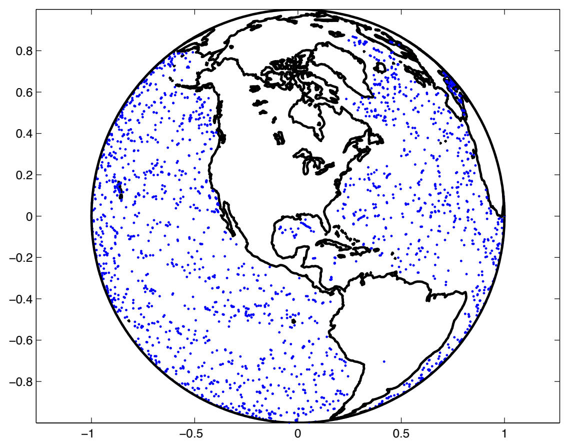
# Assignment 2: Stereographic Map Projection and the Argo Project




In this assignment, you'll create a map of the Earth that looks like a sphere in space, as shown at right. This is what's called a "stereographic projection". You'll plot both the outlines of the continents, and the current locations of a set of ocean-exploring robots called "Argo Floats".

## Argo Floats

Argo floats are ocean measuring robots, as shown at right. When dropped overboard from a ship, they sink down to a pre-programmed depth in the ocean (usually 1000 m). They drift with the ocean currents, and once every ten days they rise back up to the surface, measuring a profile of ocean temperature and salinity along the way. They radio their position and temperature/salinity data to a satellite, then sink back down again.

There are around 3600 Argo floats currently active, reporting on the changing temperature, salinity, and currents throughout the world's oceans. For more information, see

http://en.wikipedia.org/wiki/Argo_(oceanography)

The latest Argo data can be found at

http://www.usgodae.org/argo/argo.html

For this lab, we're not going to use the temperature and salinity data: we just want a report of where all the floats are located this week.

You can find on OnCourse a link to a saved data file containing the locations of every location reported by every Argo float ever deployed, as of September 12, 2018: the filename is **`ar_index_global_prof.txt`**. It is quite large — about 120 megabytes, containing 1.28 million location reports. (You can also download today's data from the usgodae.org link above, if you're patient.) This file is in compressed "zip" format on OnCourse: you can un-zip it by double-clicking on a Mac, or right clicking and choosing "extract all" on Windows.

This dataset is so big it's difficult to look at with a text editor. You can find a small sample of the file (the first 50 lines) at **`ar_index_global_prof_sample.txt`**

## Coastline Data

You can find a data file with a list of latitude, longitude coordinates for all the world's coastlines on OnCourse: the file is named coastlines.mat.

## Map Projection Mathematics

The following mathematics, derived in class, may be useful.

Suppose we have a point on a unit sphere whose latitude is $\theta$ and longitude is $\phi$. Suppose the sphere is at the origin, oriented so latitude zero, longitude zero is on the +X axis, the north pole (latitude +90°) is on the +Y axis, and longitude -90° is on the +Z axis. The *x,y,z* coordinates of the point ($\phi$, $\theta$) are as follows:

$$x = \cos(\phi)\cos(\theta)$$
$$y = \sin(\theta)$$
$$z = -\sin(\phi)\cos(\theta)$$

Suppose we take a point whose (*x,y,z* ) coordinates are known, and we rotate it about one of the three coordinate axes by an angle $\lambda$. The new coordinates of the point will be (*x', y', z'*), where:

Rotation about X axis:

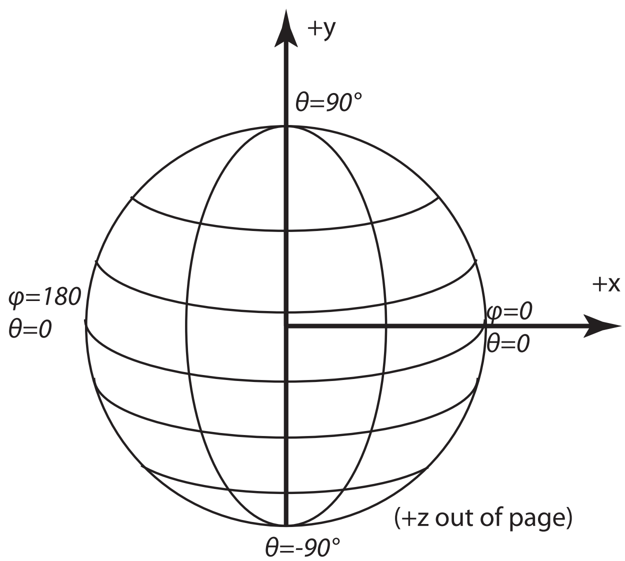
$$x' = x$$
$$y' = y\cos(\lambda) - z\sin(\lambda)$$
$$z' = z\cos(\lambda) + y\sin(\lambda)$$

Rotation about Y axis:

$$x' = x\cos(\lambda) + z\sin(\lambda)$$
$$y' = y$$
$$z' = z\cos(\lambda) - x\sin(\lambda)$$

Rotation about Z axis:

$$x' = x\cos(\lambda) - y\sin(\lambda)$$
$$y' = y\cos(\lambda) + x\sin(\lambda)$$
$$z' = z'$$

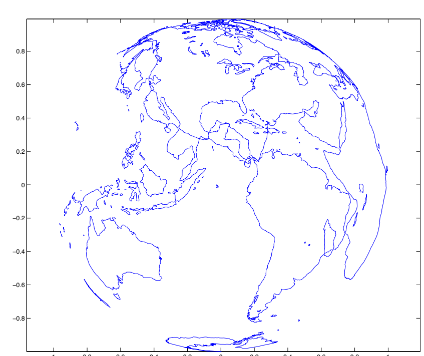



## Your Tasks

1. Load up the coastline data, and plot the coastlines as an ordinary rectangular map.

2. Write a function that will take a latitude, longitude point on a sphere and convert it to x, y, z coordinates. Write your function so that it will also work on an entire list of data points at once.

3. Send the coastline data through your function and plot x vs y, to simulate the view looking down the z axis. You should see coastlines on a spherical looking Earth, which is good, but you should see the “front” and “back” of the Earth overlapped on one another, which is ugly. Modify your code so that only points on the front of the Earth are shown, and points on the back are invisible.



4. Write a function that will rotate the globe on its axis by an arbitrary number of degrees ("spin"), and a function that will tilt the north pole of the globe toward or away from the viewer by an arbitrary number of degrees ("tilt"). Use these functions to create a plot showing the Earth from an attractive angle.

5. Read in the Argo float data. This is tricky to do: I recommend using the following line of code
   ```
   [file,date,lat,lon,ocean,prof_type,institution,date_update] = ...
   textread('ar_index_global_prof.txt','%s%n%n%n%c%s%s%n',...
   'headerlines',9,'delimiter',',');
   ```

   (The "**...**" indicates that a command continues onto the next line.) Use an editor to view the sample version of the float data file, and read the help for the "textread" command. Explain exactly what this command is doing, and how I came up with it.

6. We don't want to plot all of the Argo float data going back to the 1990s, we only want the latest positions of the floats. Figure out how dates work in the Argo dataset, and explain it. Then write code to extract only float positions from the last 10 days. (use **date**, not **date_update**.)

7. Some of this data is bad. Write code to extract only float positions whose latitude and longitude are physically possible values.

8. Make a map similar to the one at the top of the first page of this assignment, which shows the float positions as dots and the coastlines as lines. Use various options to the plot() command to make it pretty. Notice I added a circle to mark the outer edge of the sphere to make it prettier.

9. Extra credit: do something cool with this code. For instance, you might try to plot some other interesting location data, or create an animated version with a spinning globe. (hint: "**drawnow**")

## Some MATLAB Advice

1. You can use conditional tests to select values from a list. For example, if **x** and **y** are lists of numbers with the same length,
   ```
   x(x > 5)
   ```
   will return a list of all values of x that are greater than 5, and
   ```
   y(x > 5)
   ```
   will return a list of all values of y, where the corresponding value of x is greater than 5.

2. Make sure you write all your functions so that they work not just on one number, but on lists of numbers.

3. **NaN** is a special numeric value in MATLAB, called "not a number". It's the result of invalid math operations like 0 / 0, but it has a special purpose for graphics. **NaN** values will not be plotted by the **plot()** command. This is a handy way to make parts of a plot invisible.

4. It's possible to do this entire assignment in MATLAB without writing any "for" or "while" loops.

5. If your Earth looks squashed, check out the "**axis equal**" command.

6. Don't forget that MATLAB trig functions work in radians, not degrees!