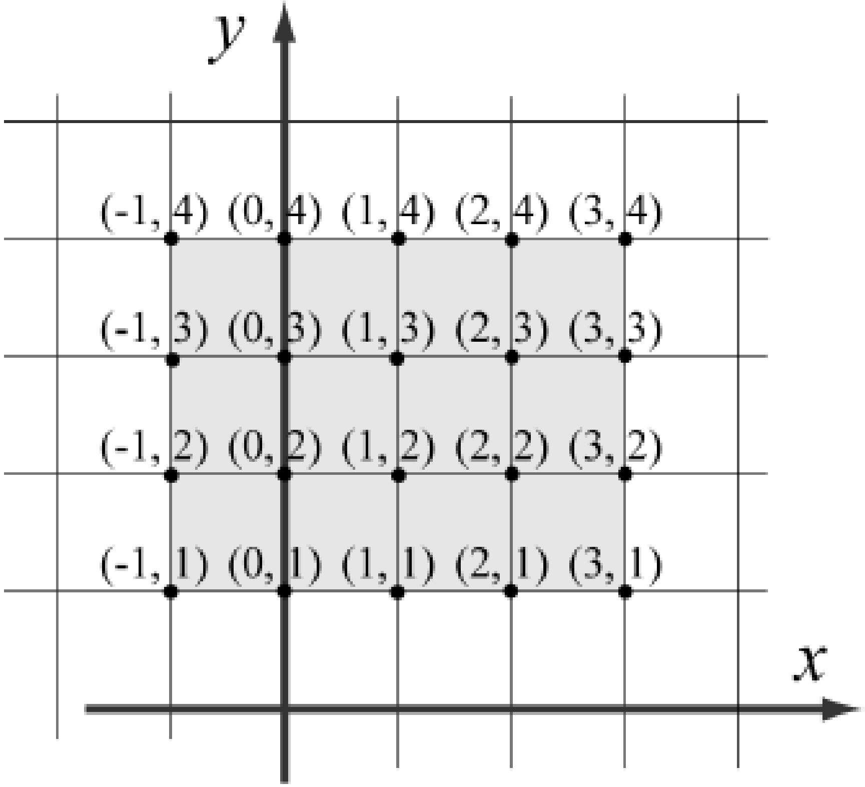
X is the matrix of the *x* coordinates of the grid points.
Y is the matrix of the *y* coordinates of the grid points.

x is a vector that divides the domain of *x*.
y is a vector that divides the domain of *y*.

10.2 Mesh and Surface Plots



```
>> x=-1:3;
>> y=1:4;
>> [X,Y]=meshgrid(x,y)
X =
     -1      0      1      2      3
     -1      0      1      2      3
     -1      0      1      2      3
     -1      0      1      2      3
Y =
      1      1      1      1      1
      2      2      2      2      2
      3      3      3      3      3
      4      4      4      4      4
```



(-1, 4) (0, 4) (1, 4) (2, 4) (3, 4)
(-1, 3) (0, 3) (1, 3) (2, 3) (3, 3)
(-1, 2) (0, 2) (1, 2) (2, 2) (3, 2)
(-1, 1) (0, 1) (1, 1) (2, 1) (3, 1)

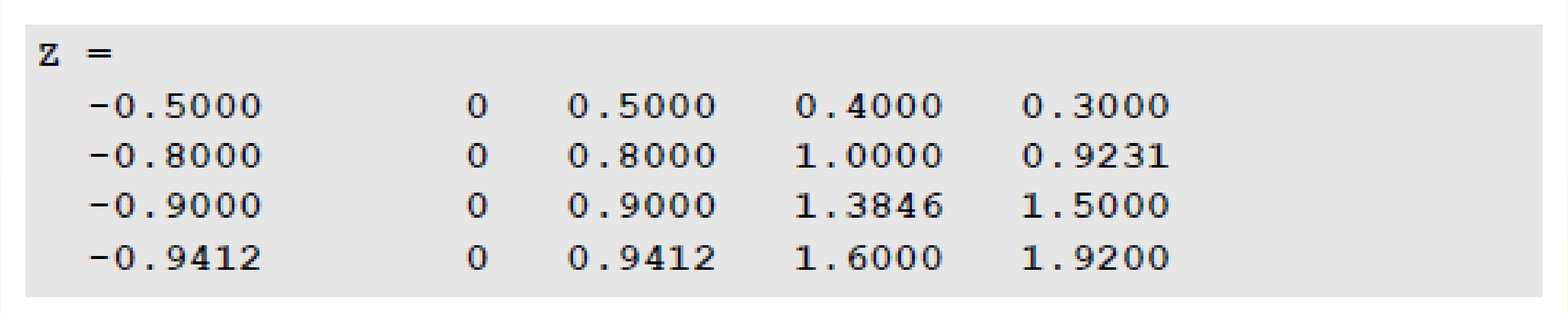
Calculating the value of *z* at each point of the grid:

Calculate value of *z* at each point by using elementwise calculations.

- X and Y must be same dimensions
- Resulting *z* will also be same dimension

For example grid and $$z = \frac{xy^2}{x^2+y^2}$$

```
>> Z = X.*Y.^2 ./ (X.^2 + Y.^2)
```



```
Z =
   -0.5000         0    0.5000    0.4000    0.3000
   -0.8000         0    0.8000    1.0000    0.9231
   -0.9000         0    0.9000    1.3846    1.5000
   -0.9412         0    0.9412    1.6000    1.9200
```

Making mesh and surface plots:

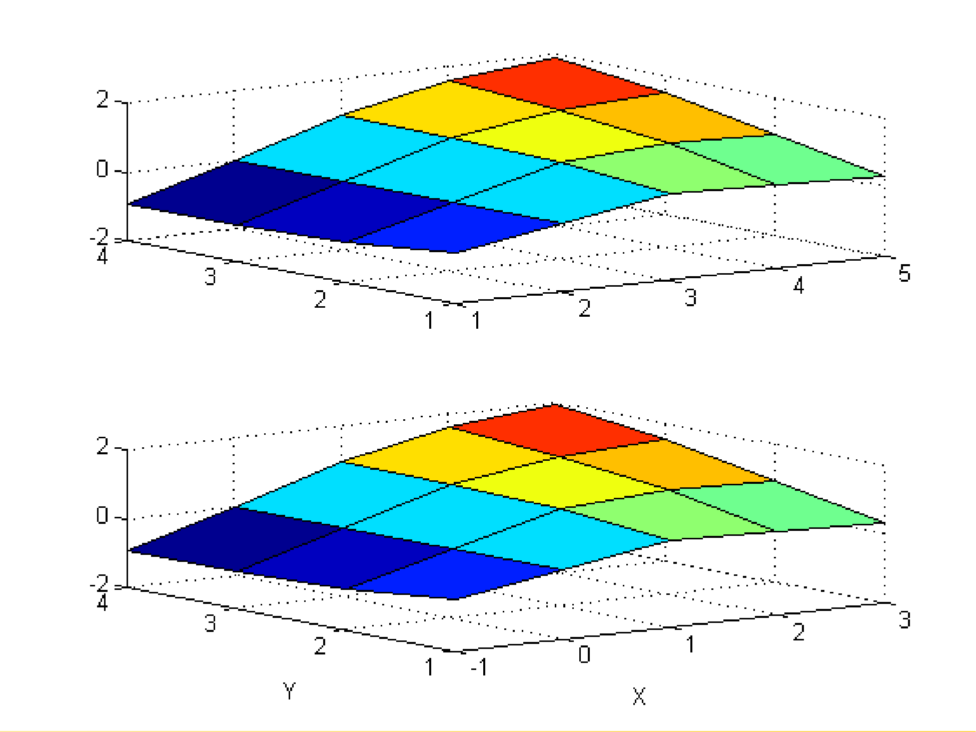
- To make mesh plot use `mesh(X,Y,Z)`
- To make surface plot use `surf(X,Y,Z)`
- mesh(Z) or surf(Z) use row index vector on the x-axis and column index vector on the y-axis

EXAMPLE

Make mesh and surface plots of $z = \frac{xy^2}{x^2+y^2}$ over domain $-1 \le x \le 3$ and $1 \le y \le 4$

10.2 Mesh and Surface Plots

```
x=-1:3;
y=1:4;
[X,Y] = meshgrid(x,y);
Z = X.*Y.^2 ./ (X.^2 + Y.^2);
figure(2)
subplot(2,1,1)
xlabel(‘columns of Z')
ylabel(‘rows of Z')
surf(Z)
subplot(2,1,2)
surf(X,Y,Z)
xlabel('X')
ylabel('Y')
```



`view` command controls direction from which you view plot. Command is

`view(az,el)` **or** `view([az el])`

- `az` – *azimuth*: angle (in degrees) in *x y* plane measured from negative *y* axis and positive in counterclockwise direction
- el – *elevation*: angle of elevation (in degrees) from *x y* plane. Positive in direction of positive *z* axis

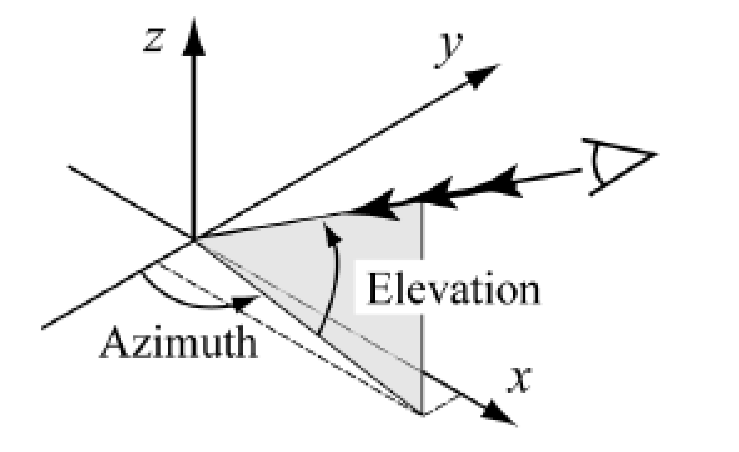
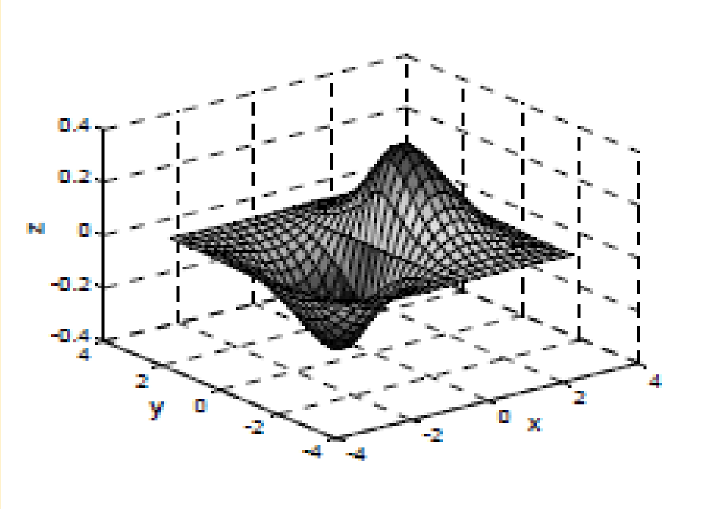
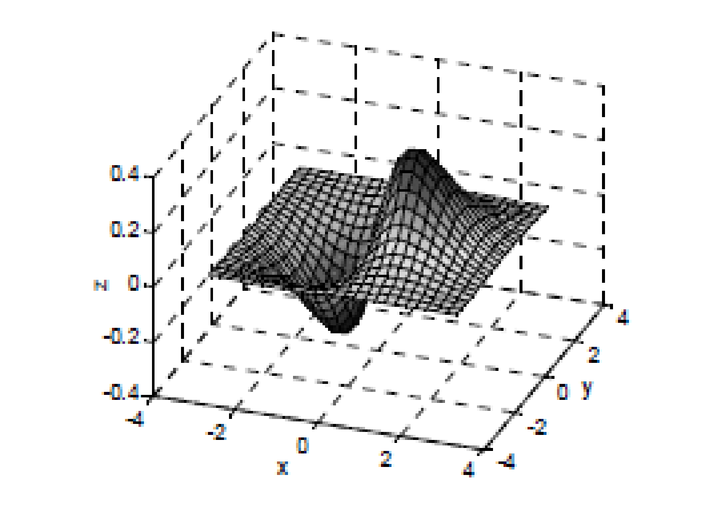


Figure 10-3: Azimuth and Elevation angles.

Default view angles are *az* = -37.5^o and *el* = 30^o

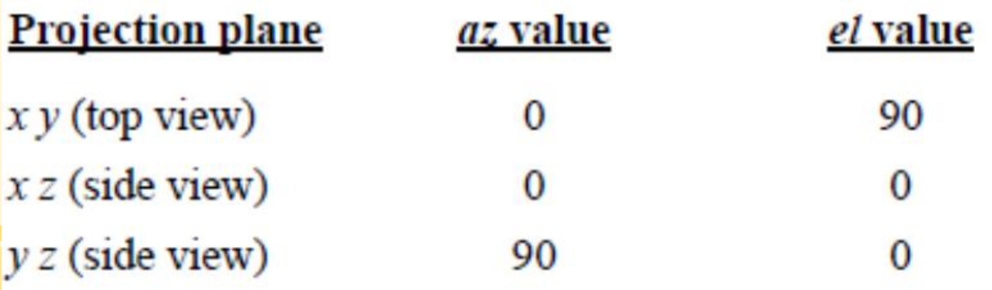


az = -37.5^o and *el* = 30^o



az = 20^o and *el* = 35^o

Can project 3-D curve onto 2-D plane by specific settings of azimuth and elevation



Projection plane	*az* value	*el* value
x y (top view)	0	90
x z (side view)	0	0
y z (side view)	90	0

`view` can also set a default view

- `view(2)` sets default to top view (projection onto x-y plane with $az = 0^o$, and $el = 90^o$
- `view(3)` sets default to standard 3-D view ($az = -37.5^o$, and $el = 30^o$)

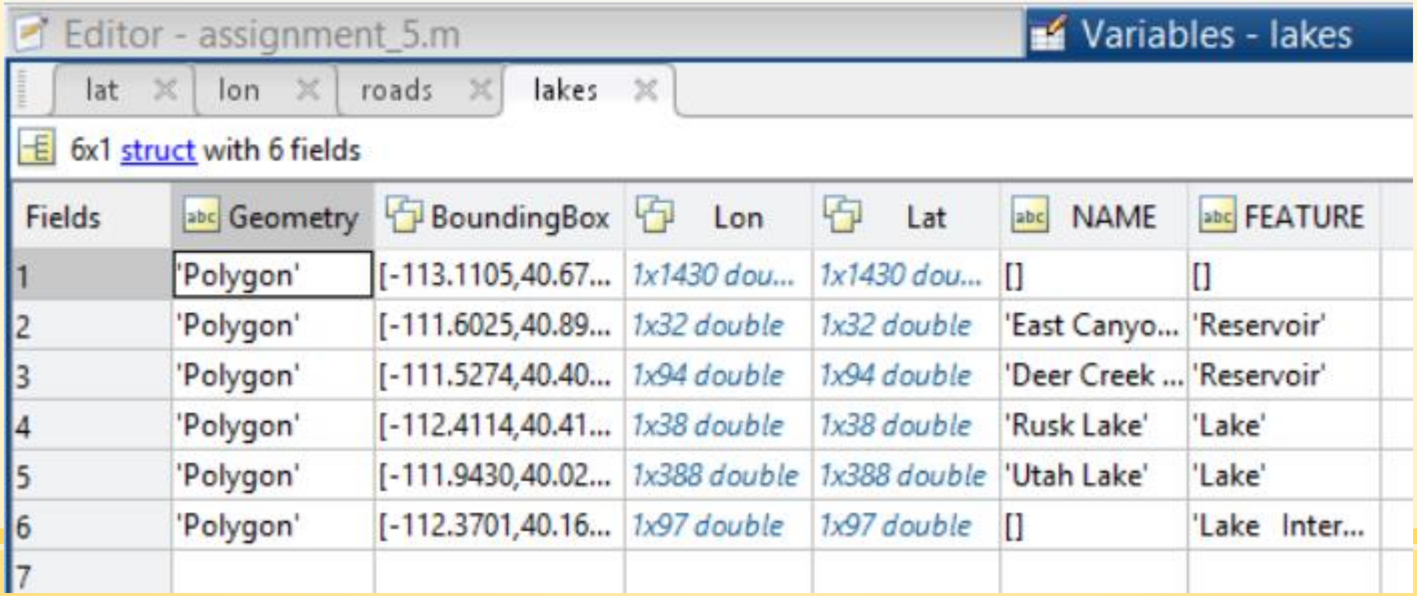
Can also set viewing direction by selecting a point in space from which to view plot

- Command has form `view([x y z])`
  - `x`, `y`, and `z` are the coordinates of the point
  - Viewing direction is direction from specified point to origin of coordinate system
  - Viewing direction independent of distance to origin, e.g., view is same with point [6 6 6] as with point [10 10 10]
  - Set top view with [0 0 1]
  - Set side view of *x*-*z* plane from negative *y* with [0 –1 0]

```
figure(3)
subplot(4,1,1)
surf(Z)
xlabel('columns of Z'); ylabel('rows of Z')
view(2)
subplot(4,1,2)
surf(X,Y,Z)
xlabel('X'); ylabel('Y')
view([2])
subplot(4,1,3)
surf(Y',X',Z')
ylabel('X'); xlabel('Y')
view([2])
subplot(4,1,4)
surf(X,Y,Z)
xlabel('X'); ylabel('Y')
zlabel('Z')
view([0 -1 0])
```

GIS shapefiles

- Shapefiles describe vector features: points, lines, and polygons
  - Shapefile shape format (.shp)
  - Shapefile shape index format (.shx)
  - Shapefile attribute format (.dbf)

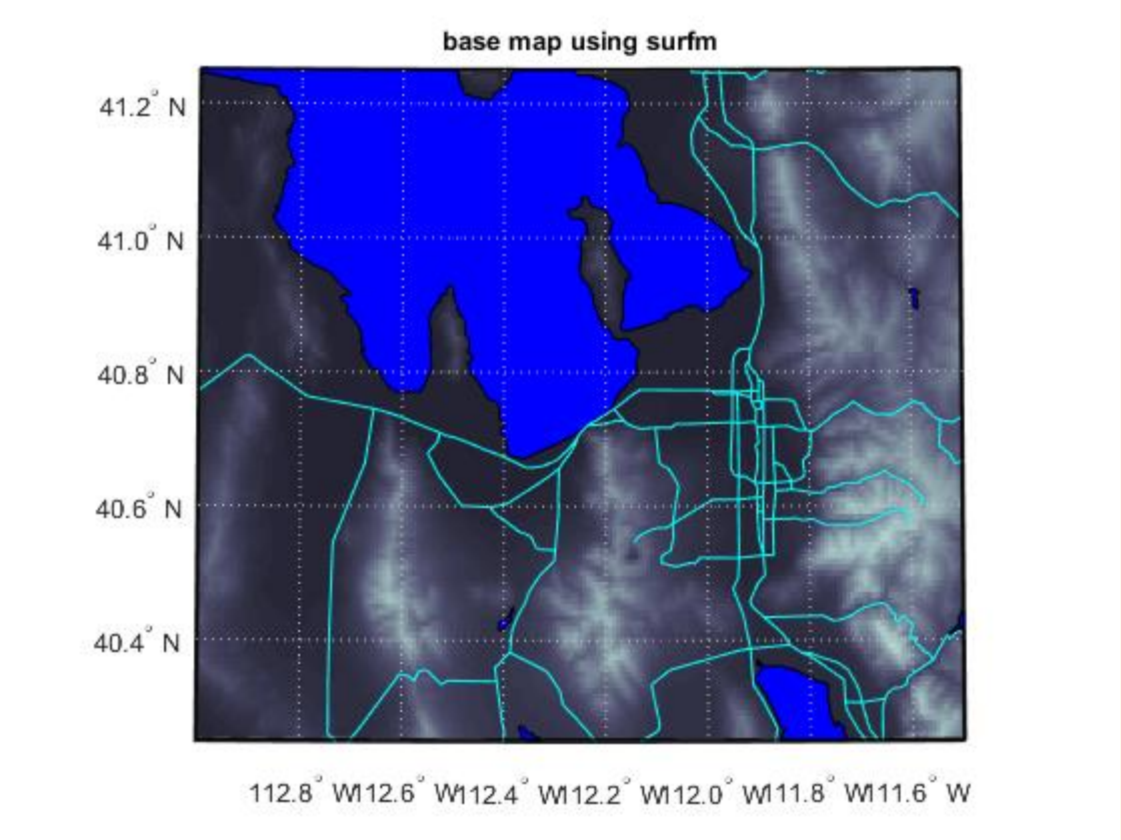


6x1 struct with 6 fields

Fields	Geometry	BoundingBox	Lon	Lat	NAME	FEATURE
1	'Polygon'	[-113.1105,40.67...	1x1430 dou...	1x1430 dou...	[]	[]
2	'Polygon'	[-111.6025,40.89...	1x32 double	1x32 double	'East Canyo...	'Reservoir'
3	'Polygon'	[-111.5274,40.40...	1x94 double	1x94 double	'Deer Creek ...	'Reservoir'
4	'Polygon'	[-112.4114,40.41...	1x38 double	1x38 double	'Rusk Lake'	'Lake'
5	'Polygon'	[-111.9430,40.02...	1x388 double	1x388 double	'Utah Lake'	'Lake'
6	'Polygon'	[-112.3701,40.16...	1x97 double	1x97 double	[]	'Lake Inter...
7						

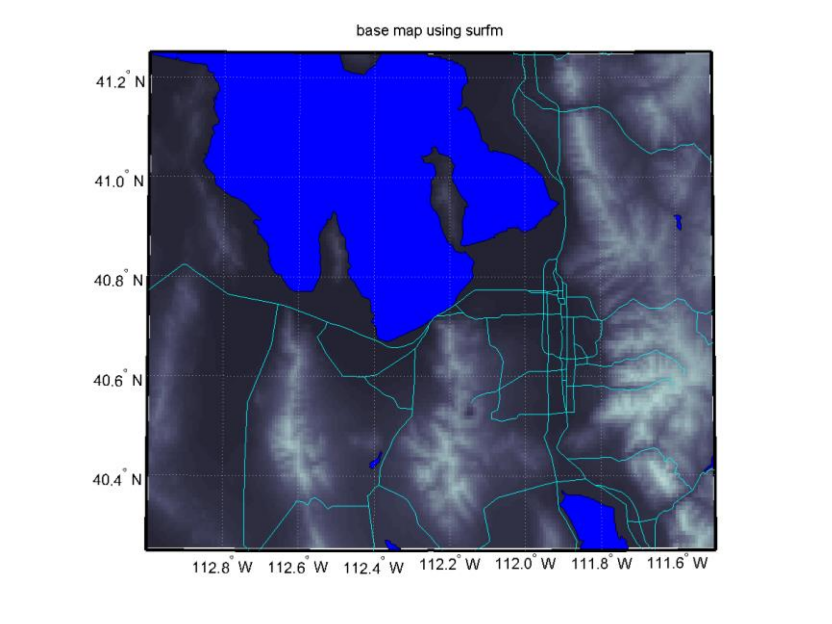
Netcdf

- NetCDF (Network Common Data Form): software self-describing, machine-independent data formats to support creation, access, and sharing of array-oriented scientific data
- http://www.unidata.ucar.edu/software/netcdf/


base map using surfm
41.2° N
41.0° N
40.8° N
40.6° N
40.4° N
112.8° W
112.6° W
112.4° W
112.2° W
112.0° W
111.8° W
111.6° W

High resolution elevation data surrounding Salt Lake radar in netcdf format

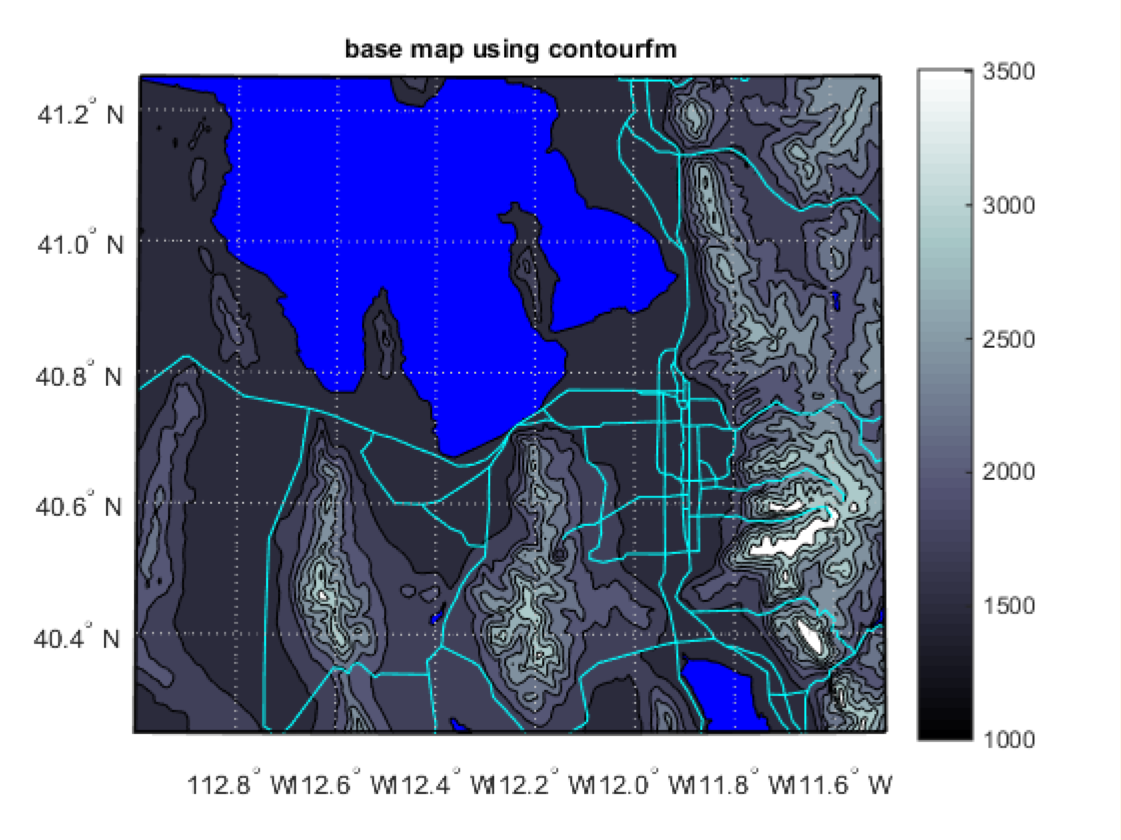
- Obtain lat's (588), lon's (804) and elevations (804x588)
- Matlab has lat's as rows and lon's as columns, so need to take transpose



usamap and setm

- ax= usamap(latlim, lonlim) assigns axis handle for Lambert Conformal map axes for a region of the U.S. defined by its latitude and longitude limits in degrees.
  - latlim and lonlim are two-element vectors of form [southern_limit northern_limit] and [western_limit eastern_limit]
- setm: set properties of map axes and graphics objects

Contourfm: slower plotting than surfm



Thursday Sept 22 : What you should be doing:

- Finish assignment 4. Due Friday at noon
- Complete/redo Assignment 3 by Tuesday 3:30
- Last Check Your Understanding: Chapter 10 next Tuesday
- Stay tuned for info on Assignments 5 & 6 (last ones!)
- covering Chapter 10 plotting