

Femsii

1.0.0

Generado por Doxygen 1.5.9

Mon Jun 28 10:22:43 2010

Índice general

1. Índice de clases	1
1.1. Lista de clases	1
2. Índice de archivos	3
2.1. Lista de archivos	3
3. Documentación de las clases	5
3.1. Referencia de la Clase G2f	5
3.1.1. Descripción detallada	5
3.1.2. Documentación del constructor y destructor	6
3.1.2.1. G2f	6
3.1.3. Documentación de las funciones miembro	6
3.1.3.1. getOrder	6
3.1.4. Documentación de los datos miembro	6
3.1.4.1. fileData	6
3.1.4.2. fileEdge	6
3.1.4.3. fileEle	7
3.1.4.4. fileIn	7
3.1.4.5. fileNode	7
3.1.4.6. fileNodef	7
3.1.4.7. N_E	7
3.1.4.8. N_N	7
3.2. Referencia de la Clase Sistema_ecuaciones	8
3.2.1. Descripción detallada	9
3.2.2. Documentación del constructor y destructor	9
3.2.2.1. Sistema_ecuaciones	9
3.2.2.2. ~Sistema_ecuaciones	9
3.2.3. Documentación de las funciones miembro	9

3.2.3.1.	aBinario	9
3.2.3.2.	dirichletLin	9
3.2.3.3.	dirichletPto	10
3.2.3.4.	enrutar	10
3.2.3.5.	getStiffness	10
3.2.3.6.	kelvin	10
3.2.3.7.	local_matrix	10
3.2.3.8.	mkdir	10
3.2.3.9.	paresHomologos	10
3.2.3.10.	print_global	10
3.2.3.11.	regiones	11
3.2.3.12.	regNodos	11
3.2.3.13.	rightSide	11
3.2.4.	Documentación de los datos miembro	11
3.2.4.1.	conductividad	11
3.2.4.2.	frec	11
3.2.4.3.	listaRegNodos0	11
3.2.4.4.	listaRegNodos1	11
3.2.4.5.	N_E	11
3.2.4.6.	N_ED	11
3.2.4.7.	N_N	12
3.2.4.8.	permeabilidad	12
3.2.4.9.	pNode	12
3.2.4.10.	ptrConducti	12
3.2.4.11.	ptrDensidadCorri	12
3.2.4.12.	ptrDirichlet	12
3.2.4.13.	ptrExtra	12
3.2.4.14.	ptrKelvin	12
3.2.4.15.	ptrMesh	12
3.2.4.16.	ptrOpEjecu	12
3.2.4.17.	ptrPermeabi	13
3.2.4.18.	ptrRutaArchivo	13
3.2.4.19.	ptrTemporal	13
3.2.4.20.	ruta	13
3.2.4.21.	VACIO	13
4.	Documentación de archivos	15

4.1. Referencia del Archivo femsii_pos.cpp	15
4.1.1. Descripción detallada	16
4.1.2. Documentación de las funciones	16
4.1.2.1. archivoGmshBxyEle	16
4.1.2.2. archivoGmshPotEle	16
4.1.2.3. enrutar	16
4.2. Referencia del Archivo femsii_solve.cpp	17
4.2.1. Descripción detallada	17

Capítulo 1

Índice de clases

1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

G2f (Clase principal G2f)	5
Sistema_ecuaciones (Clase principal Sistema_ecuaciones)	8

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

femsii_pos.cpp (Este programa acepta (obligatorios) dos parametros de entrada: -f Archivo binario que contiene el vector solucion generado por solve.bin. -r Ruta donde seran escritos los archivos generados por pos.bin)	15
femsii_solve.cpp (Este programa acepta un sistema de ecuaciones, lo resuelve segun las indicaciones que se le dan, y escribe un ventor de resultados)	17
g2f.h	??
P_sistemaEc.h	??

Capítulo 3

Documentación de las clases

3.1. Referencia de la Clase G2f

Clase principal [G2f](#).

```
#include <g2f.h>
```

Métodos públicos

- [G2f](#) (const char *, const char *)
Constructor de [G2f](#).
- void [getOrder](#) (short unsigned=16)

Atributos privados

- ifstream [fileIn](#)
- ofstream [fileNode](#)
- ofstream [fileEdge](#)
- ofstream [fileEle](#)
- ofstream [fileNodef](#)
- ofstream [fileData](#)
- unsigned [N_N](#)
- unsigned [N_E](#)

3.1.1. Descripción detallada

Clase principal [G2f](#).

La clase [G2f](#) se encarga de asumir un archivo de mallado Gmsh y extraer a partir de el archivos especificos para nodos, archivos de elementos interiores (3 nodos), archivo de elementos contorno (2 nodos) y un archivo extra con la cantidad de lineas de cada archivo. [G2f](#) es llamado por P_sistemaEc pues los archivos que genera [G2f](#) los utiliza directamente para crear la matriz de rigidez y el vector de terminos independientes.

NOTA suplementaria: Gmsh siempre genera nodos consecutivos unos de otros, esto es un problema a la hora de generar un sistema de ecuaciones. Por ellos existe [G2f](#).

NOTA procedimiento general: -Se empieza a leer el archivo generado por gmsh. -Se ignoran los primeros 34 caracteres. -Se coloca en la variable `N_N` el numero de nodos. OJO, los nodos pueden estar desordenados.

-Con un bucle "for" se leen tantas lineas como nodos existan. Se lee del archivo origen el INDICE, coordenada X, coordenada Y y la ultima columna que se ignorara. Se escribe en el archivo destino el valor cardinal "i" del bucle, y los datos coordenada X, coordenada Y, y fin de linea. Se crea la variable tipo MAP para almacena pares (viejo nodo, nuevo nodo) -Se ignoran los siguientes 18 caracteres. -Se coloca en la variable `N_E` el numero de elementos. OJO, puesto que se han rebautizado los nodos con un nuevo indice, habra que cambiar los nodos asociados a cada elemento. -Con un bucle for se barren todos los elementos. Se lee el indice y el tipo de elemento. Segun el caso (mediante un switch) se trata cada elemento. En cada caso se leen los nodos y los atributos que se desean conseguir, el resto se pasa a ignore. En cada caso se escriben los datos en el archivo destino, respetando el indice original. Mediante la relacion map "newNode" se colocan los nuevos numeros de nodos que forman cada elemento. En las variables `N_nf`, `N_edg`, `N_ele` se van contando el numero de elementos de cada tipo.

-Se termina cada archivo con un "end" y se escribe el sumario en otro archivo.

3.1.2. Documentación del constructor y destructor

3.1.2.1. `G2f::G2f (const char * ruta, const char * temporal)`

Constructor de [G2f](#).

El CONSTRUCTOR se encarga de crear todas las cadenas de caracteres que apuntan a los archivos que creara esta clase.

3.1.3. Documentación de las funciones miembro

3.1.3.1. `void G2f::getOrder (short unsigned precision = 16)`

[getOrder\(\)](#) representa la funcion principal de [G2f](#), y es la que se encarga de leer el archivo .msh y extraer y separar en distintos archivos los datos necesarios para Femsii. Segun va leyendo el archivo va mallado, va escribiendo en el archivo adecuando asi que no necesita reservar memoria. Al final de cada archivo incluye una "F" de fin. Ademas de un archivo extra con el tamanhio de cada archivo generado. Si posteriormente el programa que lea los archivos encuentra que tras completar el numero de interacciones marcadas por el archivo extra, no se encuentra una F, el archivo estara corrupto.

3.1.4. Documentación de los datos miembro

3.1.4.1. `ofstream G2f::fileData [private]`

Variable tipo flujo de salida utilizada para trabajar con el archivo que recuenta el numero de lineas de cada uno de los otros archivos generados por [G2f](#).

3.1.4.2. `ofstream G2f::fileEdge [private]`

Variable tipo flujo de salida utilizada para trabajar con el archivo de elementos de contorno.

3.1.4.3. ofstream G2f::fileEle [private]

Variable tipo flujo de salida utilizada para trabajar con el archivo de elementos interiores.

3.1.4.4. ifstream G2f::fileIn [private]

Variable tipo flujo de entrada utilizada para trabajar con el archivo de mallado generado por Gmsh.

3.1.4.5. ofstream G2f::fileNode [private]

Variable tipo flujo de salida utilizada para trabajar con el archivo de nodos.

3.1.4.6. ofstream G2f::fileNodef [private]

Variable tipo flujo de salida utilizada para trabajar con el archivo de elementos punto (finalmente no se ha utilizado en Femsii).

3.1.4.7. unsigned G2f::N_E [private]

variable que se encarga de almacenar el numero de elementos.

3.1.4.8. unsigned G2f::N_N [private]

variable que se encarga de almacenar el numero de nodos.

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- g2f.h
- g2f.cpp

3.2. Referencia de la Clase Sistema_ecuaciones

Clase principal [Sistema_ecuaciones](#).

```
#include <P_sistemaEc.h>
```

Métodos públicos

- [Sistema_ecuaciones](#) (const char *)
Constructor de [Sistema_ecuaciones](#) .
- [~Sistema_ecuaciones](#) ()
Destructor de [Sistema_ecuaciones](#) .
- int [getStiffness](#) (int=0)
- void [print_global](#) ()
- int [rightSide](#) ()
- int [dirichletPto](#) ()
- int [dirichletLin](#) ()
- int [paresHomologos](#) ()
- int [kelvin](#) ()
- int [aBinario](#) ()
- int [mkdir](#) (const char *)
- int [enrutar](#) (const char *)

Métodos privados

- int [local_matrix](#) (double, double, double, double, double, double, int)
- int [regiones](#) ()
- int [regNodos](#) (int **, int, int)

Atributos privados

- int [N_N](#)
- int [N_E](#)
- int [N_ED](#)
- float [frec](#)
- double [VACIO](#)
- double ** [pNode](#)
- const char * [ruta](#)
- char * [ptrMesh](#)
- char * [ptrTemporal](#)
- char * [ptrConducti](#)
- char * [ptrPermeabi](#)
- char * [ptrDensidadCorri](#)
- char * [ptrDirichlet](#)
- char * [ptrKelvin](#)
- char * [ptrExtra](#)
- char * [ptrOpEjecu](#)

- char * [ptrRutaArchivo](#)
- map< unsigned, double > [permeabilidad](#)
- map< unsigned, double > [conductividad](#)
- list< int > [listaRegNodos0](#)
- list< int > [listaRegNodos1](#)

3.2.1. Descripción detallada

Clase principal [Sistema_ecuaciones](#).

La clase [Sistema_ecuaciones](#) contiene todas las funciones y variables que se utilizan para el programa pre.bin.

3.2.2. Documentación del constructor y destructor

3.2.2.1. Sistema_ecuaciones::Sistema_ecuaciones (const char * *entrada*)

Constructor de [Sistema_ecuaciones](#) .

El CONSTRUCTOR crea las cadenas de caracteres (ptrMesh, ptrConducti, etc) con las direcciones de los diferentes archivos de trabajo. Ademas se llama a la funcion getOrder() de de la clase [G2f](#) para crear archivos ordenados con los datos del archivo GMSH. Por ultimo se llama a la funcion regiones, que crear variables de tipo map globales con valores de permeabilidad y conductividad.

3.2.2.2. Sistema_ecuaciones::~~Sistema_ecuaciones ()

Destructor de [Sistema_ecuaciones](#) .

El destructor de [Sistema_ecuaciones](#) se encarga de eliminar de la memoria la matriz principal de rigidez y el vector de terminos independientes. Ademas lanza la orden de finalizar PetscFinalize().

3.2.3. Documentación de las funciones miembro

3.2.3.1. int Sistema_ecuaciones::aBinario ()

aBINARIO() crea un archivo binario especifico PETSC con el contenido de una matriz, respetando la estructura y el tipo. Asi mismo tambien se crea un binario con el contenido del vector de terminos independientes. Estos archivos se crean para ser leidos posteriormente por solve.bin .

3.2.3.2. int Sistema_ecuaciones::dirichletLin ()

DIRICHLET-LIN() aplica dirichlet en Pm_global a patir de valores en lineas. Esta funcion debe realizar un barrido sobre el archivo dirichlet.lin para encontrar las regiones donde existe de antemano un valor dado del potencial vector magnetico. Para ello busca los nodos de las regiones en el archivo g2f.edge, donde se encuentran los elementos de tipo linea. Una vez encontrados los nodos asociados fuerza al sistema de ecuaciones (Pm_global y Pv_righside) para asignar el valor adecuado al nodo correspondiente.

3.2.3.3. `int Sistema_ecuaciones::dirichletPto ()`

DIRICHLET-PTO() aplica dirichlet en Pm_global a partir de valores en puntos. ESTA funcion da problemas de rendimiento. Por ahora se omite el uso de Dirichlet sobre una linea. CUIDADO: funcion prototipo con vistas a ser implementada en un futuro proyecto. No implementada en Femsii 1.0

3.2.3.4. `int Sistema_ecuaciones::enrutar (const char * archivo)`

ENRUTAR() es una funcion auxiliar que coloca en ptrRutaArchivo la suma de las cadenas de caracteres ruta y *archivo.

3.2.3.5. `int Sistema_ecuaciones::getStiffness (int verb = 0)`

La funcion GET-STIFFNESS() utiliza los archivos GMSH ordenados por getOrder() para calcular la matriz global del sistema, tanto parte real como parte imaginaria. Para ello utiliza un bucle que llama a la funcion [local_matrix\(\)](#) que genera la matriz local de cada elemento, que luego es ensamblada en la matriz global. La funcion [local_matrix\(\)](#) necesitara las coordenadas de los nodos que forman el elemento, asi como la region a la que pertenece el elemento.

3.2.3.6. `int Sistema_ecuaciones::kelvin ()`

Funcion para implementar las condiciones kelvin en Femsii atendiendo a la salida de [paresHomologos\(\)](#). Cuidado, funcion prototipo EN DESARROLLO, no incluido en Femsii 1.0

3.2.3.7. `int Sistema_ecuaciones::local_matrix (double xi, double yi, double xj, double yj, double xk, double yk, int region) [private]`

La funcion LOCAL_MATRIX() es mayormente el trozo de algoritmo que cambia al cambiar el campo de aplicacion del programa (magnetismo, distribucion termica, mecanica). En esta funcion está reflejada en gran medida la funcion fisica que representa la base del programa Femsii. Se encarga de calcular la matriz local de cada elemento, aceptando la posicion de cada uno de los tres puntos, y propiedades aplicables a cada elemento/region. En nuestro caso la permeabilidad magnetica.

3.2.3.8. `int Sistema_ecuaciones::mkdir (const char * ptrTemporal)`

MKDIR() crea la carpeta que viene en la cadena *ptrTemporal y notifica si lo consigue.

3.2.3.9. `int Sistema_ecuaciones::paresHomologos ()`

PARES-HOMOLOGOS() Se trata de leer un archivo en el que hay un listado de regiones homologas. Con ello se debe crear un puntero doble con los nodos relacionados. El procedimiento es leer el archivo que relaciona las regiones y buscar en el archivo g2f.edge los nodos que pertenecen a esa region. Cuidado, funcion prototipo EN DESARROLLO, no incluido en Femsii 1.0

3.2.3.10. `void Sistema_ecuaciones::print_global ()`

PRINT_GLOBAL() muestra en la salida estandar la matriz de rigidez global.

3.2.3.11. int Sistema_ecuaciones::regiones () [private]

REGIONES() Se trata de que esta funcion llene las variables de tipo map "permeabilidad" y "conductividad" con los valores de conductividad y permeabilidad en cada region que da el usuario. Necesario para la funcion [local_matrix\(\)](#).

3.2.3.12. int Sistema_ecuaciones::regNodos (int ** ptrNodoEdge, int region, int homo) [private]

REG-NODOS() es llamado varias veces por [paresHomologos\(\)](#) busca los nodos de que pertenezcan a una region y los coloca en listaRegNodos0 en caso de que la llamada sea para los nodos de la region origen, y listaRegNodos1 si los nodos son de la region homologa. Cuidado, funcion prototipo EN DESARROLLO, no incluido en Femsii 1.0

3.2.3.13. int Sistema_ecuaciones::rightSide ()

RIGHT-SIDE() genera Pv_rightside a partir del archivo de densidades de corriente. En el archivo de densidades de corriente se nos proporcionara una lista de regiones y densidades correspondientes. Mediante el archivo g2f.ele se obtienen los nodos asociados a cada region. Tener en cuenta que el archivo de densidades de corriente incluye siempre parte real y parte imaginaria para cada region.

3.2.4. Documentación de los datos miembro**3.2.4.1. map<unsigned,double> Sistema_ecuaciones::conductividad [private]**

Variable tipo mapa para almacenar pares region-conductividad. Lo utiliza [local_matrix\(\)](#)

3.2.4.2. float Sistema_ecuaciones::frec [private]

variable que se encarga de almacenar la frecuencia de la densidad de corriente.

3.2.4.3. list<int> Sistema_ecuaciones::listaRegNodos0 [private]

variable lista para almacenar la primera columna de los pares de nodos homologos. (en desarrollo)

3.2.4.4. list<int> Sistema_ecuaciones::listaRegNodos1 [private]

variable lista para almacenar la segunda columna de los pares de nodos homologos. (en desarrollo)

3.2.4.5. int Sistema_ecuaciones::N_E [private]

variable que se encarga de almacenar el numero de elementos.

3.2.4.6. int Sistema_ecuaciones::N_ED [private]

variable que se encarga de almacenar el numero de elementos de contorno.

3.2.4.7. int Sistema_ecuaciones::N_N [private]

variable que se encarga de almacenar el numero de nodos.

3.2.4.8. map<unsigned,double> Sistema_ecuaciones::permeabilidad [private]

Variable tipo mapa para almacenar pares region-permeabilidad. Lo utiliza [local_matrix\(\)](#).

3.2.4.9. double Sistema_ecuaciones::pNode [private]**

variable que almacena los nodos del mallado, la utilizan [getStiffness\(\)](#), [regiones\(\)](#) y [archivoGmshPot\(\)](#).

3.2.4.10. char * Sistema_ecuaciones::ptrConducti [private]

Cadena de caracteres que almacenara la direccion del archivo que define las conductividades conductividad.reg .

3.2.4.11. char * Sistema_ecuaciones::ptrDensidadCorri [private]

Cadena de caracteres que almacenara la direccion del archivo que define las densidades de corriente desidadCorri.reg .

3.2.4.12. char * Sistema_ecuaciones::ptrDirichlet [private]

Cadena de caracteres que almacenara la direccion del archivo que define las condiciones dirichlet dirichlet.lin (especificar otro para dirichlet.pto).

3.2.4.13. char * Sistema_ecuaciones::ptrExtra [private]

Cadena de caracteres que almacenara la direccion del archivo que define la frecuencia de la alimetnacion alterna, y el valor de la permeabilidad en el vacio que se va a utilizar.

3.2.4.14. char * Sistema_ecuaciones::ptrKelvin [private]

Cadena de caracteres que almacenara la direccion del archivo que define las regiones homologas en caso de utilizar la transformacion de kelvin.

3.2.4.15. char* Sistema_ecuaciones::ptrMesh [private]

Cadena de caracteres que almacenara la direccion del archivo de mallado mesh.msh.

3.2.4.16. char * Sistema_ecuaciones::ptrOpEjecu [private]

Cadena de caracteres que almacenara la direccion del archivo que carga las opciones de ejecucion del resolutor de PETSc (ya no corresponde a pre.bin).

3.2.4.17. char * Sistema_ecuaciones::ptrPermeabi [private]

Cadena de caracteres que almacenara la direccion del archivo que define las permeabilidades permeabilidad.reg .

3.2.4.18. char * Sistema_ecuaciones::ptrRutaArchivo [private]

Cadena de caracteres comodin que utiliza la funcion [enrutar\(\)](#) para escribir archivos en determinados directorios.

3.2.4.19. char * Sistema_ecuaciones::ptrTemporal [private]

Cadena de caracteres que almacena la direccion de la carpeta temporal que se creara en la carpeta de proyecto.

3.2.4.20. const char* Sistema_ecuaciones::ruta [private]

En esta variable se almacena el parametro introducido al ejecutar pre.bin, que debe ser la direccion de la carpeta de proyecto.

3.2.4.21. double Sistema_ecuaciones::VACIO [private]

variable que se encarga de almacenar el valor de la permeabilidad magnetica en el vacio.

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- P_sistemaEc.h
- P_sistemaEc.cpp

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo femsii_pos.cpp

Este programa acepta (obligatorios) dos parametros de entrada: -f Archivo binario que contiene el vector solucion generado por solve.bin. -r Ruta donde seran escritos los archivos generados por pos.bin.

```
#include <fstream>
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include "petscksp.h"
```

Definiciones

- #define **PI** 3.14159265358979323846264338327950288419716939937510

Funciones

- int [archivoGmshPotEle](#) ()
- int [archivoGmshBxyEle](#) ()
- int [enrutar](#) (const char *)
- int **main** (int argc, char **args)

Variables

- Vec **Pv_solution**
- int **N_N**
- int **N_E**
- double ** **pNode**
- const char * **ruta**
- char * **ptrRutaArchivo**

4.1.1. Descripción detallada

Este programa acepta (obligatorios) dos parametros de entrada: -f Archivo binario que contiene el vector solucion generado por solve.bin. -r Ruta donde seran escritos los archivos generados por pos.bin.

DESCRIPCION DE `main(int argc, char **args)`: Asume los parametros de entrada. Busca los archivos temporados para rellenar las variables `N_N` (numero de nodos), `N_E` (numero de elementos) y `pNode` (vector doble para pares de nodos). Luego lanza las funciones `archivoGmshPotEle()` y `archivoGmshBxyEle()`.

DESCRIPCION DE `archivoGmshPotEle()`: Crea los archivos de potencial vector magnetico para Gmsh: `mshPotReal.veg` y `mshPotImag.veg`

DESCRIPCION DE `archivoGmshBxyEle()`: Realiza la operacion rotacional de forma numerica (para cada elemento) y crea los archivos de flujo magnetico para Gmsh: `mshBxyReal.veg` y `mshBxyImag.veg`

4.1.2. Documentación de las funciones

4.1.2.1. `int archivoGmshBxyEle ()`

`ARCHIVO-GMSH-Bxy-ELE()` crea en la carpeta "ruta" un archivo de resultados GMSH con los valores de induccion magnetica resultantes por elemento. Utilizando el formato VT. Es necesario utilizar el formato VT para que GSMH pueda realizar operaciones vectoriales.

4.1.2.2. `int archivoGmshPotEle ()`

`ARCHIVO-GMSH-POT-ELE()` crea en la carpeta "ruta" un archivo de resultados GMSH con los valores de potencial magnetico resultantes por elemento. Utilizando el formato VT. Es necesario utilizar el formato VT para que GSMH pueda hacer el rotacional.

4.1.2.3. `int enrutar (const char * archivo)`

`ENRUTAR()` coloca en `ptrRutaArchivo` la suma de `*ruta` y `*archivo`.

4.2. Referencia del Archivo femsii_solve.cpp

Este programa acepta un sistema de ecuaciones, lo resuelve segun las indicaciones que se le dan, y escribe un vector de resultados.

```
#include "petscksp.h"
```

Definiciones

- `#define __FUNCT__ "main"`

Funciones

- `int main (int argc, char **args)`

4.2.1. Descripción detallada

Este programa acepta un sistema de ecuaciones, lo resuelve segun las indicaciones que se le dan, y escribe un vector de resultados.

Los archivos de entrada son 4: -f0 -> opciones de ejecucion, aplicables a la resolucion del sistema. -f1 -> matriz de coeficientes. -f2 -> vector de terminos independientes. -f3 -> archivo de salida para femsii_x con la resolucion del sistema.

DESCRIPCION DE `main(int argc, char **args)`: Se encarga de entender los parametros de entrada, rescatar la matriz de rigidez y el vector de terminos independientes, y resolver el sistema de ecuaciones. `solve()` lee el archivo `opciones.ejecu` para cargar las opciones con las que se lanzara el resolutor de subespacios de Krilov. Tras crear un vector de resultados lo escribe en la carpeta temporal del proyecto. La estructura de la funcion y de el codigo de este archivo en general ha sido creado pensando en crear un codigo siguiendo las directrices generales de la documentacion de PETSc para el caso de sistemas paralelizables. De esta manera `solbe.bin` puede ser ejecutado en varios procesos concurrentes.

Índice alfabético

- ~Sistema_ecuaciones
 - Sistema_ecuaciones, 9
- aBinario
 - Sistema_ecuaciones, 9
- archivoGmshBxyEle
 - femsii_pos.cpp, 16
- archivoGmshPotEle
 - femsii_pos.cpp, 16
- conductividad
 - Sistema_ecuaciones, 11
- dirichletLin
 - Sistema_ecuaciones, 9
- dirichletPto
 - Sistema_ecuaciones, 9
- enrutar
 - femsii_pos.cpp, 16
 - Sistema_ecuaciones, 10
- femsii_pos.cpp, 15
 - archivoGmshBxyEle, 16
 - archivoGmshPotEle, 16
 - enrutar, 16
- femsii_solve.cpp, 17
- fileData
 - G2f, 6
- fileEdge
 - G2f, 6
- fileEle
 - G2f, 6
- fileIn
 - G2f, 7
- fileNode
 - G2f, 7
- fileNodef
 - G2f, 7
- frec
 - Sistema_ecuaciones, 11
- G2f, 5
 - fileData, 6
 - fileEdge, 6
 - fileEle, 6
 - fileIn, 7
 - fileNode, 7
 - fileNodef, 7
 - G2f, 6
 - getOrder, 6
 - N_E, 7
 - N_N, 7
- getOrder
 - G2f, 6
- getStiffness
 - Sistema_ecuaciones, 10
- kelvin
 - Sistema_ecuaciones, 10
- listaRegNodos0
 - Sistema_ecuaciones, 11
- listaRegNodos1
 - Sistema_ecuaciones, 11
- local_matrix
 - Sistema_ecuaciones, 10
- mkdir
 - Sistema_ecuaciones, 10
- N_E
 - G2f, 7
 - Sistema_ecuaciones, 11
- N_ED
 - Sistema_ecuaciones, 11
- N_N
 - G2f, 7
 - Sistema_ecuaciones, 11
- paresHomologos
 - Sistema_ecuaciones, 10
- permeabilidad
 - Sistema_ecuaciones, 12
- pNode
 - Sistema_ecuaciones, 12
- print_global
 - Sistema_ecuaciones, 10
- ptrConducti
 - Sistema_ecuaciones, 12

ptrDensidadCorri
 Sistema_ecuaciones, 12
 ptrDirichlet
 Sistema_ecuaciones, 12
 ptrExtra
 Sistema_ecuaciones, 12
 ptrKelvin
 Sistema_ecuaciones, 12
 ptrMesh
 Sistema_ecuaciones, 12
 ptrOpEjecu
 Sistema_ecuaciones, 12
 ptrPermeabi
 Sistema_ecuaciones, 12
 ptrRutaArchivo
 Sistema_ecuaciones, 13
 ptrTemporal
 Sistema_ecuaciones, 13

regiones
 Sistema_ecuaciones, 10
 regNodos
 Sistema_ecuaciones, 11
 rightSide
 Sistema_ecuaciones, 11
 ruta
 Sistema_ecuaciones, 13

Sistema_ecuaciones, 8
 ~Sistema_ecuaciones, 9
 aBinario, 9
 conductividad, 11
 dirichletLin, 9
 dirichletPto, 9
 enrutar, 10
 frec, 11
 getStiffness, 10
 kelvin, 10
 listaRegNodos0, 11
 listaRegNodos1, 11
 local_matrix, 10
 mkdir, 10
 N_E, 11
 N_ED, 11
 N_N, 11
 paresHomologos, 10
 permeabilidad, 12
 pNode, 12
 print_global, 10
 ptrConducti, 12
 ptrDensidadCorri, 12
 ptrDirichlet, 12
 ptrExtra, 12
 ptrKelvin, 12

ptrMesh, 12
 ptrOpEjecu, 12
 ptrPermeabi, 12
 ptrRutaArchivo, 13
 ptrTemporal, 13
 regiones, 10
 regNodos, 11
 rightSide, 11
 ruta, 13
 Sistema_ecuaciones, 9
 Sistema_ecuaciones, 9
 VACIO, 13

VACIO
 Sistema_ecuaciones, 13