Table of Contents

John Shuler

GEOS597 Homework #4: Sea-level rise and mapping toolbox

Due: 10/3/2016

```
close all;
clear all;
clc;
```

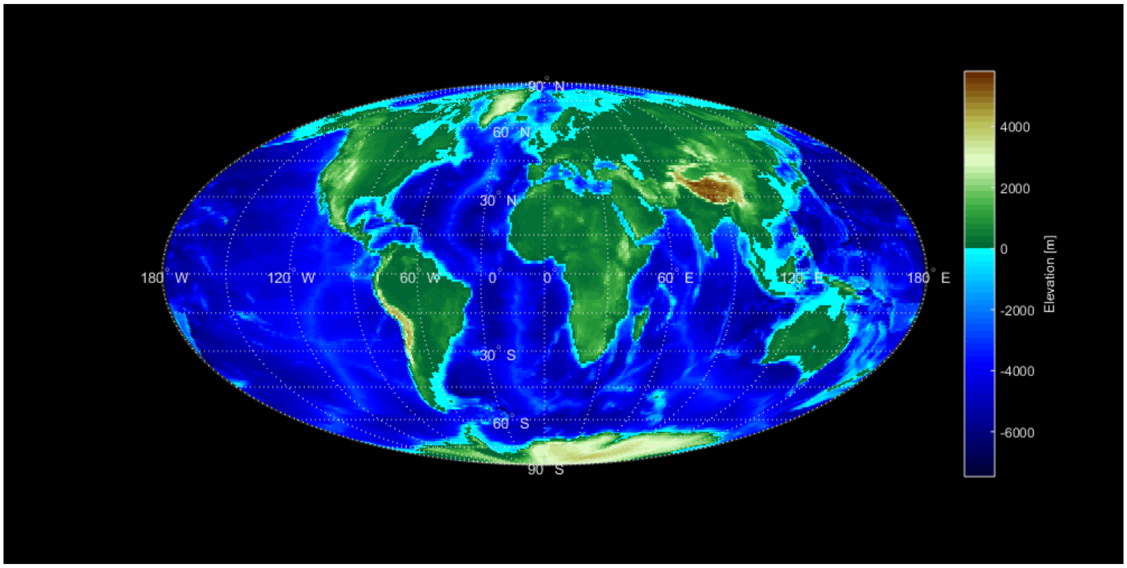
Part 1: Make a basemap and determine a crude ocean volume

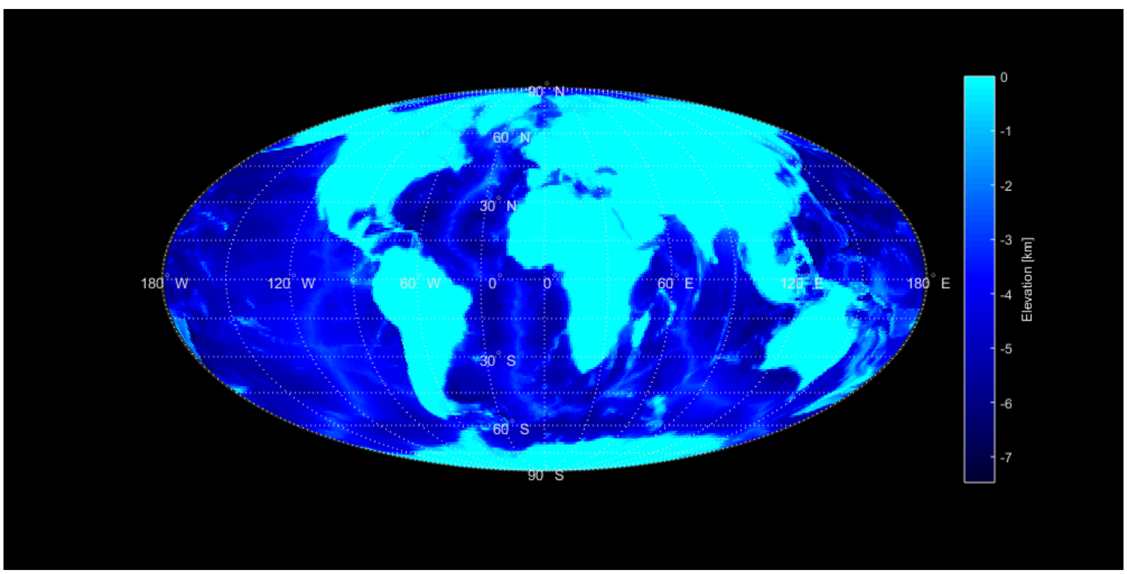
Step 1: Load the topo.mat data set

```
load('topo.mat');

% *Step 2: Plot a basemap*

h = figure;
h.InvertHardcopy = 'off'; % ensures color of saved fig. matches
 display
h.Color = 'k'; % (1 pt.) % changes fig. color to black
h.Position = [100 100 1000 500]; % specifies location/size of fig's
 drawable area
h.PaperPositionMode = 'auto'; % preserves fig's aspect ratio when
 printing
ax = axesm('Mollweid','Frame', 'on', 'Grid', 'on'); % sets map
 projection, inserts globe outline, inserts grid
setm(ax,'MLabelLocation',60); % sets lon. labels to every 60 degrees
setm(ax,'PLabelLocation',30); % sets lat. labels to every 30 degrees
mlabel('MLabelParallel',0); % sets lon. label location to the equator
plabel('PLabelMeridian',-25); % sets lat. label location to prime
 meridian
axis('off'); % prevents axes display
setm(ax,'FontColor',[0.9 0.9 0.9]); % brightens text
setm(ax,'GColor',[0.9 0.9 0.9]); % brightens grid
load('coastlines'); % load built-in MATLAB data called coastlines
plotm(coastlat, coastlon);
LAT = topolatlim(1):topolatlim(2);
LON =  topolonlim(1):topolonlim(2);
[lon, lat] = meshgrid(LON,LAT); % compute the lat/lon of every grid
 point in topo
```

```
pcolorm(lat,lon,topo); % plot the matrix of elevations on the map
demcmap(topo); % give it a better colormap
c = colorbar('color', [0.9 0.9 0.9]);
c.Label.String = 'Elevation [m]';
title('Current Sea Level')

% *Step 3: Modify the _topo_ matrix to represent ocean depth*

topokm = topo/1000;   % convert to depth to km
topokm(topokm>0) = 0;
topoSea = topokm;

h = figure;
h.InvertHardcopy = 'off'; % ensures color of saved fig. matches
 display
h.Color = 'k'; % (1 pt.) % changes fig. color to black
h.Position = [100 100 1000 500]; % specifies location/size of fig's
 drawable area
h.PaperPositionMode = 'auto'; % preserves fig's aspect ratio when
 printing
ax = axesm('Mollweid','Frame', 'on', 'Grid', 'on'); % sets map
 projection, inserts globe outline, inserts grid
setm(ax,'MLabelLocation',60); % sets lon. labels to every 60 degrees
setm(ax,'PLabelLocation',30); % sets lat. labels to every 30 degrees
mlabel('MLabelParallel',0); % sets lon. label location to the equator
plabel('PLabelMeridian',-25); % sets lat. label location to prime
 meridian
axis('off'); % prevents axes display
setm(ax,'FontColor',[0.9 0.9 0.9]); % brightens text
setm(ax,'GColor',[0.9 0.9 0.9]); % brightens grid
load('coastlines'); % load built-in MATLAB data called coastlines
plotm(coastlat, coastlon);
LAT = topolatlim(1):topolatlim(2);
LON =  topolonlim(1):topolonlim(2);
[lon, lat] = meshgrid(LON,LAT); % compute the lat/lon of every grid
 point in topo
pcolorm(lat,lon,topoSea); % plot the matrix of elevations on the map
demcmap(topoSea); % give it a better colormap
c = colorbar('color', [0.9 0.9 0.9]);
c.Label.String = 'Elevation [km]';
title('Current Ocean Depth')

% *We can determine the length and width of each pixel, then use our
 known
% depth values to calculate volume for each ocean pixel. These can be
 used
% to calculate total volume.

% *Step 4: Compute the area of each pixel*

radiusEarth = 6371;     %[km]
circEarth = 2*pi*radiusEarth;
pixArea = (circEarth/360)*(circEarth/360);
```

```
% * The area of a 1 deg x 1 deg pixel is 1.2364E4 km^2*

oceanvolumeArray = abs(topoSea*pixArea);
oceanVolume = sum(oceanvolumeArray(:));

% * Total volume of the oceans in this rough estimation is 1.8226E9
 km^3.*
```

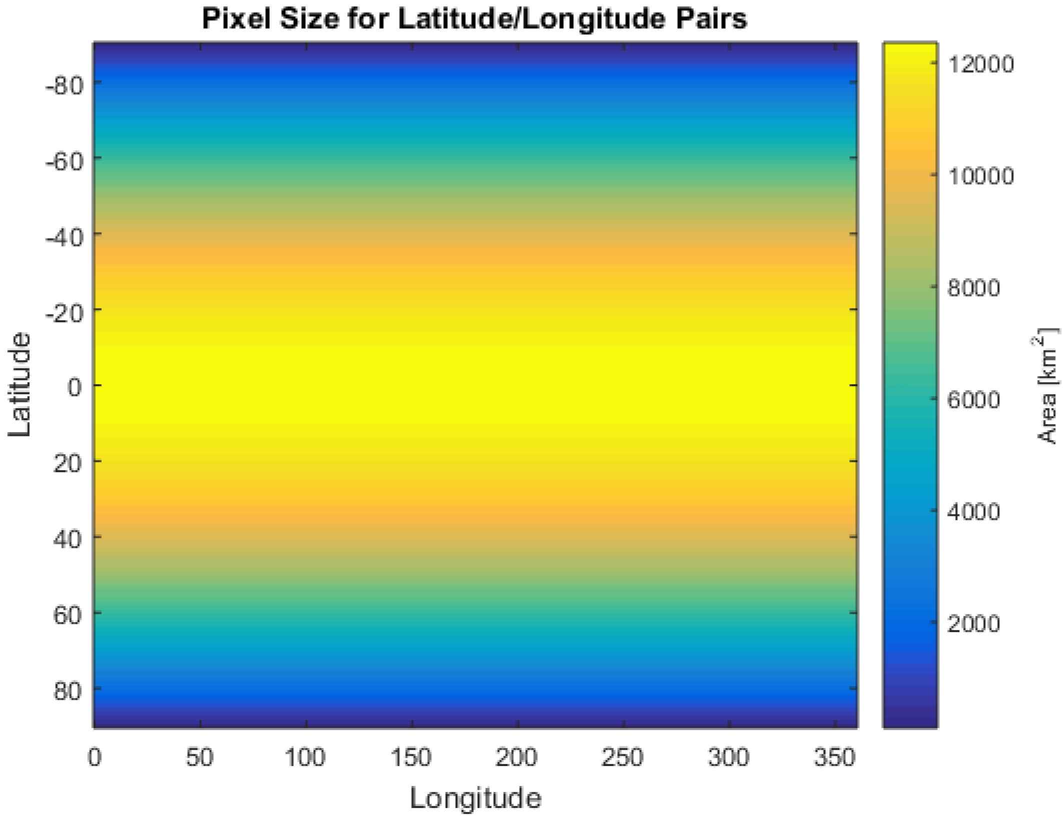




Part 2: A more accurate volume estimate

```
% *We are definitely overestimating ocean volume due to the fact that
 pixel
% area tapers toward the poles. Assuming fixed pixel size leads to
% overestimation.

% *Step 1: Compute the area between two lines of latitude*
```

```
ringArea = 2*pi*(radiusEarth^2)*abs(sind(29)-sind(30));
cellArea = ringArea/360;

% *The area of a single cell between 29-30N and two lines of latitude
 is
% 1.0761E4 km^2.

% *Step 2: Compute the area of each pixel*

earthPixels = zeros(90,360);
for i = 0:89;
    ringArea2 = 2*pi*(radiusEarth^2)*abs(sind(i)-sind(i+1));
    earthPixels(i+1,:) = ringArea2/360;
end
earthPixels2 = flipud(earthPixels);
earthPixels3(1:90,:) = earthPixels2;
earthPixels3(91:180,:) = earthPixels;

h = figure;
imagesc(LON,LAT,earthPixels3);
c = colorbar;
c.Label.String = 'Area [km^2]';
xlabel ('Longitude')
ylabel ('Latitude')
title('Pixel Size for Latitude/Longitude Pairs')

% *Step 3: Compute the ocean's volume*

oceanVolumeArray2 = abs(earthPixels3.*topoSea);
oceanVolumecorrected = sum(oceanVolumeArray2(:));

% *The total volume of the earth's oceans is estimated to be 1.3369E9
% km^3.*

oceanVolumedifference = oceanVolume-oceanVolumecorrected;

% *The difference between the two estimation methods is 4.8568E8
 km^3.*
```

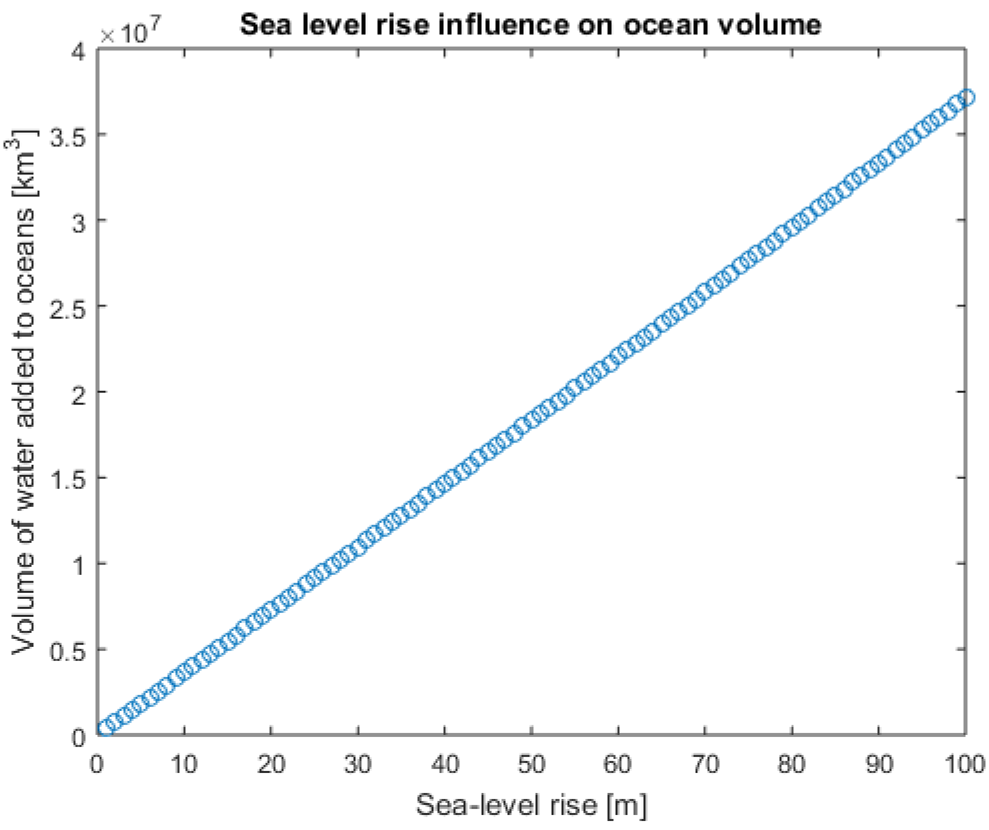


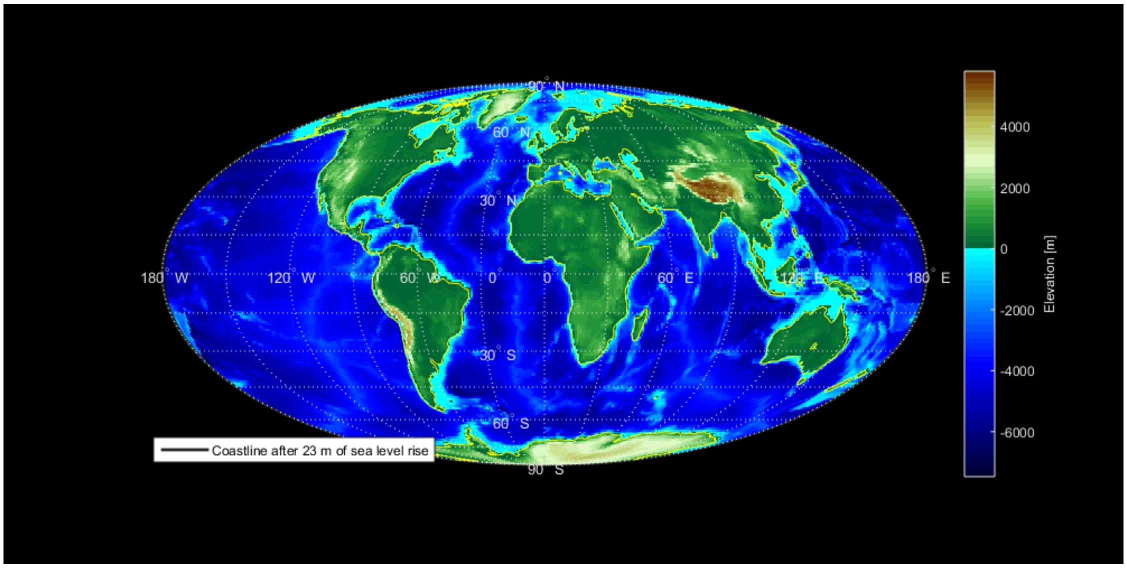
Part 3: Sea-level rise due to Antarctica

```
topom = topo;
topom(topom<0) = 0;
topoLand = topom./1000;
landvolumeArray = topoLand.*earthPixels3;
landvolumeArray2 = landvolumeArray (151:180,:);
antarcticLandvolume = sum(landvolumeArray2(:));

%*The total volume of mass above sea level below 60S is 9.1564E6 km^3.

antarcticWatervolume = antarcticLandvolume*0.9;

%*The total volume of liquid water stored in Antarctic ice is 8.2407E6
%km^3.

% *Step 2: Compute the change in total ocean volume for incremental
 changes
% in sea level height

seaRise = zeros(1,100);
oceanVolumeArray4 = zeros(1,100);
for i = 1:100;
    seaRise(i) = i;
    topo2 = topo-i;
```

```
    topo2(topo2 > 0) = 0;
    topo3 = (topo2)/1000;
    oceanVolumeArray3 = abs(topo3.*earthPixels3);
    oceanVolumeArray4(i) = sum(oceanVolumeArray3(:))-
oceanVolumecorrected;
end

h = figure;
plot (seaRise, oceanVolumeArray4,'o');
title ('Sea level rise influence on ocean volume')
xlabel('Sea-level rise [m]')
ylabel('Volume of water added to oceans [km^3]')

%*Step 3: Match the change in volume with the volume held on
 Antarctica

volumeMatch = find (oceanVolumeArray4 >= antarcticWatervolume);
volumeMatch2 = oceanVolumeArray4(23);

% *The sea level rise that roughly matches the water held in Antarctic
 sea ice is 23 m.*

h = figure;
h.InvertHardcopy = 'off'; % ensures color of saved fig. matches
 display
h.Color = 'k'; % (1 pt.) % changes fig. color to black
h.Position = [100 100 1000 500]; % specifies location/size of fig's
 drawable area
h.PaperPositionMode = 'auto'; % preserves fig's aspect ratio when
 printing
ax = axesm('Mollweid','Frame', 'on', 'Grid', 'on'); % sets map
 projection, inserts globe outline, inserts grid
hold on;
setm(ax,'MLabelLocation',60); % sets lon. labels to every 60 degrees
setm(ax,'PLabelLocation',30); % sets lat. labels to every 30 degrees
mlabel('MLabelParallel',0); % sets lon. label location to the equator
plabel('PLabelMeridian',-25); % sets lat. label location to prime
 meridian
axis('off'); % prevents axes display
setm(ax,'FontColor',[0.9 0.9 0.9]); % brightens text
setm(ax,'GColor',[0.9 0.9 0.9]); % brightens grid
load('coastlines'); % load built-in MATLAB data called coastlines
plotm(coastlat, coastlon);
LAT = topolatlim(1):topolatlim(2);
LON =  topolonlim(1):topolonlim(2);
[lon, lat] = meshgrid(LON,LAT); % compute the lat/lon of every grid
 point in topo
pcolorm(lat,lon,topo); % plot the matrix of elevations on the map
demcmap(topo); % give it a better colormap
c = colorbar('color', [0.9 0.9 0.9]);
c.Label.String = 'Elevation [m]';
hold on;
LATnew = LAT(1:180);
LONnew = LON(1:360);
```

```
V = [23 23];
contourm(LATnew,LONnew,topo, V,'y');
legend ('Coastline after 23 m of sea level
 rise','Location','southwest')

%*The boot of southern Italy and many Pacific islands would be
 permanently
%flooded in this scenario.*
%*If available, we could use a higher resolution elevation grid for
%increased ocean depth accuracy. Also, we assume that each ocean
 volume
%below a pixel is a rectangular prism, when in fact its length should
 taper
%somewhat. This results in a (perhaps negligible) overestimation of
 ocean
%volume.
%This model neglects the portion of Antarctica's above-sea-level mass
 that
%is rock rather than ice, e.g. the Transantarctic Mountains. This
%could significantly skew our estimation of liquid water held on the
%continent. Last, any conditions that result in the complete melting
 of
%Antarctica will almost certainly result in significant melting
 elsewhere
%on earth and total sea level rise will be greater than the 23 meters
%predicted here.*
```





Part 4: Sea-level rise due to Greenland

```
%*Isolating only the Greenland land mass is more difficult than
 isolating
%Antarctica, since there are other land masses at that latitude to
 contend
%with. However, we decided to draw a rough box around Greenland from
%60 N to 84 N and from 70W to 20W. Unfortunately, this box also
 includes
%parts of Canada and Iceland, so this calculation will certainly be an
%overestimation. We used these coordinates to select an array
 representing
%Greenland from our topo array. After this point, we used the same
 methods
%as we used to estimate liquid water content on Antarctica. All values
 in
%topo below zero were set to zero and the resulting matrix was
 multiplied
%by our previously calculated pixel areas to produce an array of land
%volumes. This land volume was summed and multiplied by 0.9 to account
 for
%the density difference between ice and liquid water.

greenlandArray=topo(:,110:160);
greenlandArray2 = greenlandArray(6:30,:);
greenlandArray2(greenlandArray2<0) = 0;
greenlandLand = greenlandArray2./1000;
earthPixels4 = earthPixels3(:,110:160);
earthPixels5 = earthPixels4(6:30,:);
greenlandvolumeArray = greenlandLand.*earthPixels5;
greenlandVolume = sum(greenlandvolumeArray(:));
greenlandWatervolume = greenlandVolume*0.9;

%*The liquid water volume stored on Greenland by this estimation is
```

```
%5.6121E6 km^3. As noted earlier, this is definitely an overestimation
 due to
%the rough clip of Greenland's area and how we neglected the existence
 of land where we assumed only ice.
```

Published with MATLAB® R2015a