CARLOS FABIÁN SÁNCHEZ SUÁREZ
201530455

Desarrollo taller 1 Laboratorio de Geofísica

1. Esfera, modelo de diapiros

a) Para la demostración de esta ecuación partimos de igualar la ley de gravitación universal con la segunda ley de Newton

$$F = M_1 a = \frac{GM_1M_2}{r^2}$$

$$F = a = \frac{M_2}{r^2}$$

Ahora, para conocer la masa, podemos expresar esta como el múltiplo del volumen por la diferencia de densidades:

$$M = V\Delta\rho = \frac{4}{3}\pi r^3(\rho_e - \rho_s)$$

Ahora, teniendo en cuenta el siguiente modelo geométrico para el caso:

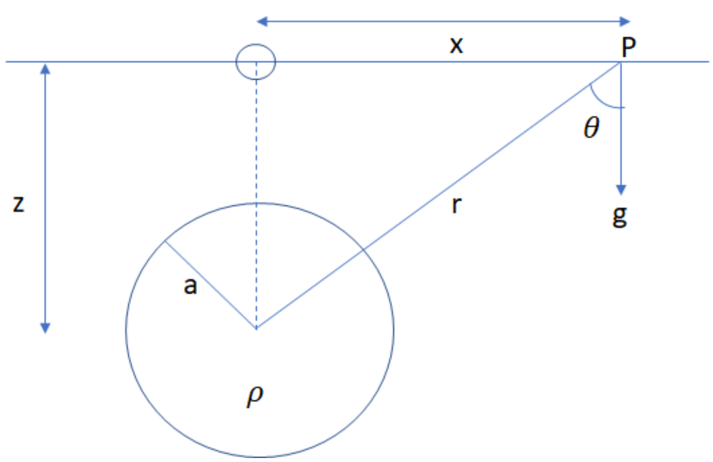

Figura 1. Modelo de Esfera, basado en el modelo de Telford W.M.

Podemos identificar lo siguiente:

$$r = \sqrt{z^2 + x^2}$$

$$g(x) = GMcos\theta$$

$$cos\theta = \frac{z}{x}$$

Para así obtener la fórmula del perfil de anomalía gravitacional:

$$g(x) = \frac{G\frac{4}{3}\pi r^3(\rho_e - \rho_s)z}{(x^2 + z^2)^{3/2}}$$

b)

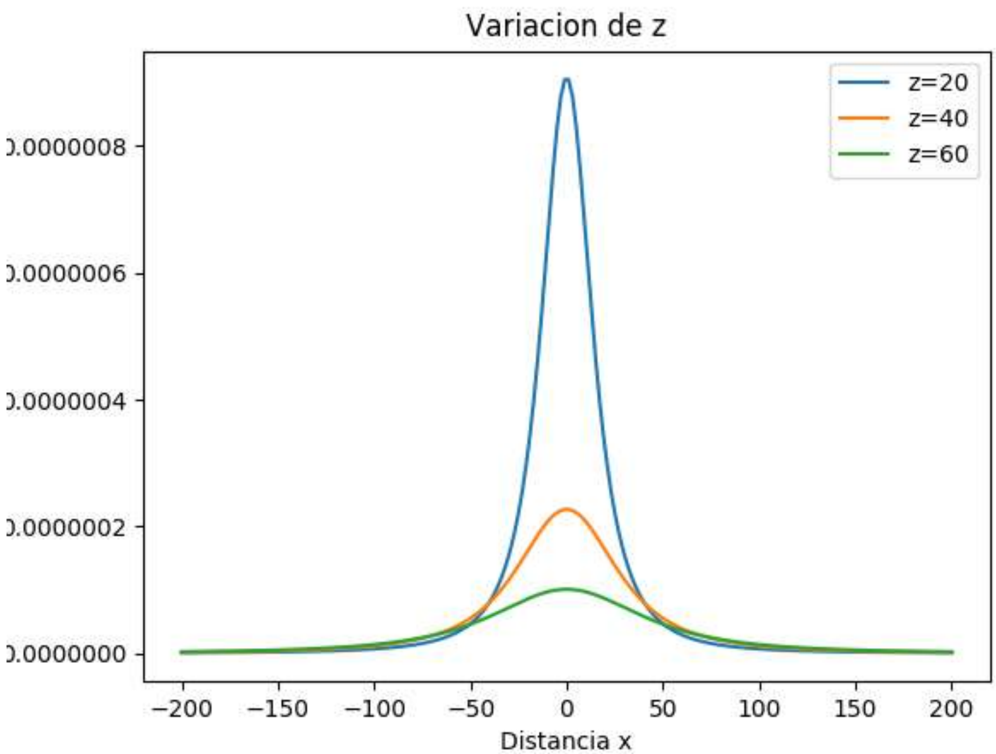

Figura 2. Gráfica de anomalías variando la profundidad z

En la gráfica anterior es posible evidenciar la manera en que afecta la profundidad en el tamaño de la anomalía, esto es, que a la menor profundidad encontramos que la curva es mucho más pronunciada que las otras más profundas, lo cual se justifica con la lejanía presente entre la fuente y el objeto de estudio.

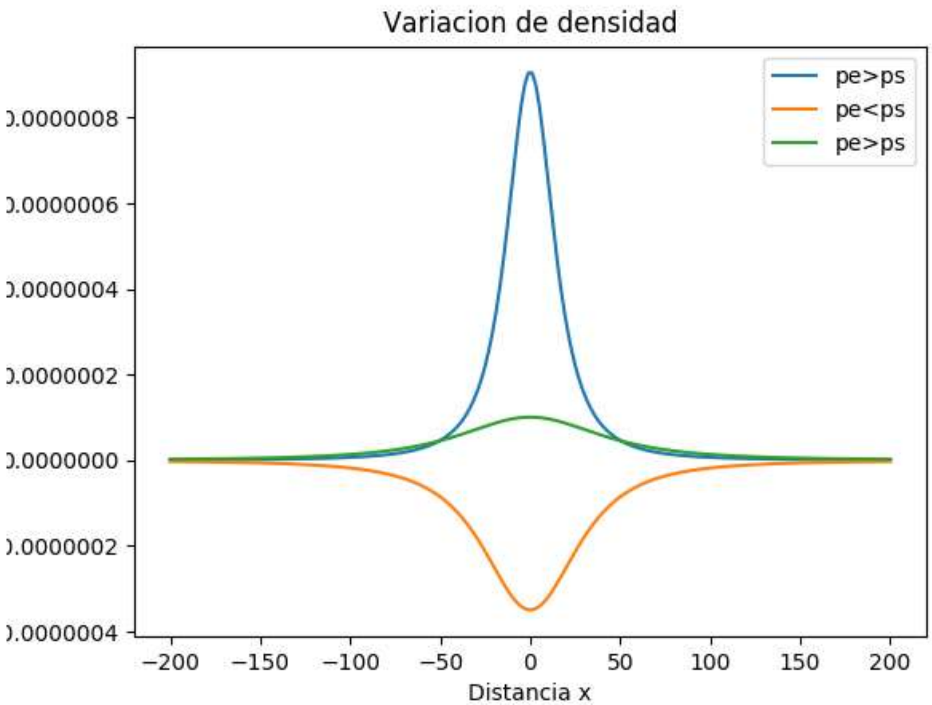

Figura 3. Gráfica de anomalías variando la densidad

En esta gráfica podemos observar el comportamiento de las anomalías, si corresponden a respuestas positivas o negativas, y se puede verificar que cuando la densidad de la esfera (ρe) es menor que la del subsuelo (ρs) se obtiene una anomalía negativa.

Esta anomalía negativa, adecuándola al problema, está relacionada con la ubicación de un diapiro a una determinada profundidad, en la medida que estos cuerpos salinos tienen una menor densidad que las rocas del subsuelo (normalmente).

c) En este punto se necesita calcular el coeficiente entre la profundidad z y $x_{1/2}$, por lo tanto, lo primero es calcular este valor. Las configuraciones distintas corresponden a las cuatro siguientes:
   1. Z=20 pe>ps
   2. Z=40 pe>ps
   3. Z=60 pe>ps
   4. Z=40 ps<ps

Para realizar las estimaciones del punto medio se procedió por medio de Python para calcular el valor mínimo o máximo de cada anomalía, y para este se obtuvieron los siguientes valores de $x_{1/2}$:

1. 15.54
2. 30.58
3. 45.61
4. 30.58

Ahora, con estos valores de x se obtiene el coeficiente:

$$C1 = \frac{20}{15.54} = 1.29$$

$$C2 = \frac{40}{30.58} = 1.30$$

$$C3 = \frac{60}{45.61} = 1.31$$

$$C4 = \frac{40}{30.58} = 1.30$$

A partir de los resultados anteriores se puede evidenciar que la tendencia de este coeficiente arroja valores bastante cercanos entre sí, que en este caso son valores alrededor de 1.3, con lo cual se puede concluir que este coeficiente presenta un valor constante de aproximadamente eso.

d) En este ejercicio se busca calcular el valor de $X_{1/2}$ utilizando el coeficiente anterior, para con él hallar el valor de la profundidad z.

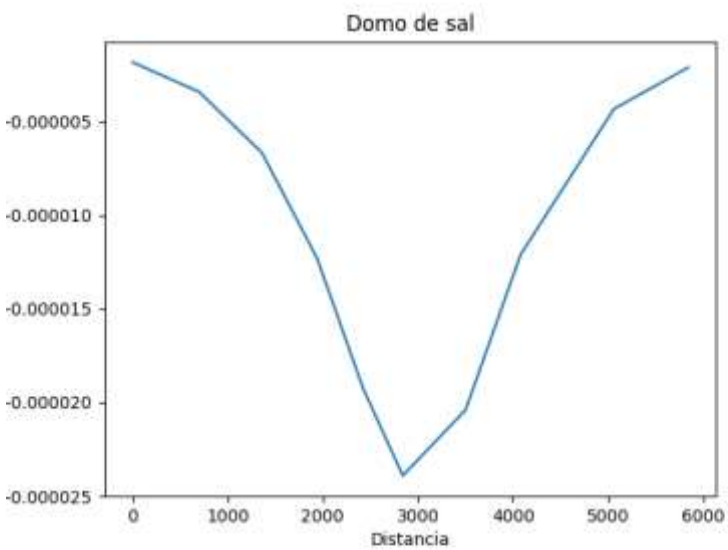

Figura 4. Anomalía negativa del domo de sal del ejercicio

El valor de $X_{1/2}$ obtenido fue de 1240 m, y para z se calculó:

$$z = 1240 * 1.3 = 1612\ m$$

Y para calcular el tamaño del diapiro despejamos r de la ecuación inicial:

$$-2.39 * 10^{-5} = \frac{4}{3}\pi * 6.674 * 10^{-11} * r^3 * (2200 - 2700) * \frac{1612}{(2840^2 + 1612^2)^{\frac{3}{2}}}$$

$$r = 3315.83\ m$$

Por lo tanto, el tamaño del diapiro es de 3315.83m – 2840m = 475.83 m

2. Losa horizontal semi-infinita: modelo para una falla vertical

a) Al realizar las variaciones pedidas se obtuvieron las siguientes gráficas:

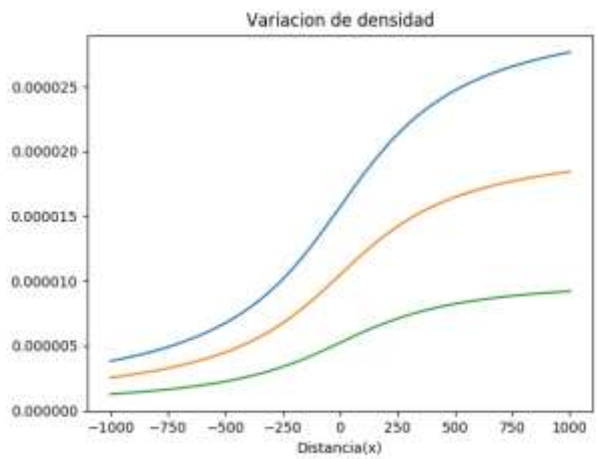

Figura 5. Variación de densidad

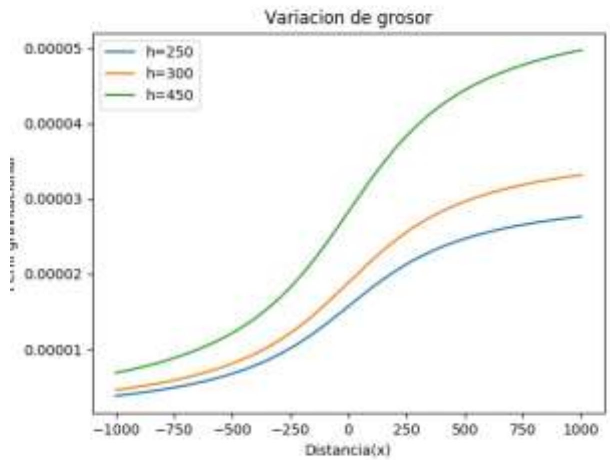

Figura 6. Variación de grosor

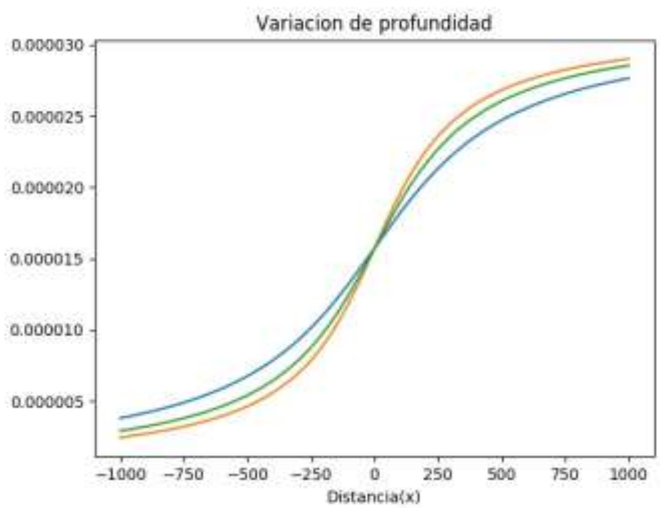

Figura 7. Variación de profundidad

A partir de estas gráficas es posible ver la variación gravitacional presente frente a la presencia de la losa, como se puede notar, esta varía hacia la parte derecha en aumento, lo cual sugiere que la ubicación de esta se halla en ese sector y que agrega una cantidad considerable de masa capaz de generar la respuesta que se puede apreciar allí con esa curva predominante y creciente hacia la parte derecha.

b) En esta parte fue necesario hacer la derivación hasta segundo grado de cada uno de los perfiles obtenidos, los resultados se muestran a continuación:

- Primeras derivadas

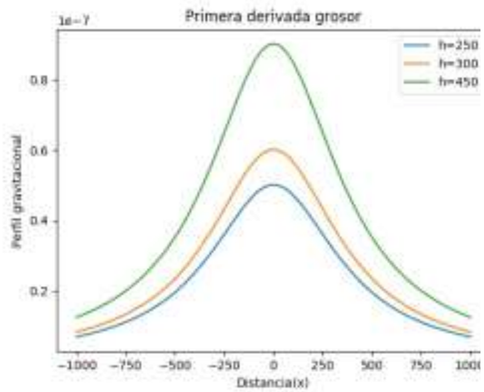

Figura 8. Primera derivada del cambio de grosor

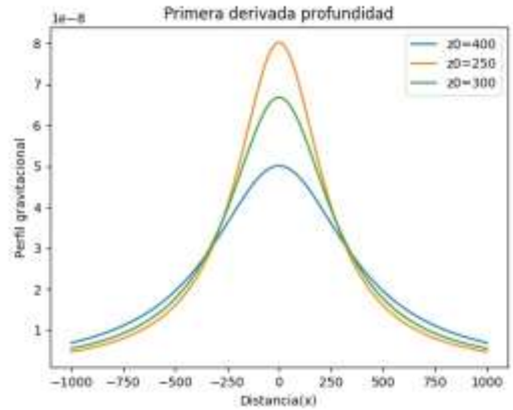

Figura 9. Primera derivada del cambio de profundidad

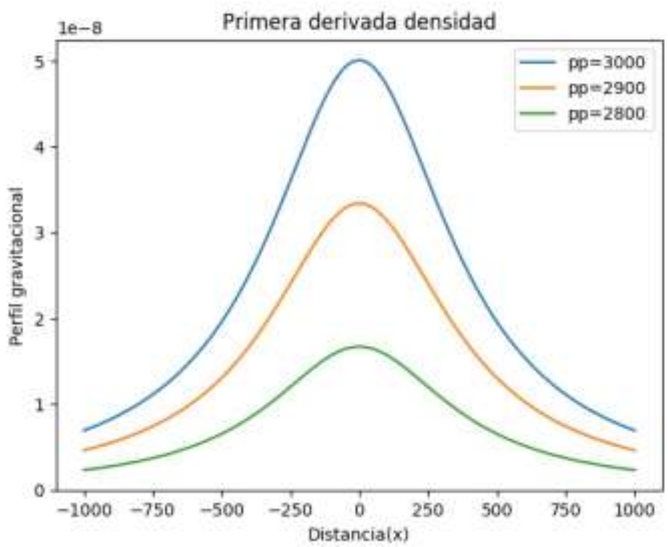

Figura 10. Primera derivada del cambio de densidad

La primera derivada representa la tasa de cambio del perfil gravitacional con respecto a una distancia x, en la cual puede verse que los máximos se hallan hacia el punto medio del arreglo de distancia. La forma obtenida en cada uno de los perfiles resulta bastante parecida entre estos, con la intersección de algunas curvas en los perfiles de grosor y profundidad, esta forma sugiere la capacidad del perfil gravitacional que se obtuvo y la presencia de algún cuerpo que altera dicho perfil (por las elevaciones).

- Segundas derivadas

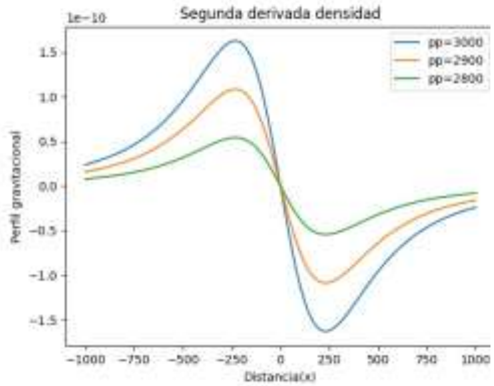

Figura 11. Segunda derivada del cambio de densidad

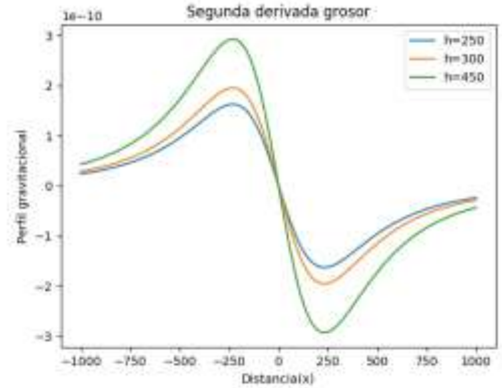

Figura 12. Segunda derivada del cambio de grosor

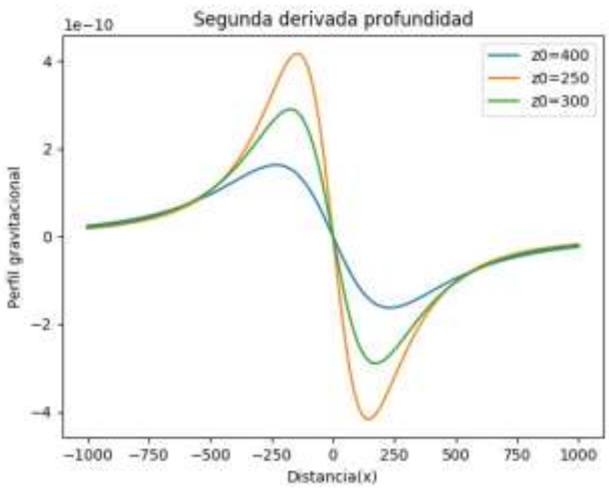

Figura 13. Segunda derivada del cambio de profundidad

La segunda derivada me muestra el comportamiento que tiene una función en un determinado periodo de análisis, en este campo es posible reconocer que el perfil gravitacional que se genera presenta un cambio asociado al cuerpo en estudio. Este cambio se puede apreciar por la intersección de las curvas en la mitad del arreglo de distancia x y porque presenta puntos de crecimiento y decrecimiento, lo cual me indica algún cambio.

ANEXO (CÓDIGOS DE PYTHON)

```
import numpy as np
import matplotlib.pyplot as plt
import math


G=6.674*10**-11
ps=2700.0

h=250.0
h2=300.0
h3=450.0

pp=3000.0
pp2=2900.0
pp3=2800.0

z0=400.0
z01=250.0
z02=300.0

x=np.linspace(-1000,1000,1000)


#Grosor

def gravitacionalg1():
```

```
    g=2.0*G*h*(pp-ps)*((math.pi/2.0)+np.arctan(x/z0))
    return g

def gravitacionalg2():
    g=2.0*G*h2*(pp-ps)*((math.pi/2.0)+np.arctan(x/z0))
    return g

def gravitacionalg3():
    g=2.0*G*h3*(pp-ps)*((math.pi/2.0)+np.arctan(x/z0))
    return g

"""
plt.figure()
plt.plot(x, gravitacionalg1(), label="h=250")
plt.plot(x, gravitacionalg2(), label="h=300")
plt.plot(x, gravitacionalg3(), label="h=450")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.title("Variacion de grosor")
plt.legend()
plt.savefig("grosor.jpg")
plt.show()
"""



#Densidad

def gravitacionald1():
    g=2.0*G*h*(pp-ps)*(math.pi/2.0+np.arctan(x/z0))
```

```
    return g

def gravitacionald2():
    g=2.0*G*h*(pp2-ps)*(math.pi/2.0+np.arctan(x/z0))
    return g

def gravitacionald3():
    g=2.0*G*h*(pp3-ps)*(math.pi/2.0+np.arctan(x/z0))
    return g

"""
plt.figure()
plt.plot(x, gravitacionald1(), label="pp=3000")
plt.plot(x, gravitacionald2(), label="pp=2900")
plt.plot(x, gravitacionald3(), label="pp=2800")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.title("Variacion de densidad")
plt.savefig("densidad.jpg")
plt.legend()
plt.show()
"""

#Profundidad

def gravitacionalp1():
    g=2.0*G*h*(pp-ps)*(math.pi/2.0+np.arctan(x/z0))
    return g
```

```
def gravitacionalp2():
	g=2.0*G*h*(pp-ps)*(math.pi/2.0+np.arctan(x/z01))
	return g

def gravitacionalp3():
	g=2.0*G*h*(pp-ps)*(math.pi/2.0+np.arctan(x/z02))
	return g

"""
plt.figure()
plt.plot(x, gravitacionalp1(), label="z0=400")
plt.plot(x, gravitacionalp2(), label="z0=250")
plt.plot(x, gravitacionalp3(), label="z0=300")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.title("Variacion de profundidad")
plt.savefig("profundidad.jpg")
plt.legend()
plt.show()
"""

#Derivadas

x2=np.linspace(-1000,1000,999)
x3=np.linspace(-1000,1000,998)

#Grosor

d1g1=np.diff(gravitacionalg1())
```

```
d1g2=np.diff(gravitacionalg2())
d1g3=np.diff(gravitacionalg3())
d2g1=np.diff(d1g1)
d2g2=np.diff(d1g2)
d2g3=np.diff(d1g3)


#Densidad


d1d1=np.diff(gravitacionald1())
d1d2=np.diff(gravitacionald2())
d1d3=np.diff(gravitacionald3())
d2d1=np.diff(d1d1)
d2d2=np.diff(d1d2)
d2d3=np.diff(d1d3)


#Profundidad


d1p1=np.diff(gravitacionalp1())
d1p2=np.diff(gravitacionalp2())
d1p3=np.diff(gravitacionalp3())
d2p1=np.diff(d1p1)
d2p2=np.diff(d1p2)
d2p3=np.diff(d1p3)


#Graficas
```

```
#Primeras derivadas
plt.figure()
plt.plot(x2, d1g1, label="h=250")
plt.plot(x2, d1g2, label="h=300")
plt.plot(x2, d1g3, label="h=450")
plt.title("Primera derivada grosor")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.legend()
plt.savefig("primeraderivadagrosor.jpg")
plt.show()


plt.figure()
plt.plot(x2, d1d1, label="pp=3000")
plt.plot(x2, d1d2, label="pp=2900")
plt.plot(x2, d1d3, label="pp=2800")
plt.title("Primera derivada densidad")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.legend()
plt.savefig("primeraderivadadensidad.jpg")
plt.show()


plt.figure()
plt.plot(x2, d1p1, label="z0=400")
plt.plot(x2, d1p2, label="z0=250")
plt.plot(x2, d1p3, label="z0=300")
```

```
plt.title("Primera derivada profundidad")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.legend()
plt.savefig("primeraderivadaprofundidad.jpg")
plt.show()


#Segundas derivadas

plt.figure()
plt.plot(x3, d2g1, label="h=250")
plt.plot(x3, d2g2, label="h=300")
plt.plot(x3, d2g3, label="h=450")
plt.title("Segunda derivada grosor")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.legend()
plt.savefig("segundaderivadagrosor.jpg")
plt.show()


plt.figure()
plt.plot(x3, d2d1, label="pp=3000")
plt.plot(x3, d2d2, label="pp=2900")
plt.plot(x3, d2d3, label="pp=2800")
plt.title("Segunda derivada densidad")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
```

```
plt.legend()
plt.savefig("segundaderivadadensidad.jpg")
plt.show()


plt.figure()
plt.plot(x3, d2p1, label="z0=400")
plt.plot(x3, d2p2, label="z0=250")
plt.plot(x3, d2p3, label="z0=300")
plt.title("Segunda derivada profundidad")
plt.xlabel("Distancia(x)")
plt.ylabel("Perfil gravitacional")
plt.legend()
plt.savefig("segundaderivadaprofundidad.jpg")
plt.show()
```