Lab 6

Plotting and fitting with ARGO data

Objectives: plot realistic data in 2D (oxygen from an ARGO float) and fit a linear model to a subset

Marine Modelling February 11, 2019

Katja Fennel
Oceanography
Dalhousie University

New Commands Overview

- `find`: as the name suggests finds things (data that satisfy given logical requirements)
- `pcolor`: plots 2D data in color
- `datevec, datenum, datestr`: are useful when dealing with dates
- `set, get`: allow you to customize plots

FIND

The `find` command finds a list of positions (indices) of the elements of a vector that satisfy a given logical condition.

For example:

```
>> x = -1:.01:1;
>> y = sin(5*pi*x).*exp(-x.^2);
>> plot(x,y,’:’)
>> k = find(y > 0.2);
>> hold on
>> plot(x(k),y(k),’o’)
```

pcolor

`pcolor`

is useful for plotting 2D data. Suppose matrix C contains some 2D variable, e.g., some property over two spatial dimensions (bathymetry, surface ocean chlorophyll or temperature) or over time and space dimensions (temporal evolution of a vertical profile of nutrients etc.), then `pcolor(C)` will make a 2D color plot

For example:

```
>> pcolor(peaks) % peaks is a predefined surface
```

pcolor

`pcolor`

is useful for plotting 2D data. Suppose matrix C contains some 2D variable, e.g., some property over two spatial dimensions (bathymetry, surface ocean chlorophyll or temperature) or over time and space dimensions (temporal evolution of a vertical profile of nutrients etc.), then `pcolor(C)` will make a 2D color plot

For example:

```
>> pcolor(peaks) % peaks is a predefined surface
```

Note: Grid lines will be plotted by default (this option is called `'faceted'`, but can be turned off with command `shading interp`). Try it!

```
>> shading interp
```

A more general way of calling pcolor is by providing X and Y data as well

```
>> pcolor(X,Y,C) % X and Y can be vectors or matrices
```

dates

datenum

converts a date into a single number (called date number; it’s essentially julian day); e.g.

```
>> datenum('19-May-2000')  % ans = 730625.
>> datenum(2001,12,19) % ans = 731204.
>> datenum(2001,12,19,18,0,0) % ans = 731204.75.
```

dates

datenum

converts a date into a single number (called date number; it's essentially julian day); e.g.

```
>> datenum('19-May-2000')  % ans = 730625.
>> datenum(2001,12,19) % ans = 731204.
>> datenum(2001,12,19,18,0,0) % ans = 731204.75.
```

`datevec` goes the opposite direction, e.g.

```
>> datevec(731204)
ans =    2001    12   19    0      0      0
>> datevec(731204.75)
ans =    2001    12   19   18      0      0
```

dates

`datestr`

turns date numbers into strings (useful for titles and labels on plots), e.g.

```
>> datestr(731204.75) % ans =19-Dec-2001 18:00:00
>> datestr(731204) % ans = 19-Dec-2001
```

dates

datestr

turns date numbers into strings (useful for titles and labels on plots), e.g.

```
>> datestr(731204.75) % ans =19-Dec-2001 18:00:00
>> datestr(731204) % ans = 19-Dec-2001
```

Note: There many different predefined formats, e.g.

```
>> datestr(731204) % ans =19-Dec-2001
>> datestr(731204,2) % ans =12/19/01
>> datestr(731204,3) % ans =Dec
>> datestr(731204,4)% ans =D
>> datestr(731204,5) % ans =12
```

Look at `help datestr` for a complete list.

set

`set`

allows you to change the appearance of your plots by changing the values of properties of your plot objects, e.g. the axes or figure.

General syntax:

```
set(h,’PropertyName’,PropertyValue)
```

or

```
set(h,’PropertyName1’,PropertyValue1,...
’PropertyName2’,PropertyValue2,...)
```

set

`set`

allows you to change the appearance of your plots by changing the values of properties of your plot objects, e.g. the axes or figure.

General syntax:

```
set(h,'PropertyName',PropertyValue)
```

or

```
set(h,'PropertyName1',PropertyValue1,...
'PropertyName2',PropertyValue2,...)
```

Example:

```
set(gcf,'DefaultTextColor','red')
set(gca,'XTick',[0 10 100])
```

Note: `get(h)` will display a complete list of properties and current values of h In the above example `gca` stands for the axis object, `gcf` for the figure object.

Recall: `linfit.m`

Requires three vector arguments: *x*, *y* and *sy* (the σ_{y_i}s)

```
[a,sa,cov,r] = linfit(x,y,sy);
```

Returned are the slope and intercept (in vector *a*), the uncertainties in those coefficients (*sa*), the covariance of slope and intercept (*cov*), and the correlation coefficient *r*

ARGO data

ARGO Program website: http://www.argo.ucsd.edu

Data from http://www.nodc.noaa.gov/argo/floats_data.htm for float 4900093 (a float with oxygen sensor).

Data was in netcdf format (which can be imported to MATLAB if the right toolboxes are installed; latest MATLAB release contains netcdf toolbox per default). Converted into MATLAB format (filename: `4900093.mat`).

Look at your data sheet handouts for the format of the given variables, especially pres, time, doxy, level.

Loading and QC

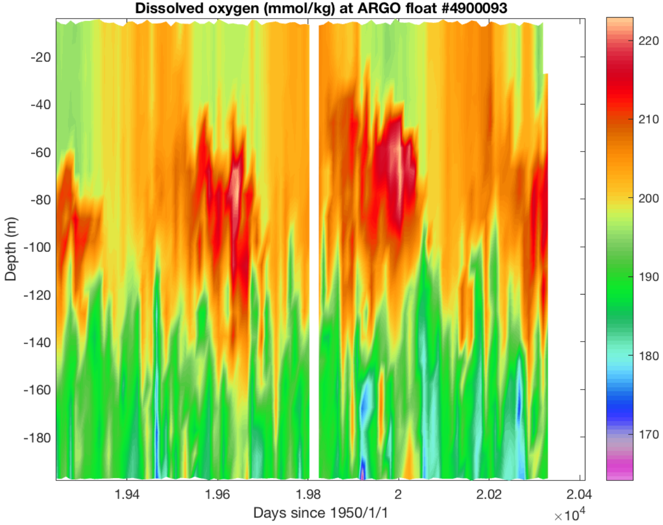
```
% 1. load data
load 4900093

% 2. plot oxygen data in the top 200 m
%  (levels 1 - 21)
% 2.1 restrict variables to layers of
% interest only
pres = pres(:,1:21);
doxy = doxy(:,1:21);

% 2.2 quality control pressure and oxygen
% data (replace 99999 with NaNs)
bad = find(pres==99999);
pres(bad) = nan;
doxy(bad) = nan;
```

More QC

```
% 2.3 test min and max of both variables;
%  replace unrealistic values with
%  NaNs if necessary
min(doxy(:))
max(doxy(:))
% remove negative oxygen values
bad = find(doxy<0);
doxy(bad) = nan;
```

Plot

```
% 2.4 plot and label axis
figure
pcolor(time*ones(1,21),-pres,doxy);
shading interp
colorbar
title([’Dissolved oxygen (mmol/kg) at ’ ...
        ’ARGO float #4900093’])
ylabel(’Depth (m)’)
xlabel(’Days since 1950/1/1’)
```


Dissolved oxygen (mmol/kg) at ARGO float #4900093
Depth (m)
-20
-40
-60
-80
-100
-120
-140
-160
-180
1.94
1.96
1.98
2
2.02
2.04
×10^4
Days since 1950/1/1
220
210
200
190
180
170

Time axis

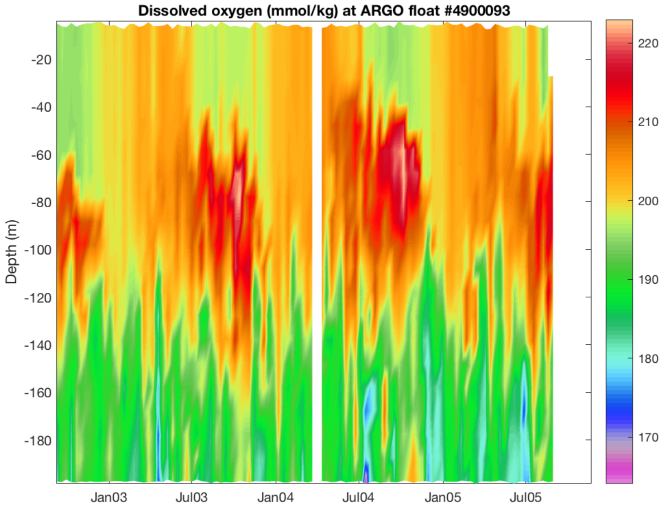
```
% Note that our time axis is not very useful.
% Use the commands datevec, datenum
% and datestr to create tick-labels
% with dates:

% first: covert the days after 1950-1-1 to
%  absolute days since Jan-1-0000
N0 = datenum([1950 1 1]);
Ntimes = time+N0;

% When does our time series begin and end?
date_beg = datevec(Ntimes(1))
date_end = datevec(Ntimes(end))
```

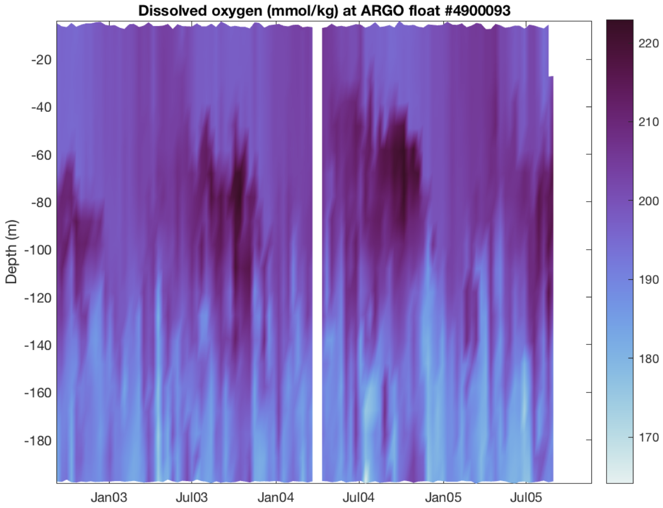
Time axis

```
% suppose you want tick-marks at Jan-1 and July-1
% of 2003 to 2005
% create vector of N values for those dates
tick_dates = [2003 1 1; 2003 7 1;
              2004 1 1; 2004 7 1;
              2005 1 1; 2005 7 1];
Nticks = datenum(tick_dates);
Nticklabels = datestr(Nticks,12);
```

Time axis

```
% suppose you want tick-marks at Jan-1 and July-1
% of 2003 to 2005
% create vector of N values for those dates
tick_dates = [2003 1 1; 2003 7 1;
              2004 1 1; 2004 7 1;
              2005 1 1; 2005 7 1];
Nticks = datenum(tick_dates);
Nticklabels = datestr(Nticks,12);

% plot again:
figure
pcolor(Ntimes*ones(1,21),-pres,doxy); shading interp
colorbar
title([’Dissolved oxygen (mmol/kg) at ’ ...
         ’ARGO float #4900093’])
ylabel(’Depth (m)’)
set(gca,’XTick’,Nticks,’XTickLabel’,Nticklabels)
```


Dissolved oxygen (mmol/kg) at ARGO float #4900093
Depth (m)
-20
-40
-60
-80
-100
-120
-140
-160
-180
Jan03
Jul03
Jan04
Jul04
Jan05
Jul05
220
210
200
190
180
170


Dissolved oxygen (mmol/kg) at ARGO float #4900093
Depth (m)
-20
-40
-60
-80
-100
-120
-140
-160
-180
Jan03
Jul03
Jan04
Jul04
Jan05
Jul05
220
210
200
190
180
170

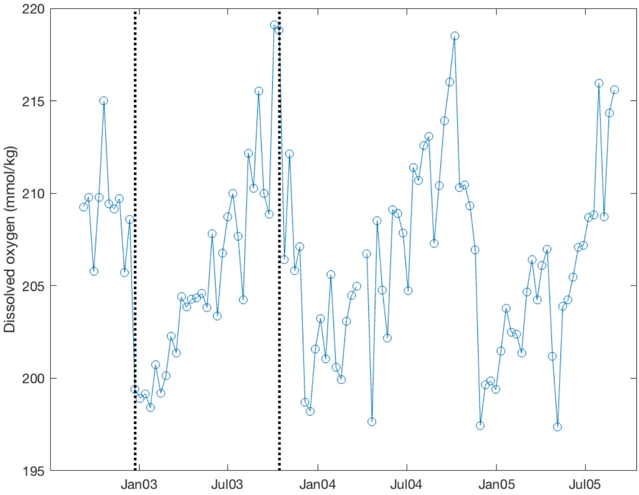
Next: linear fit

Loading and QC

```
% 1. load data
load 4900093

% 2. plot oxygen data at 90 m depth (level 10)
% 2.1 restrict variables to layer of interest
pres = pres(:,10);
doxy = doxy(:,10);

% 2.2 quality control pressure and oxygen data
% (replace 99999 with NaNs)
bad = find(pres==99999);
pres(bad) = nan;
doxy(bad) = nan;

% 2.3 remove negative oxygen values
bad = find(doxy<0);
doxy(bad) = nan;
```

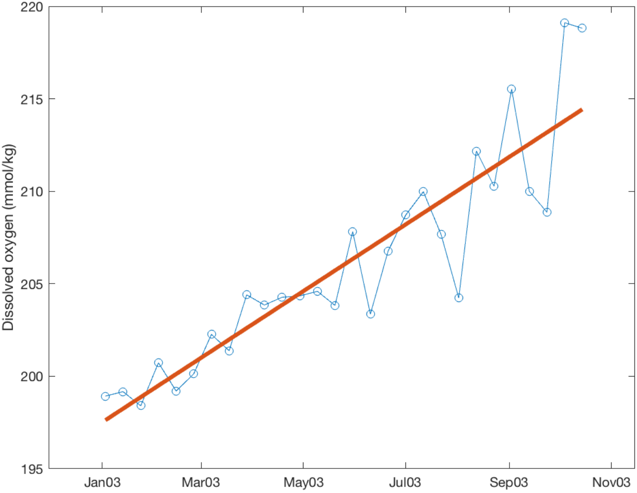
Plot subset

```
% 2.4 plot
figure
plot(time,doxy,'.:')
% do fit from time = 19350 to 19645
hold on
plot([19350 19350],[195 220],'k:')
plot([19645 19645],[195 220],'k:')
```


220
215
210
205
200
195
Dissolved oxygen (mmol/kg)
Jan03
Jul03
Jan04
Jul04
Jan05
Jul05

Linear fit

```
% 3. determine linear fit for first summer
% 3.1 prepare x and y vectors by picking only
% elements within the desired timeframe
ind = find(time>=19350 & time<= 19645);
x = time(ind);
y = doxy(ind);
% 3.2 do unweighted fit
a = linfit(x,y,0);
% 3.3 plot result
figure
plot(x,y,’.:’)
hold on
plot(x,a(1)+a(2)*x,’r-’)
% Question: What is the unit on a(2)?
```


220
215
210
205
200
195
Dissolved oxygen (mmol/kg)
Jan03
Mar03
May03
Jul03
Sep03
Nov03