test_PJ_healpix.c

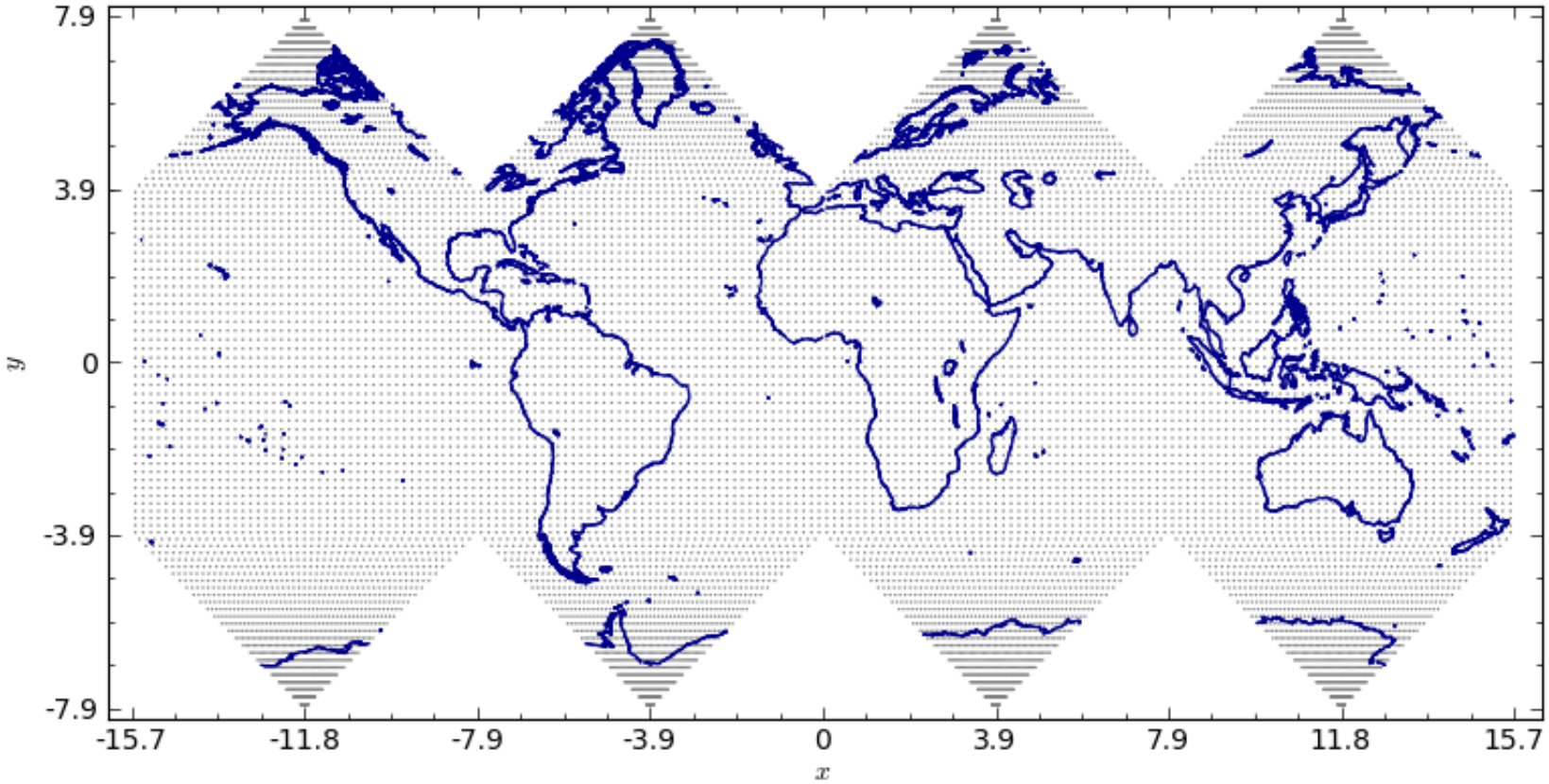
```
home = "/Users/arai021/Dropbox/scenzgrid/scenzgrid-py/0.4/src"   # Change to match your needs
#home = "/Users/raichev/Dropbox/scenzgrid/scenzgrid-py/0.4/src"   # Change to match your
needs
load_attach_path(home)
attach "ellipsoids.py"
attach "projection_wrapper.py"
import pyproj

# In your .bash_profile or .bash_rc make sure you have the line 'export SAGE_PATH=<home>'
# where <home> is `home` from above.
# This is necessary so that Sage knows where to look when it encounters import statements
# from attached files.
```

```
# Visual tests for PJ_healpix.c

for E in [Ellipsoid(R=5, radians=True), Ellipsoid(a=5, e=0.8, radians=True)]:
    E.lon_0 = 0
    print E
    ns, ss = 2, 3
    lattice = list(E.lattice())
    coast = E.coastlines(filename=home + '/coastlines.dat')
    #print 'input points:'
    #g = Graphics()
    #g = point(lattice, size=1, color='grey') + point(coast, size=1, color='darkblue')
    #g.show(aspect_ratio=1, figsize=fs, axes_labels=['$\lambda$','$\phi$'], frame=True,
axes=False, ticks=[pi/4, pi/4])
    for proj in ['healpix', 'rhealpix']:
        if proj == 'healpix':
            f = pyproj.Proj(proj=proj, a=E.a, e=E.e, lon_0=rad2deg(E.lon_0))
            #vertices = healpix_vertices(E)
```

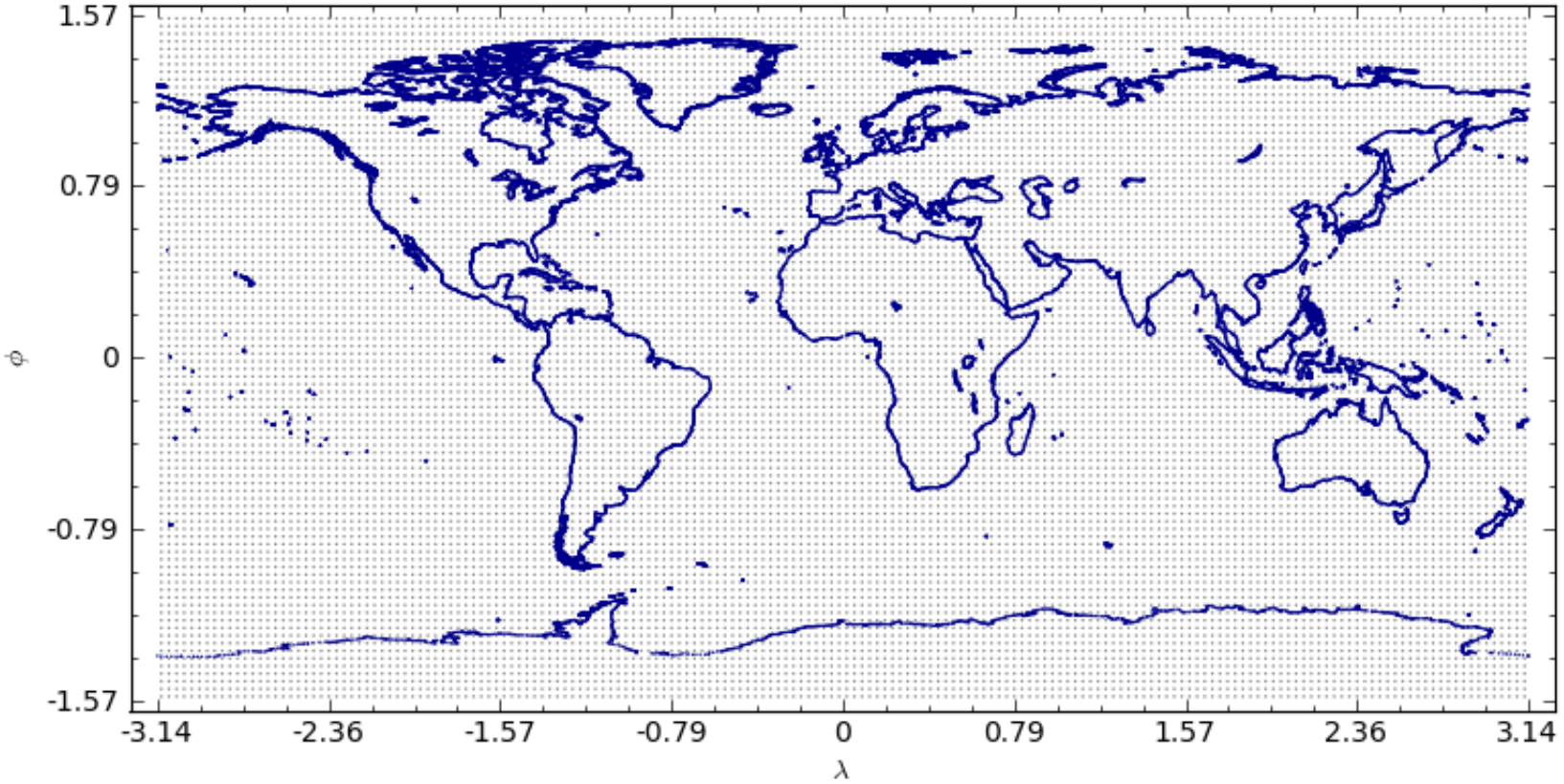
```
        elif proj == 'rhealpix':
            f = pyproj.Proj(proj=proj, a=E.a, e=E.e, lon_0=rad2deg(E.lon_0), north_square=ns,
south_square=ss)
            #vertices = rhealpix_vertices(E, ns, ss)
        print proj + ' forward:'
        g = Graphics()
        #g += line(vertices + [vertices[0]], color='black') # boundary
        f_lattice = [f(lam, phi, radians=True) for (lam, phi) in lattice]
        f_coast = [f(lam, phi, radians=True) for (lam, phi) in coast]
        g += point(f_lattice, size=1, color='grey')
        g += point(f_coast, size=1, color='darkblue')
        g.show(aspect_ratio=1, ticks=[E.R_A*pi/4, E.R_A*pi/4], axes_labels=['$x$','$y$'],
frame=True, axes=False)

        print proj + ' inverse:'
        finvf_lattice = []
        for (x, y) in f_lattice:
            p = f(x, y, radians=True, inverse=True)
            if abs(p[0]) > 2*pi:
                print "Out of bounds: x = %.15f, y = %.15f" % (x, y), p[0]
            else:
                finvf_lattice.append(p)
        finvf_coast = [f(x, y, radians=True, inverse=True) for (x, y) in f_coast]
        g = Graphics()
        g += point(finvf_lattice, size=1, color='grey')
        g += point(finvf_coast, size=1, color='darkblue')
        g.show(aspect_ratio=1, axes_labels=['$\lambda$','$\phi$'], frame=True, axes=False,
ticks=[pi/4, pi/4])
```

```
    ellipsoid:
        R = 5
        R_A = 5
        a = 5
        b = 5
        e = 0
        f = 0
```

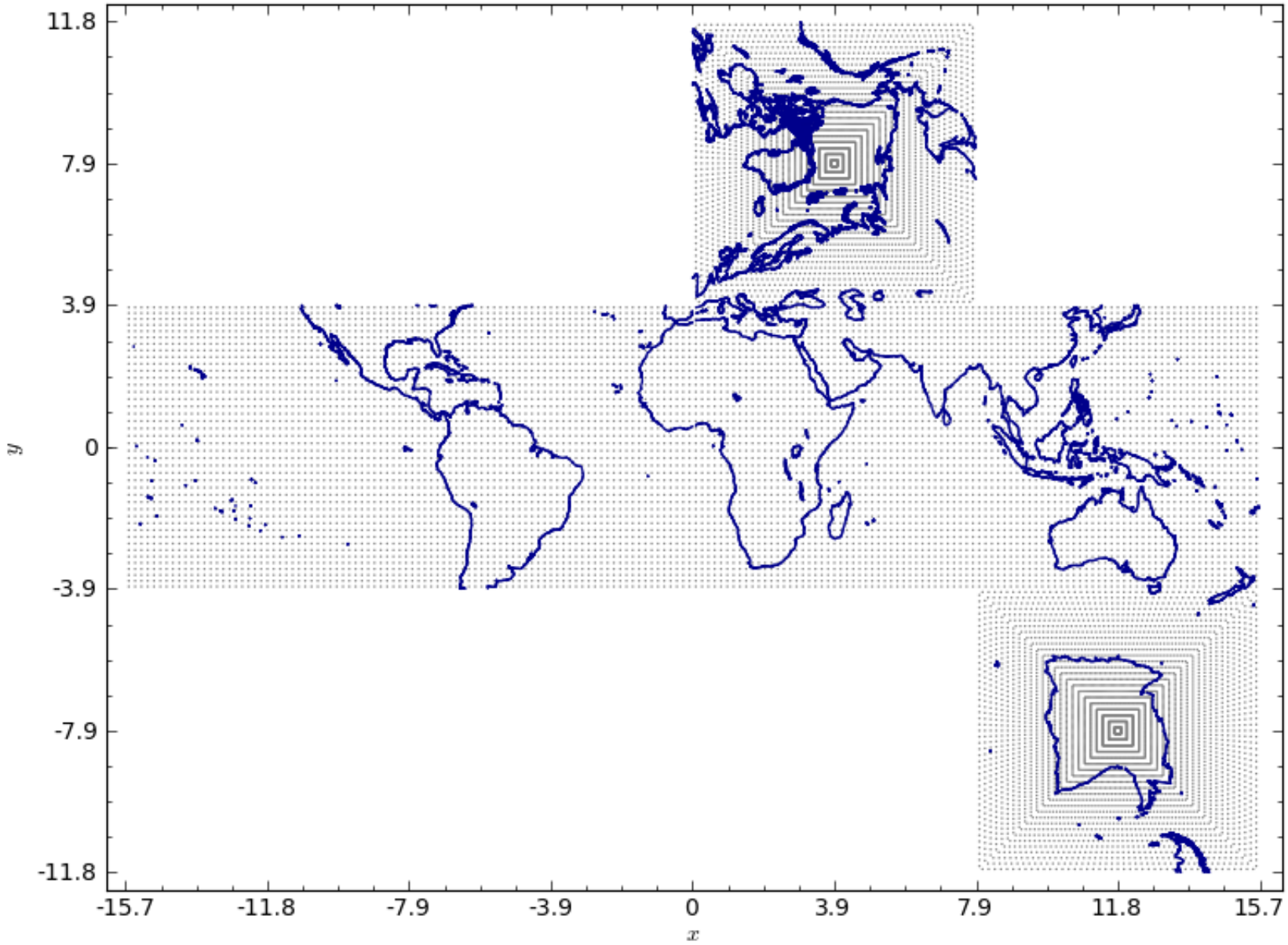
```
    lat_0 = 0
    lon_0 = 0
    radians = True
    sphere = True
healpix forward:
```



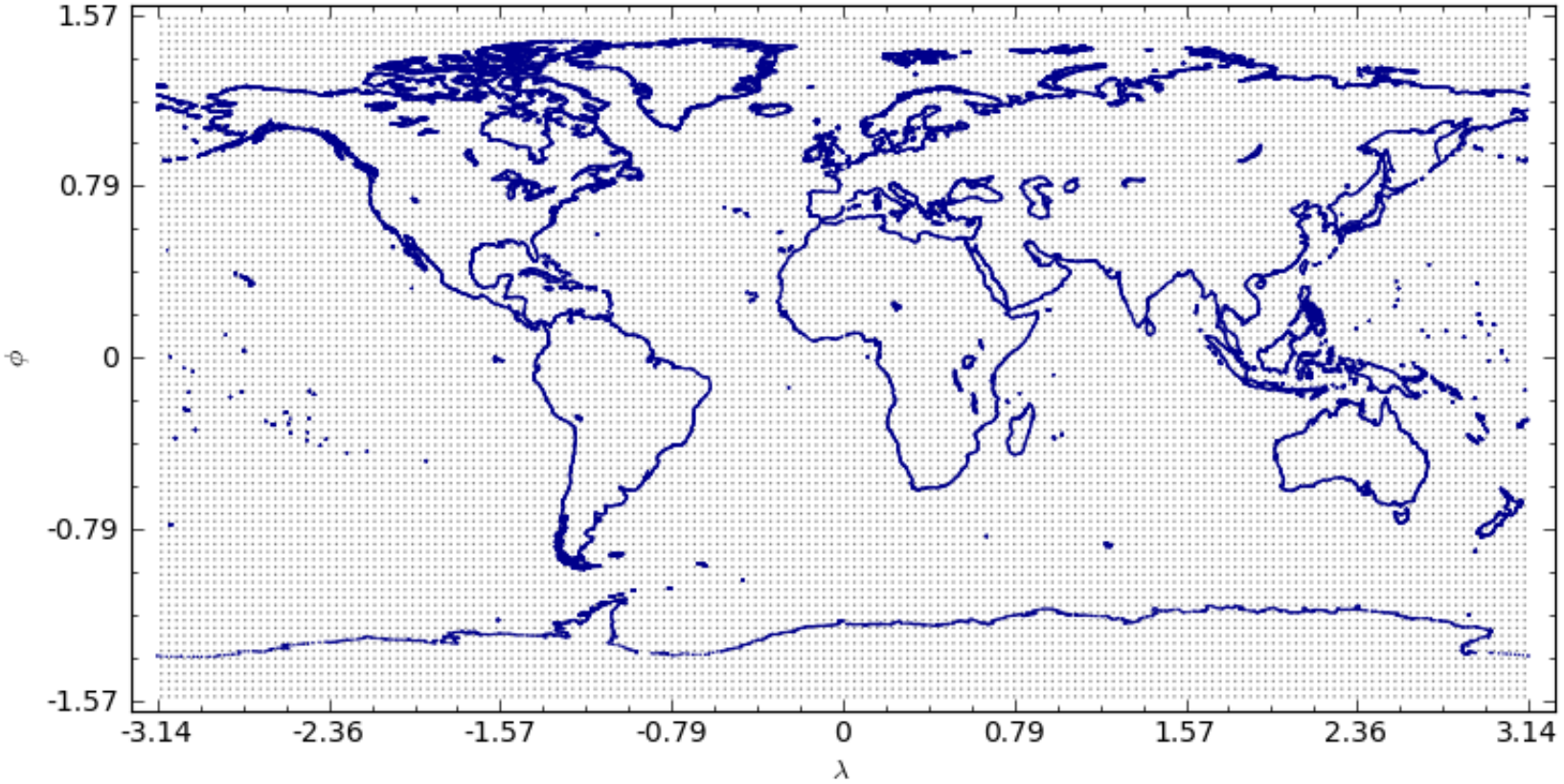

```
healpix inverse:
```



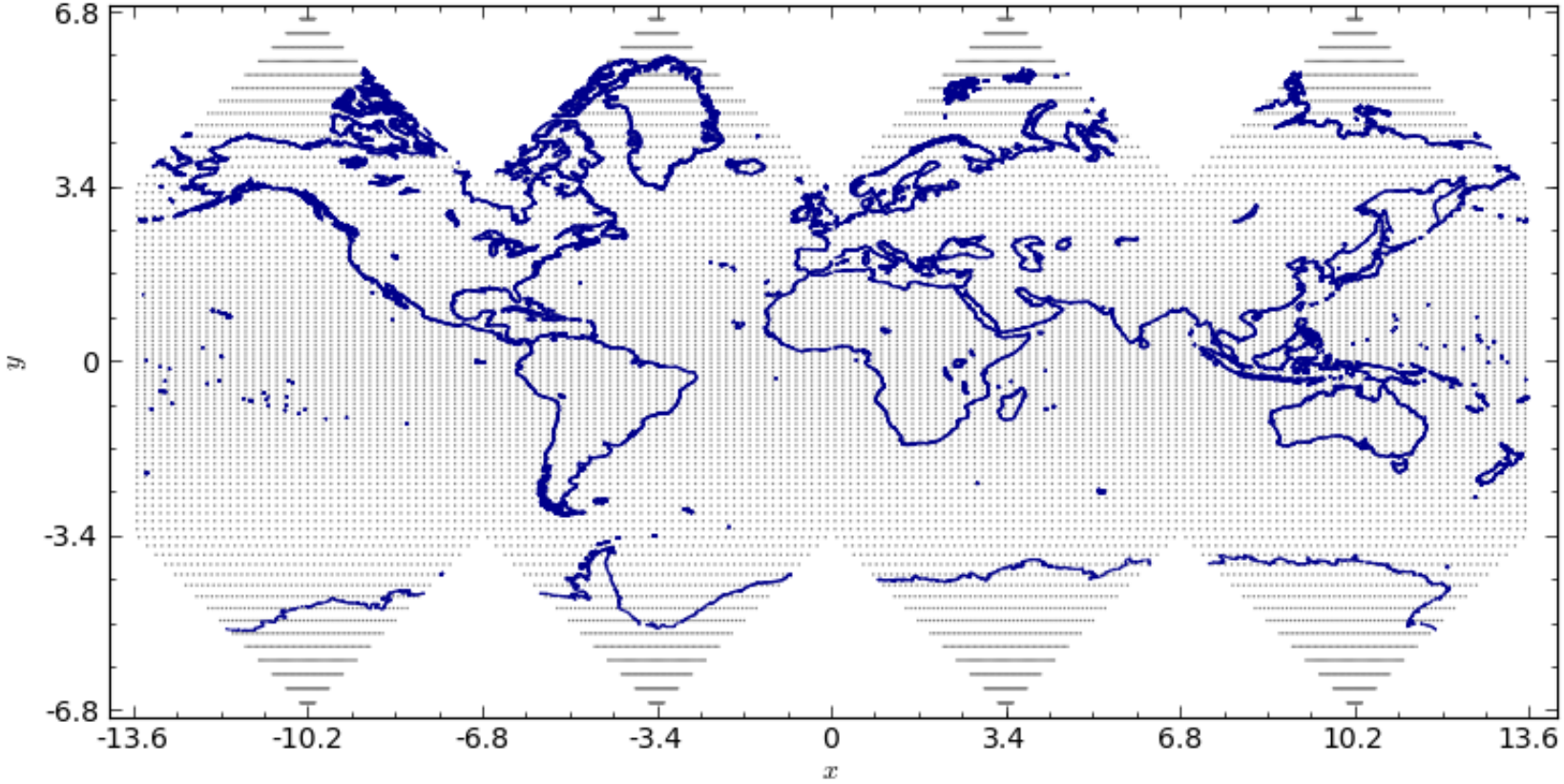
rhealpix forward:



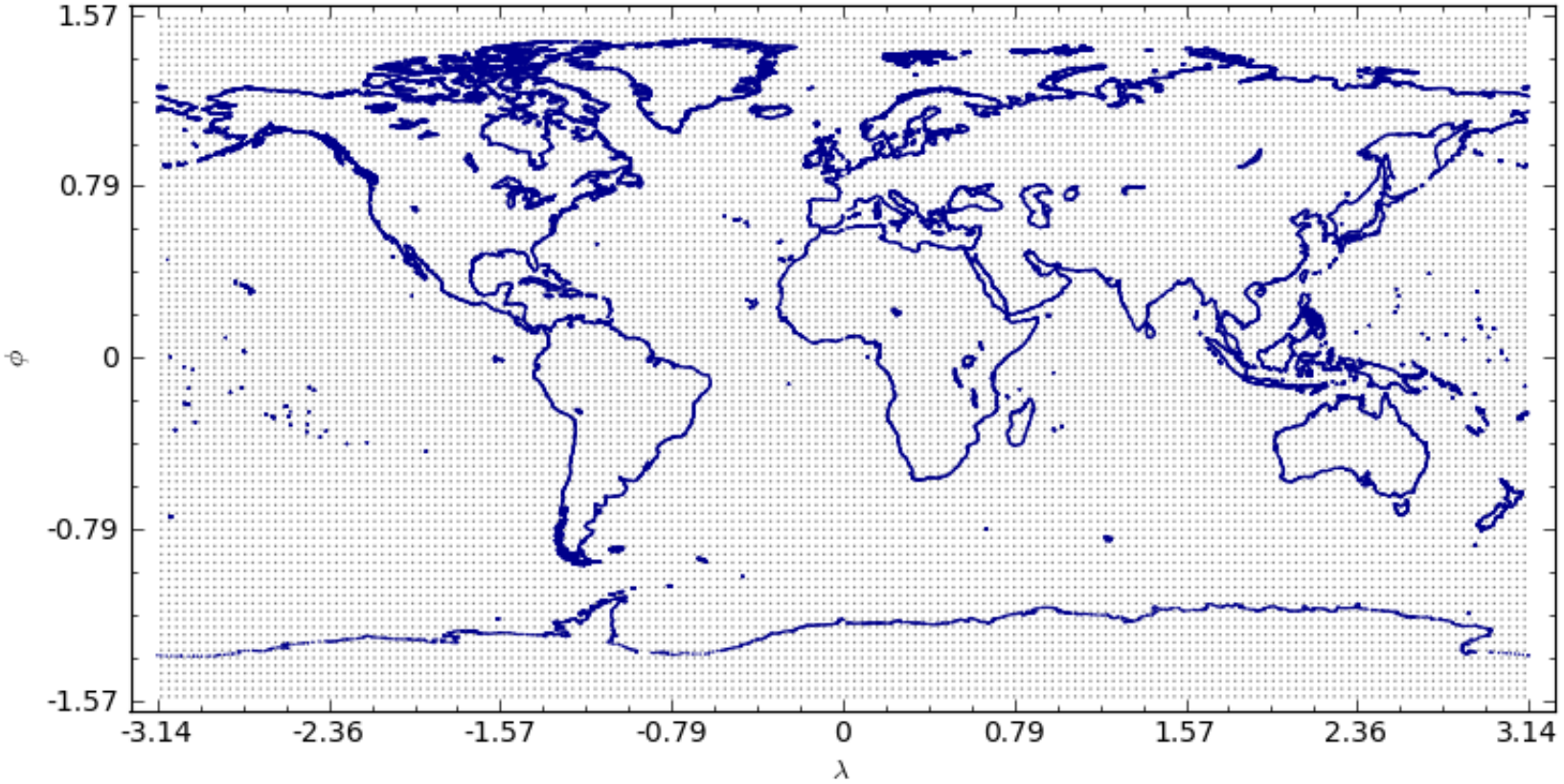
rhealpix inverse:



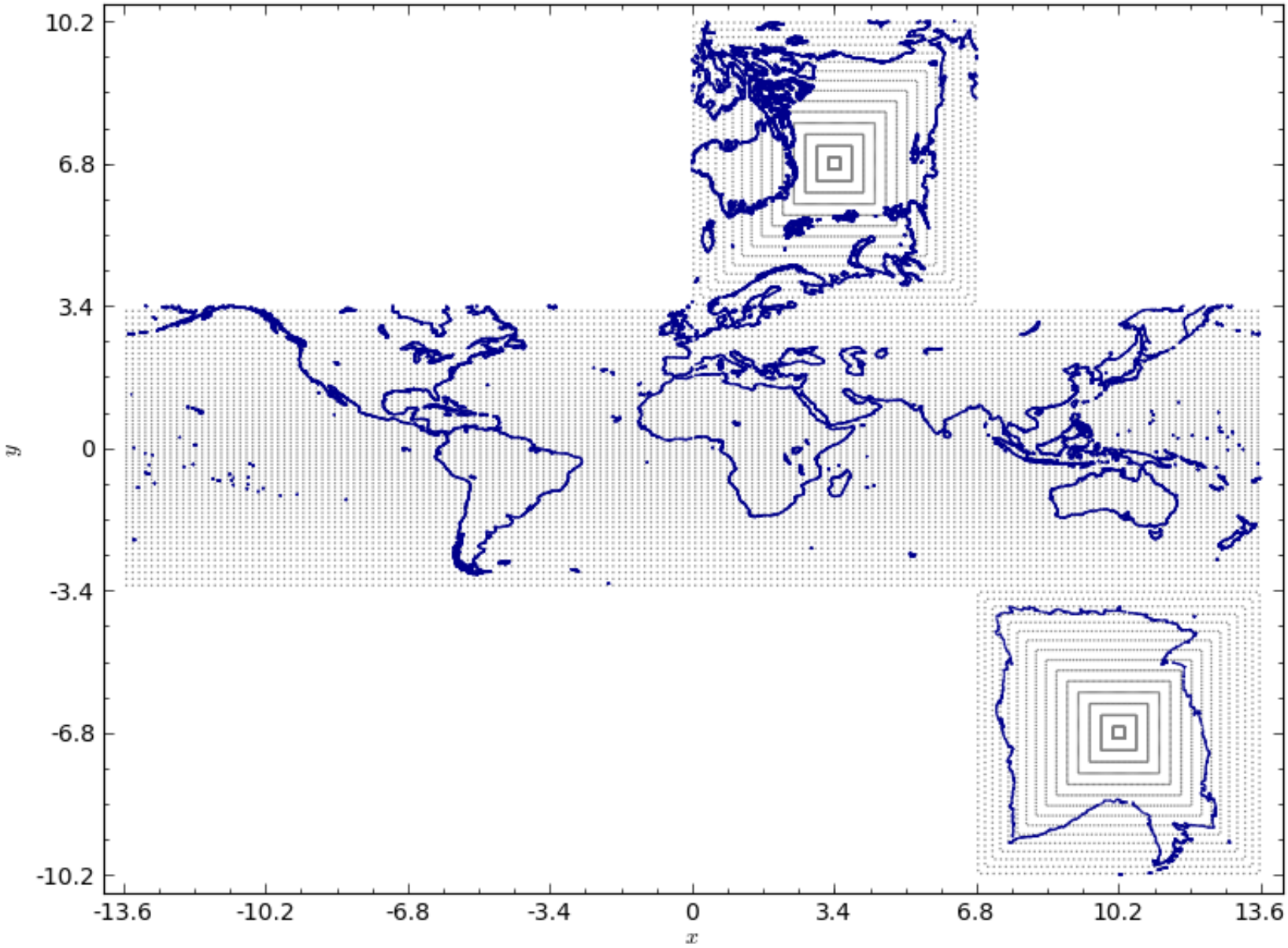

```
ellipsoid:
    R_A = 4.32200117119
    a = 5
    b = 3.0
    e = 0.800000000000000
    f = 0.4
    lat_0 = 0
    lon_0 = 0
    radians = True
    sphere = False
healpix forward:
```



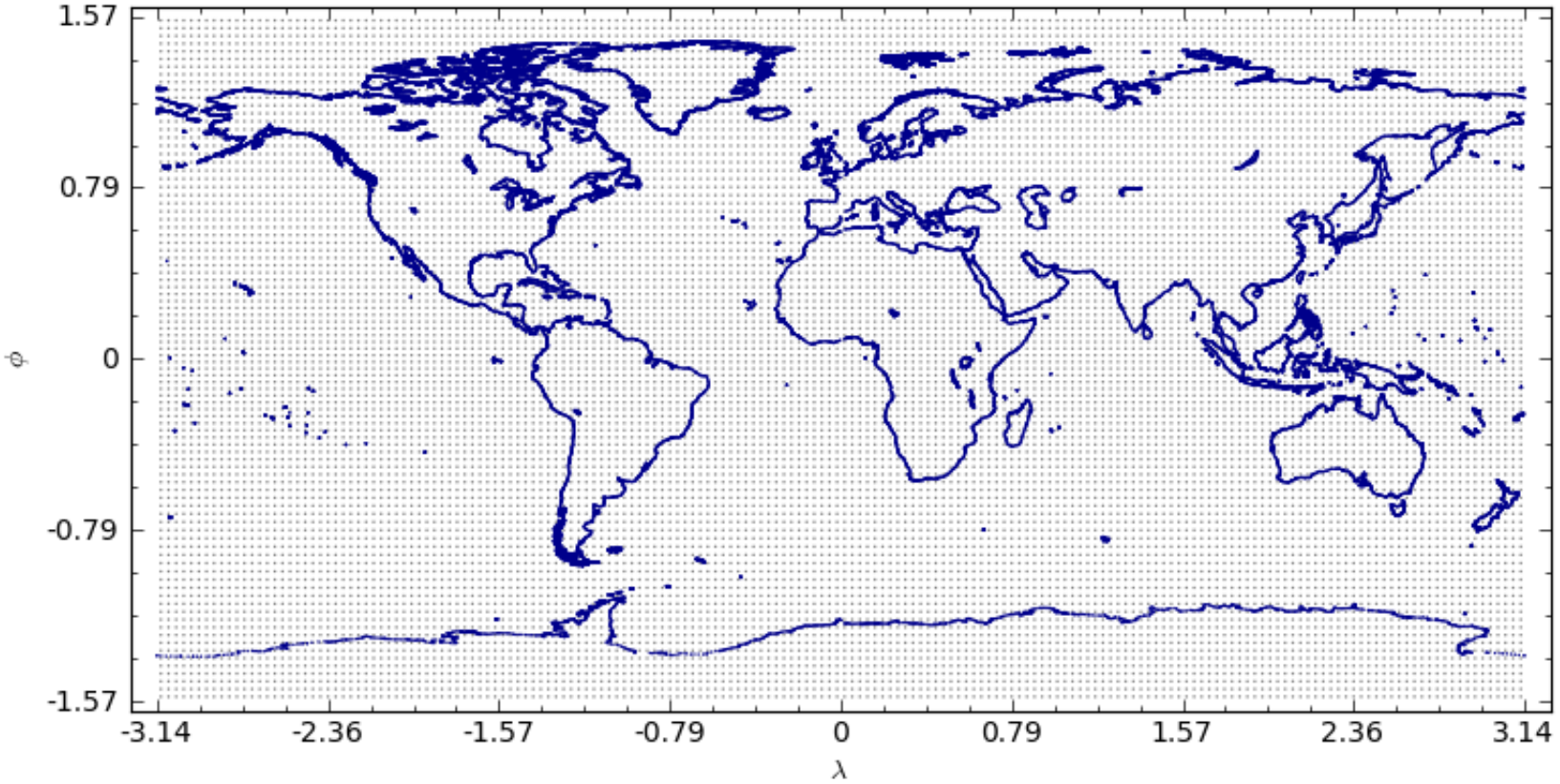
healpix inverse:



rhealpix forward:



rhealpix inverse:


1.57
0.79
0
-0.79
-1.57
ϕ
-3.14
-2.36
-1.57
-0.79
0
0.79
1.57
2.36
3.14
λ