Tarea 2 GF5013 - Métodos Inversos Aplicados a la Geofísica

Tomografía de Tiempo de Viaje
“Travel Time Tomography”

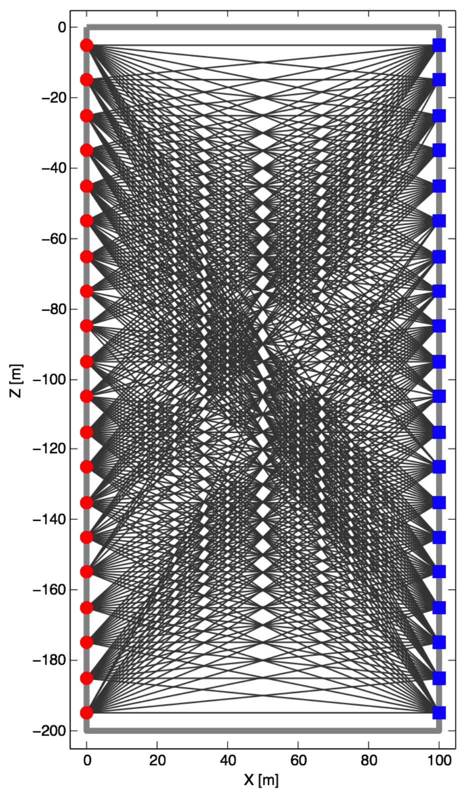
NO HACER TAREA A ULTIMA HORA
NO SE POSTERGARÁ LA FECHA DE ENTREGA

Fecha Publicación: Martes 1 de Octubre de 2019 – Fecha Entrega: Martes 22 de Octubre de 2019

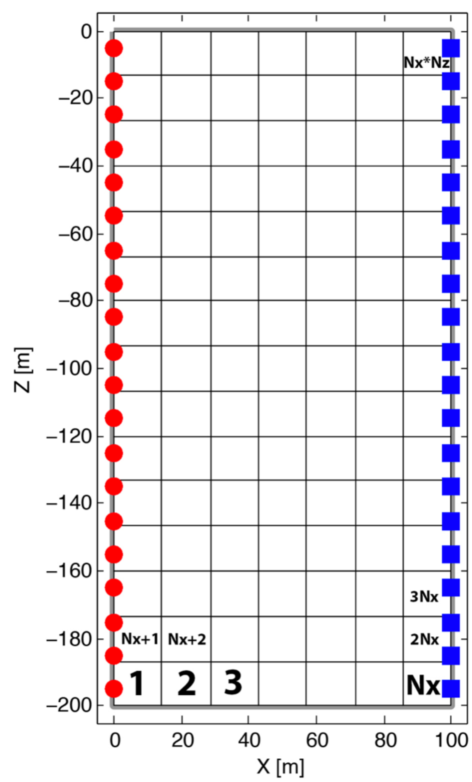
Tomografía de Tiempo de Viaje

En la Figura 1 se representan dos pozos de 200 metros de profundidad espaciados a 100 metros. En el pozo de la izquierda se sitúan fuentes explosivas (puntos rojos) y en el pozo de la derecha se ubican geófonos (cuadrados azules) que son capaces de registrar ondas sísmicas compresionales.

Se realiza el siguiente experimento: una a una se detonan las fuentes explosivas y para cada detonación se registra el tiempo de viaje de las ondas compresionales en todos los geófonos (en milisegundos). Se asume por simplicidad la aproximación de rayos rectos (lo cual implica asumir un medio relativamente homogéneo).



(a) Rayos trazados en experimento de tomografía



(b) Ejemplo de indexación de la malla

Figura 1: Configuración del problema de tomografía lineal (Rayos rectos)

En este tipo de tomografía, la observación en cada receptor es el tiempo de viaje total de la onda, el que depende de la velocidad de propagación de la onda (onda P en este caso) a lo largo del camino recorrido por el rayo. Este campo de velocidad de onda es la incógnita en el problema de tomografía. Para resolver este problema, vamos a discretizar el medio entre los pozos tal como se muestra en la Figura 1b, en donde además se muestra el esquema a utilizar para la indexación de los elementos discretos que van a conformar el medio. Nx es el número de casilleros en la dirección X y Nz es el número de casilleros en la dirección Z.

El tiempo de viaje T_{viaje} a lo largo de un rayo, en el problema discretizado, queda definido por

$$T_{viaje} = \sum_{k=1}^{N_R} \frac{L_k}{V_k} \tag{1}$$

donde N_R es el número de casilleros por los que pasa rayo, L_k y V_k son la longitud del rayo y la velocidad de propagación de ondas compresionales a lo largo del casillero k.

Las incógnitas del problema son las velocidades V_k de todos los casilleros. Si planteamos el problema inverso en función de dichas velocidades el problema es no lineal, ya que la dependencia de V_k en la ecuación (1) es no lineal. Sin embargo, si parameterizamos el problema en función de la lentitud $S_k = \frac{1}{V_k}$ (el recíproco de la velocidad) el problema planteado se vuelve lineal. De esta manera, podemos escoger el vector de incógnitas $\mathbf{m}$ de la siguiente manera

$$m_j = S_j = \frac{1}{V_j} \qquad \text{donde } j \in \{1, \dots, N_x \cdot N_z\}, \tag{2}$$

y donde el índice j en el vector de parámetros $\mathbf{m}$ sigue el mismo orden que la indexación mostrada en la Figura 1b.

Luego la ecuación del tiempo de viaje de una onda compresional a lo largo del i-ésimo rayo queda,

$$T_{viaje_i}(\mathbf{m}) = \sum_{k=1}^{N_R} L_k m_k \tag{3}$$

Para ayudarlo en la resolución del problema, se provee el paquete de python 3 “Tarea2”, en el cual hay módulos que ya están programados y listos para ser usados, y otros son plantillas que deberá completar. La estructura del paquete de python es la siguiente:

```
|----Tarea2
|-------- Mesh
|      |------ MeshTools.py
|      |------ RectangularMesh.py
|      |------ TikhonovReg.py
|      |------ __init__.py
|-------- TomoData.py
|-------- TomographyModel
|      |------ ModeloTeorico.py
|      |------ RayLinear2D.py
|      |------ SyntheticModels.py
|      |------ __init__.py
|---- __init__.py
```

Dentro del paquete Tarea2 se pueden destacar las siguientes funciones y clases:

- `Tarea2.TomoData.readTravelTimeData()`: función que permite leer el archivo de datos con la ubicación de la fuente, receptor y tiempo de viaje observado para cada rayo. (cada fila del archivo corresponde a un rayo).

- `Tarea2.Mesh`: paquete de python que contiene modulos, funciones y clases relacionadas con la malla discretizada a utilizar para el problema de tomografía. En particular, contiene la clase RectangularMesh que crea un objeto que define la ubicación de los centros y vértices de cada elemento discretizado de la malla, dada una cantidad de elementos en cada dirección X y Z respectivamente. Asimismo entrega la conectividad de los elementos vecinos.
- `Tarea2.TomographyModel.RayLinear2D`: una clase que define un objeto que representa rayos lineales en 2D. En particular, tiene una función miembro que , dado una discretización de la malla (instancia de objeto RectangularMesh), permite calcular la longitud del rayo en cada elemento discretizado (L_k) por los cuales el rayo pasa (usar esta función para calcular las funciones de Green del problema de tomografía).
- `Tarea2.Mesh.MeshTools.py`: en este módulo se definen funciones que permiten graficar un campo de valores V (escalar) en la malla discretizada.

Ver ejemplos en el notebook de Python Ejemplo_crear_y_graficar_mallas_y_cargar_datos.ipynb , y los comentarios de los códigos para mayor detalle de como usar los códigos que se proveen para esta tarea.

P1. Diseño del Problema de Tomografía Lineal: Problema Directo

Por razones de simplicidad, en la resolución de este problema se asumirá que la varianza de los tiempos de viaje de las ondas toma el valor “1” y que todas las observaciones son independientes.

El problema de tomografía lineal se puede escribir de la forma $\mathbf{Gm} = \mathbf{d}$, donde $\mathbf{G}$ es la matriz de diseño del problema, $\mathbf{m}$ el vector de parámeros del modelo descrito como en (2), y $\mathbf{d}$ las observaciones, que en este caso corresponde a los tiempos de viaje de los diferentes rayos medidos en cada receptor (puntos azules de Figura 1)

Se pide completar la función `formar_G` ubicada en el módulo `Tarea2.TomographyModel.ModeloTeorico.py` que permite calcular la matriz de diseño G (funciones de Green) del problema directo o teoría.

P2. Mal Condicionamiento del Problema de Tomografía

En esta parte de la tarea vamos a ilustrar el mal condicionamiento del problema de tomografía. Para ello vamos a utilizar una discretización del espacio lo suficientemente gruesa para que en cada casillero pase por lo menos 1 rayo del experimento.
Use $\mathtt{Nx} = 9$, $\mathtt{Nz} = 15$.

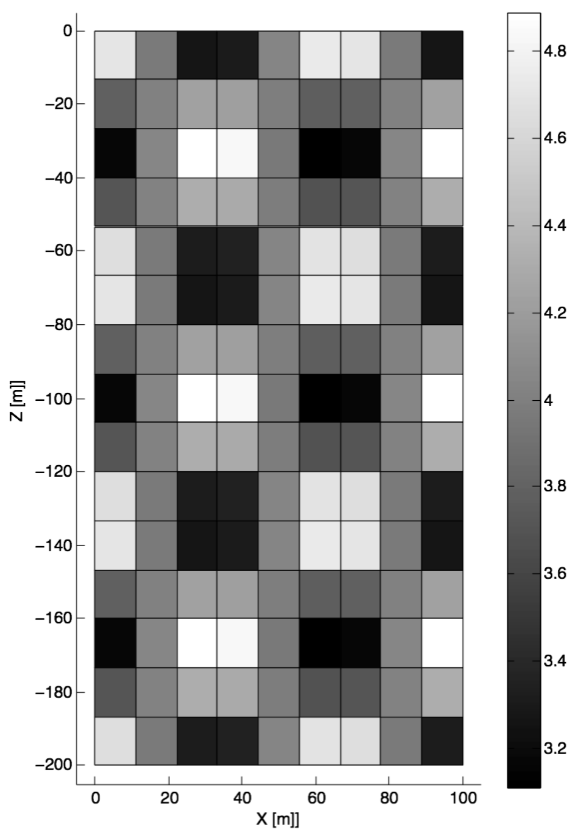


Figura 2: Modelo de velocidad sintético

Se pide completar el notebook `Tarea2_P2.ipynb`. En este parte se resuelve el problema de inversión sin regularización (por mínimos cuadrados simples),

$$\min_{\mathbf{m}} \|\mathbf{Gm} - \mathbf{d}\|_2^2 \tag{4}$$

En el notebook, primero se generan datos sintéticos, sin agregar ruido, utilizando un modelo de velocidades de tipo tablero de ajedrez como el que se muestra en la Figura 2.

Utilizando los datos sintéticos generados, se pide:

(a) Estimar el modelo de velocidades del medio utilizando Mínimos Cuadrados Simples, esto sin agregar ruido a los datos sintéticos. Grafique el modelo estimado y note que éste es muy similar al modelo utilizado para generar los datos sintéticos (Figura 2).

(b) De manera de mostrar la inestabilidad de las soluciones del problema de tomografía, se añadirá un pequeño ruido aleatorio a los datos sintéticos de tiempos de viaje. El ruido que se agrega debe tener una distribución Normal con media cero y desviación estándar A.

Estime el modelo de velocidades para cuatro niveles de ruido, $A = 0{,}0[ms]$(sin ruido), $A = 0{,}005[ms]$, $A = 0{,}02[ms]$ y $A = 0{,}04[ms]$ y grafique los modelos estimados de la velocidad del medio para los 4 valores de A .

Como aquí la idea es visualizar la influencia de la amplitud del ruido para todas las soluciones, debe tener especial cuidado en que cuando se genere el ruido en cada caso, éste sea la misma realización del generador de números aleatorios, y que sólo cambie la amplitud del ruido. Para ello use una semilla basada en su fecha de nacimiento (ver notebook), por ejemplo seed = 03101990 si nació el 3 de Octubre de 1990.

Notar que los tiempos de viaje de las ondas para el modelo sintético varían entre ~ 20 y ~ 50 $[ms]$ (milisegundos) y que los niveles de ruido A son una fracción bastante pequeña de la amplitud de las observaciones.

(c) Haga lo mismo que en (b), pero esta vez mantenga constante la amplitud del ruido en $A = 0{,}02$, y haga 4 inversiones en que cambia la realización del ruido aleatorio. Puede usar la misma semilla para reiniciar el estado del generador de números aleatorios cuando calcule la primera realización del ruido aleatorio, y luego calcular el ruido en las siguientes inversiones sin reinicializar el generador de números aleatorios, de manera que la realización del ruido sea diferente.

(d) Compare los resultados obtenidos con el modelo sintético y comente acerca de la estabilidad del método de minimos cuadrados simples no regularizado para este tipo de problemas. ¿ Cómo podría intuir la inestabilidad del problema de inversión sin tratar de resolver el problema inverso? (explique el procedimiento).

(e) ¿ Por qué para esta pregunta se necesita que en todos los casilleros pase por lo menos 1 rayo?

P3. Regularización L_2 del Problema de Tomografía Lineal

En la pregunta P2 mostramos la inestabilidad del problema de tomografía. Ahora vamos a realizar la estimación del modelo de velocidades utilizando el método de Mínimos Cuadrados Regularizado, utilizando una regularización de Tikhonov. Es decir, resolveremos el problema de estimación:

$$\min_{\mathbf{m}} \underbrace{\|\mathbf{Gm}-\mathbf{d}\|_2^2 + \varepsilon^2 \|\mathbf{Hm}\|_2^2}_{\phi(\mathbf{m})}, \tag{5}$$

donde $\mathbf{H}$ es un operador lineal que se aplica sobre $\mathbf{m}$, el término $\|\mathbf{Hm}\|_2^2$ define la regularización o información *a priori* del modelo de lentitudes $\mathbf{m}$ y ε^2 representa el peso relativo entre las ecuaciones de regularización y las del problema físico planteado.

En la resolución de este problema, escogeremos $\mathbf{H}$ como una aproximación de diferencias finitas del operador diferencial Laplaciano (∇^2) con condiciones de borde de Newmann. Luego el término $\|\mathbf{Hm}\|_2^2 = \left\|\nabla^2\mathbf{m}\right\|_2^2$ de la función objetivo $\phi(\mathbf{m})$ es una medida de la rugosidad del modelo $\mathbf{m}$ (o de que tan poco suave son las variaciones espaciales de lentitud). El stencil (o esquema de diferencias finitas) del operador Laplaciano se muestra en la Figura 3.

En esta pregunta se pide:

(a) Complete la función `order_2()` ubicada en el módulo `Tarea2.Mesh.TikhonovReg.py`, que permite calcular el operador de regularización Laplaciano (como lo vimos en clases) en función de la malla definida con el objeto de tipo `RectangularMesh`.

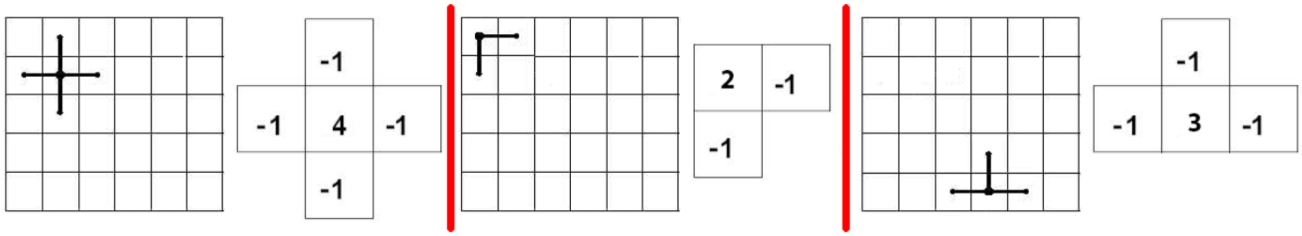


Figura 3: Sténcil del operador Laplaciano $\mathbf{H} = \nabla^{\mathbf{2}}$ con condiciones de borde de Newmann.

(b) Exprese el problema General Mínimos Cuadrados Regularizado

$$\min_{\mathbf{m}} \left\| \mathbf{W}_{\chi} \left(\mathbf{G}\mathbf{m} - \mathbf{d} \right) \right\|_2^2 + \varepsilon^2 \left\| \mathbf{W}_{h} \left(\mathbf{H}\mathbf{m} - \mathbf{h_o} \right) \right\|_2^2$$

como un problema de mínimos cuadrados simples $\mathbf{F}\mathbf{m} = \mathbf{D}$ con una matriz $\mathbf{F}$ y vector $\mathbf{D}$ adecuados (Escriba las ecuaciones). Luego escriba la función `MinCuadRegWeights(G, d, Wx, H, ho, Wh, epsilon)` ubicada en el módulo `MinCuadReg.py` (éste módulo lo debe colocar en su paquete de python `GF5013`) que resuelva el problema de mínimos cuadrados regularizado (amortiguado), como un problema de mínimos cuadrados simple equivalente, para un valor dado de la constante de amortiguamiento ε (`epsilon`). La función debe devolver el modelo estimado $\mathbf{m}^{\mathbf{est}}$ y la matriz de covarianza a posteriori de los parámetros estimados ($\mathbf{C}_{\mathbf{m}}$).

(c) Utilizando `Nx`=21, y `Nz`=43, complete el notebook `Tarea2_P3_a_Preparation_for_Inversion.ipynb` que prepara los datos y calcula la matriz de diseño $\mathbf{G}$ del problema de tomografía. Como el cálculo de $\mathbf{G}$ es lento, en este notebook se calcula la información necesaria para realizar la inversión, y se guarda en el archivo `Tomo_pre_inversion_data.pickle`, para ser utilizada posteriormente.

(d) Complete el notebook `Tarea2_P3_b_Inversion_MinCuadRegTikhonov.ipynb` y resuelva el problema inverso constreñido por datos sintéticos (generados como en la pregunta P2) en que se añade un ruido Gaussiano con una desviación estándar $A = 1[ms]$. Para ello resuelva el problema inverso varias veces, considerando diferentes valores de la constante de amortiguamiento ε (`epislon`) entre 10^{-4} y 10^{4} como límites de partida [1] (posteriormente podrá refinar este intervalo si lo estima conveniente). Guarde los resultados de la inversión en el archivo `Tomo_Result_Inversion.pickle`.

(e) En el notebook `Tarea2_P3_c_Inversion_MinCuadRegTikhonov_Procesamiento_Resultados.ipynb`, cargue los resultados y muestre una figura (use escalas logarítmicas en los ejes) comparando el valor de la norma del vector de error de ajuste ($\left\| \mathbf{G}\mathbf{m}_{\varepsilon^2}^{est} - \mathbf{d} \right\|_2$) y de la norma de la regularización ($\left\| \mathbf{H}\mathbf{m}_{\varepsilon^2}^{est} \right\|_2$) para diferentes valores de ε (curva L). Grafique 3 modelos de velocidades obtenidos para 3 valores de la constante de amortiguamiento, uno sobre-amortiguado, su modelo "favorito" (el que crea que es la mejor solución), y un modelo sub-amortiguado. Explique su razonamiento para escoger el modelo "favorito".

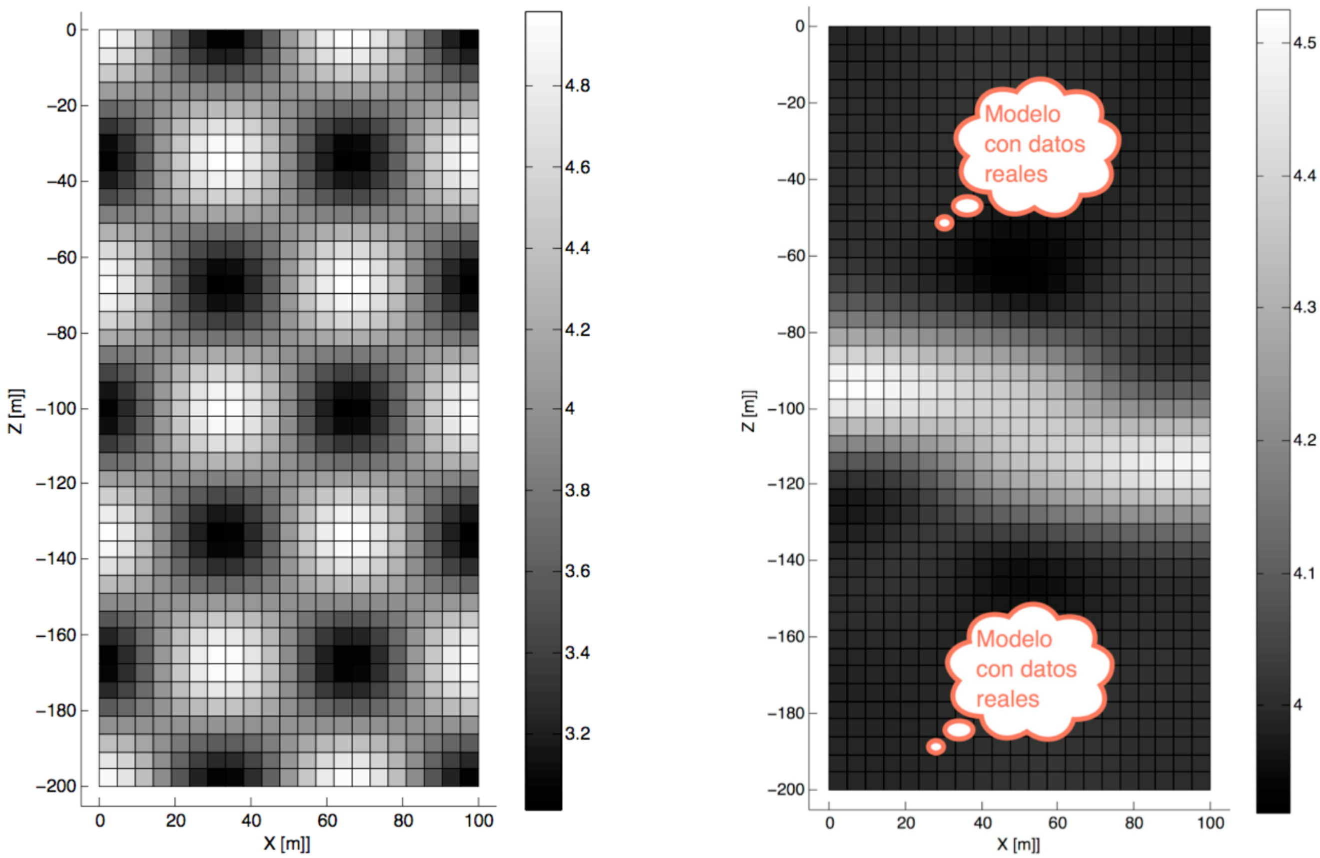
(f) El el contexto de la formulación del problema de tomografía lineal, ¿ Por qué no sería adecuado utilizar condiciones de borde de tipo Dirichlet al definir el operador de regularización que es la aproximación de diferencias finitas de ∇^2?

(g) ¿Por qué no utilizamos una matriz identidad como operador de regularización?. Justifique.

[1] Se recomienda discretizar el intervalo [0.0001, 10000] de manera logarítmica (use `numpy.logspace`). Y usar un número adecuado de valores de ε de manera de muestrear correctamente la curva L.

P4. Inversión con datos reales

Ahora utilizando los datos reales (archivo `datosTomografia.txt`), encuentre la mejor estimación del modelo de velocidades del experimento y grafique el modelo de velocidades. Debe encontrar un resultado similar (puede que un poco diferente) al de la Figura 4b. Justifique su elección de metodología de inversión, tamaño de la discretización y de la selección del mejor modelo. Debe entregar 3 notebooks, uno para preparar la inversión (`P4_a.ipynb`), otro para realizar la inversión (`P4_b.ipynb`) y otro para los resultados (`P4_c.ipynb`).



(a) Modelo sintético para pregunta P3

(b) Ejemplo de solución a obtener con datos reales

Figura 4

Instrucciones

- Todo el material necesario para realizar la tarea se encuentra en una carpeta llamada `Tarea2_NombreAlumno`.
- Copiar la estructura de carpetas en una carpeta con su nombre (Ej: `Tarea2_FranciscoOrtega/*`), dentro de esa carpeta, se encuentra la carpeta `PythonNotebook`, donde se encuentran los notebook de Python en que se deben responder cada una de las preguntas.
- En la carpeta, `PaquetesPython` se encuentran dos paquetes `GF5013` y `Tarea2`, cualquier código que no se encuentre directamente escrito en el notebook, debe ser programado como parte de uno de esos dos paquetes.
- Al entregar la tarea deben entregar, el informe debe ser escrito dentro de los notebook de python que se les entrega, esto de manera ordenada y se entienda claramente los gráficos y las respuestas a cada parte de la tarea. De manera alternativa, si lo desea puede escribir un informe en Latex, Word o el software de su elección, siguiendo los mismos lineamientos. A u-cursos se debe subir un archivo comprimido que tenga la estructura de carpetas `Tarea2_NombreAlumno` con los códigos y notebooks desarrollados al interior (y el informe en formato PDF si no respondió directamente en los notebooks).
- No es necesario una introducción teórica en el informe, pero deben mostrar el desarrollo cuando se pida encontrar expresiones matemáticas o formulaciones particulares de los problemas (escribir las ecuaciones correspondientes y como las transforman cuando aplique). Si una celda del notebook esta en modo "MarkDown", en ella se puede escribir comandos de latex encerrados entre dobles signos pesos. Por ejemplo,

  ```
  $$ \frac{1}{2} \mathbf{G} $$
  ```

  (los espacios son importantes) queda $\frac{1}{2}\mathbf{G}$.
- En la clase auxiliar se dedicará tiempo a responder preguntas que se tengan acerca del desarrollo de la tarea. De la misma manera estaré disponible los miércoles de 12:00 a 13:30 en mi oficina para consultas.
- Asimismo es una excelente práctica el escribir códigos que sean modulares de manera de poder reutilizar la mayor cantidad de veces ese código (obviamente sin abusar).
- Ver el notebook `Ejemplo_crear_y_graficar_mallas_y_cargar_datos.ipynb` para ayuda en como utilizar los archivos de datos y funciones varias provistas en esta tarea.
- En la clase auxiliar se dedicará tiempo a responder preguntas que tengan acerca del desarrollo de la tarea. Además les recomiendo que empiecen desde ya a trabajar en la tarea, ya que los códigos se van a demorar en correr **ESPECIALMENTE EN LA P3 y P4** .
- Se entrega una copia del paquete GF5013.optim en donde se encuentra programada la función MinCuadSimple de manera optimizada (usando las funciones de numpy). Debe ocupar esa función en sus cálculos.
- **Se aceptan entregas atrasadas pero por cada día hábil de atraso se resta 0.5 pts a la nota final de la tarea.**

Exito!!!..