

La herramienta de mallado GMSH

20 de junio de 2010

Índice

1. Introducción	1
2. Instalación	2
3. Funcionalidades de GMSH	2
4. GMSH y Femsii	3
5. Módulo geométrico	3
5.1. Definir geometría con ficheros de texto	4
5.2. Archivos de otros programas	7
6. Módulo de mallado	7
6.1. La triangulación Delaunay	10
6.2. Meshadapt	11
6.3. Archivo de mallado *.msh	11
7. Módulo de pro-procesado	13
7.1. Rotacional (Curl)	13
7.2. Combinar distribuciones de vectores	13
7.3. HarmonicToTime	16
7.4. CutParametric	17
8. Guiones GMSH (scripts)	17
9. Conclusión	19

1. Introducción

GMSH es un software para generar el mallado de figuras bidimensionales y tridimensionales automáticamente.

Christophe Geuzaine y Jean-François son los creadores de este software. GMSH fue concebido como un software académico para resolver problemas

académicos. Sin embargo a lo largo de los años se ha mejorado lo suficiente para ser utilizado fuera de entornos universitarios.

Actualmente GMSH incluye módulos de pre-procesado (incluido diseño CAD), procesado (llamadas a resolutores externos, y pos-procesado (visualización de resultados, operaciones vectoriales, etc). Para el presente proyecto se utilizan principalmente los módulos de diseño, mallado, visualización y herramientas de calculo vectorial, diferencial y complejo.

En el presente documento nos concentraremos en las propiedades de mallado de GMSH, fundamental para obtener buenos resultados con el método de los elementos finitos. Posteriormente veremos algunas capacidades en el apartado de pos-procesado.

2. Instalación

Desde la web oficial de GMSH <http://geuz.org/gmsh/> se ofrecen ejecutables binarios para Windows, Mac y Linux (32 bit). Descargarlos y instalarlos directamente en la manera más sencilla de hacer funcionar GMSH en un equipo.

Alternativamente existe la posibilidad de compilar el código, que es ofrecido también desde la página oficial <http://geuz.org/gmsh/#Download>. Para poder compilarlo existen unas dependencias, los archivo de desarrollo de las bibliotecas gsl, fltk, jpeg, y el programa cmake.

En Linux basta configurar las opciones de compilación e instalación con cmake (las opciones se pueden alterar atendiendo a los escrito en los archivos README e INSTALL presente en el código fuente), para luego ejecutar “make” y por último “make install”.

Señalar que GMSH hace uso de las capacidades de aceleración de las tarjetas gráficas 3D, lo cual es conveniente tener instalados controladores que nos permitan hacer uso de estas capacidades, sobre todo en el caso de que se traten de visualizar figuras con gran número de elementos.

3. Funcionalidades de GMSH

Como se ha dicho, GMSH es un software que ha ido creciendo con el paso de los años. Aunque ha sido conocido principalmente por sus capacidades de diseño y mallado, muchas otras funcionalidades han sido incorporadas.

Por una parte existe el modulo de diseño para definir geometría, que nos permite dibujar visualmente una figura. Resulta intuitivo pero tiene capacidades limitadas. De hecho el modulo de geometría subyacente es muy extenso, y es posible crear archivos script (archivos de texto con lenguaje propio GMSH) donde definir paramétricamente figuras.

El módulo de mallado se apoya en las definiciones geométricas (incluidas la longitudes características) para crear mallados. Es posible hacer mallados en una, dos y tres dimensiones, creado elementos con formas varias (líneas, triángulos, tetraedros, prismas, hexaedros y pirámides). Además es posible definir

que tipo de algoritmo de mallado queremos.

El módulo de resolución “solver” implica la llamada a ciertos sistemas externos con los que interactuar. No se ha incluido ninguna herramienta de esta tipo internamente, pero en la documentación de GMSH se dan algunas instrucciones para programar un resolutor con el que interactuar.

El módulo de pos-procesado (ligado a la visualización) incluye gran número de operaciones diferenciales, vectoriales, herramientas para trabajar con números complejos y funcionalidades para extraer datos de las distribuciones de datos de entrada.

4. GMSH y Femsii

Femsii está pensado para utilizar Gmsh principalmente como mallador y como visualizador de resultados. Como mallador para crear archivos .msh que son los que acepta de entrada Femsii; como visualizador para representar los resultados, pues Femsii genera un archivo de campo de vectores que puede leer GMSH.

Aunque no es necesario, es posible realizar el diseño de las figuras en GMSH (tiene algunas limitaciones), con lo que crear el archivo de definición geométrica .geo . En cualquier caso este paso también es posible realizarlo directamente en un archivo de texto ASCII.

Por otra parte GMSH también ofrece extensiones con los que hacer cálculos vectoriales. El programa Femsii en sí trabaja con la magnitud potencial vector magnético, por lo que para obtener el flujo magnético es necesario aplicarle la operación rotacional. Este paso se puede hacer con una extensión de GMSH.

Por último mencionar que Femsii solo trabaja con modelos de simetría plana en dos dimensiones, por lo que por ahora los volúmenes quedan fuera de esta introducción a GMSH.

5. Modulo geométrico

El módulo geométrico permite dibujar puntos, líneas, superficies y volúmenes. Además permite seleccionar qué elementos están físicamente presentes en las figuras, puesto que en ocasiones es necesario definir elementos auxiliares sin sentido físico.

A partir de la definición de puntos en el espacio es posible ir construyendo líneas, círculos y eclipses. Y con estos elemento se pueden ir definiendo las superficies que constituyen la figura que buscamos. Una vez conseguida es muy recomendable definir los puntos, líneas y superficies que tienen sentido físico con “Fiscal Groups”, pues así indicaremos al mallador qué partes discretizar.

En la web oficial se incluyen algunos vídeo-tutoriales con los que familiarizarse en el uso de la aplicación gráfica.

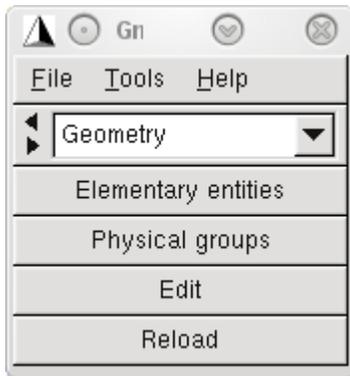


Figura 1: Módulo geométrico de GMSH

5.1. Definir geometría con ficheros de texto

La manera más potente de crear geometrías en GMSH es crear archivos de texto. Utilizando un lenguaje propio en ASCII se pueden crear figuras parametrizadas, o incluso incluir bucles.

Las funciones de GMSH son muchas y pueden ser encontradas todas en el manual oficial de GMSH. En esta sección nosotros vamos a mostrar un ejemplo.

Valga un ejemplo. Se va a mostrar ahora un cable coaxial definido en un archivo de ASCII parametrizado. Los radios r_1 , r_2 y r_3 que definen los límites de los conductores y el dieléctrico que los separa:

```

1 // Dibujo GMSH de un cable coaxial.
2 // Sistema formado por: conductor interior ,
   // dielectrico , conductor exterior y aire.
3
4 // centro del dibujo , punto auxiliar no fisico
5 Point(1) = {0, 0, 0};
6
7 // parametrizando
8 r1=1; // contorno exterior del primer conductor
9 r2=r1*2; // contorno exterior del dielectrico
10 r3=r1*3; // contorno exterior del segundo
   // conductor
11 r_dominio=r3*6; // contorno exterior (limite) del
   // dominio
12 lc1=0.4; // longitud carecteristica del mallado
   // de los puntos de la figura
13 lc2=lc1*10; //longitud carecteristica del mallado
   // de los puntos del limite de contorno
14
15

```

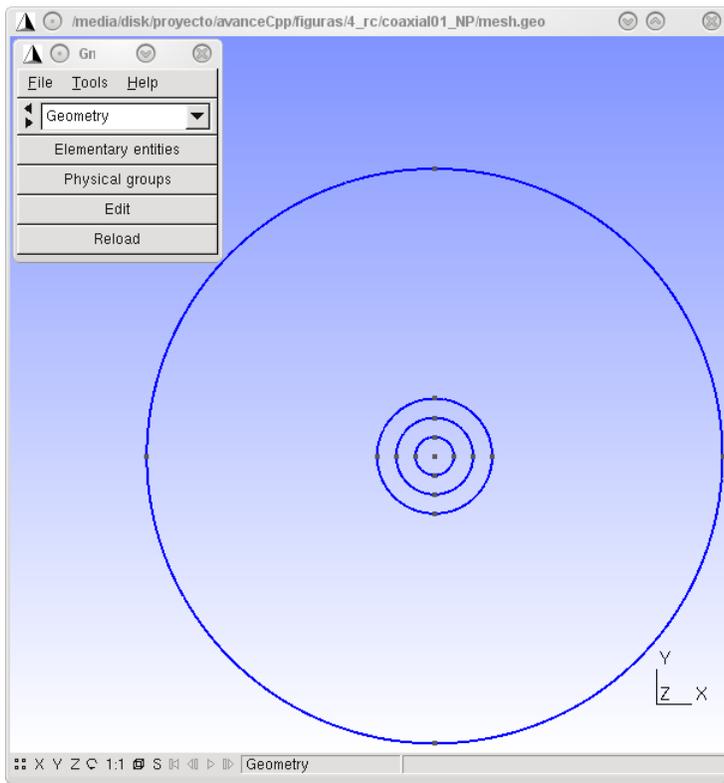


Figura 2: Geometría de un cable coaxial visualizada en GMSH

```

16 // definiendo circunferencia entre conductor
    interno y dielectrico
17 Point(2) = {r1, 0, 0};
18 Point(3) = {0, -r1, 0};
19 Point(4) = {-r1, 0, 0};
20 Point(5) = {0, r1, 0};
21 Circle(1) = {5, 1, 2};
22 Circle(2) = {2, 1, 3};
23 Circle(3) = {3, 1, 4};
24 Circle(4) = {4, 1, 5};
25
26 // definiendo circunferencia entre dielectrico y
    conductor externo
27 Point(6) = {r2, 0, 0};
28 Point(7) = {0, -r2, 0};
29 Point(8) = {-r2, 0, 0};
30 Point(9) = {0, r2, 0};
31 Circle(5) = {9, 1, 6};
32 Circle(6) = {6, 1, 7};
33 Circle(7) = {7, 1, 8};
34 Circle(8) = {8, 1, 9};
35
36 // definiendo circunferencia entre conductor
    externo y aire
37 Point(10) = {r3, 0, 0};
38 Point(11) = {0, -r3, 0};
39 Point(12) = {-r3, 0, 0};
40 Point(13) = {0, r3, 0};
41 Circle(9) = {13, 1, 10};
42 Circle(10) = {10, 1, 11};
43 Circle(11) = {11, 1, 12};
44 Circle(12) = {12, 1, 13};
45
46 // definiendo circunferencia del contorno
    dirichlet
47 Point(14) = {r_dominio, 0, 0};
48 Point(15) = {0, -r_dominio, 0};
49 Point(16) = {-r_dominio, 0, 0};
50 Point(17) = {0, r_dominio, 0};
51 Circle(13) = {17, 1, 14};
52 Circle(14) = {14, 1, 15};
53 Circle(15) = {15, 1, 16};
54 Circle(16) = {16, 1, 17};
55
56 // definiendo areas
57 Line Loop(17) = {2, 3, 4, 1};

```

```

58 Plane Surface(18) = {17}; // conductor interno
59 Line Loop(19) = {6, 7, 8, 5};
60 Plane Surface(20) = {17, 19}; // dielectrico
61 Line Loop(21) = {10, 11, 12, 9};
62 Plane Surface(22) = {19, 21}; // conductor
    externo
63 Line Loop(23) = {14, 15, 16, 13};
64 Plane Surface(24) = {21, 23}; // aire
65
66 // definiendo regiones de coherencia fisica
67 Physical Surface(25) = {18};
68 Physical Surface(26) = {20};
69 Physical Surface(27) = {22};
70 Physical Surface(28) = {24};
71 Physical Line(29) = {15, 16, 13, 14};
72
73 // asignando precision del mallado segun puntos
74 Characteristic Length {11, 7, 3, 5, 9, 13, 12, 8,
    4, 2, 6, 10} = lc1;
75 Characteristic Length {14, 15, 16, 17} = lc2;

```

Las longitudes características de la densidad de mallado también se pueden parametrizar, como se hace en el ejemplo para tener varias distintas para cada zona, pero todas relacionadas por una relación de multiplicidad.

En un archivo como el que se ha mostrado, realizar una figura variado el tamaño de algún elementos no requiere modificar todos los puntos, solo sería necesario cambiar los valores de los parámetros.

5.2. Archivos de otros programas

GMSH permite utilizar definiciones de geometría procedentes de otros programas, concretamente archivos con formato IGES, Step y BRep.

También es posible importar (mediante una herramienta externa) archivos dxf, pero en tal caso solo se asumirán puntos y líneas, así que será necesario definir superficies y elementos físicos dentro del propio GMSH.

6. Módulo de mallado

GMSH crea mallados pensados para poder trabajar con elementos finitos, con elementos punto, línea y triángulo (hay otros, pero estos son los que utiliza Femsii), por lo que en nuestro caso aplicaremos el mallado 2D a nuestras figuras.

En GMSH es posible modificar multitud de parámetros a la hora de crear un mallado. En la ventana de opciones, sección “Mesh” las encontraremos. Desde modificar el algoritmo de mallado hasta tratar de eliminar los nodos asociados a pocos elementos.

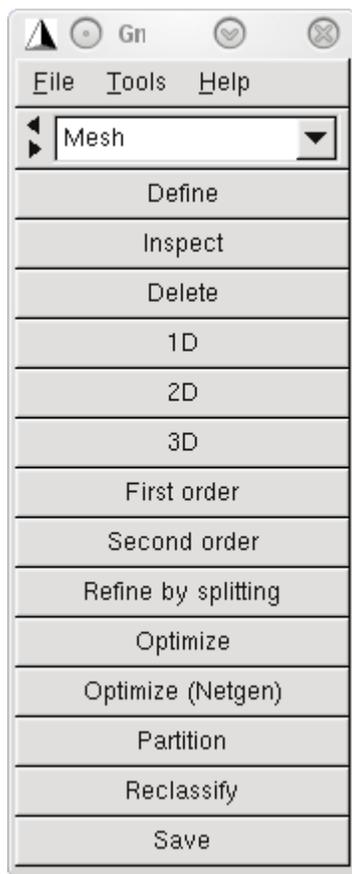


Figura 3: Módulo mallado de GMSH

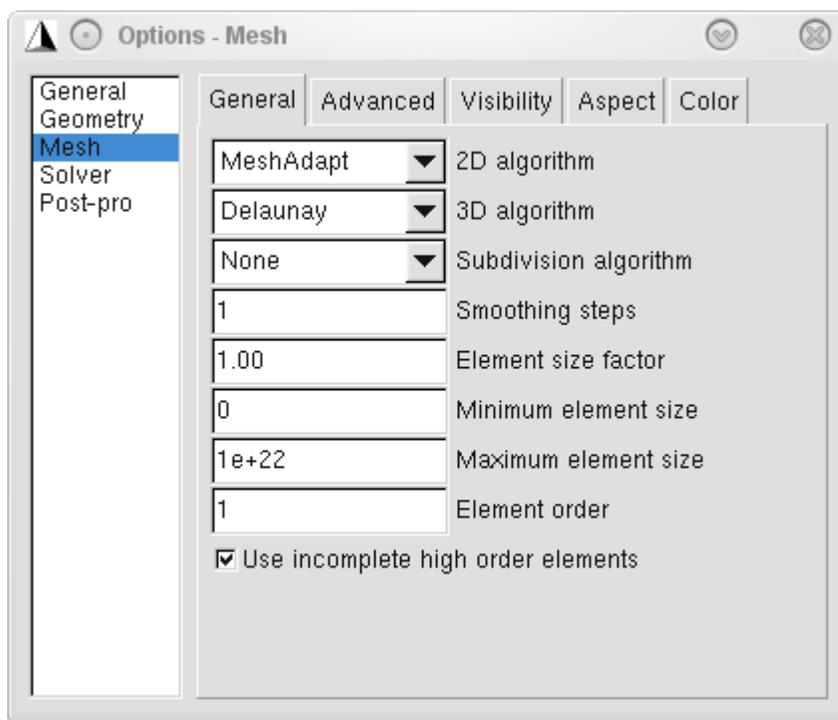


Figura 4: Opciones de mallado en GMSH

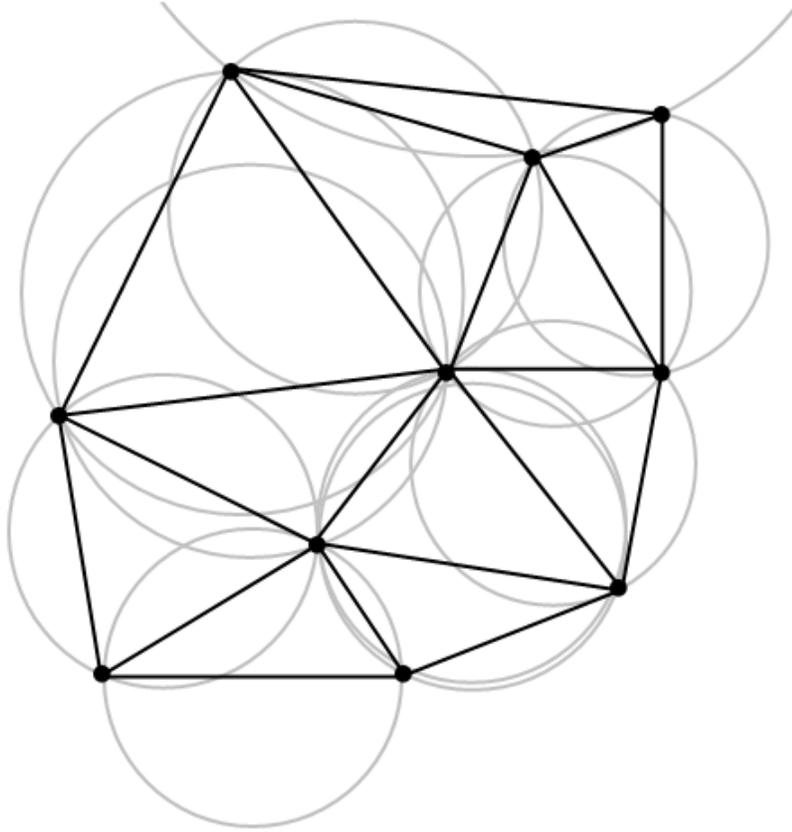


Figura 5: Triangulación Delaunay: Se puede ver que al comprobar las circunferencias circunscritas de todos los triángulos, éstas solo incluyen los nodos de propio elemento. Por tanto se dice que es una triangulación Delaunay.

En lo referente a los tipo de algoritmos de mallado, tradicionalmente GMSH ha utilizado el método Delaunay, aunque se ha ido añadiendo alguno más.

Todas las opciones de mallado de GMSH permiten el mallado adaptativo, lo quiere decir que es posible señalar zonas con mayor precisión de mallado, y zonas con elementos más grandes.

6.1. La triangulación Delaunay

El mallado principal que lleva a cabo el software GMSH utiliza el método postulado por el matemático ruso Boris Nikolaevich Delone, la triangulación Delaunay (1934). Tal método estipula que la circunferencia circunscrita de cada triángulo del mallado no debe contener ningún vértice de otro triángulo.

El triangulación Delaunay es un mallado no estructurado, está formado por

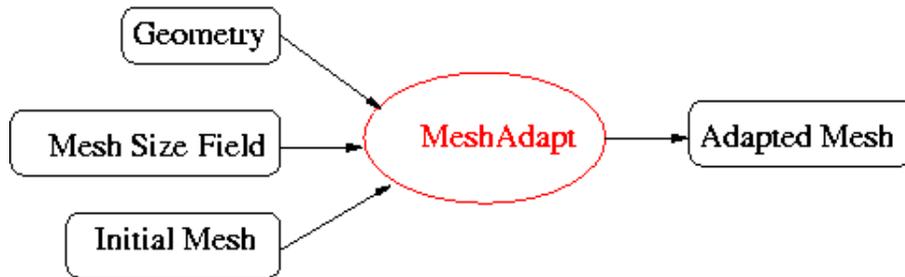


Figura 6: Diagrama básico de funcionamiento esquematizado por el desarrollador de Meshadapt.

triángulos o tetraedros y, en cualquier caso, no tiene un orden matricial. Este tipo de mallado es sencillo de ajustar a formas complejas, aunque es costoso de calcular.

La definición original de la triangulación Delaunay se postuló para dos dimensiones, aunque es posible aplicarla a espacios tridimensionales, utilizando tetraedros en tal caso.

Tras realizar una triangulación Delaunay obtendremos un mallado que maximiza los ángulos interiores de los triángulos que lo conforman, es decir, maximiza la extensión del ángulo más pequeño de la red. Por tanto, el ángulo mínimo dentro de todos los triángulos esta maximizado.

6.2. Meshadapt

GMSH se apoya también en el software de mallado Meshadapt, desarrollado en el instituto politécnico de Rensselaer, Nueva York. Este software acepta archivos de mallado, geometría y precisión de discretización para conseguir un nuevo mallado mejor adaptado.

El método consiste en aceptar un archivo de geometría donde se conocerán las líneas que definen una figura, un mallado inicial básico de esa figura (requiere un pre-mallado de baja calidad), y un mapa de tamaño de mallado según zonas. Esto último elemento consiste en una descripción de la precisión del mallado según zonas (normalmente más precisión en las zonas donde las variables en estudio tienen mayor tendencia a variar).

6.3. Archivo de mallado *.msh

Cuando se ha creado un mallado con GMSH, éste se guarda como archivo de tipo *.msh . Un archivo de texto tipo ASCII en que se enumeran los puntos y elementos de la discretización.

En el archivo se definirá primeramente la versión del programa, para luego comenzar a enumerar los nodos. Tras ello se enumeran los elementos, primero los elementos punto, luego los elementos línea y por último los elementos triángulo.

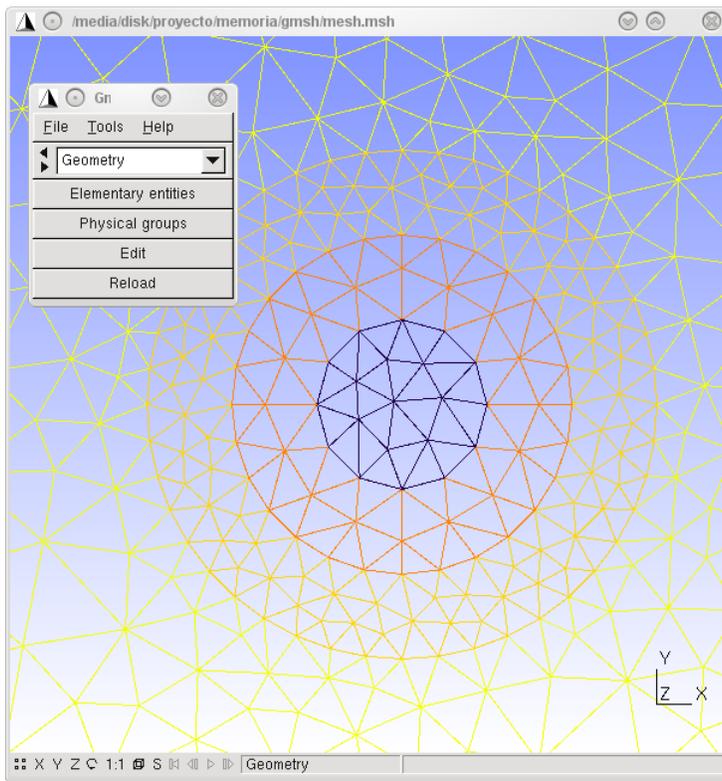


Figura 7: Mallado creado con GMSH.

Este tipo de archivo es el que asume directamente Femsii para crear el sistema de ecuaciones (una ecuación por nodo).

```
1 $MeshFormat
2 2.1 0 8
3 $EndMeshFormat
4 $Nodes
5 448
6 1 1 0 0
7 2 0 -1 0
8 ...
9 447 0.9947310172682052 -3.376678531900242 0
10 448 -0.9947310172682052 3.376678531900242 0
11 $EndNodes
12 $Elements
13 894
14 1 1 3 29 13 0 16 89
15 2 1 3 29 13 0 89 90
16 ...
17 29 2 3 25 18 0 3 115 22
18 30 2 3 25 18 0 19 20 114
19 ...
20 893 2 3 28 24 0 244 228 352
21 894 2 3 28 24 0 352 438 244
22 $EndElements
```

7. Módulo de pro-procesado

El módulo de pos-procesado de GMSH está asociado a los archivos de visualización, de hecho solo nos dará opciones en caso de tener abierto alguno.

Mencionaré tres funciones de pos-procesado útiles en el contexto de Femsii.

7.1. Rotacional (Curl)

El rotacional está incluido como extensión de GMSH (plugin), muy útil pues transforma nuestro campo vectorial de vector potencial magnético en un campo vectorial de flujo magnético. GMSH realiza esta operación internamente con cada elemento, que contiene tres valores (uno por nodo) de potencial vector magnético.

7.2. Combinar distribuciones de vectores

Es posible combinar dos campos de vectores en uno solo, internamente para GMSH serán dos campos en la misma vista. Esto es muy útil en la caso de

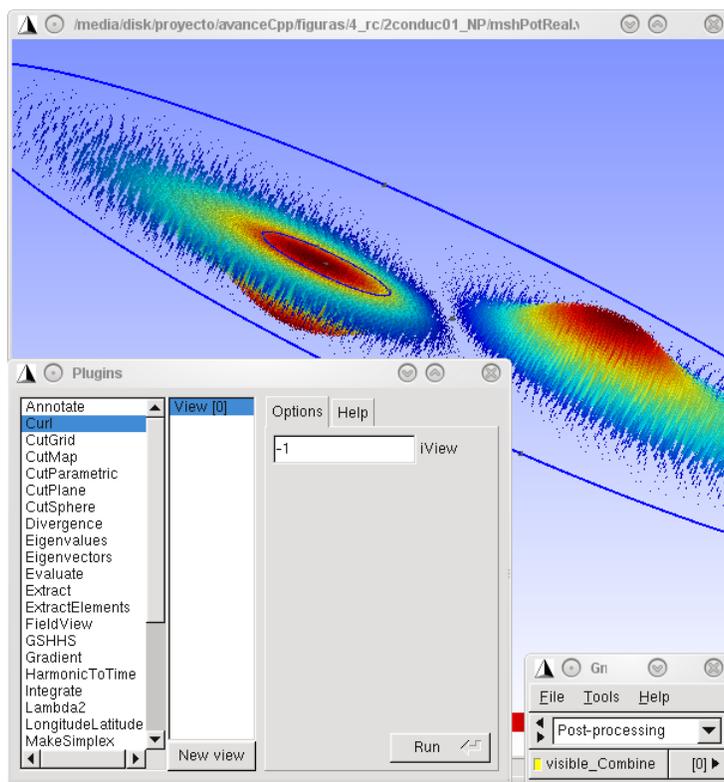


Figura 8: Se va a aplicar a este campo de vectores la operación rotacional.

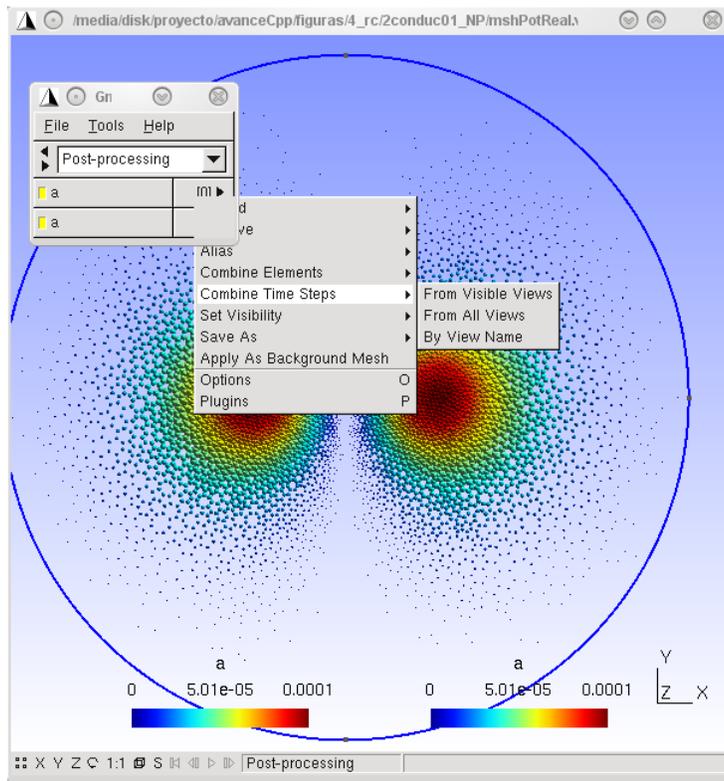


Figura 9: Dos archivos de visualización abiertos simultáneamente. A continuación se van a combinar.

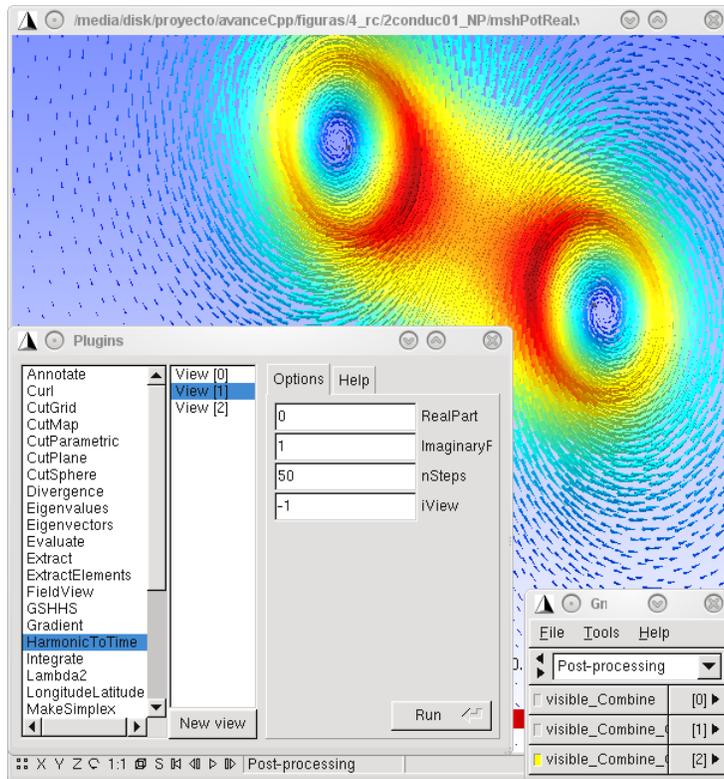


Figura 10: Aplicando HarmonicToTime a una campo de flujo magnético complejo.

que queramos trabajar con número complejos, una distribución vectorial para la parte real, y otra para la parte imaginaria.

Cada vez que se aplique otra función de pos-procesado, se aplicarán a todos los campos de una misma vista.

7.3. HarmonicToTime

“HamonicToTime” convierte un par de distribuciones de vectores (asimilados como armónicos con sus partes reales e imaginarias) al dominio del tiempo mediante la función:

$$t_n = (\text{parteReal}) \times \cos\rho - (\text{parteImaginaria}) \times \sen\rho$$

Donde

$$\rho = 2 \times \Pi \times n/nTotal$$

Para

$$n = 0 \longrightarrow n = nTotal$$

Donde nTotal es el numero total de pasos que se van a crear.

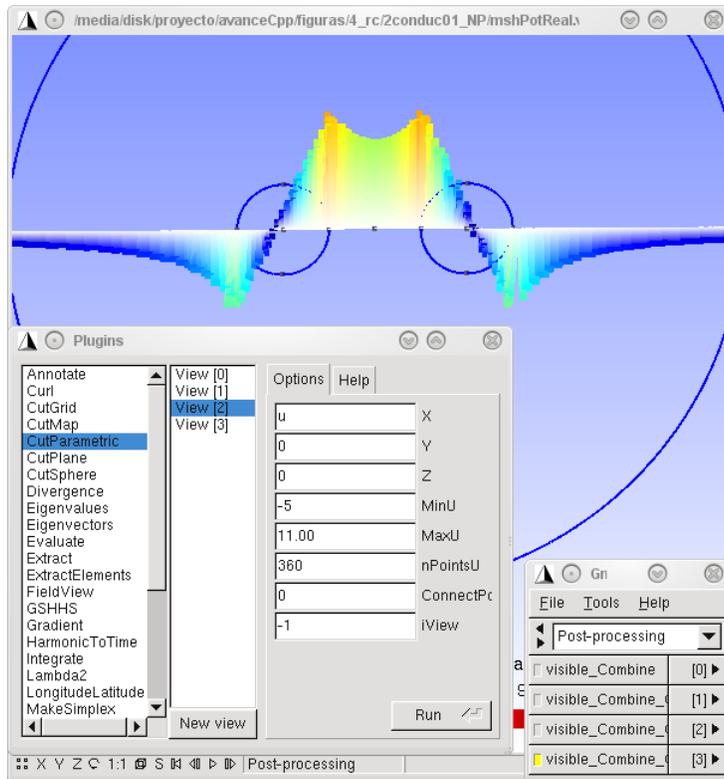


Figura 11: Tras aplicar CutParametric.

Una vez creada la nueva vista, se crearán tantas distribuciones de vectores como pasos totales se hayan creado. Podemos visualizar el movimiento completo pulsando en el botón avanzar de la parte baja de la ventana de GMSH.

7.4. CutParametric

CutParametric corta la vista actual por una función dada y representa los valores del campo de vectores en esa sección. Esta función debe definirse como dejando la variable “u” en función de: $(X(u), Y(u), Z(u))$. En el caso por ejemplo de querer simplemente una línea que corte una figura 2D por el origen se podría especificar $Y(u)=0$, $Z(u)=0$, $X(u)=u$.

8. Guiones GMSH (scripts)

El lenguaje ASCII propio de GMSH permite una gran versatilidad. Con el mismo código con el que se pueden crear figuras parametrizadas, también es

posible incluir bucles, que son muy útiles en el caso de querer dibujar objetos de estructura cíclica.

Es posible además incluir a otros archivos GMSH en cualquier punto, con lo que el código puede ser fragmentado para aumentar la simpleza de cada parte o para facilitar la reutilización de los proyectos ya desarrollados.

Además de poder incluir funciones típicas de diseño de figuras los guiones GMSH pueden ser utilizados para definir una serie de órdenes a funciones propias del programa. Entre las opciones en esta campo está la posibilidad de cargar archivos, redimensionar la escala de la visualización, hacer ejecutar una extensión concreta o incluso realizar una captura de la imagen mostrada en la ventana de GMSH.

Se muestran algunas de las opciones en el siguiente trozo de código GMSH:

```
1 Include "mesh.geo" ;
2 Include "mshPotReal.veg" ;
3 Plugin (Curl) . Run;
4 Delete View[0] ;
5 General.GraphicsWidth = 400;
6 General.GraphicsHeight = 600;
7 General.ScaleX = 3.6;
8 General.ScaleY = 3.6;
9 General.ScaleZ = 3.6;
10 Print Sprintf("render.jpg");
11 Exit;
```

Donde se incluyen las instrucciones:

Include Importa un archivo GMSH.

Plugin() Llama a un plugin concreto.

Delete Borra un elemento.

General.GraphicsWidth Escala la ventana a una altura dada.

General.GraphicsHeight Escala la ventana a una anchura dada.

General.ScaleX Escalar la figura en el eje X.

General.ScaleY Escalar la figura en el eje Y.

General.ScaleZ Escalar la figura en el eje Z.

Print Lanza la vista actual a un archivo.

Exit Cierra GMSH.

El código anterior capta una geometría y un mapa de vectores. Les aplica el rotacional y borra la vista original. Escala ventana, escala la figura y hace un captura de la vista que se muestra. Finaliza GMSH con "Exit".

9. Conclusión

Aunque son muchas las opciones de GMSH, en esta introducción nos hemos quedado en las más importantes para Femsii. El manual oficial <http://geuz.org/gmsh/doc/texinfo/gmsh.html> muestra el resto de opciones.

En cualquier caso GMSH formaba parte de los inicios del proyecto Femsii, y su visión generalista del mallado y la representación serán muy útiles en evoluciones de Femsii y otros programas relacionados con los elementos finitos.

Referencias

- [1] <http://www.scorec.rpi.edu/~xli/MeshAdapt.html>
- [2] C. Geuzaine and J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, Volume 79, Issue 11, pages 1309-1331, 2009
- [3] <http://www.ifor.math.ethz.ch/~fukuda/polyfaq/node30.html>
- [4] http://en.wikipedia.org/wiki/Delaunay_triangulation