

GG250 Lab 11

World maps

In addition to working on the line-fitting problem, our hero Joe Blow was also asked to develop some simple Matlab code to plot world maps that could be used as backgrounds for plotting geographic data sets. Joe attacked this problem with his usual vigor and developed a set of functions that simplified the process by using function handles and global variables. It is now a simple matter to add more map projection. Joe wrote one script and a main function:

<code>geomap.m</code>	Main script plots world maps by calling <code>geoworldmap.m</code>
<code>geoworldmap.m</code>	Draws a single world map. It contains several sub-functions:
<code>geoinit</code>	Function that initializes the chosen map projection
<code>geofill</code>	Function that paints a polygon using the given color
<code>geowagner</code>	Function that implements the Wagner projection
<code>geosinusoidal</code>	Function that implements the Sinusoidal projection
<code>geohammer</code>	Function that implements the Hammer projection

Your tasks are as follows:

1. Copy all the lab files to your directory, rename `geoworldmap.m` and `geomap.m` so they end with `_name`, and change all internal references to `geoworldmap` accordingly.
2. Add the Eckert projection to the package. It is defined as

$$x = 2\sqrt{2/(3\pi)}R\lambda[1 - |\phi|/\pi]$$
$$y = 2\sqrt{2/(3\pi)}R\phi$$

where (λ, ϕ) is longitude and latitude (in radians), R is the radius of the (spherical) Earth (6371.008 km), and (x, y) is the resulting map position. Do this by creating a sub-function `geoeckert` (model it after the other projection sub-functions) and then make appropriate additions to Joe's code. Add a call in `geomap_name.m` to create a yellow Eckert map in the 4th window.

3. Joe never implemented gridlines. We want to see meridians and parallels drawn on the map, not the rectangular x/y grid generated by "grid on". Append a sub-function called `geogrid` that is called at the end of the main `geoworldmap` function. It should take three input arguments (`delta_lon`, `delta_lat`, `color`) and draw meridians every `delta_lon` and parallels every `delta_lat` in latitude (there are no output arguments). Use 30° for both intervals and a green line color. Generate the coordinates needed to do this using an interval of 1° for the $[\pm 180^\circ, \pm 90^\circ]$ domain. To do the actual drawing, create a new sub-function `geoplot` using `geofill` as your model. The `geoplot` sub-function should take the same arguments as `geofill`, but instead of filling a polygon it simply should draw a colored line using the Matlab `plot` command.

Submit `geomap_name.m` and `geoworldmap_name.m`.