



Background field removal usando el Método de Elementos Finitos

Estudiante: Tabita Catalán Muñoz
Lugar: Centro de Imágenes Biomédicas UC
Evaluador: Cristián Tejos

Fecha de inicio: 18 de diciembre de 2019
Fecha de término: 31 de enero de 2020
Santiago, Chile

Índice de Contenidos

1. Resumen	1
2. Contexto	2
2.1. Quatitative Susceptibility Mapping	2
2.2. Background field removal	3
3. Generar la malla	5
3.1. Malla no regular	5
3.2. Malla regular	6
4. Resolver la ecuación con FEniCS	8
5. Visualizar resultados y comparar	10
6. Análisis de resultados	11
6.1. Erosión de 5 vóxeles	11
6.1.1. FEM: Malla no regular	11
6.1.2. FEM: Malla regular	12
6.2. Sin erosión	13
6.2.1. FEM: Malla no regular	13
6.2.2. FEM: Malla regular	14
6.3. Comparación mediante RMSE	15
6.4. Tiempos de ejecución	15
7. Conclusiones	15
8. Anexos	16
8.1. Datos	16
8.2. Parámetros para malla no regular	16

Índice de Figuras

1. Etapas para obtener un mapa de susceptibilidad a partir de una adquisición.	3
2. Malla generada por <code>example_meshing.m</code>	6
3. Idea para generar la malla regular	6
4. Ejemplo de máscara problemática	7
5. Malla regular generada por <code>Example_regular_mesh.m</code>	7
6. Una solución vista en Paraview	9
7. <i>Phase tissue</i> obtenida con LBV, en ppm, con erosión de 5 vóxeles.	11
8. <i>Phase tissue</i> obtenida con FEM, en ppm, con erosión de 5 vóxeles.	11
9. Diferencia de <i>phase tissue</i> FEM - LBV, en ppm, con erosión de 5 vóxeles.	12
10. <i>Phase tissue</i> obtenida con FEM, en ppm, con erosión de 5 vóxeles.	12
11. Diferencia de <i>phase tissue</i> FEM - LBV, en ppm, con erosión de 5 vóxeles.	12
12. <i>Phase tissue</i> obtenida con LBV, en ppm, sin erosión.	13

13.	<i>Phase tissue</i> obtenida con FEM, en ppm, sin erosión.	13
14.	Diferencia de <i>phase tissue</i> FEM - LBV, en ppm, sin erosión.	13
15.	<i>Phase tissue</i> obtenida con FEM, en ppm, sin erosión.	14
16.	Diferencia de <i>phase tissue</i> FEM - LBV, en ppm, sin erosión.	14

Índice de Tablas

1.	Comparación por RMSE entre LBV y FEM	15
2.	Parámetro de <code>unstructured_meshing</code> para malla no regular	16

1. Resumen

La remoción del campo de fondo (*background field removal*) es una etapa importante para generar mapas de susceptibilidad magnética mediante QSM. Una forma de lograrlo es la propuesta por Zhou, Liu, Spincemaille y Wang en [9], que aprovecha la armonicidad del campo de fondo al interior de la región de interés (ROI) para plantear una Ecuación de Laplace. Las condiciones de borde para el campo de fondo son desconocidas, pero pueden aproximarse por las del campo total, debido a que este es uno o dos órdenes de magnitud superior al campo local. En [9], la ecuación se resuelve mediante el Método de Diferencias Finitas, utilizando un esquema multigrilla. El objetivo del presente trabajo es resolver la ecuación mediante el Método de Elementos Finitos.

El trabajo realizado se separa en tres etapas:

1. **Generar la malla:** Para resolver la ecuación con Elementos Finitos se necesita una malla de tetrahedros. Esta se genera en Matlab a partir de una *máscara* del cerebro, obtenida a partir de los datos de MRI medidos. Se trabaja con dos tipos de malla: regular y no regular. La malla regular se crea replicando una triangulación de un cubo en cada agrupación cúbica de 8 vóxels. Para la malla no regular se usa el toolbox *iso2mesh* de Matlab.
2. **Resolver la ecuación con FEniCS:** *FEniCS* es una plataforma computacional de código abierto, con interfaz de alto nivel en Python. Permite resolver EDPs con el método de elementos finitos a partir de la formulación variacional en una malla de tetrahedros.
3. **Visualizar resultados y comparar:** Finalmente, los resultados obtenidos con FEniCS se importan a Matlab, donde se termina el proceso de QSM con la etapa de Reconstrucción. Para estudiar la eficacia del método, se usan datos de [3] y se comparan con métricas como *Root Mean Squared Error* (RMSE).

Los códigos escritos pueden encontrarse en [4]. El presente informe pretende ser una síntesis de lo realizado, así como una guía para entender y usar los códigos escritos.

El trabajo se llevó a cabo en el Centro de Imágenes Biomédicas UC, a cargo del profesor Cristián Tejos del Departamento de Ingeniería Eléctrica de la Universidad Católica.

2. Contexto

2.1. Quatitative Susceptibility Mapping

Lo siguiente es una síntesis de la introducción de [8].

La susceptibilidad magnética es una medida de cuánto se magnetiza un material al aplicar un campo magnético. Dentro del contexto de Imagen por Resonancia Magnética (MRI), *Quatitative Susceptibility Mapping* (QSM) es una técnica que busca generar un mapa de susceptibilidad, mediante la aplicación de un campo magnético B_0 (que supondremos s.p.g. en la dirección z) a los tejidos, y la medición de una señal. Las diferencias en el campo magnético que excita a cada molécula se traducen en un cambio en la señal medida.

Si se cuenta con la distribución de susceptibilidad magnética χ , es posible calcular el desfase en la señal mediante:

$$\Delta\phi = d * (\chi - \chi_{\text{aire}}) \quad (1)$$

donde $*$ es el operador de convolución y d es el campo del kernel dipolar, del cual conocemos su transformada de Fourier:

$$\mathcal{F}d[k] = D[k] = \gamma B_0 \cdot TE \cdot \left(\frac{1}{3} - \frac{k_z^2}{k^2} \right),$$

donde γ es la constante giromagnética del hidrógeno y TE es el tiempo de eco, un parámetro de adquisición conocido. A esto se le conoce como **problema directo**.

El desafío está en el **problema inverso**: encontrar la distribución de susceptibilidad a partir de las diferencias de fase. Uno podría simplemente aplicar Transformada de Fourier en (1), lo que transforma la convolución en multiplicación puntual. Lamentablemente, en el dominio de Fourier hay una superficie (llamada cono mágico) donde el kernel dipolar vale 0, lo que genera valores indefinidos al dividir. Además, el problema está “mal puesto”, por lo que el ruido tiende a amplificarse al invertir.

Es por esto que el problema se resuelve en varias etapas (ver Figura 1).

1. La primera dificultad es que la fase en bruto (*raw phase*) está “envuelta”; esto pues los ángulos se miden en un rango de $-\pi$ a π , mientras que el verdadero rango podría ser mucho mayor. El algoritmo más común para corregir esto es el *Laplacian unwrapping*, dando lugar a la fase sin envolver (*unwrapped phase*).
2. La siguiente dificultad es eliminar las contribuciones al campo magnético inducidas por fuentes externas (el resto del cuerpo, por ejemplo). Esto incluye cavidades de aire, como los senos nasales o la boca. Esta etapa se conoce como *Background field removal*, y es la que nos interesa en este trabajo.
3. La última etapa consiste en reconstruir la susceptibilidad a partir de la fase, para lo cual hay varios métodos como TKD o COSMOS.

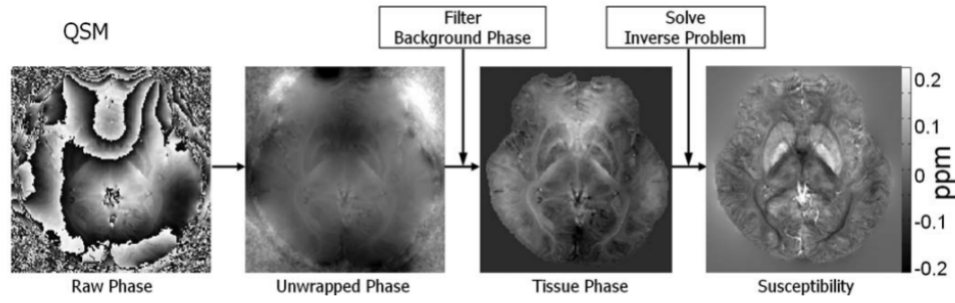


Figura 1: Etapas para obtener un mapa de susceptibilidad a partir de una adquisición.

2.2. Background field removal

Los métodos actuales para realizar *Background Removal* pueden dividirse en 3 tipos:

- Basados en la propiedad del Valor Medio para funciones armónicas (SHARP) (ver [5]).
- Basados en resolver la Ecuación de Laplace (LBV).
- Basados en proyección sobre la base ortogonal de las funciones armónicas (PDF) (ver [6]).

Uno de los mejores métodos hasta la fecha es el descrito en [9], el cual resuelve la Ecuación de Laplace con el Método de Diferencias Finitas, usando un esquema multigrilla (FMG). A este método lo llamaremos LBV de ahora en adelante. El objetivo del trabajo es resolver la Ecuación de Laplace mediante el Método de Elementos Finitos (FEM), y comparar los resultados obtenidos con los de [9], idealmente superándolos.

Las etapas anteriores de QSM permiten obtener datos del campo magnético total f_T . Se busca separar el campo magnético local f_L , que es producido dentro de la región de interés (ROI), que llamaremos Ω , del campo magnético de fondo f_B , que es producido por tejidos fuera de la ROI, y del cual se sabe que es armónico (su laplaciano es 0). Luego:

$$\begin{cases} f_T = f_L + f_B \\ \Delta f_B = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) f_B = 0 \end{cases} \quad \text{en } \Omega$$

Para resolver esto necesitamos condiciones de borde para f_B , las que no están fácilmente disponibles. Sin embargo, es sabido que el campo local f_L es en muchos casos uno o dos órdenes de magnitud menor que el campo de fondo f_B ($f_L \ll f_B$). Luego, podemos aproximar $f_B|_{\partial\Omega} = f_T|_{\partial\Omega}$.

Con esto, planteamos finalmente el problema que buscamos resolver:

$$\begin{cases} \Delta f_B = 0 & \text{en } \Omega \\ f_B = f_T & \text{en } \partial\Omega \end{cases}$$

Finalmente, lo obtenido a partir de las etapas anteriores de QSM es lo que llamamos *fase desenrollada*, que denotamos ϕ_T . Se sabe que $\phi \propto f_T$, y debido a la linealidad del problema, podemos simplemente buscar la fase que viene del campo de fondo ϕ_B :

$$\begin{cases} \Delta \phi_B = 0 & \text{en } \Omega \\ \phi_B = \phi_T & \text{en } \partial\Omega \end{cases}$$

3. Generar la malla

Para resolver la ecuación con Elementos Finitos se necesita una malla de tetrahedros. Esta se genera en Matlab a partir de una *máscara* del cerebro, obtenida a partir de los datos medidos. Se trabaja con dos tipos de malla: regular y no regular.

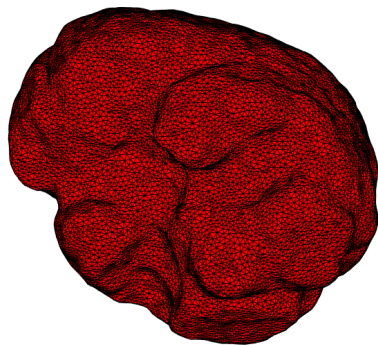
3.1. Malla no regular

La generación de la malla no regular usa fuertemente el toolbox `iso2mesh` de Matlab, y se condensa en la función `unstructured_meshing`. Esta, en el atributo `mask`, toma una *máscara* binaria de la ROI y crea la malla en varias etapas:

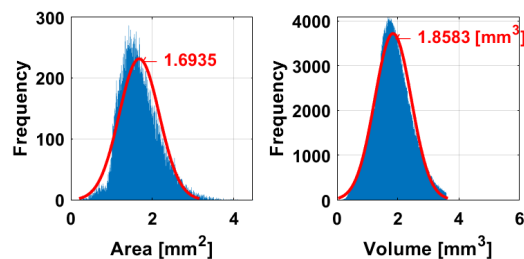
1. Suavizar la máscara con un kernel de convolución. Se hace con la función `smooth3` de Matlab.
2. Hacer una malla de la frontera de la ROI. Esto se hace a través de la función `mesh_boundary`, la cual encuentra el borde viéndolo como una isosuperficie de valor 0.5. Esto pues el borde es la zona donde los valores de la máscara cambian de 0 a 1.
3. Remallar la frontera usando los parámetros `gridsize`, `closesize`, `elemsize`, a través de la función `remeshsurf` de `iso2mesh`. Ver su documentación para más detalle de los parámetros.
4. Suavizar la malla de la frontera con un filtro laplaciano. Se hace mediante la función `smoothsurf` de `iso2mesh`.
5. Crear la malla de la ROI a partir de la malla de la frontera, usando los parámetros `keep_ratio` (proporción de elementos que se mantendrán) y `vol_factor` (el volumen máximo de cada elemento será $0.12 \cdot \text{vol_factor}$. 0.12 es el volumen de un tetrahedro regular de lado 1). Se hace mediante la función `s2m` de `iso2mesh`.

Un ejemplo de cómo usar la función se encuentra en el script `example_meshing.m`. Antes de ejecutarlo se deben considerar lo siguiente:

1. La línea `addpath(' ../iso2mesh/')` debería cambiarse para que coincida con el path a `iso2mesh`, o eliminarse si ya está en el path de Matlab.
2. El script usa una máscara erosionada (ver la subsección Datos de Anexos). Se pide ingresar el `peel` adecuado, dependiendo de la máscara que se está usando: 0 para `mask_p0.mat`, 1 para `mask_p1.mat` y 5 para `mask_p5.mat`. Esto solo influye en el nombre de la carpeta donde serán guardados los datos.
3. Se generará una nueva carpeta al interior de `results/`, cuyo nombre dependerá de los parámetros usados, mediante la función `generate_folder_name`. El nombre de esta carpeta también quedará guardado en `examples/folder_name.txt`, y será usado posteriormente por el script `examples/example_solving.py`
4. En esta carpeta se guardará un archivo `mesh_data.vtu` que contiene los datos de la malla creada y de la fase, obtenida de `phs_unwrap.mat`. Se guarda también un histograma del tamaño de los elementos de la malla, una imagen de la malla (ver Figura 2) y un archivo `log_mesh.txt` que contiene la salida en consola generada por `iso2mesh` al crear la malla.



(a) Malla



(b) Histogramas: tamaño de los elementos

Figura 2: Malla generada por `example_meshing.m`

3.2. Malla regular

La malla regular se crea replicando una triangulación de Delauney de un cubo en cada agrupación cúbica de 8 nodos vecinos (ver Figura 3). Esto se hace solamente para las agrupaciones completas, es decir, aquellas cuyos 8 nodos corresponden a un valor 1 es la máscara binaria.

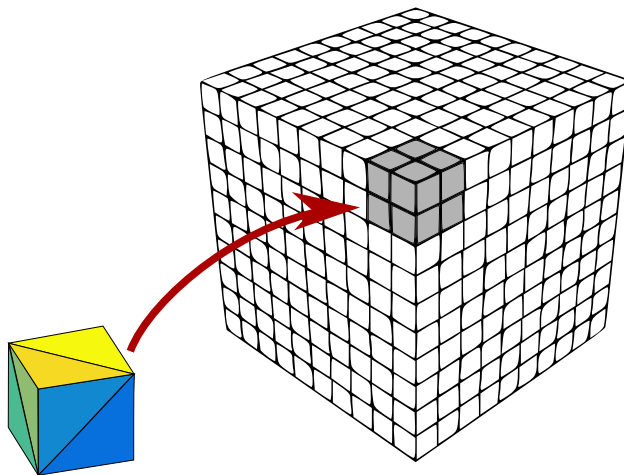


Figura 3: Idea para generar la malla regular

Se hace notar esto puede producir pérdida de datos en la frontera de la malla, lo que no se logró corregir. Para ilustrar esto basta considerar una máscara como la de la Figura 4, donde todas las posibles agrupaciones de 8 vecinos están incompletas, por lo que no es posible generar una malla regular de esta manera.

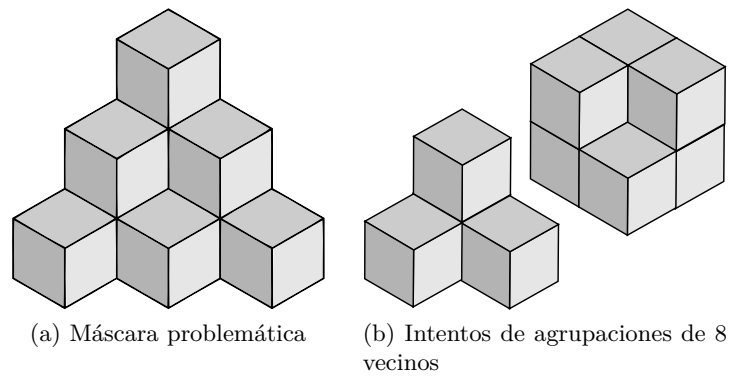
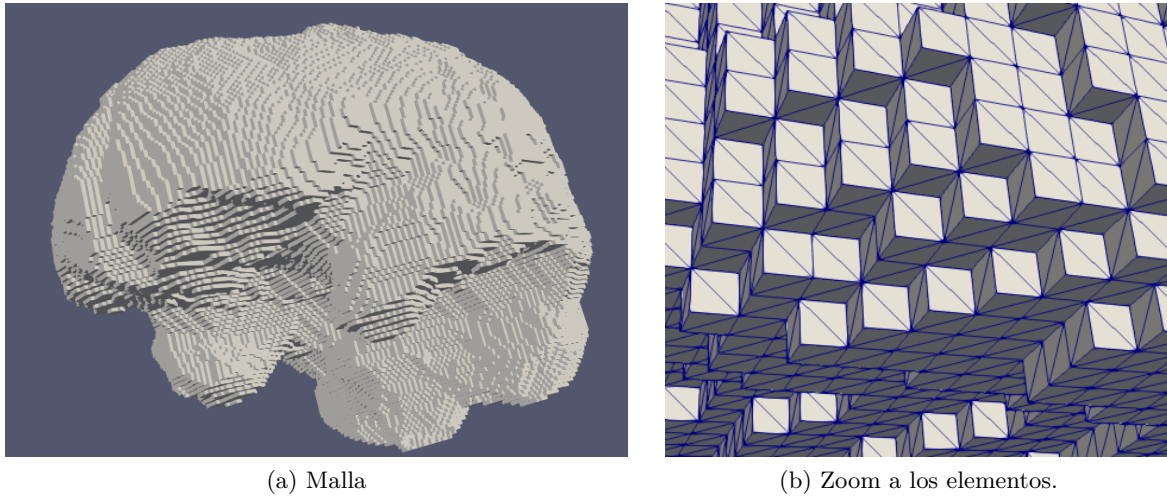


Figura 4: Ejemplo de máscara problemática

Puesto que no se creó como función de Matlab, el mallado regular solo se encuentra disponible a través del script `Example_regular_mesh.m`. La máscara debe quedar guardada en la variable `SEG`.

Figura 5: Malla regular generada por `Example_regular_mesh.m`

4. Resolver la ecuación con FEniCS

FEniCS es una plataforma computacional de código abierto, con interfaz de alto nivel en Python. Permite resolver EDPs con el método de elementos finitos a partir de la formulación variacional en una malla de tetrahedros. Debe usarse desde una terminal de Linux. Se puede usar desde otro sistema operativo mediante **Docker** o **Windows Subsystem for Linux**.

Una vez que contamos con una malla de en formato **.vtu**, esta puede ser importada a FEniCS. La función para hacer esto es `solve_laplace`, del archivo `solve_laplace.py` de [4]. Esta función tiene por argumentos `path`; un `str` con la dirección a la carpeta donde se encuentra la malla, y `filename`; el nombre del archivo **.vtu** donde está guardada la malla (sin extensión). La función obtendrá los datos de la malla,

```
geometry = meshio.read(path + filename + ".vtu")
```

los pasará a un archivo **.xdmf** que quedará guardado en `path`, y luego creará una malla que puede ser usada por **Dolfin** para resolver el problema usando Elementos Finitos. Se usan elementos de Lagrange de grado 1, lo que se expresa en la línea:

```
V = FunctionSpace(mesh, "CG", 1)
```

Suponemos además que los datos de la fase ϕ_T están guardados en el archivo **.vtu**, y serán leídos en la instrucción

```
point_data = geometry.point_data
```

para guardarse en un archivo **.xdmf**, que después es leído, y los datos quedan guardados en la función `upha`.

Para el problema

$$\begin{cases} -\Delta u = 0 & \text{en } \Omega \\ u = \phi_T & \text{en } \partial\Omega \end{cases}$$

podemos multiplicar por una función test v , integrar en Ω e integrar por partes para obtener su formulación variacional:

$$\begin{cases} a(u, v) := \int_{\Omega} \nabla u \nabla v - \int_{\partial\Omega} \nabla u \cdot n v = 0 =: L(v) & \forall v \in V \\ u = \phi_T & \text{en } \partial\Omega \end{cases}$$

donde n es la normal al borde $\partial\Omega$ de la ROI, apuntando hacia afuera. Para resolver el problema en FEniCS escribimos:

```
n = FacetNormal(mesh) # normal

u = TrialFunction(V)
v = TestFunction(V)
F = inner(grad(u), grad(v))*dx - dot(n, grad(u))*v*ds

# Separate left and right sides of equation
a, L = lhs(F), rhs(F)
```

Las condiciones de borde Dirichlet se imponen de la siguiente forma:

```
class Boundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary
```

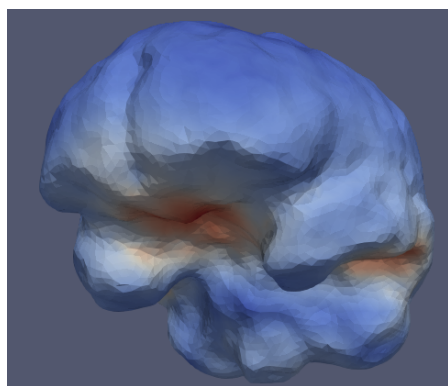
```
bc = DirichletBC(V, upha, Boundary())
```

Para resolver el problema se eligió usar el solver `cg` (conjugate gradient method), que funciona bien para resolver la ecuación de Laplace. Para más opciones ver [2].

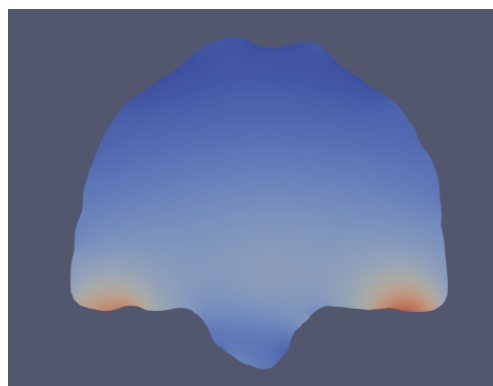
```
u = Function(V)
solve(a == L, u, bc, solver_parameters={'linear_solver': 'cg', 'preconditioner': 'ilu'})
```

Es posible correr esto en paralelo (ver [1]), pero antes se debe cambiar el preconditionador `ilu` por uno que sí se pueda usar junto a `mpirun`.

Finalmente, la función guardará la solución en un archivo `Solution.pvd`, dentro del directorio entregado en el argumento `path`. Este archivo puede ser visualizado usando herramientas como `ParaView` (ver Figura 6).



(a) Malla



(b) Un corte

Figura 6: Una solución vista en Paraview

Puede verse un ejemplo del uso de la función `solve_laplace` en `examples/example_solving.py`. El script supone que la malla está guardada en alguna carpeta dentro de `results/`, y que el nombre de esa carpeta está escrito en el archivo `examples/folder_name.txt`.

5. Visualizar resultados y comparar

Una vez se ha obtenido una solución con FEniCS, se importa a Matlab, donde se termina el proceso de QSM con la etapa de Reconstrucción. Para estudiar la eficacia del método, se usan datos de [3] y se comparan los resultados con los obtenidos con LBV ([9]) del `MEDI_toolbox`. Esto puede hacerse mediante métricas como *Root Mean Squared Error* (RMSE), o visualmente, restando el resultado con un *ground truth* y estudiando la imagen obtenida. Esto se realizó mediante el script `example_read_compare_solution.m`

Algunas consideraciones:

1. Al cargar los datos, se espera que tanto la máscara como la fase usada tengan el mismo nivel de erosión (ver subsección Datos en Anexos). Esto significa usar `mask_p0.mat` junto a `phs_lbv_p0.mat`, `mask_p1.mat` junto a `phs_lbv_p1.mat` etc. Los datos elegidos deben quedar en las variables `mask` y `phs_lbv`.
2. El script supone que existe un archivo `Solution000001.vtu` al interior de la carpeta en `results` cuyo nombre aparece, al igual que antes, en `examples/folder_name.txt`. Esto es cierto si se ejecutó `examples/example_solving.py` antes.

El script está basado en [3] y [7], y utiliza varias funciones sacadas de [3] como `compute_rmse`, `imagesc3d2`, `polyfit3D_NthOrder` y `TKD`. Un resumen de lo que hace:

1. Obtiene el ruido armónico a partir de `'Solution000001.vtu'`.
2. Interpola el ruido desde la malla a la grilla original de $160 \times 160 \times 160$.
3. Calcula la *phase tissue* y la escala por un valor que depende del campo magnético B_0 al que fueron sometidos los tejidos, el radio giromagnético γ y el tiempo de eco TE .
4. Se ajusta un polinomio 3D de orden 4 a los datos, con el fin de remover la *transmit phase* (ver [3]).
5. Se reconstruye usando `TKD`.
6. Se calcula el error RMSE con `compute_rmse`.

El mismo procesamiento se hace para la fase que se obtiene a partir de método LBV.

6. Análisis de resultados

Se aplican los dos métodos de *background removal*: por un lado, LBV con el esquema FMG (a este método lo llamaremos simplemente LBV), a través de la implementación del **MEDI toolbox** ([9]); por otro lado, el método planteado FEM para las diferentes mallas obtenidas. Estos métodos permiten obtener una *phase tissue*, a la que se le quitó además un polinomio 3D de orden 4 ajustado a los datos (como se muestra en [3]), y se normalizó por $\gamma \cdot TE \cdot B_0$.

Se prueban distintos niveles de erosión: se desea una erosión mínima o nula que permita estudiar la corteza cerebral, pero al resolver el problema sin erosión se incrementa mucho el error.

6.1. Erosión de 5 vóxeles

Con una erosión de 5 vóxeles, LBV obtiene los resultados de la Figura 7.

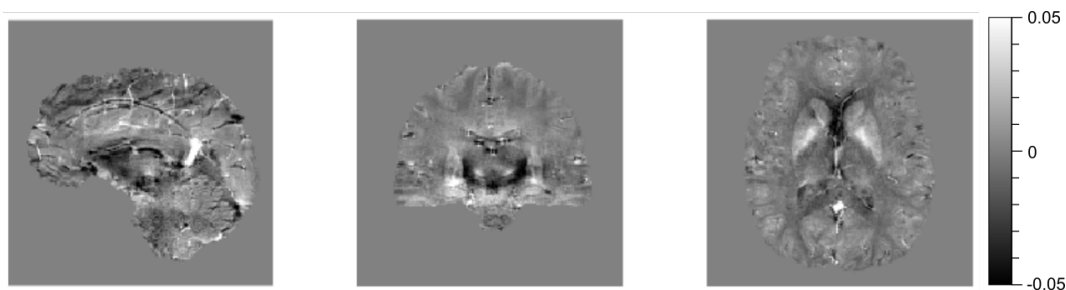


Figura 7: *Phase tissue* obtenida con LBV, en ppm, con erosión de 5 vóxeles.

6.1.1. FEM: Malla no regular

Usando FEM con una malla no regular (ver ?? en Anexos) se obtienen los resultados de la Figura 8, los cuales son bastante similares a los logrados por LBV. Las flechas rojas, sin embargo, apuntan a artefactos que no están presentes en LBV, los cuales se originan probablemente en vóxeles perdidos al interpolar datos a la malla.

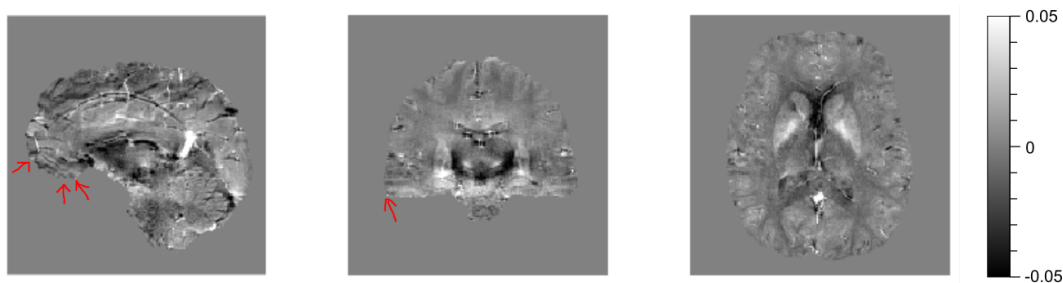


Figura 8: *Phase tissue* obtenida con FEM, en ppm, con erosión de 5 vóxeles.

Las mayores diferencias entre FEM y LBV se encuentran en el borde, como muestra la Figura 9.

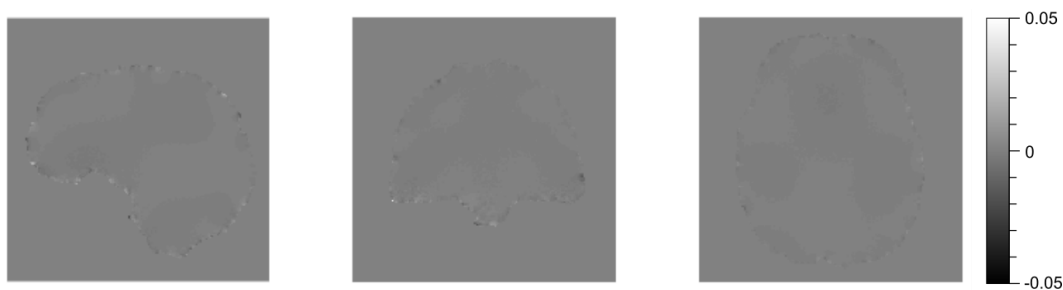


Figura 9: Diferencia de *phase tissue* FEM - LBV, en ppm, con erosión de 5 vóxeles.

6.1.2. FEM: Malla regular

Usando una malla regular vemos los resultados de la Figura 10. En la figura central es especialmente notoria una zona negra en el borde superior derecho de la imagen, la que se debe probablemente al problema que se mencionó con el mallado regular: la pérdida de vóxeles en los grupos incompletos de 8 nodos vecinos.

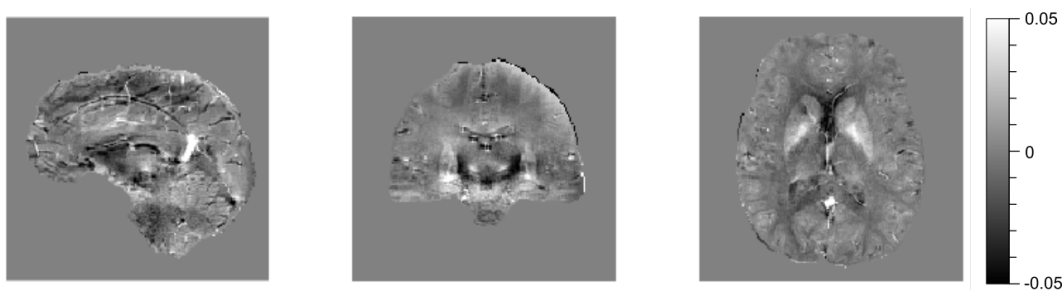


Figura 10: *Phase tissue* obtenida con FEM, en ppm, con erosión de 5 vóxeles.

Esta vez en la Figura 11 vemos claramente una componente suave en la diferencia entre los métodos. Lamentablemente no se logró seguir estudiando esto, por lo que no se sabe cuál de los dos métodos la eliminó correctamente. En caso de que haya sido FEM, daría un motivo para seguir estudiando el método.

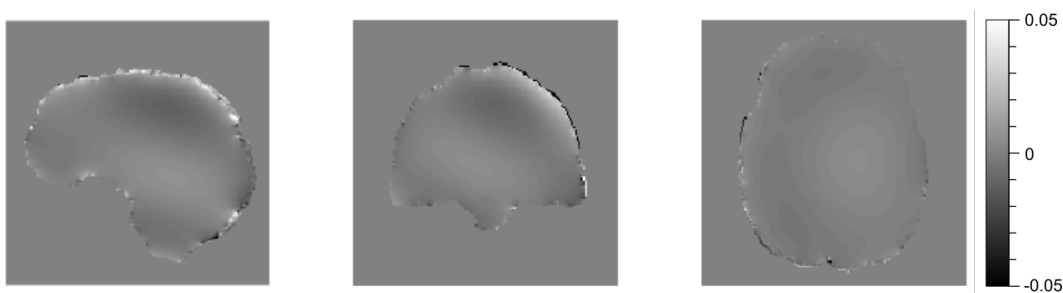


Figura 11: Diferencia de *phase tissue* FEM - LBV, en ppm, con erosión de 5 vóxeles.

6.2. Sin erosión

Sin erosión, LBV obtiene los resultados de la Figura 12. Como se esperaba, el ruido es bastante mayor que antes.

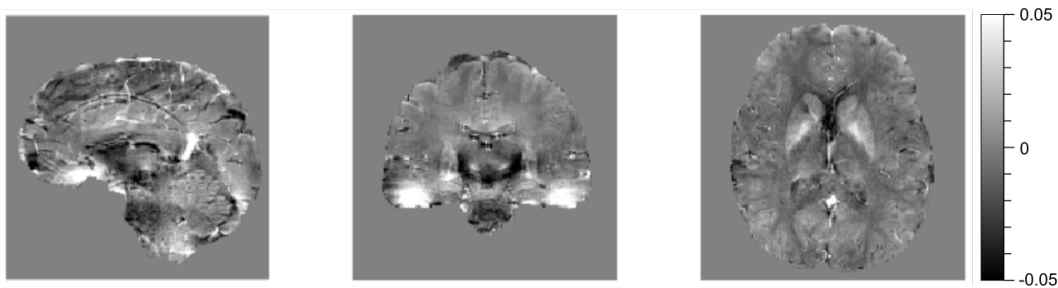


Figura 12: *Phase tissue* obtenida con LBV, en ppm, sin erosión.

6.2.1. FEM: Malla no regular

Usando una malla no regular (igual que antes ver Parámetros para malla no regular en Anexos), vemos los resultados de la Figura 13. Notamos que el ruido en la imagen es mucho más visible que para LBV.

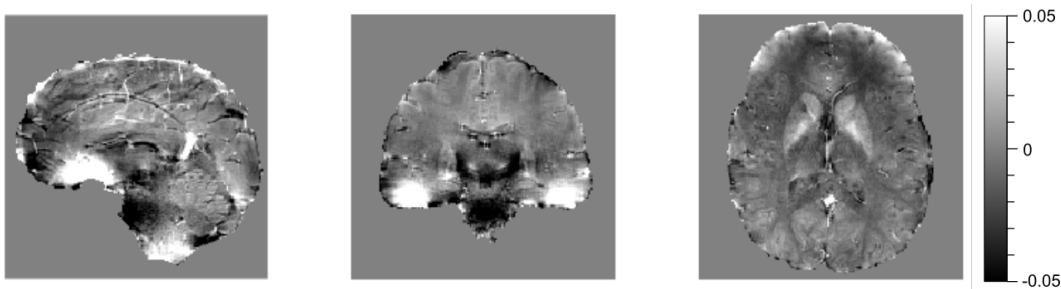


Figura 13: *Phase tissue* obtenida con FEM, en ppm, sin erosión.

Nuevamente, las mayores diferencias entre FEM y LBV se encuentran en el borde, y esta vez la Figura 14 también muestra una componente suave en la diferencia, una que alguno de los métodos está fallando en eliminar.

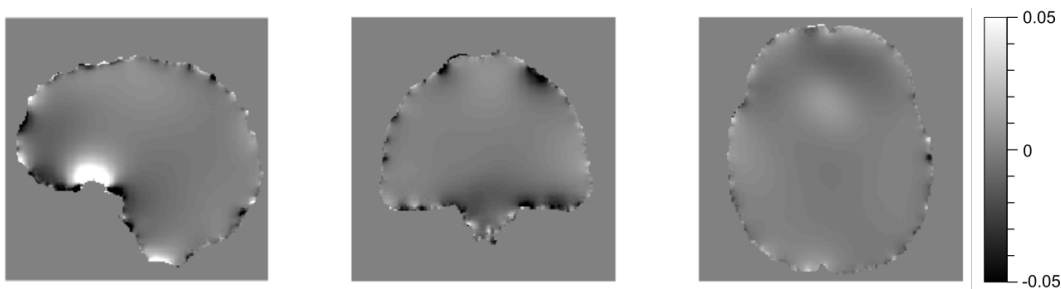


Figura 14: Diferencia de *phase tissue* FEM - LBV, en ppm, sin erosión.

6.2.2. FEM: Malla regular

Para el caso sin erosión, nuevamente vemos mucho error, pero el resultado obtenido es mejor que el logrado con la malla no regular. Nuevamente vemos una zona negra en la parte superior derecha de la imagen central, aunque, en contra de lo esperable, es de menor tamaño que en el caso con mayor erosión.

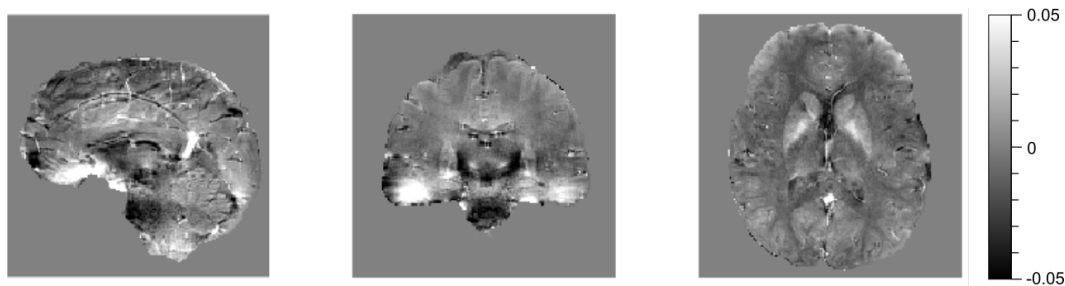


Figura 15: *Phase tissue* obtenida con FEM, en ppm, sin erosión.

Nuevamente, en la Figura 16 se observa la componente suave que no fue removida por alguno de los métodos.

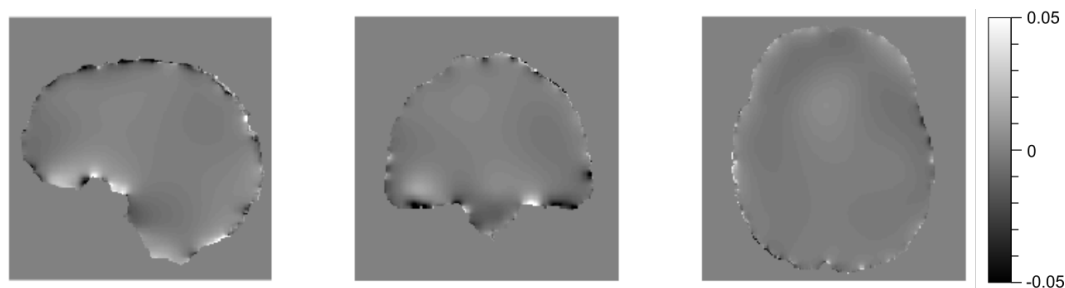


Figura 16: Diferencia de *phase tissue* FEM - LBV, en ppm, sin erosión.

6.3. Comparación mediante RMSE

Un indicador de la calidad de un método es *Root mean square error* (RMSE) (mientras más cercano a 0 mejor). Una comparación entre los dos métodos usando este indicador se encuentra en la Tabla 1.

Para obtener estos valores se realizó la reconstrucción usando *Thresholded K-space Division* (TKD), que consiste en cambiar los 0s del cono mágico por un valor $\varepsilon > 0$, lo que permite realizar la inversión.

Tabla 1: Comparación por RMSE entre LBV y FEM

Erosión	RMSE		Tipo de malla
	LBV	FEM	
5 vóxeles	70.4	71.8	No regular
		144.1	Regular
Sin erosión	129.4	218.5	No regular
		165.2	Regular

Los altos valores de RMSE se explican porque no se prestó especial atención a la etapa de reconstrucción, y solo se usó TKD, el método menos sofisticado.

Vemos que LBV consigue resultados mejores en todos los casos. La malla no regular funciona bastante mal en el caso sin erosión. Al pasar de erosión de 5 vóxeles a trabajar sin erosión aumenta mucho el error tanto en LBV como en la malla no regular, pero el error crece mucho menos para la malla regular.

6.4. Tiempos de ejecución

Usar FEM resulta bastante más costoso: en un notebook con Intel i3-4005U con dos procesadores a 1.70GHz y 8GB de RAM, LBV tarda menos de 10 segundos, mientras que con FEM, generar la malla lleva unos 15 segundos en el caso regular y unos 4 minutos en el caso no regular. Además, exportar los datos para trabajar en FEniCS e importarlos de regreso se hace de una forma muy lenta, ya que Matlab no trabaja con archivos *.vtu* y estos deben escribirse manualmente. Esto tarda unos 3 o 4 minutos extra. Esto se podría corregir resolviendo todo en Matlab.

7. Conclusiones

Se logra implementar *Background removal* usando Elementos Finitos, pero en general FEM resulta peor que LBV, tanto en tiempo de ejecución como en calidad de los resultados.

Se esperaba mejorar la reconstrucción en los bordes de la ROI, sin embargo, con los dos tipos de malla usados se pierden algunos vóxeles cerca de la frontera, lo que empeora la calidad en esa zona. Quedó pendiente averiguar de dónde procedía la componente suave al interior de la ROI que aparecía en la diferencia de las *phase tissue* entre FEM y LBV.

8. Anexos

8.1. Datos

La carpeta `data/` contiene varios archivos. A continuación se dará un breve descripción de ellos. A menos que se indique lo contrario, los datos fueron obtenidos de [3].

- `Mask_bet.mat`: máscara del cerebro obtenida mediante BET (*Brain Extraction Tool*, un método para segmentar imágenes del cerebro obtenidas por resonancia magnética).
- `msk.mat`: una máscara del cerebro, obtenida a partir de erosionar `Mask_bet.mat` en 5 vóxeles.
- `phs_unwrap.mat`: fase después de *Laplacian unwrapping*, y enmascarada por `Mask_bet.mat`.
- `mask_p0`, `mask_p1`, `mask_p5`, `phs_lbv_p0.mat`, `phs_lbv_p1.mat`, `phs_lbv_p5.mat`: obtenidos corriendo el script `examples/example_get_mask_from_LBV.m`, el cual obtiene las máscaras usadas por el método LBV de `MEDI_toolbox`.

8.2. Parámetros para malla no regular

La malla no regular usada para obtener los resultados se obtuvo mediante el script `example_meshing.m`, utilizando la función `unstructured_meshing` con los parámetros de la Tabla 2.

Tabla 2: Parámetro de `unstructured_meshing` para malla no regular

Parámetro	Valor
<code>gridsize</code>	0.4
<code>closesize</code>	0
<code>elemsize</code>	1.6
<code>keep_ratio</code>	0.6
<code>vol_factor</code>	10

Referencias

- [1] Running programs with the `mpirun` command. <https://docs.oracle.com/cd/E19708-01/821-1319-10/ExecutingPrograms.html>. [Online; accedido el 7 de abril de 2020].
- [2] Working with linear solvers. https://fenicsproject.org/pub/tutorial/html/._ftut1017.html. [Online; accedido el 7 de abril de 2020].
- [3] BILGIC, B. Data and code for the QSM 2016 Reconstruction Challenge. <http://martinos.org/~berkin/software.html>. [Online; accedido el 7 de abril de 2020].
- [4] CATALÁN, T. Repositorio en Github: QSM-BackgroundRemoval-LVB-FEM. <https://github.com/tabitaCatalan/QSM-BackgroundRemoval-LVB-FEM>. [Online; accedido el 7 de abril de 2020].
- [5] KAN, H., ARAI, N., TAKIZAWA, M., OMORI, K., KASAI, H., KUNITOMO, H., HIROSE, Y., AND SHIBAMOTO, Y. Background field removal technique based on non-regularized variable kernels sophisticated harmonic artifact reduction for phase data for quantitative susceptibility mapping. *Magnetic Resonance Imaging* 52 (2018), 94 – 101.
- [6] LIU, T., KHALIDOV, I., DE ROCHEFORT, L., SPINCEMAILLE, P., LIU, J., TSIOURIS, A. J., AND WANG, Y. A novel background field removal method for mri using projection onto dipole fields (pdf). *NMR in Biomedicine* 24, 9 (2011), 1129–1136.
- [7] MILOVIC, C. FANSI-toolbox. <https://gitlab.com/cmilovic/FANSI-toolbox>. [Online; accedido el 7 de abril de 2020].
- [8] MILOVIC, C. *Fast and robust reconstructions for quantitative susceptibility MRI measurements*. PhD thesis, Pontificia Universidad Católica de Chile, School of Engineering, 2018.
- [9] ZHOU, D., LIU, T., SPINCEMAILLE, P., AND WANG, Y. Background field removal by solving the laplacian boundary value problem. *NMR in biomedicine* 27 (03 2014).