basemap

As Randy has shown, matplotlib has a lot of functionality. There are times when you want to take it further. This is especially true when you want to alter geographic projections, plot multiple data sets, and interact with web mapping services.

In [1]:
```
%matplotlib inline

from mpl_toolkits.basemap import Basemap #also have Cartopy
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Polygon
```

In [2]:
```
# setup Lambert Conformal basemap.
# set resolution=None to skip processing of boundary datasets.
m = Basemap(width=12000000,
            height=9000000,
            projection='lcc',
            resolution=None,
            lat_1=45.,
            lat_2=55,
            lat_0=50,
            lon_0=-107.)
m.bluemarble()
plt.show()
```

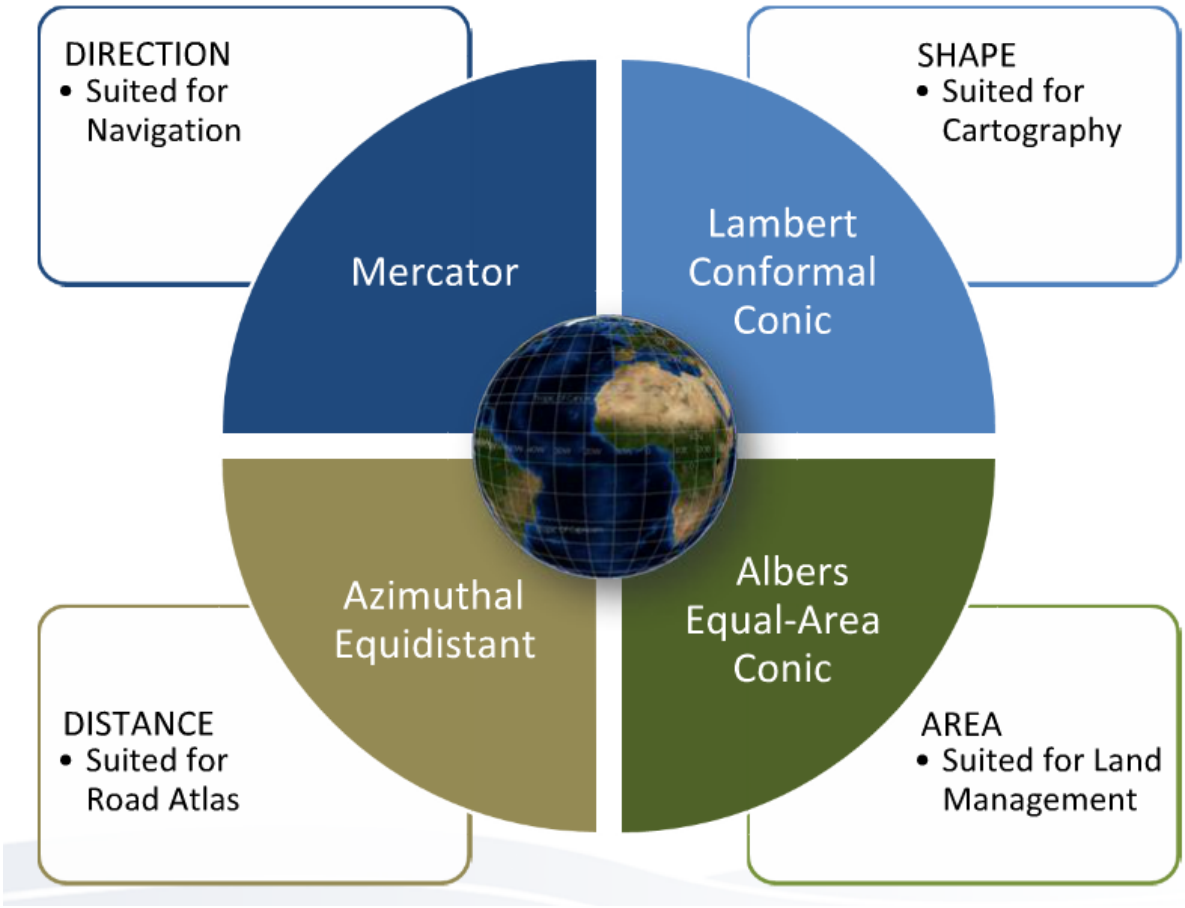
shaded relief

In [3]:
```
%matplotlib inline

from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
# setup Lambert Conformal basemap.
# set resolution=None to skip processing of boundary datasets.

m = Basemap(width=12000000,height=9000000,projection='lcc',
            resolution=None,lat_1=45.,lat_2=55,lat_0=50,lon_0=-107.)
m.shadedrelief()
plt.show()
```

drawlsmask()

In [4]:
```
%matplotlib inline

from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
# setup Lambert Conformal basemap.
# set resolution=None to skip processing of boundary datasets.
m = Basemap(width=12000000,height=9000000,projection='lcc',
            resolution=None,lat_1=45.,lat_2=55,lat_0=50,lon_0=-107.)
m.drawlsmask()
plt.show()
```

Geographic reference frame: treats the earth as a sphere



projections ¶

experment by changing the useProj variable

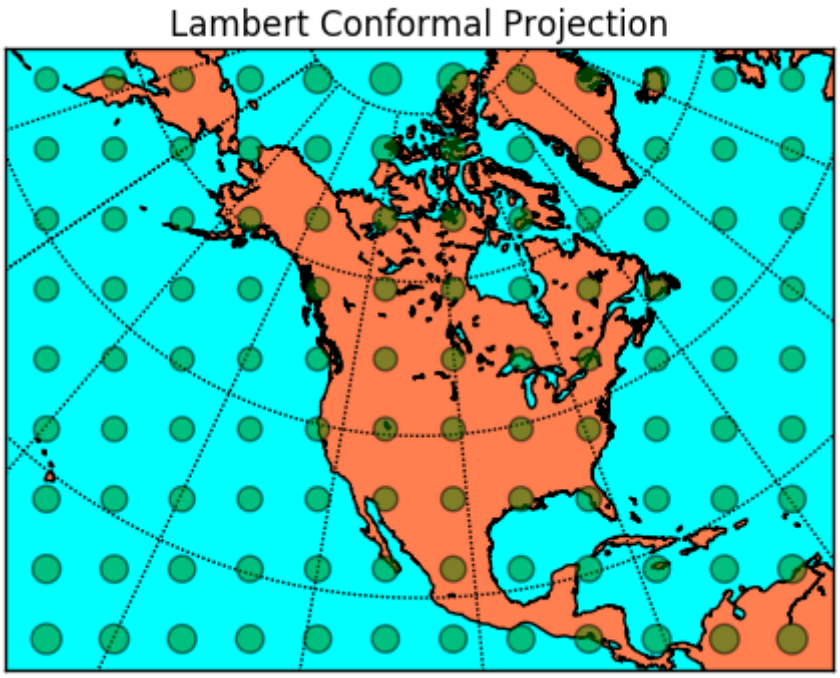
References

this USGS website (http://egsc.usgs.gov/isb//pubs/MapProjections/projections.html#lambert) is very useful, and I often come back to it.

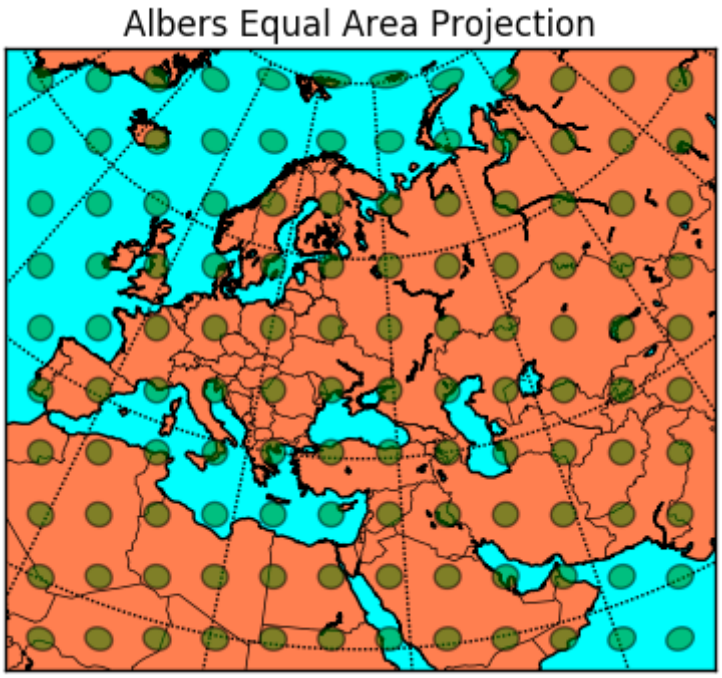
using this list (http://matplotlib.org/basemap/users/mapsetup.html) of projections, explore how different projections preform using a Tissot's indicatrix (https://en.wikipedia.org/wiki/Tissot%27s_indicatrix)

In [ ]:

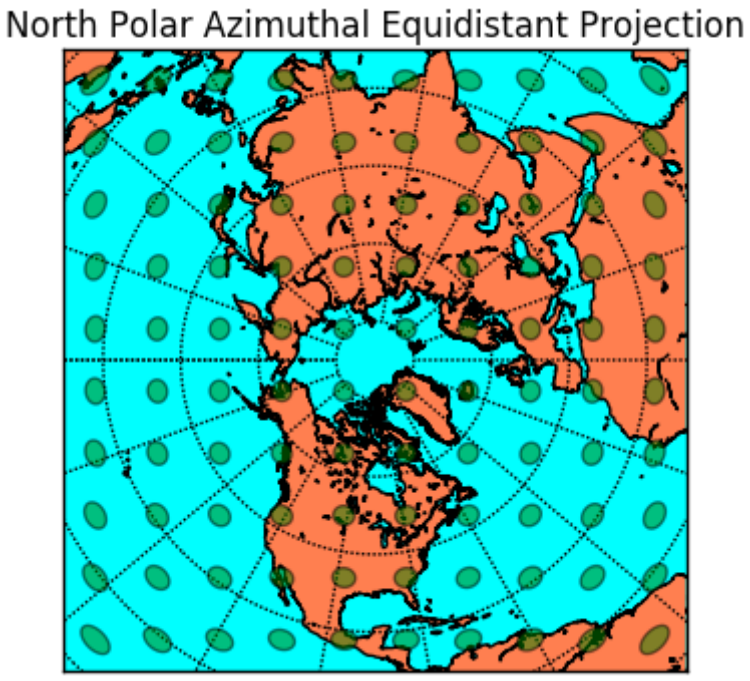
In [5]:
```
# setup lambert conformal basemap.
# lat_1 is first standard parallel.
# lat_2 is second standard parallel (defaults to lat_1).
# lon_0,lat_0 is central point.
# rsphere=(6378137.00,6356752.3142) specifies WGS4 ellipsoid
# area_thresh=1000 means don't plot coastline features less
# than 1000 km^2 in area.
m = Basemap(width=12000000,height=9000000,
            rsphere=(6378137.00,6356752.3142),\
            resolution='l',area_thresh=1000.,projection='lcc',\
            lat_1=45.,lat_2=55,lat_0=50,lon_0=-107.)
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
# draw parallels and meridians.
m.drawparallels(np.arange(-80.,81.,20.))
m.drawmeridians(np.arange(-180.,181.,20.))
m.drawmapboundary(fill_color='aqua')
# draw tissot's indicatrix to show distortion.
ax = plt.gca()
for y in np.linspace(m.ymax/20,19*m.ymax/20,9):
    for x in np.linspace(m.xmax/20,19*m.xmax/20,12):
        lon, lat = m(x,y,inverse=True)
        poly = m.tissot(lon,lat,1.5,100,\
                        facecolor='green',zorder=10,alpha=0.5)
plt.title("Lambert Conformal Projection")
plt.show()
```



In [6]:
```
# setup albers equal area conic basemap
# lat_1 is first standard parallel.
# lat_2 is second standard parallel.
# lon_0,lat_0 is central point.
m = Basemap(width=8000000,height=7000000,
            resolution='l',projection='aea',\
            lat_1=40.,lat_2=60,lon_0=35,lat_0=50)
m.drawcoastlines()
m.drawcountries()
m.fillcontinents(color='coral',lake_color='aqua')
# draw parallels and meridians.
m.drawparallels(np.arange(-80.,81.,20.))
m.drawmeridians(np.arange(-180.,181.,20.))
m.drawmapboundary(fill_color='aqua')
# draw tissot's indicatrix to show distortion.
ax = plt.gca()
for y in np.linspace(m.ymax/20,19*m.ymax/20,10):
    for x in np.linspace(m.xmax/20,19*m.xmax/20,12):
        lon, lat = m(x,y,inverse=True)
        poly = m.tissot(lon,lat,1.25,100,\
                        facecolor='green',zorder=10,alpha=0.5)
plt.title("Albers Equal Area Projection")
plt.show()
```



In [7]:
```
# setup north polar aimuthal equidistant basemap.
# The longitude lon_0 is at 6-o'clock, and the
# latitude circle boundinglat is tangent to the edge
# of the map at lon_0.
m = Basemap(projection='npaeqd',boundinglat=10,lon_0=270,resolution='l')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
# draw parallels and meridians.
m.drawparallels(np.arange(-80.,81.,20.))
m.drawmeridians(np.arange(-180.,181.,20.))
m.drawmapboundary(fill_color='aqua')
# draw tissot's indicatrix to show distortion.
ax = plt.gca()
for y in np.linspace(m.ymax/20,19*m.ymax/20,10):
    for x in np.linspace(m.xmax/20,19*m.xmax/20,10):
        lon, lat = m(x,y,inverse=True)
        poly = m.tissot(lon,lat,2.5,100,\
                        facecolor='green',zorder=10,alpha=0.5)
plt.title("North Polar Azimuthal Equidistant Projection")
plt.show()
```



In [ ]:

In [ ]: