Working with gridded atmospheric and oceanographic data.

The first set of notes on working with NCEP atmospheric data (posted on the website and attached below) introduces you to some basic plots with Matlab. Complete problem 1 and any other problem from the list. Write these up (preferably electronically but you can hand in a paper copy if you like). Typical length of writing: 5 pages text (1.5 line spacing) plus innumerable figures.

1. Plot sections of the zonal velocity on the Northern Hemisphere (y, p) plane. Look for the core of the jet streams (sometimes you see a separate low-latitude jet associated with the end of the Hadley cell as well as the high-latitude ‘eddy driven’ jet that is strongest in winter). Look at its latitude variation. Now, plot meridional sections contouring $\theta$ at some of the same longitudes (overplotting these on color contours using pcolor(.,),shading interp or contourf(.,) of u). Describe what you see, particularly looking for steeply sloping surfaces of constant $\theta$ (and T). Pick out a few locations and test the thermal-wind equation, connecting north-south density gradient with vertical gradient of $u$ (vertical shear). At this point it is important that your variables are numerically correct; check this against textbook or journal plots, or maps on www.atmos.washington.edu/~ovens/loops.

We discussed below the technical problem of calculating thermal wind from data stored at widely spaced levels. Here is a bit more about that.

The $\partial \rho / \partial y$ part is normally calculated along constant $z$ surfaces, but it is more accurately calculated along constant pressure levels, as we pointed out in class. So the stored data can be used as it is stored on pressure levels.

Calculating derivatives for the thermal wind equation is obviously an approximate process, in which $\partial u / \partial z \approx (u(z_2) - u(z_1)) / (z_2 - z_1)$. The data is stored on constant pressure levels, rather than $z$-levels, but the dynamic height $z(p)$ tells you the altitude of those levels and thus can be used to give the $z$ values. Alternatively, the thermal wind equation in pressure coordinates can be used without needing to know $z$: $f_u_p = -(1/\rho^2)\rho_y$ and $f_v_p = (1/\rho^2)\rho_x$.

(Note that there was a sign error in this pressure-coordinate equation in the lecture notes, corrected last week.) While $\rho$ is not a stored variable, pressure $p$ and temperature, $t_e$, are stored, and $\rho = p/RT$. Because the vertical differentiation should be centered at the same $z$ (or $p$) level as the calculation of $\partial \rho / \partial y$ it is best to average two adjacent. We are not looking for fantastic accuracy here, however, but it is good to see the limitations of gridded datasets. With more care we would interpolate between the 17 pressure levels using a spline fit.

Another approach is to work in this testing of thermal wind with the vertical integral of the equation, which is the equation for dynamic height difference (see Gill §7.7): for example $\int [u(p_2) - u(p_1)] = - \partial/\partial y [\Phi(p_2) - \Phi(p_1)]$

$\Phi = gz(p)$ is the geopotential and its difference is ‘thickness’, proportional to average temperature over the vertical interval, (average temperature = $\int RT/\rho \ dp \ averaged \ with \ respect \ to \ ln(p)$ which is close to geometric height): see Gill p.216. Recall that geostrophic balance in terms of geopotential is simply $-fv = -\Phi_x, fu = -\Phi_y$
o Overplot the Earth topography, and pick longitudes of major topographic features (Rockies, Greenland, Andes, Himalayan plateau), with the zonal velocity and $\theta$ sections.

o Integrate the velocity over the region of westerly winds on one of these sections to find the mass transport of the westerly winds in Sverdrups ($10^6$ kg sec$^{-1}$ or megatonnes sec$^{-1}$). For comparison, the major ocean currents have (water) mass transport of 30 to 150 Sverdrups. Note that with $z(p)$ stored on pressure levels, the mass transport $\int_{\text{region}} \rho u \, dz \, dy = -\int_{\text{region}} u \, dp \, dy / g$ because $\rho \, dz = -dp/g$ from hydrostatic balance...so you don’t need to use the density field to get mass transport.

o Make vertical profiles (line plots versus pressure, $p$) of potential temperature $\theta$ and velocity in one of the developing cyclonic eddies in this winter’s data. For contrast do a couple of these with $z$ as the vertical coordinate, for example compare $\theta(z)$ with $\theta(p)$ profiles. Describe the vertical structure of the horizontal wind in these storms.

o Then, make horizontal maps of height contours $z(\text{lon, lat})$ at 1000, 250 and 30 HPa (near surface, upper troposphere, stratosphere). With snapshots or videos, describe the relationships you see between geostrophic velocity among these three layers. As seen in class, eddy and wave activity can be seen reaching high in the atmosphere, even though the heat engine of the atmosphere creates large horizontal temperature gradients, and hence baroclinic thermal wind shear, $u(z)$, $v(z)$.

2. The buoyancy frequency $N = (g \theta^{-1} \partial \theta / \partial z)^{1/2}$ is a key variable for waves and circulation, for example the Rossby deformation radius for rotating, stratified flows in the baroclinic mode is $L_d = NH/f$ where $H$ is the vertical length scale of the motion of interest (like the inverse vertical wavenumber $1/m$, of a wave).

So, make plots of $L_d$ and $N$ for the atmosphere, as functions of $y$ and $p$. Use finite-difference approximations based on the temperature data $T(p)$, corrected for the effect of pressure.

{ These combine with velocity data to give important parameters that suggest where flow instability can occur...both small-scale instability (‘billows’) due to vertical shear ($\partial u / \partial z$) and large-scale baroclinic instability (the subject of GFD-2).

You could on to map these parameters:
the Richardson number $N^2/((\partial u / \partial z)^2 + (\partial v / \partial z)^2)$ and the ‘Eady index’, $(f/N)\partial u / \partial z$ which has dimensions 1/sec, and is a rough estimate of baroclinic instability (the doubling time for an ‘Eady wave’ being roughly $N/(f \partial u / \partial z)$). An even rougher estimate of Eady index is $\sim U / L_d$ where, again, the internal, baroclinic Rossby radius is $L_d = c_0 / f$ and $c_0 = NH$, $H$ being the vertical scale of the unstable wave, which you could take to be simply the scale height of the density field $RT/g$. As a precursor to our study in internal gravity waves, $c_0$ is an estimate of the horizontal phase speed of hydrostatic internal gravity waves without rotation ($f=0$).}
mountains!). This is a challenging problem, but you can easily look at v(z, p) and even calculate the mass transport north-south in Sverdrups.

4. Using the ‘quiver’ command plot u,v velocity vectors on top of dynamic height contours, which is a crude test of geostrophic balance. From the u,v field calculate a finite difference estimate of the vertical vorticity and make a map of it, superimposed on the z height field (as in the maps available on atmos. washington.edu weather loops).

5. In the previous problem, look at the specific humidity field and calculate the mass transport of water vapor (in Sverdrups) north-south in individual sections and in zonal- and time-averages. Now make some horizontal maps of humidity and see how it relates to storms and storm tracks. What is the vertical structure of humidity in a developing cyclone? You could calculate the north-south transport of moisture at a fixed latitude by multiplying humidity by v-velocity and air density, and integrating vertically over all pressure levels...how does this flux of moisture vary east-west, around the whole globe? How many Sverdrups of water are moving meridionally at any given moment?

6. Using the 4 times daily data to plot the Z dynamic height maps at 1000 HPa, 250 HPa and 30 HPa (near surface, top of troposphere roughly and stratosphere) and follow cyclonic storm systems, the jet stream and the stratospheric polar vortex above. Look for relationships between these three levels. Animations are particularly useful. Then make summary plots on the longitude-time plane (x, t) and the latitude-time plane (y, t) as Hovmöller plots at individual pressure surfaces of, for example, meridional velocity v. These space/time (x/t) plots at constant latitude are particularly valuable. With pcolor contouring look for the movement of individual storms and groups of storms. This can span both Pacific and Atlantic sectors and sometimes the ‘billiard-ball’ effect can be seen in the downstream development of low-pressure cyclonic systems and clusters of storm.

Technical notes:

I like to use pcolor and shading interp or contourf for contour plots (with colorbar and adjusting the colors with caxis). Then to be more quantitative you can superimpose contour lines of the same or another field, using hold on, contour(….. ’k’). The ‘k’ specifies black contours. Matlab will sometimes rebel and wipe out your pcolor field and to cure this you need to reset the color palette after the plot is made (for example use the command caxis([280 380]) for potential temperature theta roughly in the range 280K-380K. Another very useful technique is to overplot line plots at a few latitudes or levels, on top of pcolor plots. You may here also have to reset your caxis([-])

In horizontal mapping it is very interesting to plot two or more contour maps corresponding to different pressure levels. If pcolor is working properly you can add a base color plot too.

Contour plots like contour(lon, p, squeeze(z(:,15,:,:,24))’) give you snapshot plots for latitude 15 (55N). You can specify the number of contour levels or the exact choice of contour levels (use ‘help contour’) . The z(…)’ is there to swap the indices of the matrix. -p is there to make the pressure decrease upward. The p variable needs to be in your data-set (it is the vector of pressure values for the 17 levels in NCEP data. Thus,

\[ p = [1000 925 850 700 600 500 400 300 250 200 150 100 70 50 30 20 10]; \]

You often need to reassure matlab that the matrix is 2-dimensional by putting squeeze(z(:,5,:,:,24))’ as in the plot command above. If the data is stored as single precision or integers then for pcolor and contour you need to convert to double precision by putting in the plot statement
squeeze(double(z(:, 5, : , 24))). This is much faster and saves space, compared with converting the whole data matrix to double before plotting.

Space: for calculating derivatives d/dy, d/dz, using finite differences, you need to have the right units: latitude is stored in degrees, so multiply these by 1.11e5 to get meters (111 km per degree of latitude). Longitude is trickier: multiply by 1.11e5 cos(lat*2*pi/360) to get east-west distance in m. For d/dz derivatives for variables stored on constant pressure surfaces we take the difference z(iz) - z(iz-1) for example, where iz is the index for the level of interest. There is the problem of centering the differences, which suggests using z(iz+1) - z(iz-1) as dz for comparison with a variable at the iz level. You can see that the data resolution begins to be poor for such calculations: high resolution modeling is the only solution for the future. In some cases it may be better to integrate vertically, as in \( \int \partial \theta / \partial z \, dz \) for comparison with the difference between v-velocity between the two heights. In terms of temperature the thermal wind equation is

\[
\begin{align*}
\frac{f}{\partial z} & = (g/T)\frac{\partial T}{\partial x} |_{p=\text{const}} \\
\frac{f}{\partial z} & = (g/T)\frac{\partial T}{\partial y} |_{p=\text{const}}
\end{align*}
\]

where \( f = 2 \Omega \sin(\text{latitude}) = 2 \Omega \sin(2\pi \text{lat}/360) \), converted to radians.

Time: the variable t is hours since 1-1-1800. To convert this to something intelligible use
timestring=datestr(t(it)/24+datenum('1-Jan-1800'))
where you choose the index it, for example the beginning it=1; or end, it=size(t);

For horizontal mapping, it is often very good to use m_map projections (the ‘satellite’ projection is my favorite), as described in the notes below

Almost any spatial plot (profiles, sections, maps) can be animated in time with great interest. For the full-year datasets you can go through the seasons and through many ‘storms’ and wave events. You can abstract some of this information on single Hovmoeller plots. Movie making is described in the notes at the end, or there are numerous descriptions on the Web.

Datasets:
We have on the website ... single precision versions, for winter 2014/15 and monthly means for earlier years.

NCEP atmospheric data
- ncep-2015-z-te-u-v-omega1.mat
  1Nov2014-1Feb2015 4xdaily data (necessary for following fast-moving storms), all levels, stored in individual files for Z, u, v, T, omega velocity, specific humidity 434Mb (big!)
- ncep-2015-subset2.mat same as above but daily means rather than 4 times daily 143Mb.
- ncep-monthly1.mat NCEP data Sep1996-Jan2015 monthly means 434Mb (big!)
- ncep-monthly2.mat" NCEP data May2003-May2004 monthly means 28Mb (small!)

Earth topography

ETOPO20 low-resolution (20 nautical mile); higher resolution data easily found on the web.

contourf color 'none' gives grid
mb reverse srt(gca, 'YDir', 'reverse') inverts y-axis

%%% m-file describing basic plots (as posted on website)

%%% handout 10 Feb 2015 GFD1 UW P.Rhines
The Matlab files
ncep-2015-z-u-v-te-omega1.mat  1 Nov 2014- 1 Feb 2015 4x daily 477Mb
ncep-2015-subset2.mat         % same as above but daily averages 143Mb
and potential temperature theta is included.
for the variables z dynamic height, u, v velocities, omega vertical velocity,
temperature, lat, lon in degrees, t is time in hrs since 1-Jan-1800.
from NCEP reanalysis data at www.cdc.noaa.gov

load ncep-2015-z-te-u-v-omega1.mat
% To get the calendar data from the time variable ' t(it) ' use
timestring=datestr(t(it)/24+datenum('1-Jan-1800')) % test t(1) for 1 Nov 2015
% where it is the index of the time axis (from 1 to 369 for the file loaded).
% check dimensions of arrays
whos          % note variables are single precision to save space
size(z)       % ...lon lat level time .. 144 73 17 365
lat           % prints out latitudes
lon           % note these are east longitudes from the Greenwich meridian

% test plots
% near surface dynamic height (level 1 is 1000 HPa):
contour(lon, lat, squeeze(double(z(:,:,1,1))))', 25); grid
% NOTE: transpose ()' gets lat,lon correct
% upper troposphere (150 HPa):
contour(lon, lat, squeeze(double(z(:,:,9,1))))', 25); grid
% color shading: a fast kind of plot:
pcolor(lon,lat,squeeze(double(z(:,:,9,1))))'),shading interp; colorbar
% note: the ' grid ' command does not work on top of a pcolor plot
% so I laboriously draw grid lines:
hold on; for ilat = 1:8:73,plot([0 360],[lat(ilat),lat(ilat)]),end;
    forilon = 1:10:133,plot([lon(ilon),lon(ilon)],[90 90]),'k'),end;
hold off
% north-south plot at 1 Nov 2014 and 60W longitude (lon=300E which is index 121)
plot(lat,squeeze(z(121,:,:,length(t)))); grid % squeeze is often needed
    % to keep Matlab happy
    xlabel('LATITUDE');ylabel('PRESSURE');title('HEIGHT FIELD 60W 1 Feb 2015')
    % bring out detail by stretching your plot vertically
    % with mouse
% movie of dyn height z at jet stream level  250 HPa
figure
hold off
pcolor(lon,lat,squeeze(double(z(:,:,9,1))))'),shading interp
caxisfixed=caxis; colorbar
for it=1:369 % number of records for this 4x daily data
    pcolor(lon,lat,squeeze(double(z(:,:,9,it))))'),shading interp
caxis=caxisfixed;
timestring=datestr(t(it)/24+datenum('1-Jan-1800'));
title(timestring)
drawnow
end
```matlab
% plot u-wind at all levels, longitude 60W on 1 Jan 2010
plot(squeeze(u(121,:,:length(t)))), grid
title('zonal velocity at each pressure level, 1 Feb 2015 60W')
xlabel('LATITUDE')
% color contour it
pcolor(lat,-p,squeeze(double(u(121,:,:length(t))))'),shading interp,colorbar
title('zonal velocity vs. latitude and pressure, 1 Feb 2015, 60W longitude')
hold on, contour(lat,-p,squeeze(double(z(121,:,:length(t))))',25,'k')

% to emphasize the signal, stretch your plot vertically with mouse

%%% getting a good map projection and continental outlines with M_MAP %%%
%%% M_MAP set of m-files version 1.4 %%%
%%% http://www.eos.ubc.ca/~rich/map.html or the m-files are posted on the GFD-1 website

%%% 1. create a folder m_map
%%% 2. download m_map from web or gfd1 website and put into your m_map folder
%%% 2. add this m_map folder to your matlab path (use ' addpath ')...try help m_contour to see that it's there
%%% 3. visit the m_map website of Rich Pawlowicz to learn more
%%% 4. set up the plot with commands similar to the following:

%%% Plot 1000 HPa dyn height field in colors with contours of 250 Hpa dyn height
m_proj('satellite', 'lon',300,'lat', 65);  %% sets type of projection and
m_pcolor(lon,lat,squeeze(double(z(:,:,1,length(t))))'); shading interp
colorbar
caxisretain=caxis;
% m_grid('xaxislocation','middle');     %% draws grid: erratic behavior under some Matlab releases. Try it
m_coast('color','b','linewidth',1);        %% draw coastlines
hold on
m_contour(lon,lat,squeeze(double(z(:,:,9,length(t))))','k');
    %% it seems necessary to specify color (here black 'k').
hold off     %% the contouring will change the color settings, so reset them
    caxis(caxisretain);
% m_grid('xaxislocation','middle');     %% draws grid: erratic behavior under some Matlab releases. Try it
    axis equal

%%% explore shading
spinmap      %% explore shading
caxis       %% find limits of color palette

%%% make animation
for it=1:size(t)
    m_pcolor(lon,lat,squeeze(double(te(:,:,1,it))))', shading interp; axis equal
    colorbar
caxis([230 310]);                          % fix color scheme degrees K
    m_coast('color','b','linewidth',1);       % draw coastlines
    hold on
    m_contour(lon,lat,squeeze(double(z(:,:,1,it))'),20,'k')
    m_grid('xaxislocation','middle');         % draws grid
    hold off
timestring=datestr(t(it)/24+datenum('1-Jan-1800'));  % seems to work
    title(['1000 HPa temperature (colors) and Z250 upper troposphere dyn height time ' timestring ])
drawnow
end