Variance Ellipses

In this lab, you'll be working with current meter data, that is, observations of the horizontal ocean currents made from a fixed point. The main analysis tool we will learn are variance ellipses, a fundamental tool for analyzing velocity datasets or any kind of bivariate (or complex-valued) data. You'll see their relationship to the two-dimensional histogram, and will learn some associated analysis and visualization ideas, in particular the value of finding a suitable coordinate rotaiton.

For this assignment, we are going to use a mooring from the Labrador Current on the west side of the Labrador Sea known as the 'm1244' mooring. Please download my version of it here and put it on your Matlab search path if you have not done so already. (This is included in the full distribution of the course materials.)

This notebook requires my jlab toolbox to be installed. You will also need to have set up Jupyter Lab to work with Matlab, following these instructions.

A Quick Look at the Data

Now let's load and examine the data.

In [1]:
```
set(groot,'defaultfigurepaperposition',[1 1 7 5]) %set default figure size

clear
load m1244
m1244
```

```
m1244 =
  struct with fields:

    description: '1244 mooring 1994-1995'
           link: 'https://www.nodc.noaa.gov/woce/woce_v3/wocedata_1/cmdac/netcdf/acm29/acm29.htm'
        creator: 'Created by the L_M1244 script by J.M. Lilly'
      timestamp: '18-Oct-2018 05:40:00'
            lat: 55.478599548339844
            lon: -53.654998779296875
         depths: [201 1001 1501 2751]
            num: [7371x1 double]
              p: [7371x4 double]
              t: [7371x4 double]
             cv: [7371x4 double]
```

The data is organized as a structure with different fields. The field variables are as follows:

depths --- Depths of 4 different currents meters, in meters
num -- date in Matlab's "datenum" format
p --- Pressure in decibar
t --- Temperature in Centigrade
cv --- Complex velocity u+iv in cm/s, u = eastward, v = northward

Ordinarily, you would access these with the notation "m1244.cv", and so forth. However, jLab has a function called "use" that will map all the fields in a structure into variables in memory.

In [2]:
```
use m1244
whos
```

```
  Name                Size              Bytes  Class     Attributes

  creator             1x43                 86  char
  cv                7371x4             471744  double    complex
  depths              1x4                  32  double
  description         1x22                 44  char
  lat                 1x1                   8  double
  link                1x78                156  char
  lon                 1x1                   8  double
  m1244               1x1               1004678  struct
  num               7371x1                58968  double
  p                 7371x4              235872  double
  t                 7371x4              235872  double
  timestamp           1x20                 40  char
```

As you can see all of the fields now exist as variables. This makes it convenient to have different datasets with the same variable names, and swap out which one we are working with at the

moment, thus keeping our variable names short and consistent.

Before we proceed let's find out the sampling interval and duration of the time series.

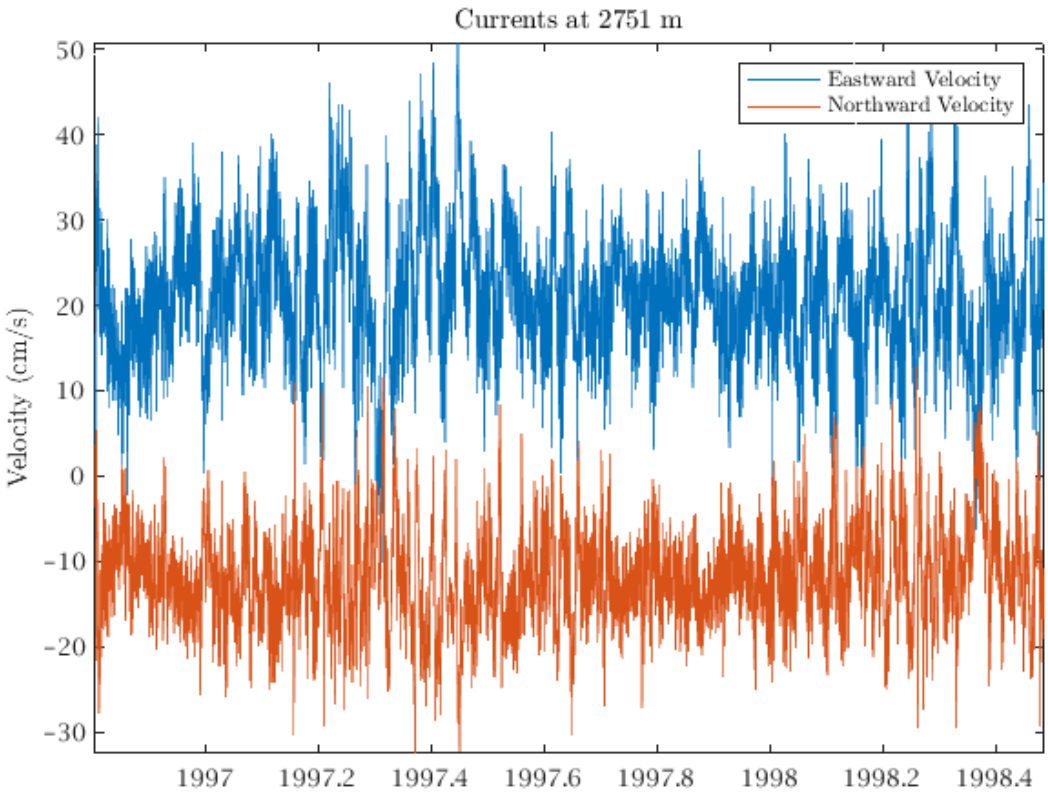
In [3]:
```
(num(2)-num(1))*24    %Sampling interval in days converted to hours
(num(end)-num(1))/365 %Duration in days converted to years
```

```
ans =
   2.000000000931323
ans =
   1.682648401826378
```

So we have a roughly 1.7 year record of the currents sampled every two hours.

Let's take a look at the currents the deepest depth. Note that I often like to use year.fraction as a simple time axis for time periods of a year or more.

In [4]:
```
%Note that I'll be adjusting the 'paperposition' property to have the figures display
uvplot(yearfrac(num),cv(:,4)),axis tight
%uvplot is a jlab function that plots the real and imaginary part of a complex variabl
legend('Eastward Velocity','Northward Velocity')
ylabel('Velocity (cm/s)')
title(['Currents at ' int2str(depths(4)) ' m'])
```



Here a mean value is readily apparent in both the eastward and northward components. In addition, we see a lot of small-scale variability, possibly with a greater amplitude in the eastward component. Hints of multiple timescales of variability are present, with a fine noise-like variability superposed on somewhat longer timescales.

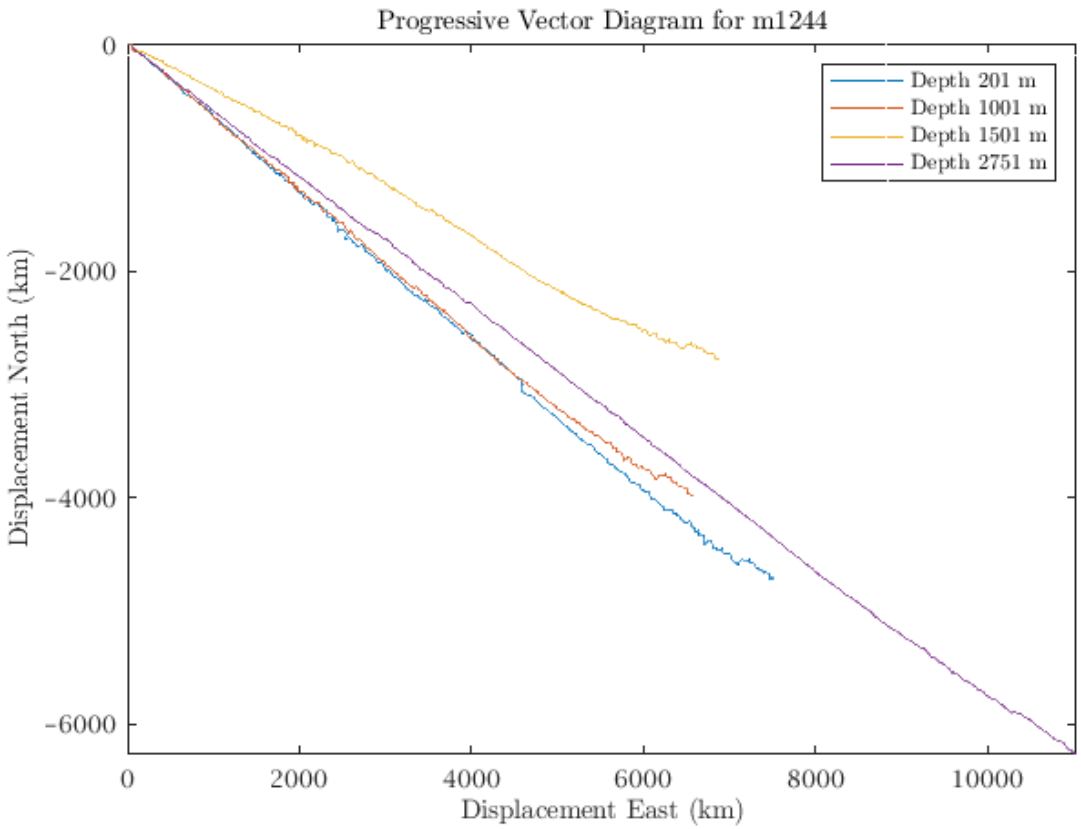
It is a little difficult to make sense of this plot, however, because it is not rotated into the most useful coordinate system. Let's figure out a sensible rotation angle.

Choosing a Coordinate Rotation

To do this, we will look at the progressive vector diagram. For a current measurement at a point, the progressive vector diagram is defined as a plot of its time integral. In other words, the progressive vector diagram shows the displacement that would occur if a particle were advected with the given currents.

In [5]:

```
dt=num(2)-num(1);    %sampling interval in days
dt=dt*3600*24;       %sampling interval in seconds
cx=cumsum(cv)*dt;    %complex displacement x+iy in centimeters
cx=cx/100/1000;      %complex displacement x+iy in kilometers

plot(cx)   %Matlab plots the real and imaginary parts against each other
xlabel('Displacement East (km)')
ylabel('Displacement North (km)')
xtick([-10:2:10]*1000),ytick([-10:2:10]*1000),axis tight

%Create a legend for the figure lines
clear str
for i=1:4
    str{i}=['Depth ' int2str(depths(i)) ' m'];%use a cell array for the labels
end
legend(str)
title('Progressive Vector Diagram for m1244')
```

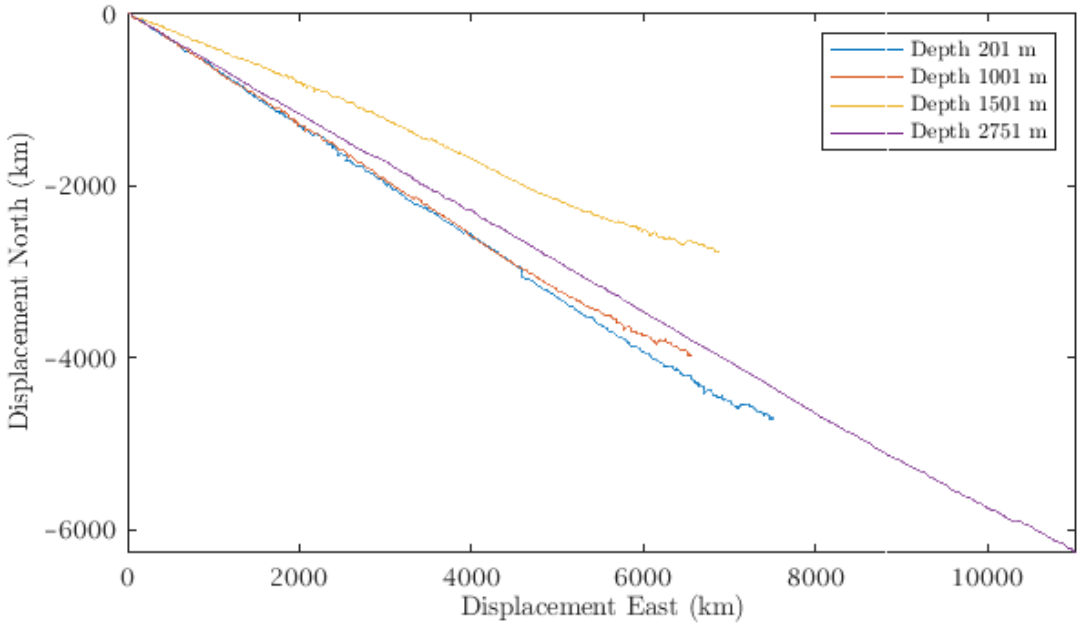


Here, the currents appear to be directed due southeast. However, there is a problem with this plot: the x and y axes, which are the same physical quantity of displacement, correspond to different physical lengths! If we re-scaled the figure window, the direction of the mean flow would appear to change. Thus our perspective on the comparison between the eastward and northward currents is distorted.

This problem is fixed by setting the data aspect ratio. Not setting the data aspect ratio correctly is actually a common problem, even, regrettably, in many published papers.

We'll now remake the above figure with the correct aspect ratio.

In [6]:
```
set(gcf,'paperposition',[1 1 7 4]) %set suitable figure size
plot(cx)   %Matlab plots the real and imaginary parts against each other
xlabel('Displacement East (km)')
ylabel('Displacement North (km)')
xtick([-10:2:10]*1000),ytick([-10:2:10]*1000),axis tight
axis equal %This sets it the data aspect ratio to 1:1   <--------------------

%Create a legend for the figure lines
clear str
for i=1:4
    str{i}=['Depth ' int2str(depths(i)) ' m']; %use a cell array for the labels
end
legend(str)

%In jlab the progressive vector diagram with the correct aspect ratio can also
%be plotted with the function "provec".
```



Now we see a strong mean flow at all depths directed to the east-southeast. This matches what we saw above with the line plot, where the positive mean of the eastward currents is roughly twice as large as the negative mean of the southward currents.

Rotating by the Mean Flow Direction

It's natural to rotate our current meter data so that the east-southeastward direction of the mean flow corresponds to the first velocity component, and the direction normal to that to the second velocity component. In other words, we will rotate our (u,v) data to become $)\tilde u,\tilde v)$ with $\tilde u$ corresponding to the 'downstream' direction and $\tilde v$ corresponds to the 'cross-stream' direction. We will take a little time to do this so we understand how rotations work.

Firstly we find the direction of the mean flow at the deepest current meter, where the flow is strongest.

In [7]:
```
angle(mean(cv(:,4)))*360/2/pi  %angle in radians converted to degrees
```

```
ans =
 -29.602806827016341
```

So the angle is about 30 degrees clockwise from due east. That looks about right!

Rotation of a complex-valued number $z=u+iv$ are straightforward. To rotate $z=|z|e^{i\varphi}$ through some angle ϕ, we simply multiply by $e^{i\phi}$. This leads to a new version of z , denoted $\tilde z =\tilde u+ i\tilde v=|z|e^{i(\varphi+\phi)}$.

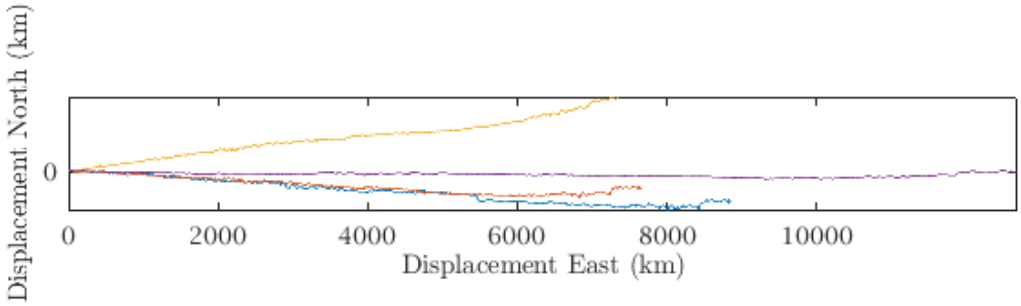
In this case, we want to choose the rotation angle ϕ as the *negative* of the angle of the mean flow. Let's look at the mean real and imaginary components of the velocity before and after this rotation.

In [8]:
```
mean(cv(:,4))
phi=-angle(mean(cv(:,4)));
mean(cv(:,4)*exp(1i*phi))
%note in Matlab, it's best to use 1i and not i for sqrt(-1), since we often use i for
```

```
ans =
 20.785806533575666 -11.809328448895153i
ans =
 23.906275152459880 - 0.000000000000001i
```

This shows that after the rotation, the mean of the cross-stream velocity (the imaginary part) is zero, as expected. Now let's re-plot the progressive vector diagram.

In [9]:
```
set(gcf,'paperposition',[1 1 7 2]) %set suitable figure size
plot(cx*exp(1i*phi))   %Matlab plots the real and imaginary parts against each other
xlabel('Displacement East (km)')
ylabel('Displacement North (km)')
xtick([-10:2:10]*1000),ytick([-10:2:10]*1000),axis tight
axis equal %This sets the data aspect ratio to 1:1
```

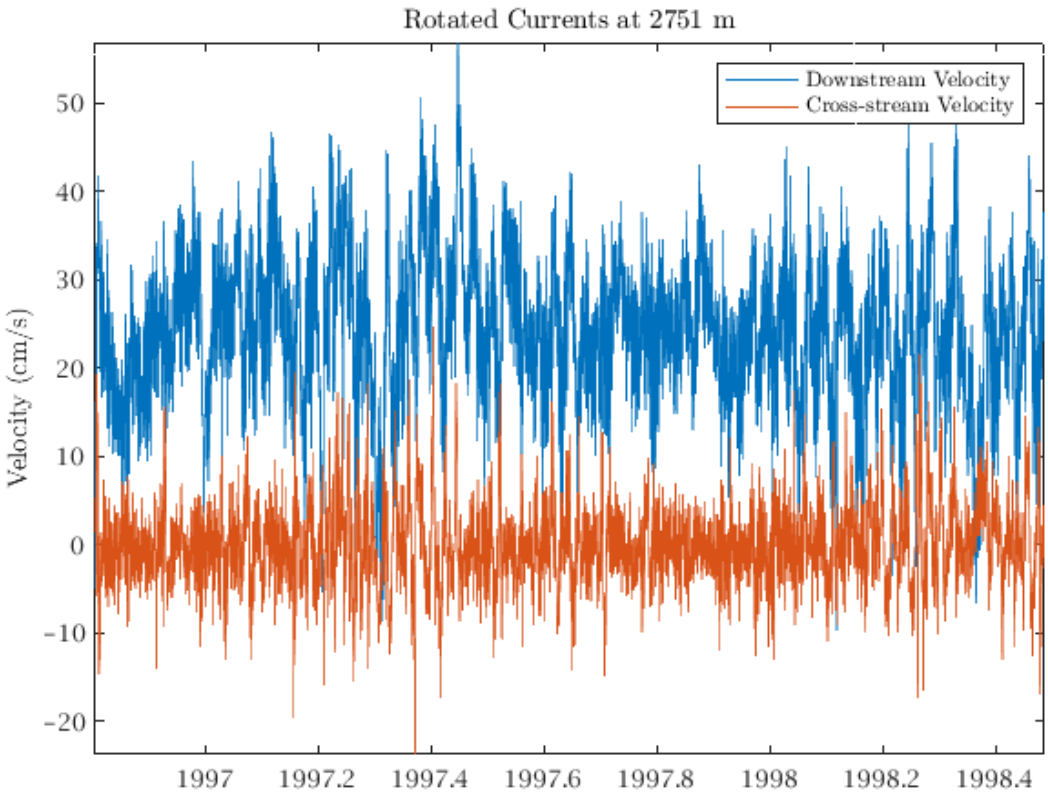


At first glance, this plot appears rather boring. But actually, boring can be a good sign because it means we've found a way to look at our dataset in such a way that it simplifies!

Let's return to the line plot we made earlier, but now make it for the rotated velocity data.

In [10]:
```
rcv=ml244.cv*exp(1i*phi);  %Define a rotated version of the velocity
uvplot(yearfrac(num),rcv(:,4)),axis tight,
legend('Downstream Velocity','Cross-stream Velocity')
```

```
ylabel('Velocity (cm/s)')
title(['Rotated Currents at ' int2str(depths(4)) ' m'])
```

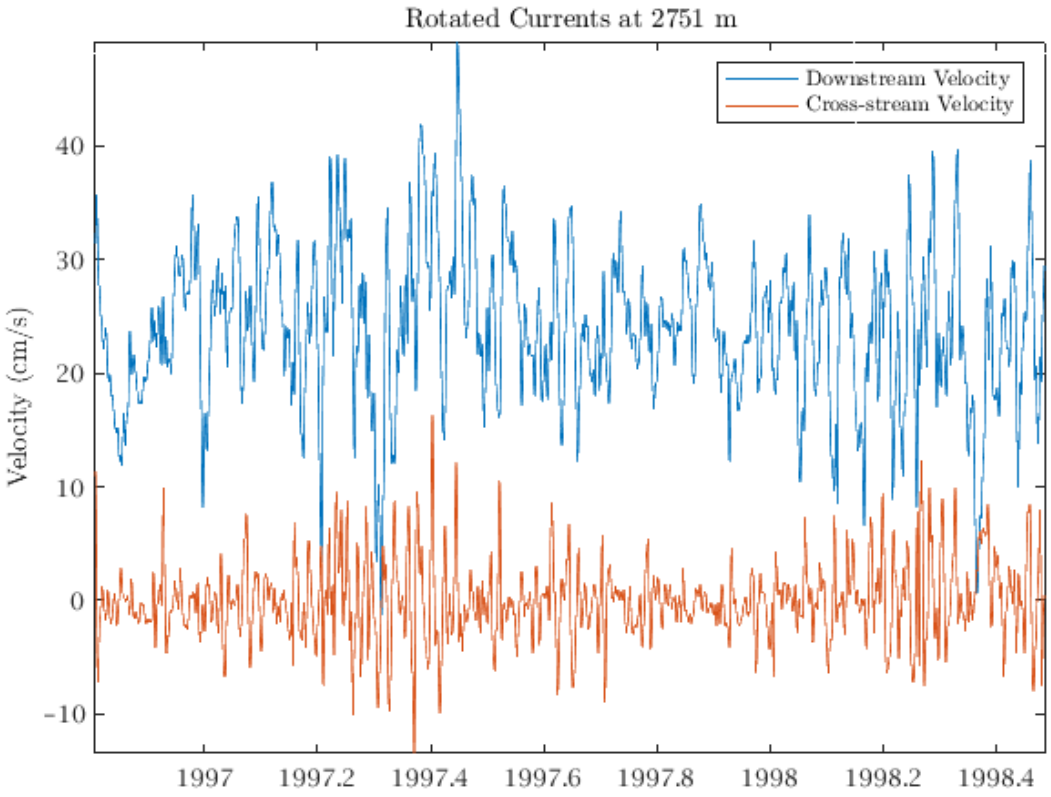


The downstream and cross-stream components are seen to be distinctly different. The cross-stream flow oscillates about a mean of zero, which is by construction, and also presents higher-frequency variability than does the downstream flow. Or, perhaps more accuately, it appears that the cross-stream flow is *lacking* an intermediate-timescale component that is present in the downstream flow.

Let's take a closer look at the timescales present in this dataset with a simple lowpass filter. We'll use a 24-point filter, which corresponds to a 2-day running mean since the sampling interval is 2 hours.

In [11]:

```
uvplot(yearfrac(num),vfilt(rcv(:,4),24)),axis tight
%vfilt is a jlab routine for filtering that uses, by default, an N-point Hanning filte
legend('Downstream Velocity','Cross-stream Velocity')
ylabel('Velocity (cm/s)')
title(['Rotated Currents at ' int2str(depths(4)) ' m'])
```



Here we can see clearly that the downstream flow presents an intermediate-timescale variability that is not present in the cross-stream flow.

Our rotation, designed just based on the *mean*, has thus also revealed distinctions in *variability*. The variability of the flow is now seen as being *anisotropic*, that is, lacking the property of being the same in all directions. This was not visible before the rotation because the downsteam and cross-stream components were mixed.

In general, finding ways to "rotate" a dataset, perhaps in an abstract way, in such a way that variability presents anisotropic structure is a quite simple yet powerful approach we can use to unlock its information content.

Here, the different timescales between the two components is not what one would expect if the variability were entirely due to the advection of eddies past the mooring; the timescales in that case would be the same. So this plot informs the physical hypotheses we would frame regarding the nature of the variability.

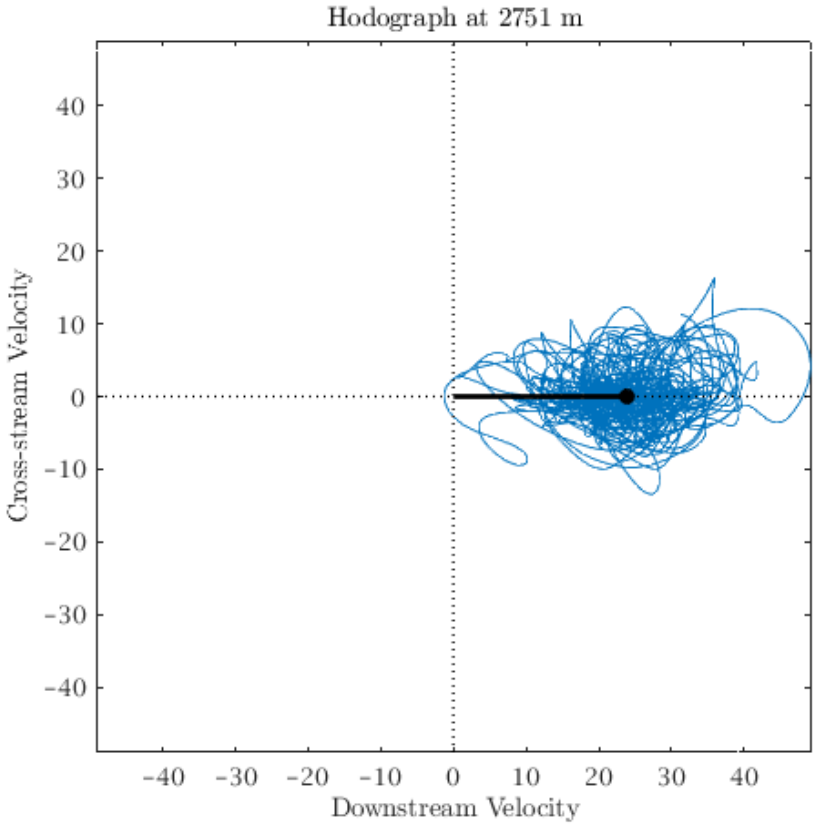
Two-Dimensional Histograms

Next we will summarize the statistics of the velocity through looking at its properties on the u,v plane. First, we'll make a simple line plot of $\tilde u$ versus $\tilde v$---a type of plot known as a *hodograph*---for the rotated versions of the currents we created earlier. We'll work with the lowpassed version of the time series for the moment.

In [12]:

```
plot(vfilt(rcv(:,4),24))
axis(49*[-1 1 -1 1]),xtick([-40:10:40]),ytick([-40:10:40])
vlines(0,'k:'),hlines(0,'k:')
```

```
%Next, plot the mean as a heavy line ending in a circle
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'ko','markerfacecolor','k')
axis equal %Don't forget to set the aspect ratio!
axis tight
xlabel('Downstream Velocity'),ylabel('Cross-stream Velocity')
title(['Hodograph at ' int2str(depths(4)) ' m'])
```



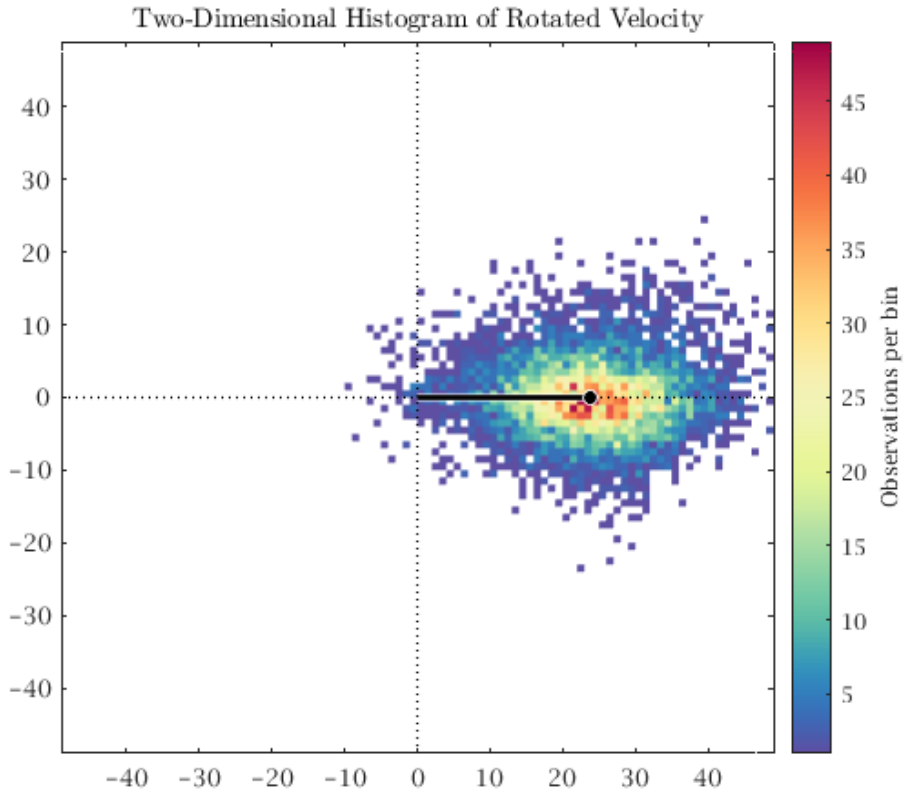
The mean flow is plotted for reference. Note that the time integral of this plot is the progressive vector diagram.

At first glance, this plot is not very informative. It basically tells us that there is varability about the mean, with greater variability along the downstream axis then the cross-stream axis, which we already knew. But we can use statsitics to nicely summarize what we are seeing.

To do this we will look at two-dimensional distributions of the velocity. Firstly, we plot the two-dimensional histogram at the deepest depth.

In [13]:

```
%using two2hist, a jlab routine for finding a two-dimensional histogram
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:50],[-50:50]); %the last two arguments
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
jpcolor(xmid,ymid,mat);hold on
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])
title('Two-Dimensional Histogram of Rotated Velocity')
```



This plots shows the number of observations within each bin, which we have specified to be 1×1 cm/s bins ranging from -50 to 50 cm/s in both axes. The white bins are never observed.

The histogram of the velocity has a roughly elliptical shape, elongated along the x-axis. We now have a picture of a "cloud" of possibilities, with, generally speaking, bins closer to the location of the mean flow occurring more frequently.

Note this plot contains more information than was visible in the line plot, because while the line plot also showed us the *shape* of the distribution, it did not provide us with information as to the relative frequency of occurrence of velocity values.

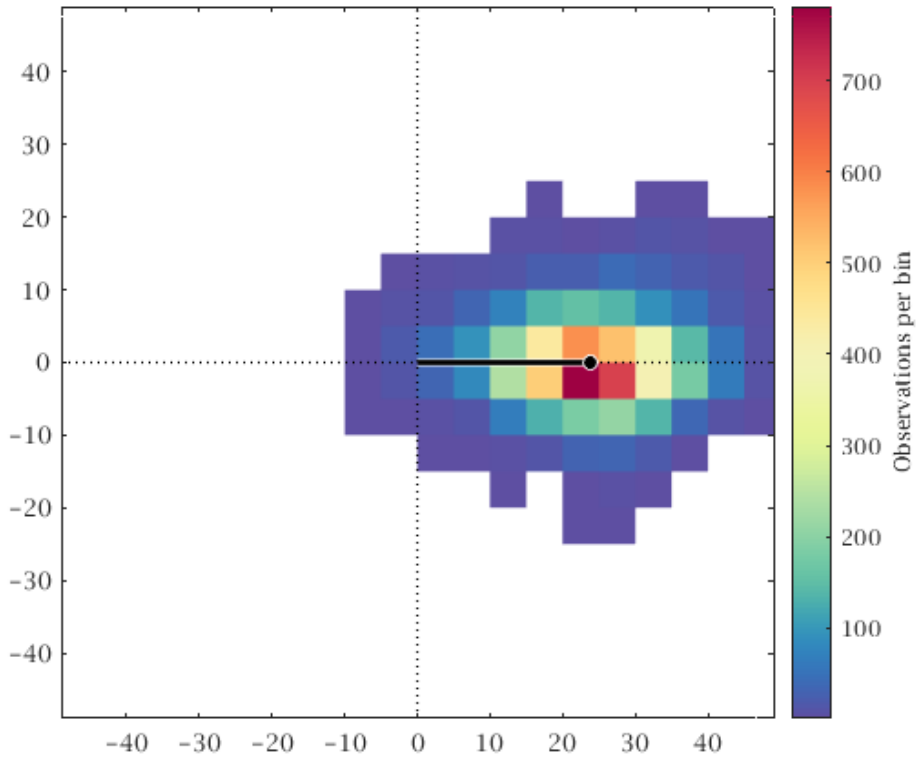
Adjusting 2D Histogram Plots

Two-dimensional statistics are an exceedingly powerly analysis tool. Before proceeding, let's take a look at how we can choose our settings to get the most out of them.

First, you'll want to pay attention to your choice of bin size.

In [14]:
```
%same plot as above but with x- and y-bins of [-50:5:50]
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:5:50],[-50:5:50]); %the last two argu
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
jpcolor(xmid,ymid,mat);hold on
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
```

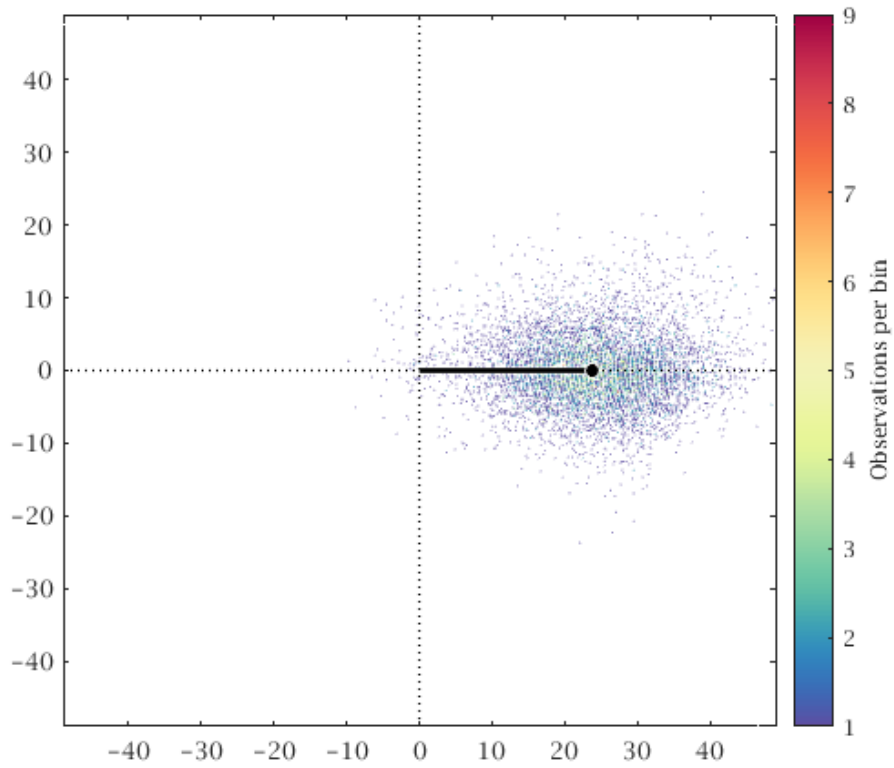
```
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])
```



Here, the bins have been chosen to be too coarse, so we can't see much structure, and the plot has a blocky look to it.

In [15]:

```
%same plot as above but with x- and y-bins of [-50:.2:50]
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:.2:50],[-50:.2:50]); %the last two ar
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
jpcolor(xmid,ymid,mat);hold on
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])
```

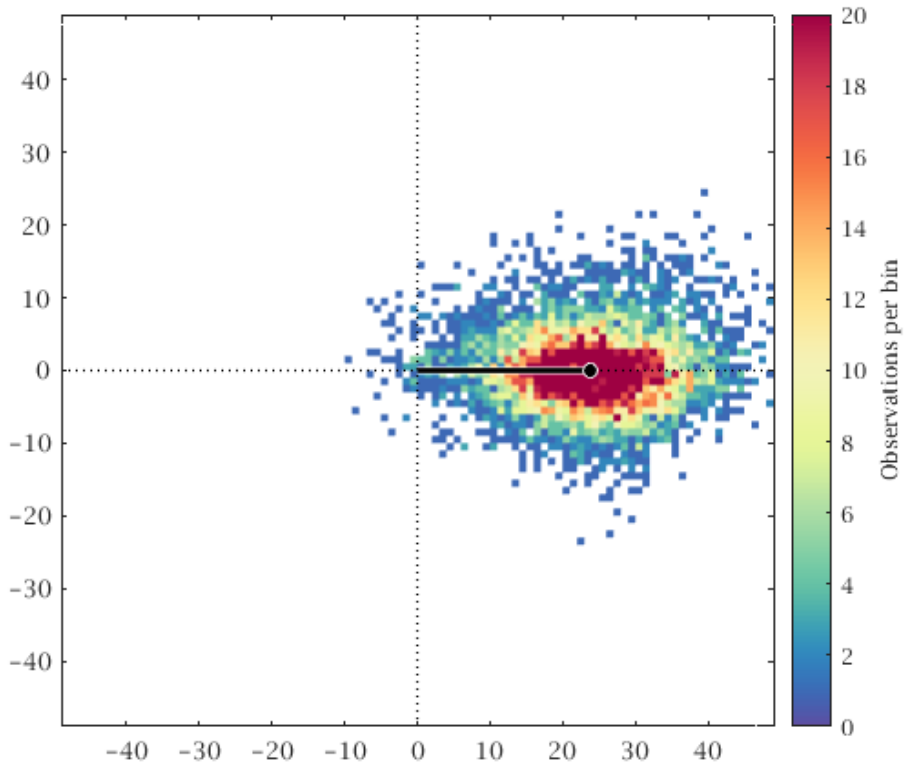


Here, the bins have been chosen to be too fine, so the "cloud" of points has been reduced to a mist, and again, we can't see much. Thus, you'll want to play around with your bin sizes to choose one that seems to reveal the most structure. The 1 cm/s bins we used earlier seem ideal for this dataset.

You'll also want to pay attention to your choice of color scale.

In [16]:

```
%same plot as above but with a different color axis
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:50],[-50:50]); %the last two arguments
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
jpcolor(xmid,ymid,mat);hold on
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])
caxis([0 20])
```

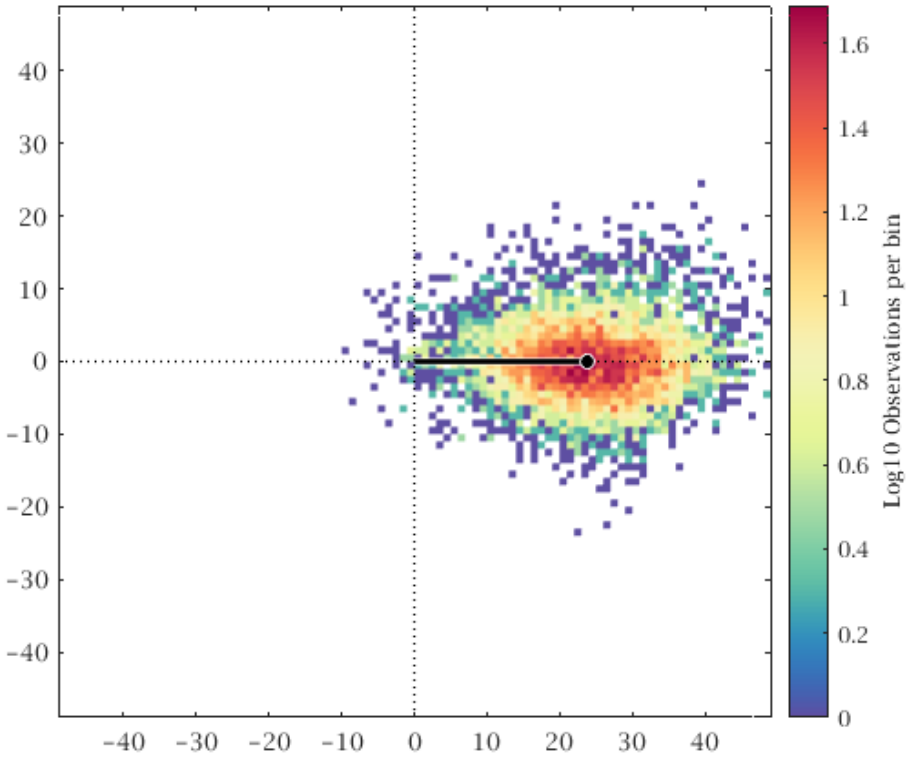


If you see a splotch of all one color in your dataset, one likely explanation is that the color axes limits have been set inappropriately, wiping out some of the information. Thus you'll want to check the color axes and see if an improvement could be made.

Finally, one useful trick is to plot not the number of observations per bin, but rather the *logarithm* of the number of observations, like so:

In [17]:

```
%same plot as above but with a logarithmic color axis
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:50],[-50:50]); %the last two argument
jpcolor(xmid,ymid,log10(mat));hold on
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Log10 Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])
```



This has the effect of allowing regions with radically different densities to be visualized together. The logarithm "flattens" the plot, since for exmaple, log10(100)=2, log10(10)=1, and log10(1)=0.

In this plot, the structure on the flanks of the distribution, where very few data points occur, now appears more visible. In general, you'll want to play around to see if linear or logarithmic histograms are more informative.

The Variance Ellipse

Finally we are ready to look at a quantity called the variance ellipse. We will study this more in the lectures; here we will just get a feeling for how it works.

First we form the terms in the covariance matrix for the rotated velocities at the deepest depth. Then, we diagonalize the covariance matrix with an eigenvalue decomposition.

In [18]:
```
u=real(rcv(:,4));u=u-mean(u);
v=imag(rcv(:,4));v=v-mean(v);
cuu=mean(u.*u);
cvv=mean(v.*v);
cuv=mean(u.*v);

[a2,b2,theta]=specdiag(cuu,cvv,cuv);%specdiag is a jlab routine for diagonalizing a cc
sqrt(a2),sqrt(b2),theta

jmat2(theta)*[a2 0;0 b2]*jmat2(-theta)
[cuu cvv cuv]
```

```
ans =
   8.451475023398675
ans =
```

```
   4.808861637794671
theta =
    -2.469712673485169e-04
ans =
  71.427427124943478  -0.011929274777806
  -0.011929274777806  23.125153197641410
ans =
  71.427427124943478  23.125153197641399  -0.011929274777800
```

Because the velocity consists of a *pair* of variables $\tilde u$ and $\tilde v$, its second-order statistics are not just the variances of $\tilde u$ and $\tilde v$ separately. One must also take into account their covariance.

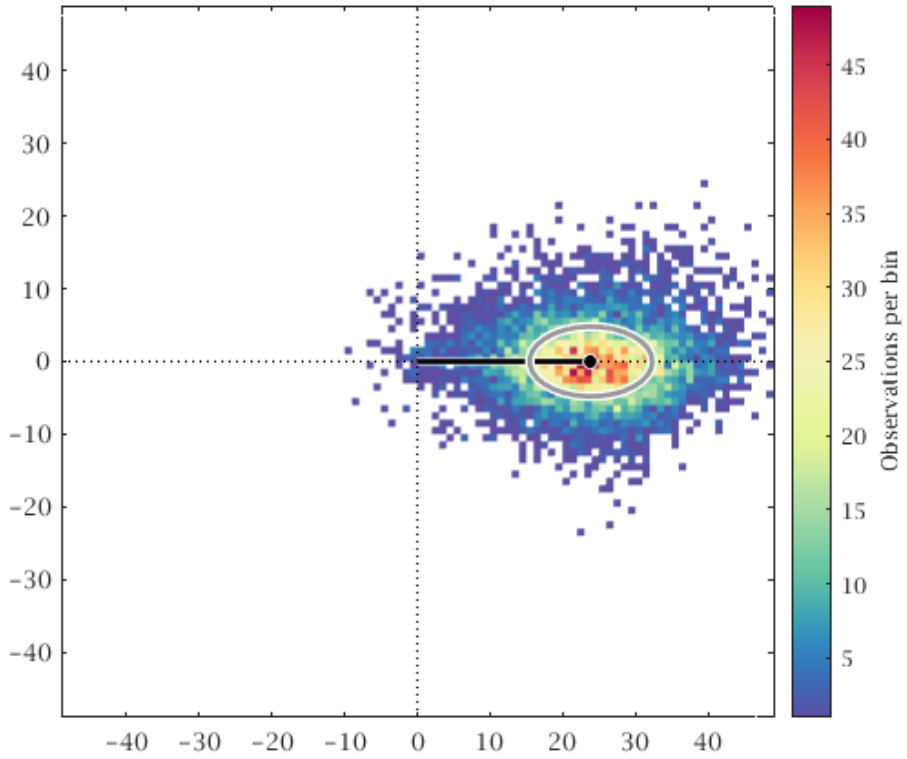
The diagonalization of the covariance matrix expresses this same information in a different way. It will be shown in class that a2 and b2 are the squares of the semi-major and semi-minor axes of an ellipse, respectively, while theta is the orientation of the major axis of the ellipse with respect to the x-axis.

This quantity is called the variance ellipse. Just as the variance is the fundamental second-order statistical quantity for a single or univariate time series, such as u, the variance ellipse is the fundamental second-order statistical quantity for a pair of time series (u,v).

Now we return to the 2D histogram presented earlier, adding a plot of the variance ellipse.

In [19]:

```
%same plot as above using 1x1 cm/s bins
[mat,xmid,ymid]=...
    twodhist(real(rcv(:,4)),imag(rcv(:,4)),[-50:50],[-50:50]); %the last two arguments
mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appear wi
jpcolor(xmid,ymid,mat);hold on
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Observations per bin';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])

%now we add a plot of the variance ellipse
%we'll plot it twice, with an outer white ring, for variability
[kappa,lambda]=ab2kl(sqrt(a2),sqrt(b2));%jlab code to convert ellipse axis lengths int
h=ellipseplot(kappa,lambda,theta,mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color','w'),set(h,'linewidth',4)
h=ellipseplot(kappa,lambda,theta,mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color',[1 1 1]*0.6),set(h,'linewidth',2)
```



You can see that the basic shape of the velocity distribution is indeed capture by the variance ellipse. Variance ellipses are a powerful tool for describing the basic features of velocity fluctuations, as we shall see.

In the above, rather than work with the ellise semi-major axis length a and semi-minor axis length b, we have used the parameters

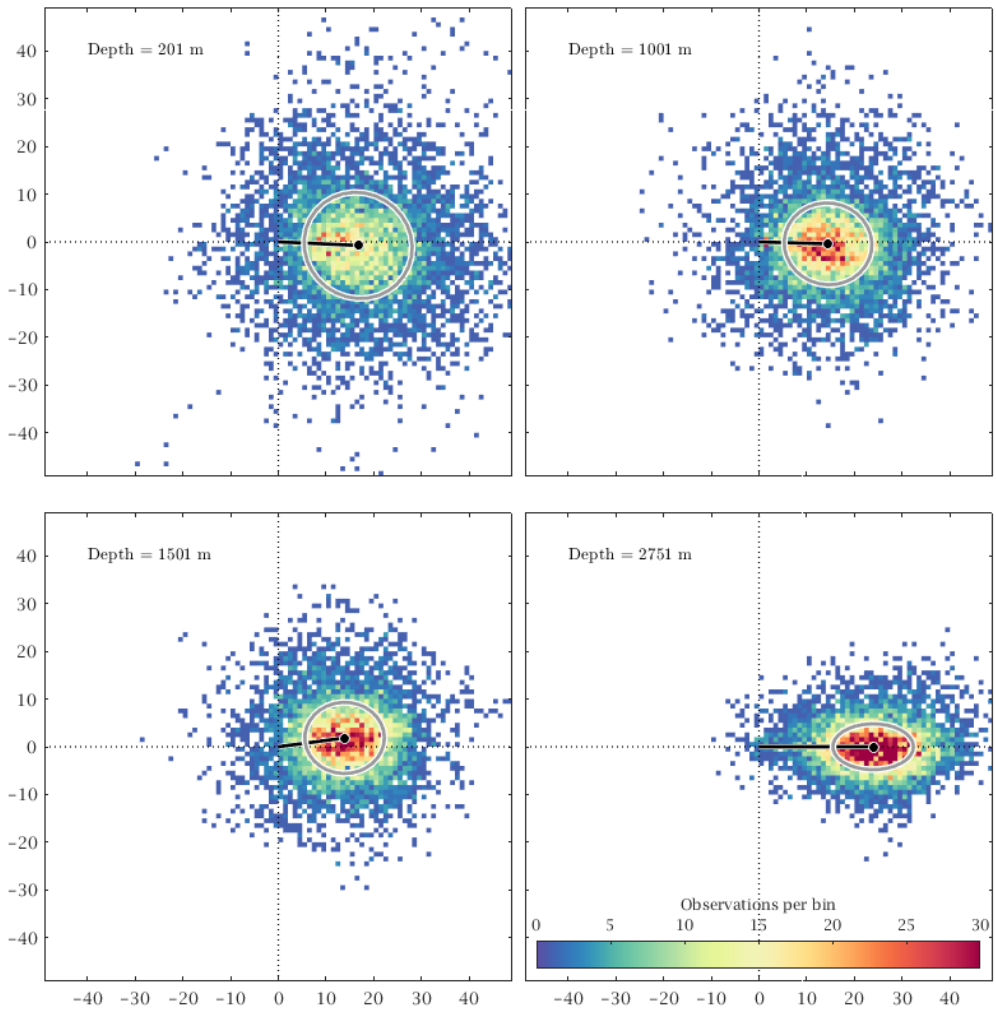
\begin{equation} \kappa^2\equiv \frac{a^2+b^2}{2},\quad\quad\quad \lambda\equiv\frac{a^2-b^2}{a^2+b^2} \end{equation}
which describe the ellipse size and ellipse shape, respectively. κ is the ellipse root-mean-square axis length, while λ is a measure of the ellipse shape which, like the eccentricity, is equal to zero for a circle ($a=b$) and one for a straight line ($b=0$).

Next we will make the nicer version of the 2D histogram for all four depths, by just looping over the code blocks we have already used.

In [20]:

```
kappa=zeros(1,4);lambda=zeros(1,4);theta=zeros(1,4);
for i=1:4
    u=real(rcv(:,i));u=u-mean(u);
    v=imag(rcv(:,i));v=v-mean(v);
    [a2,b2,theta(i)]=specdiag(mean(u.*u),mean(v.*v),mean(u.*v));
    [kappa(i),lambda(i)]=ab2kl(sqrt(a2),sqrt(b2));
end

set(gcf,'paperposition',[1 1 10 10])   %set suitable figure size
for i=1:4
    subplot(2,2,i)
    %this is the same code block we used for the plot just above,
    %but for the ith depth instead of the 4th depth
    [mat,xmid,ymid]=...
        twodhist(real(rcv(:,i)),imag(rcv(:,i)),[-50:50],[-50:50]); %the last two argu
    mat(mat==0)=nan; %swap zero values in the histogram for NaNs, so they don't appea
```

```
    jpcolor(xmid,ymid,mat);hold on
    colormap lansey, axis equal, axis(49*[-1 1 -1 1])
    vlines(0,'k:'),hlines(0,'k:')

    if i==4
        h=colorbar('South');
        h.Label.String='Observations per bin';
    end

    plot([0+0*1i mean(rcv(:,i))],'color','w','linewidth',3) %plotting a white edge arc
    plot([0+0*1i mean(rcv(:,i))],'color','k','linewidth',2)
    plot(mean(rcv(:,i)),'wo','markerfacecolor','k')
    xtick([-40:10:40]),ytick([-40:10:40])

    %now we add a plot of the variance ellipse
    h=ellipseplot(kappa(i),lambda(i),theta(i),mean(rcv(:,i)));%jlab code to plot ellip
    set(h,'color','w'),set(h,'linewidth',4)
    h=ellipseplot(kappa(i),lambda(i),theta(i),mean(rcv(:,i)));%jlab code to plot ellip
    set(h,'color',[1 1 1]*0.6),set(h,'linewidth',2)

    text(-40,40,['Depth = ' int2str(depths(i)) ' m'])
    caxis([0 30])
end

packfig(2,2) %jlab routine to remove space between subplots
```



This shows that as we proceed down in depth, the current variability becomes increasingly anisotropic.

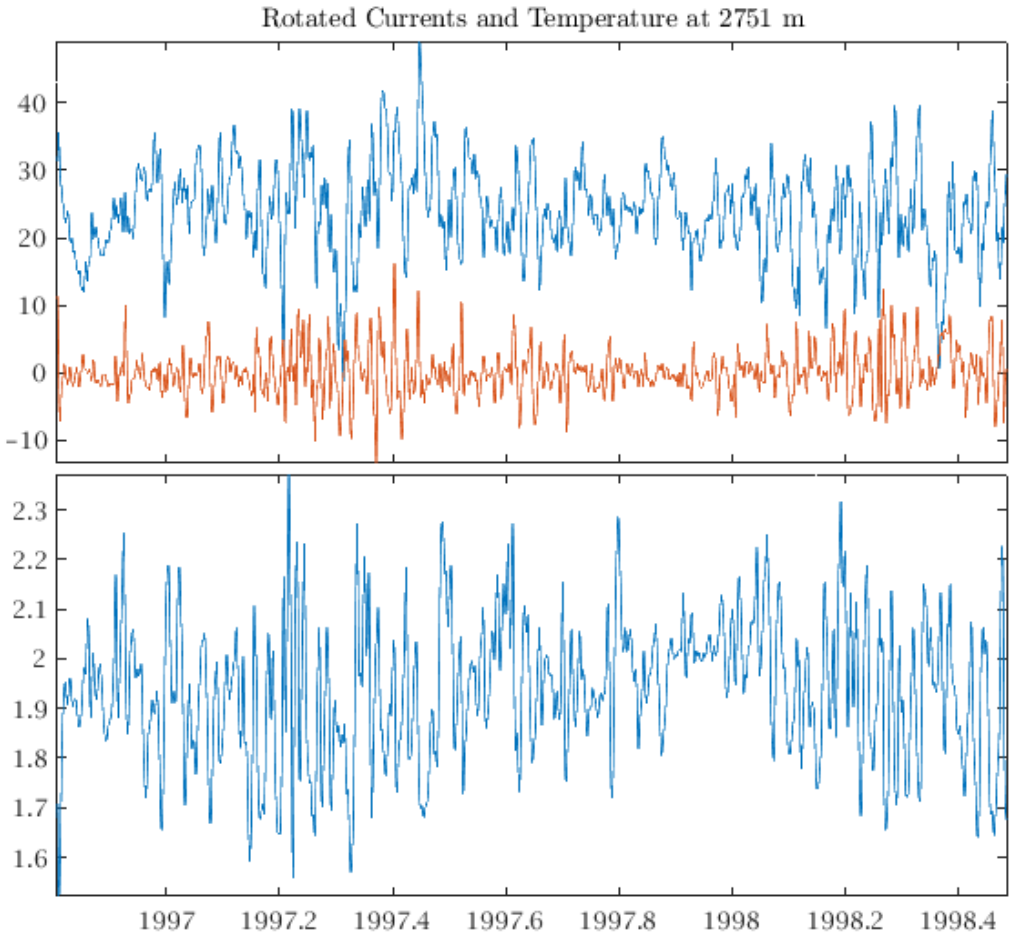
Near the surface, the variabilty has no preferred orientation, and the variance ellipse is nearly ciruclar. Moving downwards, the distribution becomes increasingly "squished" in the cross-stream direction, leading to variance ellipses elongated along the x-axis. Another way to say this is that the velocity fluctuations become increasingly *polarized* with depth.

We can also see slight changes in the orientation of the mean flow relative to the mean flow at depth.

Velocity vs. Temperature Fluctuations

Let's move on now to take a look at the co-variability of temperature and velocity. First we will make a simple time series plot. This is just the same lowpass-filtered rotated velocity signal we've been looking at, with filtered temperature at the same depth plotted underneath it.

In [21]:
```
set(gcf,'paperposition',[1 1 7 6]) %set suitable figure size
subplot(2,1,1),uvplot(yearfrac(num),vfilt(rcv(:,4),24)),axis tight,
title(['Rotated Currents and Temperature at ' int2str(depths(4)) ' m'])
subplot(2,1,2),plot(yearfrac(num),vfilt(t(:,4),24)),axis tight,
packfig(2,1)
```



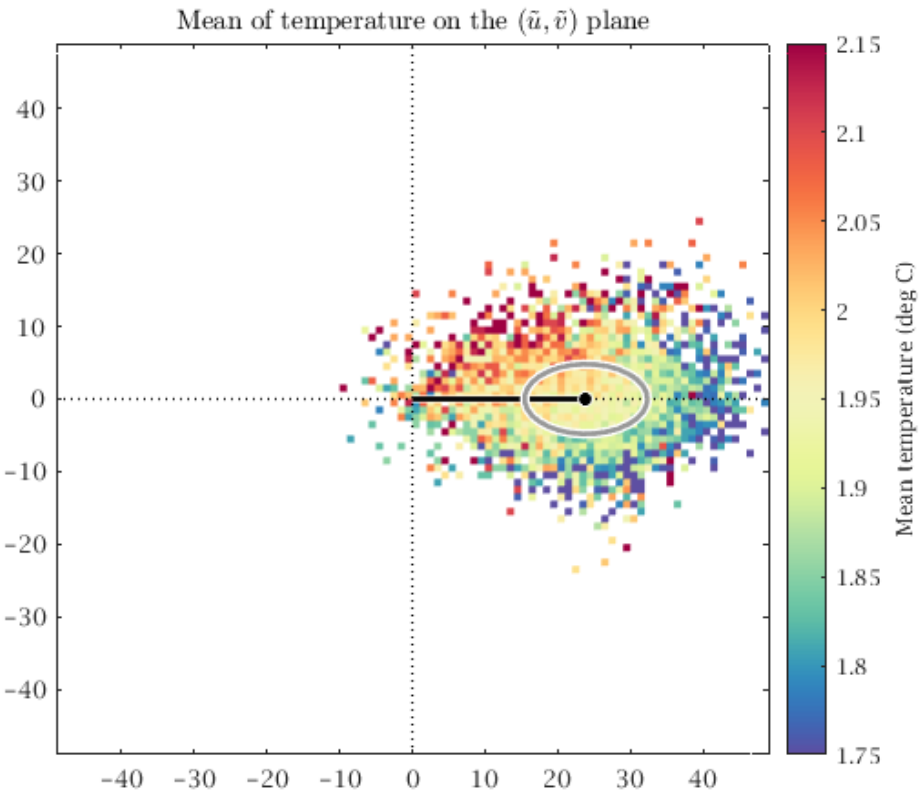
Temperature and velocity indeed present variability on similar timescales, but it's difficult to

ascertain what is going on from looking at this plot. In particular, it's not clear whether or not there is meaningful co-variability.

To examine this we are going to look at the two-dimensional mean of temperature t as a function of the $(\tilde u, \tilde v)$ plane. This is also known as the *conditional mean*, because the mean of t is taken relative to a particular range of values of $\tilde u$ and $\tilde v$.

In [22]:

```
[mz,xmid,ymid,numz,stdz]=...
    twodstats(real(rcv(:,4)),imag(rcv(:,4)),t(:,4),[-50:50],[-50:50]);
jpcolor(xmid,ymid,mz);caxis([1.75 2.15])
%plot the mean flow and set axis limits etc.
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Mean temperature (deg C)';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])

%now we add a plot of the variance ellipse
h=ellipseplot(kappa(4),lambda(4),theta(4),mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color','w'),set(h,'linewidth',4)
h=ellipseplot(kappa(4),lambda(4),theta(4),mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color',[1 1 1]*0.6),set(h,'linewidth',2)

title('Mean of temperature on the $(\tilde u,\tilde v)$ plane')
```



This shows that there is a definite pattern to temperature fluctuations with respect to velocity fluctuations. Warm temperatures tend to occur when the downstream flow is weak and the cross-stream flow is in the positive direction. Cold temperatures tend to occur when the downstream flow is strong and the cross-stream flow is in the negative direction.

The meaning of this pattern is not entirely clear, but it definitely suggests a particular relationship between the currents and temperature anomalies, perhaps associated with coherent eddies.

Moreover, we see that there is a tendency for velocity anomalies to systematically transport heat toward the positive cross-stream direction, which as will be seen later is offshore in this case.

Note that although a relationship is clear in the 2D histogram, it does not correspond to a simple correlation with downstream or cross-stream velocity. If we simply found correlation coefficients, we would miss the details of the pattern contained in this plot.

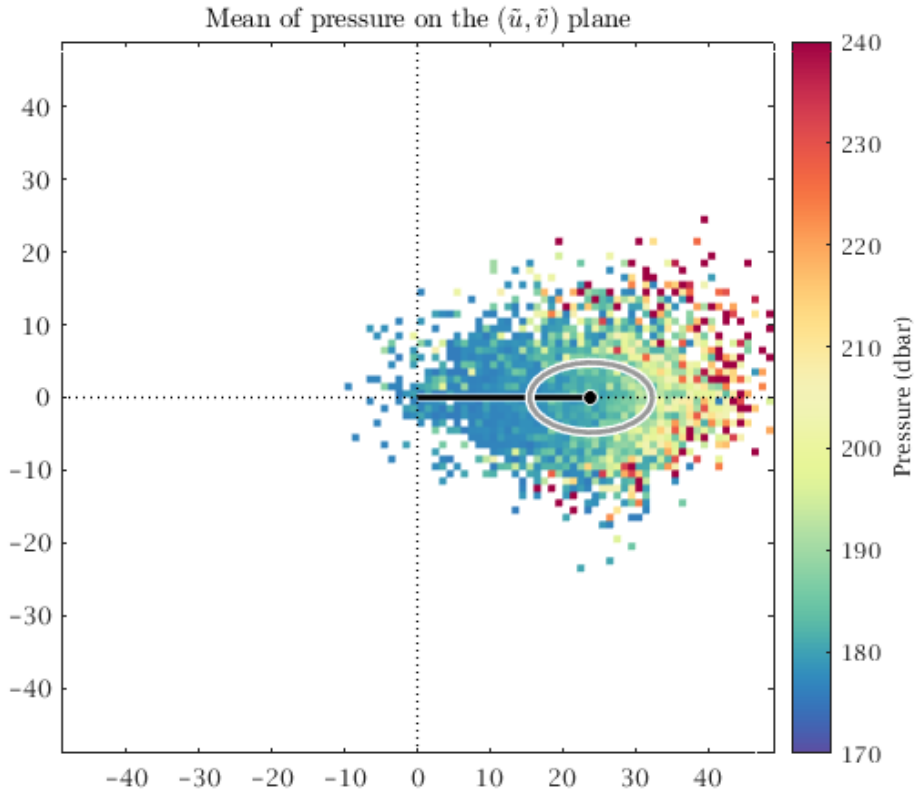
Let's examine the hypothesis that this pattern of temperature and velocity variability is an artifact of mooring towdown. That is, we conjecture that the mooring is going up and down due in the water column due to changes in drag, and this could interact with a pre-existing temperature gradient to generate the pattern we saw above as an artifact.

To examine this will make a plot as above, but this time with pressure as the variable we're taking the mean of. Because we don't have pressure at the deepest depth, we will substitute pressure at where we do have it, at the shallowest depth.

Note, in this next figure we are plotting the mean value of the pressure observed *at the shallowest depth* as a function of the currents observed *at the deepest depth*.

In [23]:

```
[mz,xmid,ymid,numz,stdz]=...
    twodstats(real(rcv(:,4)),imag(rcv(:,4)),p(:,1),[-50:50],[-50:50]);
jpcolor(xmid,ymid,mz);caxis([170 240])

%plot the mean flow and set axis limits etc.
colormap lansey, axis equal, axis(49*[-1 1 -1 1])
vlines(0,'k:'),hlines(0,'k:'),h=colorbar;
h.Label.String='Pressure (dbar)';
plot([0+0*1i mean(rcv(:,4))],'color','w','linewidth',3) %plotting a white edge around
plot([0+0*1i mean(rcv(:,4))],'color','k','linewidth',2)
plot(mean(rcv(:,4)),'wo','markerfacecolor','k')
xtick([-40:10:40]),ytick([-40:10:40])

%now we add a plot of the variance ellipse
h=ellipseplot(kappa(4),lambda(4),theta(4),mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color','w'),set(h,'linewidth',4)
h=ellipseplot(kappa(4),lambda(4),theta(4),mean(rcv(:,4)));%jlab code to plot ellipses
set(h,'color',[1 1 1]*0.6),set(h,'linewidth',2)

title('Mean of pressure on the $(\tilde u,\tilde v)$ plane')
```



This pattern is completely different. It shows that pressure is high when the speed is high, which occurs on the downstream fringe of the velocity distribution. This is just what we expect from mooring dynamics: when speeds are large, the instruments are dragged to deeper depths through the force of horizontal drag acting on the mooring line.

Indeed, had we reflected upon what the pressure should look like, we could have anticipated such a pattern. Therefore whatever is happening with the temperature is not an artifact of towdown. Rather, it appear to be physically meaningful.

Further examination of the temperature/velocity pattern will have to wait until another time. The point of this analysis has been to illustrate how simple plots, such as two-dimensional statistics, can be used to investigate physical hypotheses.

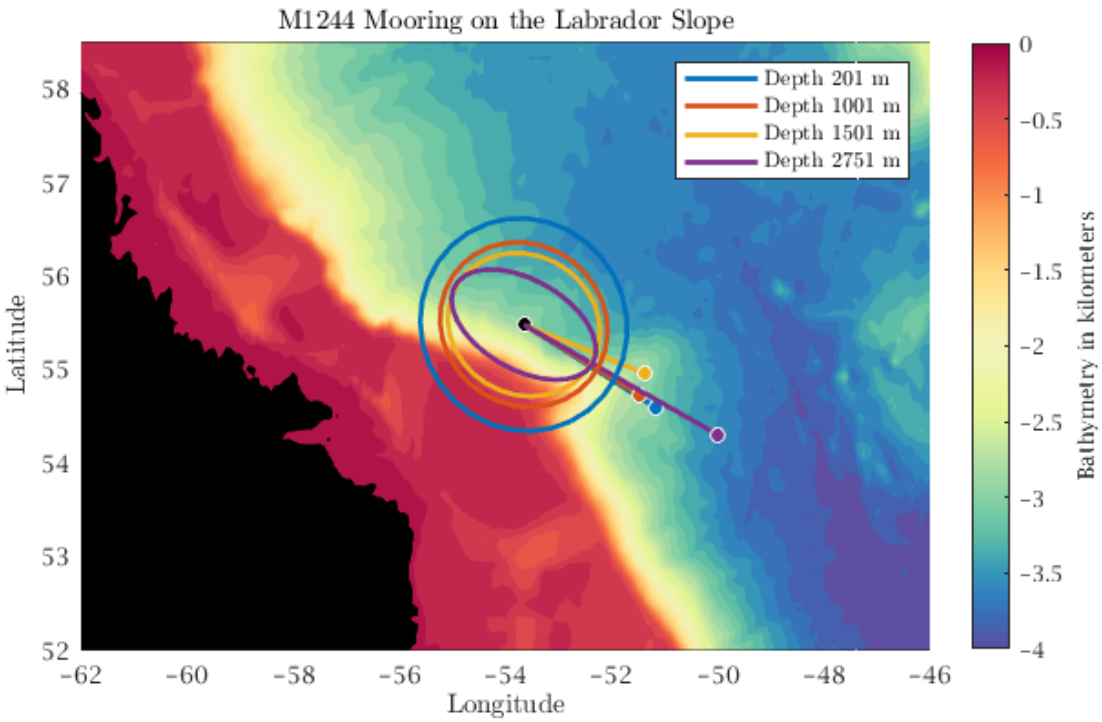
Summary Plots Using the Variance Ellipse

Now we will see how the variance ellipse can be combined with mean flow vectors to generate a summary plot. Here, we will plot the mean flow and variance ellipse of all four instruments at this mooring superposed on the bathymetry.

In [24]:

```
axis([-62 -46 52 58.5])
topoplot(axis,-5:1/8:0),hold on
topoplot continents
caxis([-4 0])
plot(lon,lat,'wo','markerfacecolor','k')
hc=colorbar;hc.Label.String='Bathymetry in kilometers';

latratio(lat)  %jlab routine to set the aspect ratio appropriate for latitude lat
ar=get(gca,'DataAspectRatio');  %the aspect ratio is now [1/cos(deg2rad(lat)) 1]
```

```
%plot mean flow vectors
sty='TUVW';%String to specify the default Matlab linestyle colors
factor=10; %A scale factor to convert cm/s to degrees, chosen for display purposes
for i=1:size(cv,2)
    %form the mean flow vector, and stretch the real and imaginary parts
    %according to the aspect ratio
    cvbar=real(mean(cv(:,i)))*ar(1)+1i*imag(mean(cv(:,i)))*ar(2);
    h=plot(lon+1i*lat+[0+1i*0 cvbar/10]); %plot mean flow vector
    linestyle(h,['2' sty(i)]) %jlab routine for setting line properties
    %finally plot matching circles at endpoints of velocity vector
    plot(lon+1i*lat+cvbar/10,'wo','markerfacecolor',get(h,'color'))
end

%plot variance ellipses using matching colors
h=ellipseplot(kappa/factor,lambda,theta-phi,lon+1i*lat,ar);
%note the last argument to ellipseplot is so it can take into account aspect ratio
set(h,'linewidth',2)
for i=1:4
    linestyle(h(i),['2' sty(i)]) %jlab routine for setting line properties
end

%Create a legend for the figure lines
clear str
for i=1:4
    str{i}=['Depth ' int2str(depths(i)) ' m']; %use a cell array for the labels
end
legend(h,str);

title('M1244 Mooring on the Labrador Slope')
xlabel('Longitude'),ylabel('Latitude')
```



Note that the mean flow is more or less aligned with the direction of the isobaths at the mooring location. This is a not uncommon result, particularly at higher latitudes where the stratification tends to be weaker and flows are more guided by bathymetry.

Then as we progress in depth variance ellipses are seen to become increasingly constrained to be parallel to the local isobaths as well, matching the orientation of the mean flow vectors. Again, it is fairly common (for oceanic currents, anyway) for the variance ellipse orientation to match the mean flow direction.

This plot is a good example of how the simple statitics of means and variances can be used to generate useful summaries of a whole dataset. It also shows how those statitics become more informative when placed into some kind of context, in this case, the context of the local topography.

The End