Assignment 1 for GEOL 1520: Ocean Circulation and Climate or, Notions for the Motions of the Oceans

Baylor Fox-Kemper

May 2, 2019

1 Contacts

The professor for this class is: Baylor Fox-Kemper
baylor@brown.edu
401-863-3979
Office: GeoChem room 133
http://fox-kemper.com/teaching, http://fox-kemper.com/1520

2 Getting Help!

I am usually available by email. You can make an appointment other times. Just check my calendar at http://fox-kemper.com/contact and suggest a time that works for you.

The most important commands in matlab are 'help' and 'lookfor'. The first one allows to to get a description of any matlab function, for example, '≫ help plot' tells you about the function named 'plot'. The second one allows to to search for keywords within a function description (in case you don't know or can't remember the name of the function).

If you are new to matlab, or just want a refresher, I recommend checking out http://www.geo.brown.edu/research/Fox-Kemper/classes/GEOL1520_19/notes/matlabprimer.pdf. There are lots of online tutorials as well.

3 Introduction

As you know, our assignments are structured to resemble a scientific journal's peer review process. So, when in doubt about procedures, try to understand the assignments in that context. I am listing together all of the pieces for the first paper at the same time here, so that you can see the full flow all at once (even though there are different due dates on each piece).

4 Homework 1

Each of the subsections here is a homework problem that you should complete, make a pdf of the results, and turn in through canvas.brown.edu. Please include screenshots or script files if that is needed.

4.1 Reading Preparation

Please read Chapters 1 & 2, Sections 3.1 and 3.2, and Appendix A of Wunsch (2015). These will help you get oriented for Paper 1!

You should be aiming to have about 5-10 papers cited in your references of Paper 1. Ideally, you can identify most of these in time for your Plans 1, having quickly skimmed them by then. You can read them more deeply as you prepare the Paper 1, but you may not need to read them completely if you only require information from one section or figure (`https://bit.ly/2DExCP8`).

4.2 Getting Analysis Software Running

If you don't have a strong preference for an alternative, download a copy of matlab (`http://software.brown.edu`) and install it on the machine you plan to use. If you plan to use an alternative (e.g., python or R), it will be up to you to get this homework done in the alternative, as well as the plans and papers, etc.

If you are using matlab for the first time, go through my matlab primer document (`http://bit.ly/2jDoEFh`). If you are still struggling with understanding, you can try some online tutorials. The HW assignment stage of each paper will include matlab steps and codes, so you will continue to be guided all semester (at increasing levels of sophistication).

4.3 Opening Files

Here is the directory for the Reid-Mantyla dataset (`http://fox-kemper.com/data/Reid-Mantyla/`). This data is from the Ocean Data View website (`http://odv.awi.de/en/data/ocean/reid_mantyla/`), where there is more description of the data. It is a global set of representative "classic" profiles of T, S, and other chemical data, hand-selected by Reid and Mantyla. Here is the description file (`http://fox-kemper.com/data/Reid-Mantyla/Reid-Mantyla.info`).

I have converted these data into three versions, which you can play with and compare. This is a matlab-native file: `http://fox-kemper.com/data/Reid-Mantyla/Reid-Mantyla.mat`. This is a netcdf version: `http://fox-kemper.com/data/Reid-Mantyla/data_from_Reid-Mantyla.nc`. This is a plain (ascii) text version: `http://fox-kemper.com/data/Reid-Mantyla/data_from_Reid-Mantyla.txt`. The goal here is to figure out how to open all three forms of the file using matlab.

Here are the homework problems (0, 1, 2, 3) related to opening the files:

0.1 Make a directory to contain the data files.

0.2 Download the 3 data files using whatever browser you like. Put them in the same directory. You might have to do a "save link as" command to get them correctly downloaded.

0.3 You should be able to browse the contents of the .txt file with a text viewer (Notepad, a web browser, emacs, vi, etc.)

0.4 Open matlab and change directories to locate the one with the files in it. You should be able to see the filenames when you type `ls` or `dir`. For example,

```
>> cd ~/Downloads/RM
>> ls
Reid-Mantyla.mat data_from_Reid-Mantyla.nc data_from_Reid-Mantyla.txt

>> dir

.                          Reid-Mantyla.mat            data_from_Reid-Mantyla.txt
..                         data_from_Reid-Mantyla.nc
```

1.1 Open the .mat file using the `load` command. Examine the size of the variables that result and their names using `who` and `whos`. For example,

```
>> who

Your variables are:

Type           date           latitude       nitrate        phosphate      station
bottom_depth   day            longitude      nitrite        salinity       temperature
```

```
cruise        depth         month         oxygen        silicate      year

>> whos
  Name                       Size                       Bytes  Class     Attributes

  Type                 301133x1                    34329162  cell
  bottom_depth         301133x1                     2409064  double
  cruise               301133x1                    36229232  cell
  date                 301133x10                    6022660  char
  day                  301133x1                     2409064  double
  depth                301133x1                     2409064  double
  latitude             301133x1                     2409064  double
  longitude            301133x1                     2409064  double
  month                301133x1                     2409064  double
  nitrate              301133x1                     2409064  double
  nitrite              301133x1                     2409064  double
  oxygen               301133x1                     2409064  double
  phosphate            301133x1                     2409064  double
  salinity             301133x1                     2409064  double
  silicate             301133x1                     2409064  double
  station              301133x1                    35310308  cell
  temperature               1x301133                2409064  double
  year                 301133x1                     2409064  double
```

1.2 In the matlab file version of the data, the casts are broken up so that each location, each depth, each time counts at a new row in every variable. Locate one "cast" that contains the 10,000th row of all of the data points, but find it at all depths from the following code snippet,

```
>> latitude(10000)

ans =

   36.2533

>> longitude(10000)

ans =

  309.1120

>> day(10000)

ans =

    23

>> depth(latitude==36.2533&longitude==309.1120&day==23)

ans =

           0
          10
          20
          30
          40
```

```
50
...
```

1.3 Using this selection approach make labeled, titled plots of salinity and temperature as a function of depth (depth on the vertical axis) for this complete cast.

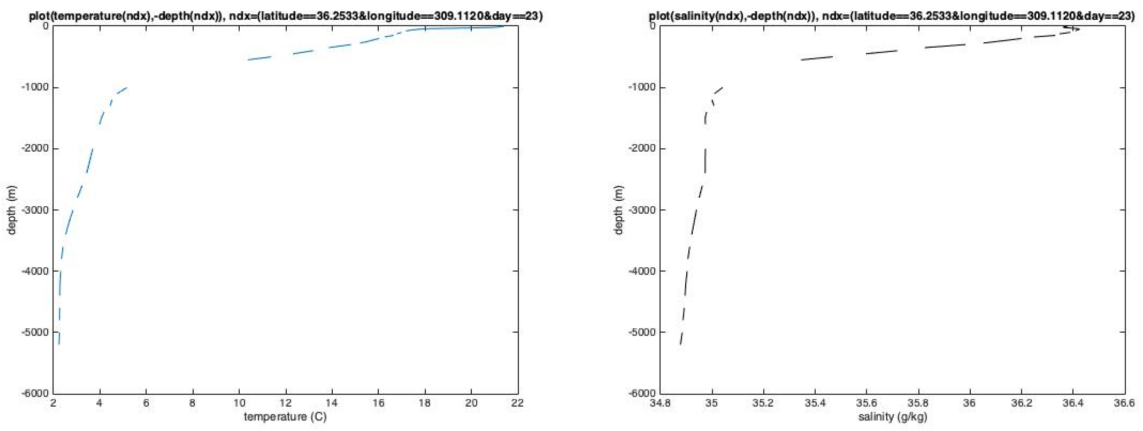

Figure 1: Temperature and salinity as a function of depth at a specified location.

2.1 Matlab is also capable of opening netcdf files natively, for information use help netcdf. Netcdf files have lots of "metadata" associated with the data, which can be accessed. Figure out how to read in the longitude, latitude, day, depth, salinity, and temperature fields from the netcdf file, and use them to make the same plot as in 1.3. Beware! All of the variables have different names in the netcdf files, so you *must use the metadata to figure out what's what*. Here's a hint,

```
>> ncdisp('data_from_Reid-Mantyla.nc')
```

OK, this is a bit tricky to sort. The dates in this dataset are totally different, as are the dimensions of the variables. Good learning example!

First, find the index of the same cast by its latitude and longitude. They won't be exactly the same as before, so just look for a location nearby, i.e.:

```
>> ndx=(abs(latitude-36.2533)<1e-2&abs(longitude-309.1120)<1e-2);
```

This finds exactly one cast on one day in this case. Next, look at the size of the temperature, salinity, and latitude and longitude variables. In the .mat file, these were all the same size, but in the netcdf file version, they are not. Note that there is only one latitude and longitude value for each cast, while there are 88 times as many temperature and salinity values! Thus, the depth at this site can be found by:

```
>> var1(:,ndx)
```

And similarly for the temperature (var2) and salinity (var3). Thus, you can make the temperature plot using

```
>> plot(var2(:,ndx),-var1(:,ndx));
```

and you'll see that it's much the same as the previous one.

3.1 As a last resort, matlab can read ascii file data, but it involves lot of character string editing and regular expression stuff, such as A Royal Pain in the A*S. Figure out a command to use matlab to display the first line of non-header data (i.e., the numerical values) from the .txt file. You don't have to figure out how to convert the text file output to an amenable format. . . save that effort until you have to do it with real data!

4.4 Bootstrapping Exercise

This portion of the homework is to get you thinking about probability distribution functions, averages, bootstrapping, and the Central Limit Theorem. If these concepts are new to you, then you might skim through Chapter 13 of my math notes (`http://bit.ly/2cpyj34`) and Appendix A of Wunsch (2015).

4.1 Open the Reid-Mantyla dataset above, and find all temperature measurements in the latitude range from 30N to 35N and longitude range from 150 to 155 and shallower than 500m depth. For example,

```
hist(temperature(find(latitude<35&latitude>30&longitude>150&longitude<155&depth<500)))
```

Make histograms of the years, depths, and temperature values of these data at 485 locations and times. (Matlab functions hist and histc).

```
>> hist(year(ndx),100)
>> title('histogram of years')
>> xlabel('year')
>> ylabel('number of occurences')
```

You can do the same thing for depths and temperatures. Note that as you increast the number of bins in the histogram, you start to see missing values, etc. It is not the case that every depth or every year gets a measurement! With some fiddling and reading the help page for hist and histc, you'll be able to plot whatever you like.

4.2 Use `http://maps.google.com` and the entry 32.5N, 152.5E to figure out where this set of data is centered. Save an image of the location.

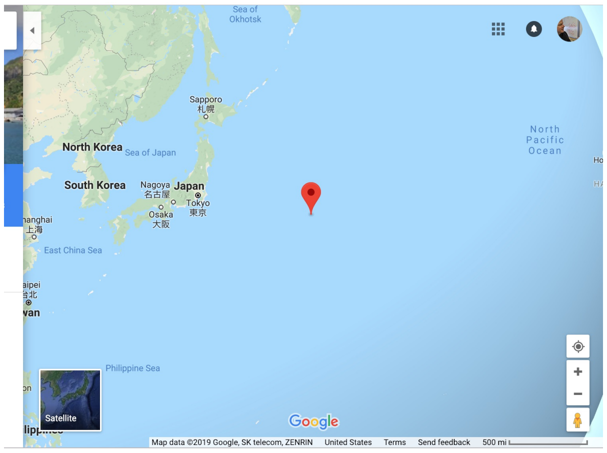

Figure 2: Location of 32.5N, 152.5E.

WIth some experimenting and googling about how to use google maps, you can figure out how to zoom in and out, add bottom topography or satellite overlays, indicate regions rather than just a point, etc.

4.3 Make a histogram representing the distribution of "average temperature of any 100 points" from this data, also known as the standard error for a 100-sample average. You can do this by bootstrapping (see matlab function bootstrp) or by calculation from the mean and standard deviation of the original 485 points and using the central limit theorem (you should get the same result).

```
>> mean(temperature(ndx))
ans =
   17.4961
>> std(temperature(ndx))
```

```
ans =
    2.7037
>> std(temperature(ndx))/sqrt(485)
ans =
    0.1228
>> std(temperature(ndx))/sqrt(100)
ans =
    0.2704
```

The mean of the whole dataset is 17.4961. If we subsampled 100 values from it, we'd expect to find nearly the same mean, but it would vary depending on which 100 we chose. The central limit theorem here tells us that the uncertainty in that should be 17.4961 ± 0.2704, or respecting the conventions of significant digits, 17.5 ± 0.3. Similarly, we expect the uncertainty in the whole 485 values to be 17.5 ± 0.1, if they are interpreted as a sample drawn from a much larger set.

Using bootstrp is more complicated, but also more powerful (since it works even when the data is not normally distributed).

```
>> m = bootstrp(100, @mean, temperature(ndx));
>> m
m =
   17.5325
   17.6651
   17.2478
   17.4016
   17.3446
   17.5102
   ....
>> mean(m)
ans =
   17.4912
>> std(m)
ans =
    0.1137
```

Your numbers will differ, since bootstrp uses random numbers to select its subsamples. Bootstrapping normally works by sampling from the original data, with replacement–i.e., you can draw the same datum multiple times, sampling as many values as the original dataset. Then the statistics of these multiple versions of the data (485 of them in this case) approximate the statistics of the original data. Thus, the bootstrp mean is very close to the overall mean (17.4961 vs. 17.4912), and the bootstrp standard deviation of the 485 values (0.1137) is quite close to the theoretical value (standard deviation divided by $\sqrt{N}$, or 0.1228).

But what about the problem of sampling 100 numbers out of the 485 and finding the statistics on that? That takes a little more work, as you have to define a function that is "average 100 values drawn from" a variable. Here is a quick version:

```
>> stats = bootstrp(100, @(x) [mean(x(1:100))], temperature(ndx))
stats =
   17.6603
   17.5169
   17.4983
   17.3831
   .....
>> mean(stats)
ans =
   17.4866
>> std(stats)
```

```
ans =
    0.2248
```

This time, we specified that instead of the mean over all of the data in the bootstrap sample, we only wanted the mean over the first 100 values by specifying the function to bootstrp

```
@(x) [mean(x(1:100))]
```

) It doesn't matter if we used 1:100 or 101:200, since the bootstrapped version is drawn in random order from the original 485 data points. Again, the mean is nearly the same as the mean of the original data, but now the standard deviation of the 100-point samples is higher (0.2248), close to the theoretical estimate of 0.2704. If you do 10,000 bootstraps of 100 point averages, it is very close to this value.

4.5 Budget Exercise

The model we will be using throughout this course is the ECCOv4r2 (Estimating the Current Climate of the Ocean, version 4, release 2). Details can be found in (Forget et al., 2015) and (Forget et al., 2016). The ECCO model is a data-assimilating ocean reanalysis using the MITgcm software (Massachusetts Institute of Technology general circulation model, Adcroft et al., 2008).

For this exercise, we'll be using a time average of the 1992-2011 state estimate. A state estimate is a "hindcast" or "reanalysis" of what happened in the past, using a model which is redirected toward agreement with observations collected at the time. See Appendix B of (Wunsch, 2015) if you are interested in the nitty-gritty details of state estimation and sections 9.5 and 9.6 for a more general discussion. An all-time average is just the average over all 20 years (here, for simplicity, I've just approximated and averaged the months together as though they were all the same length). A climatology is an average for each month, rather than the full 240 months of the state estimate. That is, January in the climatology is the average of the 20 Januarys in the state estimate, February is the average of the 20 Februaries, etc. The climatology is like a "normal year", except it is actually a bit less noisy/weather-dependent than any given year because of the averaging.

The climatology and averages are available here: `http://fox-kemper.com/data/ecco_for_las/version_4/release2`. The interp_monthly is available here (sign in as a guest user): `ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/interp_monthly` and also here: `https://bit.ly/2RcyFJp`. The latlon_monthly is available here: `http://fox-kemper.com/data/ecco_for_las/version_4/release2`.

The ECCOv4r2 uses a very complicated grid to cover the earth. It consists of 13 "tiles" that cover the Earth (see Figure). Each tile is logically arranged into a 90x90 rectangular grid. Obviously, the relationship between these grid coordinates and latitude and longitude is complicated! However, between 69°S and 56°N, the grids are chosen to align with latitudes and longitudes. So, as long as we consider things in only this region, budgets are easy to understand, as each gridcell edge aligns with either a meridian (line of constant longitude) or a parallel (line of constant latitude). Later in the class, you may use an *interpolated* version of this model, which keeps things easy even outside of this region (although the interpolation brings a few errors relative to the model's native grid where all model physics hold exactly). If you want to work on the Arctic or far Northern oceans to study certain balances you might have to use the full tiled version of the model. In later assignments, you might have to use two adjacent tiles to get a full section spanning the Pacific or Atlantic.

Each gridcell has advective (i.e., caused by the model velocity) and diffusive (i.e., caused by small-scale turbulence or other mixing) transports through each cell face. The model coordinate directions are called x, y, r, which may or may not align with North, East, and Up, because of the peculiar tiling above. In this simple example, they do correspond to these directions, though. The advective fluxes "live" on the cell faces, where the velocities are labeled in the figure. The diffusive fluxes also "live" there, while temperature, salinity, and density live at the center of the cell. Formally, it is approximate to say any of these variables "live" at any point, since the temperature really represents the average temperature within the gridcell, and the velocities represents the velocity averaged over each cell face. But, when we say the variables "live" at a location, it means their averaging region is centered on that location.

For this exercise, we'll consider the budget of temperature in one gridcell, then a column of gridcells down to the seafloor, then two neighboring gridcells.

1. Download the files SALT.nc (the salinity), ADVn_SLT.nc (northward advection of salt), ADVe_SLT.nc (eastward advection of salt), ADVr_SLT.nc (vertical advection of salt) and RF.mat (vertical grid information) from

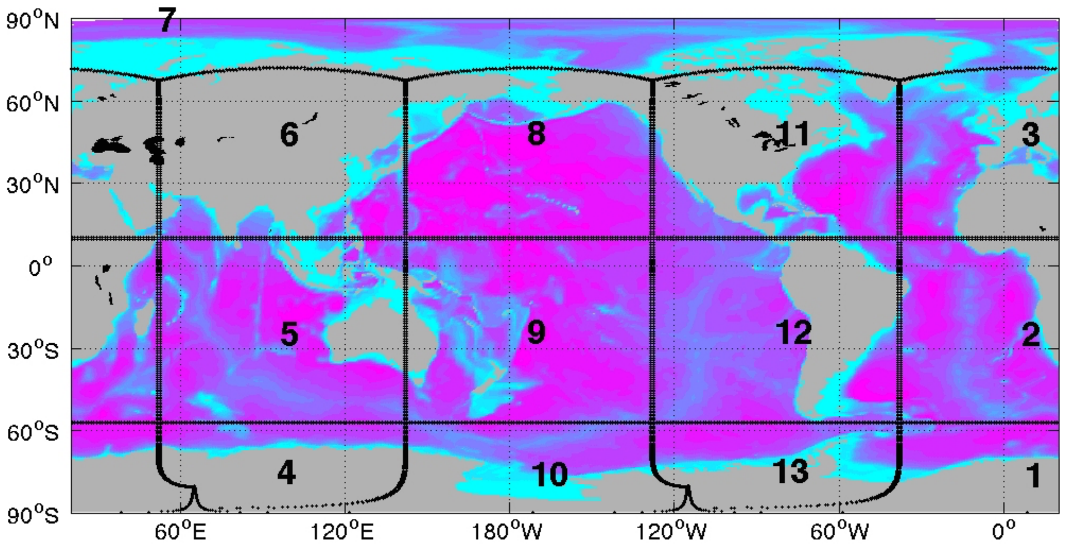

Figure 3: **ECCOv4 tiles the Earth.**

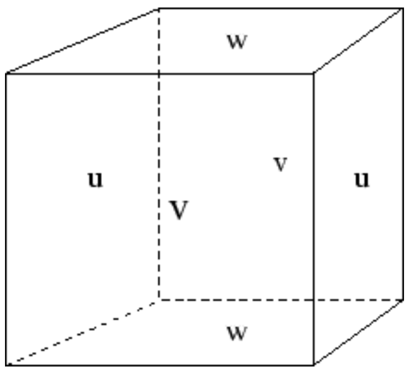

Figure 4: **MITgcm gridcell with velocity component's central location labeled.**

`http://fox-kemper.com/data/ecco_for_las/version_4/release2/latlon_avg/`. If you have chosen not to use matlab, the text version of the grid files can be found in `http://fox-kemper.com/data/ecco_for_las/version_4/release2/RF/`. These files have RC (the depth of cell centers) and RF (the depth of cell faces) and DRC and DRF, which are the distances between values of RF and RC locations.

2. Change directories in matlab to where the files are downloaded, and execute `ncdisp('ADVn_SLT.nc')`. Note how many variables are output. Find the longitude, latitude, depth, and the main variable (Northward Advective Flux of Salinity) and note their dimensions and the coordinates they depend on.

3. Load these variables into the memory of matlab, e.g., `lon=ncread('ADVn_SLT.nc','lon');`. Does it matter which file you get lon and lat from? You might use the convention of lonc, latc for cell centers and long and latg for edges.

4. You might like the simple matlab script located here `http://fox-kemper.com/data/ecco_for_las/version_4/release2/readncfile.m`. To use it, set the variable fname to the filename you want to open, e.g., `fname='latlon_avg/SALT.nc'`; then just type `readncfile`. It will read every variable in the netcdf file and make a matlab variable with the same name. BEWARE–if you already have a variable with the same name as one in the file, it will be overwritten.

5. `load RF.mat`

6. Make a pcolor surface plot of ADVn_SLT, ADVe_SLT, and ADVr_SLT.nc, e.g.,
`pcolor(lon,lat,ADVn_SLT(:,:,1));shading('flat');colorbar;title('ADVn Salt');xlabel('longitude');ylabel('latitude')`

Now we want to examine how well these fluxes balance at a point. The lon, lat, and dep values describe the location of the cell centers in Fig. 4. Let's consider a point about 500km (5 degrees of longitude) east of Providence (41.8240N, 71.4128W). So, the point we are looking for is the nearest one to (41.8240N, 66.4128W). We can do this using

```
>> [val,ndx]=min((latc(:)-41.8240).^2+(lonc(:)+66.4128).^2)
>> % This approximates the minimum squared distance.  Now for some fancy index searching!
>> lonc(ndx)
>> latc(ndx)
>> ndxx=find(abs(lonc(:,1)-lonc(ndx))<0.1);  % Note, I use less than here instead of exact ==,
>> %since sometimes roundoff errors make this more robust.
>> ndxy=find(abs(latc(1,:)-latc(ndx))<0.1);
>> lonc(ndxx,ndxy)
>> latc(ndxx,ndxy)
```

Ok, so we have located the indices of our desired point. Now, let's think about budgets for mass/volume, salt, and potential temperature into this gridcell. It is important to know that this model uses the Boussinesq approximation, which we'll talk more about in class, but it means that in all of these budgets the ocean density ρ_o, seaice density ρ_i, and snow on ice density ρ_s are all taken as constants. The values of all of these constants are given at the end of this document.

The conservation of mass (or continuity) equation for a fluid is

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}). \tag{1}$$

This just states that if there is a convergence (i.e., a negative divergence) of mass flowing into a point, then the density will increase. If we take the density as constant, then this just becomes $0 = \nabla \cdot \mathbf{u}$, which means that the velocity cannot diverge. This is sometimes called incompressibility. However, the ocean (and this ocean model) are not actually incompressible, just nearly so. This limit is the basis of the Boussinesq approximation. Under the Boussinesq approximation, we still take $0 = \nabla \cdot \mathbf{u}$ as the conservation of mass/volume, but allow the effect of gravity to still vary a small amount based on changes to pressure, salinity, and temperature.

OK, in the ocean model, all of the grid cells are of fixed volume except the surface cell. For an interior cell, then, we can integrate $0 = \nabla \cdot \mathbf{u}$ over the volume V to find

$$0 = \int_V \nabla \cdot \mathbf{u} \mathrm{d}V = \oint_A \mathbf{u} \cdot \hat{\mathbf{n}} \mathrm{d}A. \tag{2}$$

The unit vector $\hat{\mathbf{n}}$ is outward and normal to the bounding surface of the volume. The last step is just the divergence theorem, or Gauss's theorem, one of the fundamental theorems of vector calculus. So, in order to evaluate the integral of the divergence over a gridcell, we just need to evaluate the integral of the velocity through each cell face. Fig. 4 shows that the velocities are laid out to make this convenient, so we just multiply each face's velocity times that face's area.

Let's see how this works on the first subsurface gridcell at our location. These variables are available from `http://fox-kemper.com/data/ecco_for_las/version_4/release2/latlon_avg/`

```
>> fname='WVELMASS.nc'  % this is the vertical velocity
>> readncfile
>> lonc=lon;
>> latc=lat;
>> fname='NVELMASS.nc'  % this is the northward velocity, mass weighted by changes in cell height if any
>> readncfile
>> latg=lat;
>> fname='EVELMASS.nc'  % this is the eastward velocity, mass weighted by changes in cell height if any
>> readncfile
>> long=lon;
```

OK, so let's check on the velocity differences.

```
>> WVELMASS(ndxx,ndxy,2)-WVELMASS(ndxx,ndxy,3)  % this is the difference in vertical velocity from top to
>> EVELMASS(ndxx+1,ndxy,2)-EVELMASS(ndxx,ndxy,2)  % this is the difference in eastward velocity from side
>> long(ndxx+1,ndxy)
>> lonc(ndxx,ndxy)           % Note how the cell is bounded side to side by the long longitudes.
>> long(ndxx,ndxy)
>> NVELMASS(ndxx,ndxy+1,2)-NVELMASS(ndxx,ndxy,2)
```

In which direction(s) is the velocity in that direction converging? diverging?

Now, let's bring the cell face dimensions into the picture. We can calculate the cell face sizes, or look them up from mygrid. Let's calculate them ourselves.

```
>> % This is the cell volume, using the depth range of the cell (see end of document) and the convergence
>> % I neglect the fact that its not exactly a rectangle, but a trapezoid by using only the central latitu
>> H=10
>> NS=111200*(latg(ndxx,ndxy+1)-latg(ndxx,ndxy))
>> EW=111200*cosd(latc(ndxx,ndxy))*(long(ndxx+1,ndxy)-long(ndxx,ndxy))
>> H*EW*NS
>> % the following are the volume flux divergences (cross-section area * velocity diff).
>> A=(WVELMASS(ndxx,ndxy,2)-WVELMASS(ndxx,ndxy,3))*EW*NS
>> B=(EVELMASS(ndxx+1,ndxy,2)-EVELMASS(ndxx,ndxy,2))*H*NS
>> C=(NVELMASS(ndxx,ndxy+1,2)-NVELMASS(ndxx,ndxy,2))*H*EW
>> A+B+C  % This is the absolute divergence estimate
>> (A+B+C)/A % This is the estimate relative to the magnitude of one of the terms.
```

The surface gridcell is more complicated, because precipitation-evaporation (oceFWflx), sea ice thickness (SIheff), snow thickness (SIhsnow), ice sublimation (SIatmFW), and horizontal advection and diffusion of sea ice (ADV?HEFF, DF?EHEFF, ? here could be x, y, or e, n) and snow on the sea ice (ADV?SNOW, DF?ESNOW) must also be considered, plus the sea ice has a different assumed density (ρ_i) as does the snow (ρ_s). Perhaps most importantly, this cell changes its volume, through the sea surface height variable `ETAN`.

Likewise, similar approaches can be taken for salinity (S) budgets or potential temperature (T) budgets, noting that the equations for those are:

$$\frac{\partial \rho S}{\partial t} = -\nabla \cdot (\rho S \mathbf{u}) + \kappa \nabla^2 S, \tag{3}$$

$$\frac{\partial \rho T}{\partial t} = -\nabla \cdot (\rho T \mathbf{u}) + \kappa \nabla^2 T, \tag{4}$$

(5)

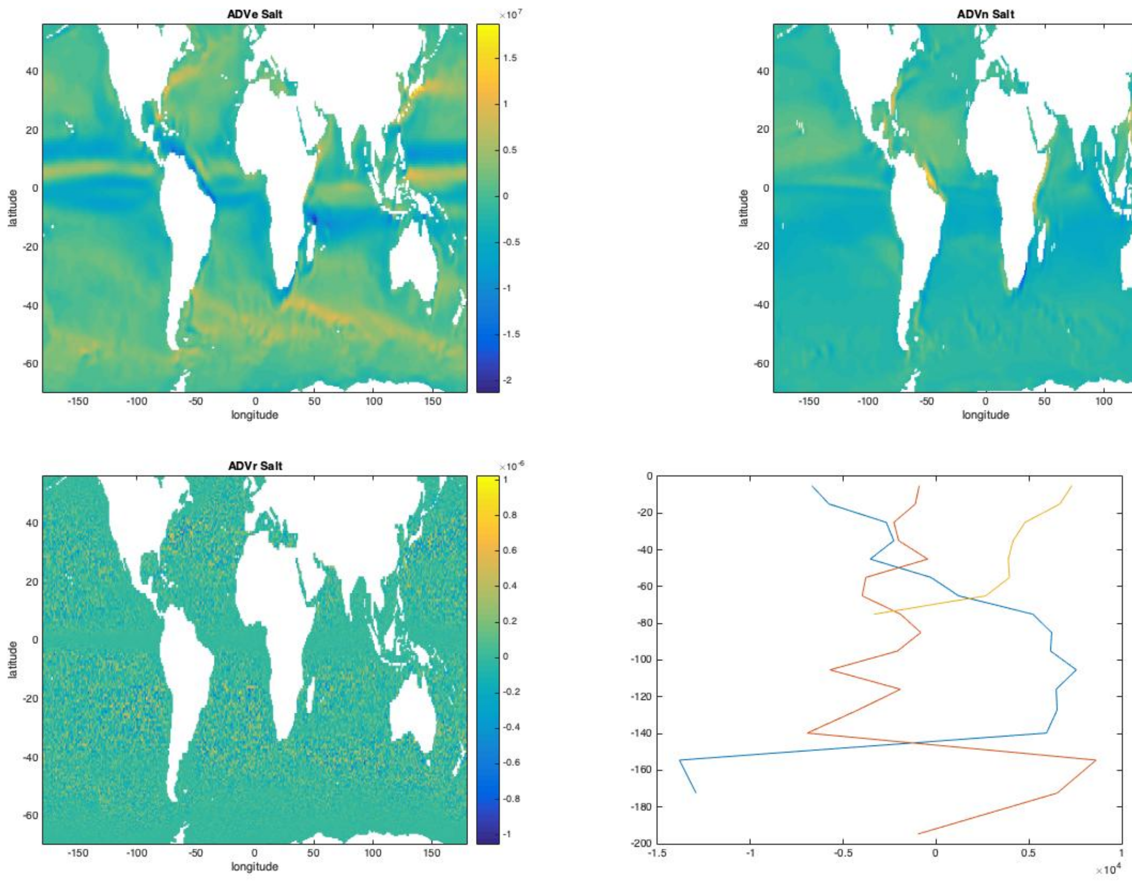

Figure 5: Budget figures from the homework.

In the salinity budget, the relevant variables are advective fluxes of salt (ADVe_SLT, ADVn_SLT, ADVr_SLT), diffusive fluxes of salt (DFeE_SLT, DFnE_SLT, DFrE_SLT+DFrI_SLT; the last two should be summed), plus the horizontal diffusive and advective fluxes of snow and ice, and surface sources (SFLUX). For the potential temperature budget, the advective and diffusive fluxes, plus the net surface fluxes (TFLUX), plus the latent heats of divergences in the snow and ice. It should be clear that these budgets are harder to perform where there is sea ice!

The last part of this assignment is to consider what happens for budgets over a collection of boxes. We can examine the horizontal flux divergences as a function of depth.

```
>> load RF.mat
>> A=(WVELMASS(ndxx,ndxy,1:end-1)-WVELMASS(ndxx,ndxy,2:end))*EW*NS
>> B=(EVELMASS(ndxx+1,ndxy,:)-EVELMASS(ndxx,ndxy,:))*H*NS
>> C=(NVELMASS(ndxx,ndxy+1,:)-NVELMASS(ndxx,ndxy,:))*H*EW
>> plot(squeeze(A),RC(1:end-1),squeeze(B+C),RC)
>> plot(squeeze(A),RC(1:end-1),squeeze(B),RC,squeeze(C),RC)
```

As you can see A, B, C are experiencing different depths, but near the surface their cancellation is good. Can you explain why they each see different depth ranges?

Finally, you may be concerned that the budgets do not exactly close. This is because our estimates for EW and NS do not include *partial cells*, that is gridcells that are shaved to account for intermediate ranges topography. I will distribute give some guidance as to how to obtain more accurate budgets in time for the plans assignment.

4.6 Additional Guidance (not part of HW)

There is a file llgrid.mat that has lots of useful variables for describing the latlon grid version of the ECCOv4r2 data. For example,

```
>> load llgrid.mat
>> pcolor(llgrid.XC,llgrid.YC,llgrid.Depth);shading('flat')
>> pcolor(llgrid.XC,llgrid.YC,WVELMASS(:,:,4));shading('flat')
>> pcolor(llgrid.XC,llgrid.YC,llgrid.basins);shading('flat')
>> % This one allows you to select a geographic basin,
>> % e.g., Pacific: llgrid.basins==1, Atlantic: llgrid.basins==2, Indian: llgrid.basins==3, etc.
```

You no longer need to load the separate lat and lon values from different files, as

```
>> latc=llgrid.YC;
>> lonc=llgrid.XC;
>> latg=llgrid.YG;
>> long=llgrid.XG;
>> dep=llgrid.XC;
```

Or, just use the values stored inside llgrid:

```
>> llgrid
>> [val,ndx]=min((llgrid.YC(:)-41.8240).^2+(llgrid.XC(:)+66.4128).^2)
>> ndxx=find(abs(llgrid.XC(:,1)-llgrid.XC(ndx))<0.1);
>> ndxy=find(abs(llgrid.YC(1,:)-llgrid.YC(ndx))<0.1);
```

Note that what we have been calling lonc can be associated with XC, latc with YC, dep with RC, etc.

Plus, there are other useful variables not included before, such as llgrid.RF which lists the locations of the gridcell tops and w velocities, and llgrid.RAC which is a more accurate version of $\mathsf{EW} * \mathsf{NS}$ from before, because it is the grid actually used by the model.

```
>> EW*NS

ans =

   6.7389e+09

>> llgrid.RAC(ndxx,ndxy)

ans =

   6.7606e+09
```

Additionally, we now have the vertical grid information in a form to make the deep grid cell values easier to compute. RF and RC contain the vertical locations of the cell faces (where vertical velocities and fluxes live) and cell centers (where temperatures and salinities live). DRF and DRC are also given, which are the distances between these locations. So, rather that incorrectly assuming that $H = 10$m as we did above, we could use:

```
>> A=(WVELMASS(ndxx,ndxy,1:end-1)-WVELMASS(ndxx,ndxy,2:end))*llgrid.RAC(ndxx,ndxy)
>> B=squeeze(EVELMASS(ndxx+1,ndxy,:)-EVELMASS(ndxx,ndxy,:)).*llgrid.DRF(:)*llgrid.DXG(ndxx,ndxy)
>> C=squeeze(NVELMASS(ndxx,ndxy+1,:)-NVELMASS(ndxx,ndxy,:)).*llgrid.DRF(:)*llgrid.DYG(ndxx,ndxy)
```

Note: squeeze is a useful matlab function that takes any single column, row, or other single-dimension array and expresses it as a column vector. It is equivalent to using the (:) index for a vector, but you can save space often with squeeze.

The climatology and averages are available here: `http://fox-kemper.com/data/ecco_for_las/version_4/release2`. The interp_monthly and nctiles_monthly is available here: `ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/interp_monthly`. The latlon_monthly is available here: `http://fox-kemper.com/data/ecco_for_las/version_4/release2`.

5 Plans 1

As you plan your plans, take a look at the primer documents, available on the webpage. Pay particular attention to the links to different datasets, as well as their descriptions in the textbook chapters.

For the plans document, you will outline what plans you have for the project, including as many details as you have collected by this time. Here are some critical points to make in the plans document, which will help me to better advise you in moving to the paper stage.

- Describe your region and timeframe of interest.
- Give a working title.
- Make a list of 1-3 hypotheses you plan to address.
- Identify specific journal articles, papers, or subsections of the book that are relevant. Briefly summarize what they say that is relevant to your planned paper, and what missing hypotheses to test that you will try to test.
- Identify datasets that you have located that are relevant. Indicate whether you have been successful in opening them (i.e., have you mastered the file format?), locating the relevant dates/locations (have you mastered the data subselection/data layout?), and which figures you intend to create (have you designed the most important metrics for your problem?).
- MOST IMPORTANT. If possible, include draft versions of the figures you plan to use in your paper, based on the real data and the real metrics you plan to calculate. Normally, if you can get to this stage for the plans, completing the paper on time is no problem.
- If you are unable to get to the preceding step, give your fallback plans if all else fails. Because of the reviewing process and timeframe, you will *not* be able to get an extension on the paper deadline if things don't go smoothly. You can include more figures in the revised version of the paper if you figure it out after you turn in the first version of the paper.

6 Paper 1

As you plan your paper, go through my paper primer document, available on the webpage at `http://bit.ly/2kzrK13`.

6.1 Topic Choice 1: Three Kinds of Lies

The goal here is to choose a region, timeframe, and straightforward diagnostic of some oceanic quantity, and then describe quantitatively the amount and quality of ocean data available. Some examples:

- Subsample a dataset to estimate the sampling error in a bulk measurement, e.g., global mean temperature.
- Compare similar measurements from two datasets (e.g., Reid & Mantyla versus eWOCE, or WOCE shipboard CTD versus WOCE drifter CTD). You can do this in a region of interest, or in a chosen bulk measurement (e.g., global temperature, Mediterranean temperature).
- Compare similar measurements but separated in time. Is there signal above the noise? How can you estimate noise (or has someone else done it for you)?

Don't be afraid to piggyback off of other published work. Use google scholar or Web of Science to locate articles by keyword, and then see what they say. Feel free to use the matlab references, and refer to similar projects from Ready to Roll, What Do You Mean Mean?, and 3 Kinds of Lies.

6.2 Topic Choice 2: Inside Joke

In this assignment you will estimate a temperature, salt, or volume budget. That is, you will take a dataset and try to figure out how to balance the incoming and outgoing sources into a region or into a volume. You will want to use the ECCOv4r2 data assimilating datasets. The climatology already touched on the HW may be sufficient for your purposes, depending on location and science question. You should choose a region whose budget is in some way "interesting," which is a main goal of your plans document to sort out. Interesting may mean that you need the full 240-month state estimate (e.g., to determine a trend in the budget over this 20-year period), or the full tiled version of the data (if you are interested in the Arctic or Antarctic).

Feel free to use the matlab references, and refer to similar projects from previous years, It's Getting Hot in Here or Inside Joke. Links to the proceedings can be found at `http://fox-kemper.com/1520`, right under the course description.

7 Reviews 1

As you plan your reviews, go through my review primer document, available on the webpage or `http://bit.ly/2jaqq4w`.

You will read and review two of your colleagues work. You can look at the peer reviewing primer for more details as to how to proceed. Be sure that you are actually critical where you see room to improve ("Everything looks great! Good work!" is not a useful review unless it is *totally* true), but do not be heedless of feelings or unfriendly. Make sure you comment on the *science* as well as the presentation and writing. For example, it is more important to catch unfounded claims that to fix spelling errors. Do you believe the metrics used? Do you understand the fundamentals of the dataset and measurement technique? Do you suspect that the data is cherry-picked? Were figures labeled in such a way as to be understandable? Did they indicate their region of interest with an annotated map? Etc.

8 Revisions 1

Learning how to respond to reviews, both positive and negative, is a crucial part of any scientist's (or just person's) training. There are a number of things to consider, which I'll just itemize here. More detail can be found in `https://bit.ly/2RnuqLj`.

- Your first paper was *not a draft*, it was a complete version of a *completed scientific work* in your opinion. Thus, if the reviewers (and editor, a.k.a. me) don't point out anything, you don't have to change anything within the paper itself (although this is very rare!).
- You DO have to upload a response to reviewers document/set of comments and a track changes version of your revised paper plus a final version. Please make each of these a pdf for upload.
- You may catch your own mistakes that were missed by the reviewers, it is OK to change these, too. Please itemize everything you change in the response to reviewers document.
- You should consider very carefully whether the reviewer is right before making changes. Generally, make all "easy" changes that are suggested by the reviewer. Things like clarifying unclear sentences or phrases, making notation more consistent, labeling figures better, etc., are all in this category. You should expect that the reviewer is reading more carefully than any other readers you are likely to get, so if they don't understand what is going on easily no one else is likely to either!
- You don't have to go through small suggestions one at a time in the response document, just say you took all minor suggestions.
- Some reviewers will make complicated suggestions, like additional computations, rearranging sections, deleting or adding figures, etc. Consider whether the outcome is guaranteed to be an improvement of the overall work. If you are sure the reviewer is wrong, just politely decline to make the change. If you aren't sure, give it a try, but don't be afraid to compare to the original before accepting the change. Whatever you do, explain your process and thinking in the response to reviewers document.

- The reviewers are not perfect, and they are equally or perhaps more likely to misunderstand what you've done than to point out something you've done incorrectly. Be careful to figure out which before you make any changes. Most often, the changes you make will just be to clarify what you've done rather than change what you've done. On the other hand, good reviewers will catch inconsistencies in your work that are important to understand or fix as you finalize your revision.
- Don't forget that ultimately it is your paper! Over-reacting to a reviewer comment can definitely make the paper worse than it was before the review. The art is to make mostly small changes, but ones that logically address the reviewers' comments.

Here are some useful constants (matching the exact constant values used in the model). Background constant ocean density $\rho_o = 1029\text{kg/m}^3$, which is used for all conversions from volume to mass. Background constant seaice density $\rho_i = 910\text{kg/m}^3$, and constant seaice salinity $S_i = 4\text{g/kg}$. Background constant snow density $\rho_s = 330\text{kg/m}^3$.

In case you are not using matlab, I include here the depth information of the model cells, which is included in the file RF.mat. List of depths of cell faces (RF) here in km: 0 -0.0100 -0.0200 -0.0300 -0.0400 -0.0500 -0.0600 -0.0700 -0.0800 -0.0900 -0.1002 -0.1105 -0.1213 -0.1330 -0.1464 -0.1625 -0.1823 -0.2072 -0.2383 -0.2767 -0.3232 -0.3782 -0.4417 -0.5133 -0.5922 -0.6773 -0.7675 -0.8615 -0.9580 -1.0563 -1.1555 -1.2555 -1.3569 -1.4614 -1.5728 -1.6956 -1.8347 -1.9936 -2.1744 -2.3780 -2.6045 -2.8540 -3.1265 -3.4220 -3.7405 -4.0820 -4.4465 -4.8340 -5.2445 -5.6780 NaN

Distances between cell faces (i.e., cell heights) in m: 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0100 10.0300 10.1100 10.3200 10.8000 11.7600 13.4200 16.0400 19.8200 24.8500 31.1000 38.4200 46.5000 55.0000 63.5000 71.5800 78.9000 85.1500 90.1800 93.9600 96.5800 98.2500 99.2500 100.0100 101.3300 104.5600 111.3300 122.8300 139.0900 158.9400 180.8300 203.5500 226.5000 249.5000 272.5000 295.5000 318.5000 341.5000 364.5000 387.5000 410.5000 433.5000 456.5000

Distances between cell centers in m: 5.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0050 10.0200 10.0700 10.2150 10.5600 11.2800 12.5900 14.7300 17.9300 22.3350 27.9750 34.7600 42.4600 50.7500 59.2500 67.5400 75.2400 82.0250 87.6650 92.0700 95.2700 97.4150 98.7500 99.6300 100.6700 102.9450 107.9450 117.0800 130.9600 149.0150 169.8850 192.1900 215.0250 238.0000 261.0000 284.0000 307.0000 330.0000 353.0000 376.0000 399.0000 422.0000 445.0000

References

Adcroft, A., J. M. Campin, S. Dutkiewicz, C. Evangelinos, D. Ferreira, G. Forget, B. Fox-Kemper, P. Heimbach, C. Hill, E. Hill, et al.: 2008, The mitgcm user manual.

Forget, G., J.-M. Campin, P. Heimbach, C. N. Hill, R. M. Ponte, and C. Wunsch: 2015, ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, **8**, 3071–3104.
URL `http://dx.doi.org/10.5194/gmd-8-3071-2015`

— 2016, ECCO version 4: Second release. *DSpace@MIT*.
URL `http://hdl.handle.net/1721.1/102062`

Wunsch, C.: 2015, *Modern observational physical oceanography: understanding the global ocean*. Princeton University Press.