GEOL 7720 - Exercise 2

September 21st 2017

Ian Deniset

Problem

Solve for the density structure of the earth using two data observations - that is, solve an underdetermined inverse problem (number of data points < number of model parameters)

Approach

To solve for the continuous density structure of the earth using only two data points (observations), the problem will be re-formulated using a priori information as constraint via Lagrange Multipliers.

For this problem, the methodology will be to minimize the L_2 norm subject to the constraints of the data through the use of the Lagrangian Function. To obtain the most realistic results, a base model will be used as a constraint and the perturbation from that base model will be solved for while minimizing the deviation from said model.

Theory and Solution

The Lagrangian function (objective function) will be:

$$||\rho(r) - \rho_0|| + 2\sum_{j=1}^{M} \alpha_j (d_j - < k_j, \rho_0 + \delta\rho >)$$

...where $\rho(r)$ is the density distribution of the earth, ρ_0 is the initial base model provided as the constraint, α_j are the Lagrange constants, and $\delta\rho$ is the deviation away from the base model.

As can be seen, the L_2 norm to be minimized ($||\rho(r) - \rho_0||$) represents the deviation of the density structure solution from the base model. The constraint used for the minimization will be the data ($d_j - < k_j, \rho_0 + \delta\rho >$) which is now parameterized in terms of the base model (ρ_0) and the perturbation from that model ($\delta\rho$).

By parameterizing the problem in this way, the original data points (observations) can now be written as:

$$d_j = < k_j, \rho_0 + \delta\rho >$$

...since the density structure of the earth in this format becomes $\rho(r) = \delta\rho(r) + \rho_0(r)$

The inner product can be expanded giving:

$$d_j = < k_j, \rho_0 > + < k_j, \delta\rho > = d_j^0 + \delta d$$

...where d_j^0 is the data given by the base model and δd is the data given by the perturbation from the base model.

Rearranging the above equation gives:

$$\delta d = d_j - d_j^0$$

...showing that subtracting the data given by the base model from the original data will provide the data deviations from the model. These data deviations can then be used to invert for the minimized perturbation from the base model, whos values can be added back to the base model to obtain the density distribution of the earth.

The first step is to find the data given by the base model d_j^0 assuming a very simple starting model that is equal to the mean density of the earth.

Finding the data given by a simple base model:

From above, the data expected from the base model is given by:

$$d_j^0 =< k_j, \rho_0 >$$

...giving...

$d_1^0 =< k_1, \rho_0 >$; where $k_1 = \frac{3}{R^3} r^2$

$d_2^0 =< k_2, \rho_0 >$; where $k_2 = \frac{8\pi}{3MR^2} r^4$

...solving the integrals...

$$d_1^0 = \frac{3\rho_0}{R^3} \int_0^R r^2 dr = \rho_0$$

$$d_2^0 = \frac{8\pi\rho_0}{3MR^2} \int_0^R r^4 dr = \frac{8\pi R^3}{15M} \rho_0$$

With the expected data from the base model achieved, the next step is to subtract it from the original data.

Subtracting base model data from original data:

Recall that the original data was the mean density of the earth, $\bar{\rho}$, and the ratio of the moment of inertia to the mass of the earth multiplied by the radius squared, $\frac{I}{MR^2}$.

Thus:

$$\delta d_1 = d_1 - d_1^0 = \bar{\rho} - \bar{\rho} = 0$$

$$\delta d_2 = d_2 - d_2^0 = \frac{I}{MR^2} - \frac{8\pi R^3}{15M} \rho_0$$

Invert the data perturbations:

From above, δd_i is known to be:

$$\delta d_i =< k_i, \delta\rho >=< k_i, \sum_{j=1}^{M} \alpha_j k_i >= \sum_{j=1}^{M} \alpha_j < k_i, k_j >= \sum_{j=1}^{M} \alpha_j \Gamma_{ij}$$

...where α_j are the Lagrange constants and Γ_{ij} represents a sensitivity matrix of inner products of the above kernel functions.

The goal then, is to solve for the Lagrange constants and subsequently solve for the density perturbations from the base model as a function of radius.

Solving for α_j then gives:

$$\delta d = \Gamma\alpha$$

$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \end{bmatrix} = \begin{bmatrix} \Gamma_{11}\Gamma_{12} \\ \Gamma_{21}\Gamma_{22} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \end{bmatrix} = \begin{bmatrix} < k_1, k_1 >< k_1, k_2 > \\ < k_2, k_1 >< k_2, k_2 > \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \end{bmatrix} = \begin{bmatrix} \frac{9}{R^6} \int_0^R r^4 dr \frac{8\pi}{MR^5} \int_0^R r^6 dr \\ \frac{8\pi}{MR^5} \int_0^R r^6 dr \frac{64\pi^2}{9M^2R^4} \int_0^R r^8 dr \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \frac{9}{5} \frac{6}{7\bar{\rho}} \\ \frac{6}{7\bar{\rho}} \frac{4}{9\bar{\rho}^2} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

Assuming a radius of $R = 6,371,000$ and a mean density of $\bar{\rho} = 5,517 \frac{kg}{m^3}$...

$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \end{bmatrix} = \begin{bmatrix} 2.8253x10^{-7} 2.4386x10^{-11} \\ 2.4386x10^{-11} 2.2919x10^{-15} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

...invert Γ and solve for α...

$$\alpha = \Gamma^{-1}\delta d$$

Solve for the Lagrange Constants

In [1]:
```
#import the usual modules
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
#a few constants first
R = 6371000.0           #radius
M = 5.972e24            #mass
meanDensity = 5517.0    #mean density
deltaData_1 = 0.0
deltaData_2 = 0.33078 - ((8*np.pi*(R**3)*5517.)/(15*M))
```

In [3]:
```
#create matrices
deltaData = np.array([[deltaData_1],[deltaData_2]])
gamma = np.array([[2.8253e-7, 2.4386e-11],[2.4386e-11,2.2919e-15]])
print("DATA: \n", deltaData)
print("GAMMA: \n", gamma)
```

```
DATA:
 [[ 0.        ]
 [-0.06949144]]
GAMMA:
 [[  2.82530000e-07   2.43860000e-11]
 [  2.43860000e-11   2.29190000e-15]]
```

In [4]:
```
#solve for constants
alphaConst = np.dot(np.linalg.inv(gamma), deltaData)
print("Lagrange Constants: \n", alphaConst)
```

```
Lagrange Constants:
 [[  3.20625472e+10]
 [ -3.71468526e+14]]
```

Calculate and plot density perturbations from base model

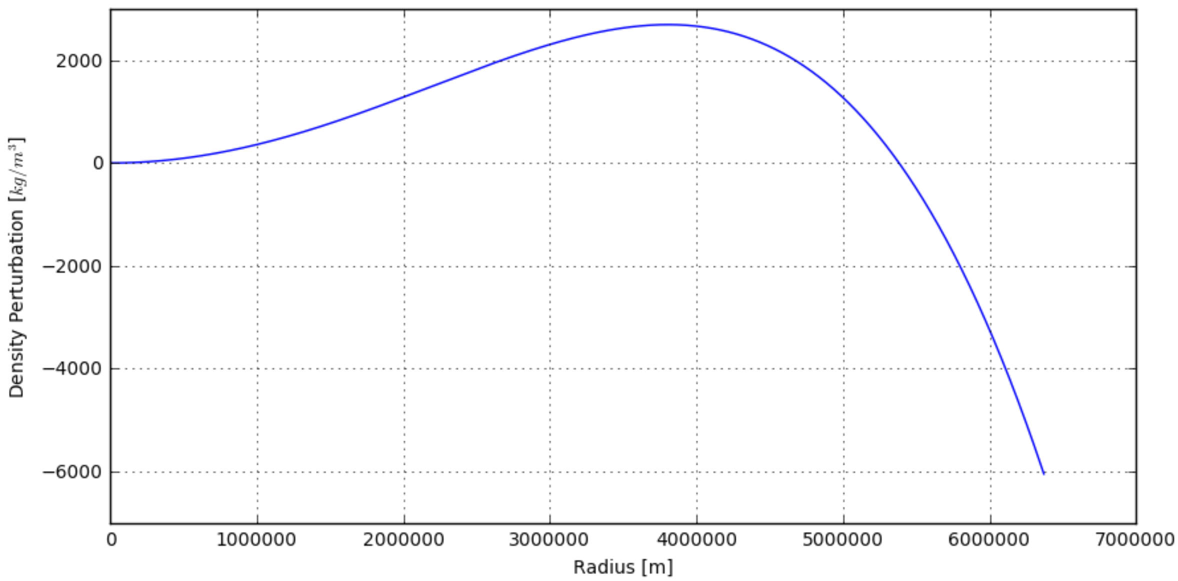
From before it is known that $\delta\rho(r) = \sum_{j=1}^{M} \alpha_j k_j$. Thus...

$\delta\rho(r) = \alpha_1 k_1(r) + \alpha_2 k_2(r)$

In [5]:
```
#a few more constants
k1 = 3./(R**3)
k2 = ((8/3)*(np.pi))/(M*(R**2))
```

In [6]:
```
#calculate deviations from base model density structure as a function of rad
radius = np.arange(0.,6371000.,1.)       #array of values for radius in step
denStructP = alphaConst[0]*k1*(radius**2) + alphaConst[1]*k2*(radius**4)
```

In [7]:
```
#plot
fig, ax = plt.subplots(figsize=(10,5))
ax.plot(radius, denStructP)
ax.set_xlabel('Radius [m]')
ax.set_ylabel('Density Perturbation [$kg/m^3$]')
ax.grid()
plt.show()
```

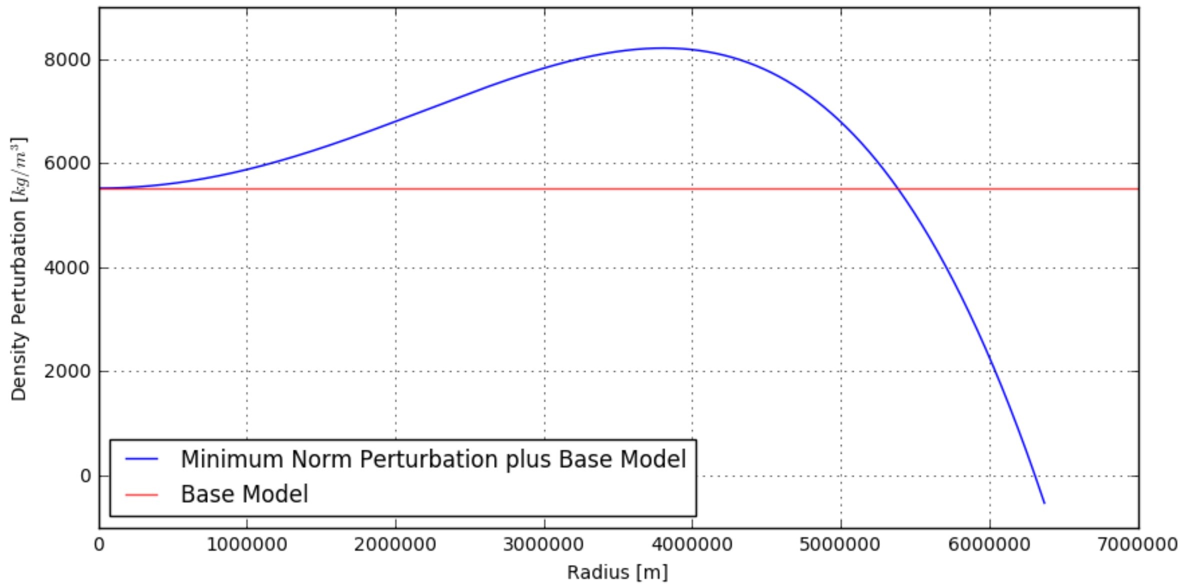


Add back base model and re-plot:

Now that the density perturbations with radius have been solved for, the last step is to add it back to the base model (constant mean density) and re-plot.

In [8]:

```
#plot
fig, ax = plt.subplots(figsize=(10,5))
ax.plot(radius, denStructP + 5517.0, label='Minimum Norm Perturbation plus I
ax.set_xlabel('Radius [m]')
ax.set_ylabel('Density Perturbation [$kg/m^3$]')
ax.grid()
ax.axhline(y=5517, c='r', alpha=0.75, label='Base Model')
ax.legend(loc=3)
plt.show()
```



Summary

Compared to solving the same problem without using a base model, this result is much more realistic - but still not perfect. Without providing the mean density of the earth as a data constraint, the original solution provided a value of zero density at the earth's center and highly unrealistic negative density values at the surface. As can be seen here, at a radius of zero, the density is a more believable value and only a small amount of the solution contains impossible negative values.

However, this solution still shows a decrease in density from the mantle to the core and has a tendency to bias most of the mass towards the surface. This is inevitable for this inversion due to how the kernel functions influence the effect of radius (more sensitive towards the surface).