



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE COMPUTACIÓN

HERRAMIENTA PARA LA SOLUCIÓN DE EDP SOBRE DOMINIOS
GENERALES EN 2-D MEDIANTE MÉTODOS ADAPTIVOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

EDUARDO MERCADER ORTA

PROFESOR GUÍA:
MARÍA CECILIA RIVARA ZUÑIGA

MIEMBROS DE LA COMISIÓN :
NANCY HITSCHFELD KAHLER
LUIS MATEU BRULÉ

Este trabajo ha sido parcialmente financiado por FONDECYT (1911220)

SANTIAGO DE CHILE
ABRIL, 2012

Resumen

En este trabajo se discute un sistema experimental para resolver ecuaciones diferenciales parciales (EDP) elípticas o parabólicas sobre dominios en 2-D mediante el método de elementos finitos (MEF). La herramienta combina algoritmos de refinamiento y desrefinamiento de triangulaciones conformes sobre dominios generales con bordes curvos en las fronteras e interfaces de medios; el MEF, usando elementos lineales; una estructura de datos adecuada; el uso de estimadores del error cometido en la resolución numérica; una interfaz gráfica sobre XWindows y un lenguaje declarativo para la definición de los problemas.

La herramienta fue desarrollada en forma modular, para permitir la incorporación de nuevas opciones, como elementos de grado mayor en el uso del MEF, y utiliza librerías externas, como son, “Sparse” y “SuperLU” para la resolución de los sistemas de ecuaciones poco densos, “pdraw” para visualización de resultados en 3D y “GLADE” para construcción de la interfaz gráfica; todas estas herramientas corresponden a software de código abierto.

La herramienta tiene un uso potencial en una amplia gama de aplicaciones, por ejemplo en cálculo estructural, y mecánica de fluidos y permite al usuario manejar en forma flexible la adaptabilidad, pudiendo definir o modificar a través de la interfaz gráfica, las triangulaciones, nodos, conexiones, moléculas, condiciones de borde y lados curvos. Esta interfaz, también permite al usuario definir o modificar tanto el problema que se desea resolver, como las regiones en las que el usuario desea dirigir ya sea el refinamiento como el desrefinamiento de la triangulación.

Los estimadores de error utilizados permiten crear indicadores que dirigen el refinamiento y desrefinamiento en forma adaptativa, para mejorar la solución con la menor interacción del usuario. Con ello solo se debe definir una triangulación inicial conforme que representa una malla gruesa y luego por medio de los mecanismos de refinamiento explícito o los procesos adaptativo, obtener una triangulación que provea de una malla mas fina, que permitirá obtener una solución de mejor calidad.

Se ilustra el uso del sistema con problemas de prueba, de solución conocida; se muestra la imagen de la malla inicial del dominio y de algunas iteraciones, la malla y solución asociada.

Se concluye que esta herramienta constituye un software general, flexible y sencillo de usar para resolver problemas de EDP sobre dominios en 2-D generales.

Dedicatoria

Dedico este trabajo

a Patricia, mi esposa, por su amor, paciencia y apoyo incondicional

a Eduardo Antonio, mi hijo, por su confianza y fe en mí

a Elena, mi madre, por su ejemplo y sus esfuerzos

y a María Cecilia, mi profesora guía.

Agradecimientos

Agradezco a María Cecilia Rivara, a Edgardo Krell, a Patricio Poblete y a todos los que me han apoyaron tanto en NIC Chile, en la Universidad de Chile y en mi familia para poder completar este proyecto.

Agradezco también a Fondecyt (Proyecto N° 1911220).

Índice general

Índice general	I
Índice de cuadros	IV
Índice de figuras	V
1. Introducción	1
1.1. Antecedentes generales	1
1.2. Justificación	3
1.3. Objetivo	3
1.4. Contenido por capítulos	4
2. Diseño de la aplicación	5
2.1. Objetivos de la aplicación	5
2.2. Componentes o módulos	5
2.2.1. Librería de rutinas de refinamiento/desrefinamiento	6
2.2.2. Resolución de ecuaciones	7
2.2.3. Control de la aplicación	7
2.2.4. Estimadores de error	8
2.2.5. Presentación de resultados	8
3. Refinamiento/Desrefinamiento de mallas de triángulos	9
3.1. Estructura Molecular Compacta	9
3.2. Definiciones	11
3.3. Algoritmos de Refinamiento	12
3.4. Algoritmo de Desrefinamiento	13
3.5. Lados curvos	15
3.5.1. Bisección de un triángulo por su lado curvo	15
3.5.1.1. Expresión para hallar el punto medio del lado curvo (primer caso)	17
3.5.1.2. Expresión para hallar el punto medio del lado curvo (segundo caso)	18
3.5.1.3. Expresión para hallar el punto medio del lado curvo (tercer caso)	18
3.5.2. Método de Newton-Raphson	18

4. Método de Elementos Finitos	20
4.1. Descripción	20
4.2. Problemas elípticos	20
4.2.1. Planteamiento	20
4.2.2. Discretización del problema	21
4.2.3. Elementos triangulares lineales	24
4.2.3.1. Funciones de forma para Triángulos lineales C^0	24
4.2.3.2. Evaluación numérica de las integrales	26
4.2.3.3. Expresión para el flujo	31
4.2.4. Resolución numérica	33
4.3. Problemas parabólicos	33
4.3.1. Planteamiento	33
4.3.2. Discretización del problema	33
4.3.3. Elementos triangulares lineales	36
4.3.3.1. Funciones de forma para Triángulos lineales C^0	37
4.3.3.2. Evaluación numérica de las integrales	37
4.3.3.3. Expresión para el flujo	38
4.3.4. Resolución numérica	39
4.4. Integración de elementos triangulares curvos.	40
5. Estimadores del Error	41
5.1. Descripción	41
5.2. Indicador del error	41
5.2.1. Definición	41
5.2.2. Aplicación	42
5.3. Estimadores del Error	42
5.3.1. Estimador basado en la solución del problema.	42
5.3.2. Estimador basado en la solución obtenida	42
5.3.3. Estimador basado en la suavidad de la solución obtenida	42
6. Problemas de prueba	44
6.1. Problema 1	44
6.1.1. Descripción	44
6.1.2. Resultados	44
6.2. Problema 2	49
6.2.1. Descripción	49
6.2.2. Resultados	49
6.3. Problema 3	50
6.3.1. Descripción	50
6.3.2. Resultados	50
6.4. Problema 4	56
6.4.1. Descripción	56
7. Aplicación	58
7.1. Introducción	58
7.2. Ventana principal	58
7.3. Ventana de consulta/creación/modificación del problema	59

7.4. Ventana de consulta/creación/modificación de la malla	59
7.5. Ventana de presentación de soluciones (3DPlot)	61
7.6. Otras ventanas	61
7.7. Ejemplo de definición de un problema	62
7.7.1. Creación de la malla inicial	63
7.7.2. Definición del problema	65
8. Conclusiones	67
Bibliografía	68
A. Estructura de datos	69
A.1. Nodo	69
A.2. Lado	70
A.3. Triángulo	70
A.4. Conexión	70
A.5. Molécula	71
A.6. Triangulación	71
A.7. Región	72
A.8. Expresión	72
B. Formato de los archivos de datos	74
B.1. Estructura del archivo de descripción del problema a resolver	74
B.2. Estructura del archivo de descripción de la malla a utilizar	74
C. Expresiones aritméticas	82

Índice de cuadros

6.1. Estadísticas ejecución problema 1	45
6.2. Estadísticas ejecución problema 2	49
6.3. Estadísticas ejecución problema 3	53
B.1. Archivo de descripción del problema (1/3)	75
B.2. Archivo de descripción del problema (2/3)	76
B.3. Archivo de descripción del problema (3/3)	77
B.4. Archivo de descripción de la malla (1/3)	79
B.5. Archivo de descripción de la malla (2/3)	80
B.6. Archivo de descripción de la malla (3/3)	81
C.1. Descripción de los elementos de la gramática de expresiones	83

Índice de figuras

2.2.1.Esquema funcional de la aplicación	6
3.1.1.Ejemplo de una Triangulación	10
3.1.2.Representación de la triangulación conforme de la figura previa	10
3.3.1.Algoritmos de refinamiento	13
3.3.2.Propagación del refinamiento	14
3.4.1.Triángulo a eliminar	15
3.4.2.Desrefinamiento	16
3.5.1.Refinoamiento de un triángulo por el lado curvo	17
4.2.1.Elemento triangular lineal C^0 y su solución de prueba	24
4.2.2.Flujo a lo largo de la frontera de dos elementos con un lado interno común	27
4.2.3.Flujo concentrado en un punto de la frontera	29
4.2.4.Flujo para un lado interno entre elementos	29
4.2.5.Ejemplo de flujo nodal	32
6.1.1.Malla inicial, problema 1	45
6.1.2.Iteración 4, problema 1	46
6.1.3.Iteración 12, problema 1	47
6.1.4.Iteración 23, problema 1	48
6.2.1.Malla inicial, problema 2	50
6.2.2.Iteración 5, problema 2	51
6.2.3.Iteración 12, problema 2	52
6.3.1.Malla inicial, problema 3	53
6.3.2.Iteración 2, problema 3	54
6.3.3.Iteración 9, problema 3	55
6.4.1.Dominio D	56
6.4.2.Problema 4	57
7.2.1.Ventana principal aplicación gxmalla	59
7.3.1.Ventana de visualización de un problema	60
7.4.1.Ventana de visualización de una malla	60
7.5.1.Presentación de la solución	61
7.6.1.Abrir archivo	61
7.6.2.Acerca de	62
7.7.1.Creación de nodos de la malla inicial	63
7.7.2.Creación de la triangulación, las moléculas y los vecinos	64
7.7.3.Creación de las condiciones de borde	64

7.7.4.Modificación de los nodos de la malla	65
7.7.5.Definición del problema	66

Capítulo 1

Introducción

1.1. Antecedentes generales

La resolución de ecuaciones diferenciales parciales (EDP) es una necesidad de las más diversas áreas de la Ciencia y la Ingeniería, pues permite entre otros modelar problemas de resistencia de materiales, estructuras, dinámica de fluidos o termodinámica.

El Método de los Elementos Finitos (MEF) es una técnica que permite resolver EDP o sistemas de EDP que modelan problemas físicos o de ingeniería.

Para la utilización del MEF, el usuario debe proveer:

- una EDP que modele su problema,
- un dominio acotado sobre el cual se encontrará la solución a la ecuación que modela el sistema,
- condiciones de borde (CB) sobre el dominio, sean
 - sobre la función, es decir, CB esenciales (CBE), o
 - sobre la primera derivada de ésta, es decir, CB Naturales (CBN).

En la utilización del método de los elementos finitos clásico, el proceso se divide en las siguientes etapas:

- En primer lugar se genera una discretización o malla del dominio dado, que corresponde a una colección finita de elementos preseleccionados. Para esto se define el tipo de elementos, nodos y propiedades geométricas. Por ejemplo, triángulos y/o rectángulos en los dominios de dos dimensiones (2-D) o tetraedros y/o hexaedros en los dominios de tres dimensiones (3-D). En forma más precisa diremos que, en el caso 2-D, los elementos serán triángulos lineales, isoparamétricos, de segundo orden, etc. Los nodos del problema corresponderán a los vértices de la malla y otros puntos auxiliares que permiten calcular funciones de forma sobre el elemento.
- En segundo lugar se construye el sistema lineal de ecuaciones sobre la discretización. Esta etapa incluye la derivación de las ecuaciones sobre los elementos geométricos, el ensamble de las ecuaciones elementales y la imposición de las condiciones de borde.

- En tercer lugar se encuentra la solución numérica, resolviendo el sistema de ecuaciones ensamblado.
- Como la precisión de la solución numérica depende de la discretización utilizada, la construcción de la malla es un aspecto clave en la utilización correcta del método de elementos finitos. De acuerdo al error estimado en la solución, se procede a repetir la primera etapa preparando una malla que sea mas fina en las zonas en donde éste estime que se requiera mejorar la precisión del resultado. El proceso de preparar una malla mas fina se llama refinamiento. Los procesos de modificación de las mallas, resultan complejos, dificultando la resolución de los problemas.

La complejidad en el uso del MEF, motiva el desarrollo de herramientas y métodos que automaticen o simplifiquen su operación. Una solución a este problema es el uso de métodos adaptivos.

Al usar software basado en métodos adaptivos, la modificación de la malla se realiza en forma dinámica durante el proceso de resolución del problema. Esto depende en forma crítica de la disponibilidad de algoritmos flexibles que permitan modificar la malla, ya sea por refinamiento y/o por desrefinamiento, de acuerdo al error estimado sobre la discretización. Este trabajo realizado en forma automática, obtendrá resultados que dependerán de que el software incluya estimadores confiables del error asociados a la resolución de la ecuación diferencial en consideración.

En forma más específica, el diseño de software adaptivo para análisis de elementos finitos, sigue el siguiente esquema general:

- En primer lugar se construye una malla inicial gruesa que representa adecuadamente la geometría del problema.
- Luego se calcula la solución de elementos finitos asociada a ésta malla.
- Se procede entonces a evaluar esta solución (discretización). Mientras la solución numérica calculada no es aceptable, se repite el siguiente ciclo:
 - En base a la evaluación de la solución se define una submalla de refinamiento, es decir, se identifica el conjunto de elementos de la malla que deberían ser refinados para mejorar la solución.
 - Se utiliza entonces esta información para refinar la malla.
 - Se calcula una nueva solución de elementos finitos, que a su vez se evalúa para decidir si es suficientemente precisa para concluir el proceso.
- En el caso de los problemas parabólicos, una vez obtenida una solución para un tiempo t_i se procede a evaluar el problema para el tiempo t_{i+1} , entonces se repite el siguiente ciclo:
 - En base a la evaluación de la solución se define una submalla de desrefinamiento, es decir, se identifica el conjunto de elementos de la malla que deberían ser desrefinados pues su pequeño tamaño no ayuda a mejorar la solución.
 - Se utiliza entonces esta información para desrefinar la malla.

- Se calcula una nueva solución de elementos finitos, que a su vez se evalúa para decidir si ya no es lo suficientemente precisa para concluir el proceso.
- Se procede a repetir el ciclo desde 1.1 en la página anterior para el tiempo t_{i+1} .

Como la solución numérica obtenida en cada paso depende de la calidad de la malla construida para discretizar el problema continuo, la evaluación de la malla se realiza en la práctica analizando dicha solución numérica.

En el contexto adaptivo, el proceso completo se realiza en forma totalmente automática usando un estimador confiable del error para evaluar la solución y un criterio que permita decidir qué elementos deberían refinarse para obtener una mejor solución (submalla de refinamiento).

En el sistema descrito en este trabajo, la generación y modificación de las mallas, se hace mediante algoritmos de refinación y desrefinación de mallas triangulares conformes basados en los algoritmos diseñados y desarrollados por Rivara y Spencer [2][4][3], una primera referencia de este trabajo fue presentada en el Congreso Internacional sobre Métodos Numéricos en Ingeniería y Ciencias Aplicadas celebrado en la Universidad de Concepción, Chile del 16 al 20 de Noviembre de 1992[5].

1.2. Justificación

En el momento del inicio de este trabajo (año 1988), no habían herramientas disponibles para la resolución de EDP sobre dominios generales en 2-D mediante mecanismos adaptativos. El desarrollo inicial evolucionó junto con el desarrollo de las tecnologías, incorporándose nuevos elementos y módulos, como GUI para los usuarios. Si bien, actualmente existen varias herramientas disponibles para la resolución de EDP sobre dominios generales en 2-D¹, la aplicación puede usarse como:

- prototipo para apreciar el funcionamiento del MEF,
- plataforma de evaluación de indicadores de error y su impacto en el control de la adaptabilidad,
- como plataforma para probar nuevos desarrollos con métodos multimalla
- o como base para el soporte de resolución de problemas en 3-D.

1.3. Objetivo

El objetivo del trabajo es desarrollar una herramienta para la solución de EDP sobre dominios generales en 2-D mediante métodos adaptivos, que utilizase los desarrollos realizados en el refinamiento y desrefinamiento de triangulaciones conformes desarrollado por María Cecilia Rivara para representar los dominios y apoyar el proceso de solución de los problemas elípticos y parabólicos en forma adaptiva.

Esta herramienta debería utilizar métodos adaptativos y permitir la evaluación de estimadores de error para estos métodos y ser una aplicación útil para su uso en Ingeniería, que dispusiera de interfaces amigables para el usuario.

¹FreeFem++, <http://www.freefem.org/> - Matlab, <http://www.mathworks.com/products/matlab/>, entre otras

1.4. Contenido por capítulos

En el capítulo 2 se describe la forma en que se diseñó e implementó el sistema. En el capítulo 3 se describen los algoritmos de refinación y desrefinación. En el capítulo 4 se hace referencia a como se implementa el M.E.F. En el capítulo 5 se presentan los estimadores del error. En el capítulo 6 se presentan problemas test y los resultados obtenidos al resolverlos con el sistema desarrollado. En el capítulo 7 se presenta la aplicación, mostrando la interfaz gráfica y un ejemplo de como utilizarla para resolver un problema. Para terminar en el capítulo 8 se discute el trabajo realizado.

Capítulo 2

Diseño de la aplicación

2.1. Objetivos de la aplicación

Se definió en primer lugar el rango de aplicaciones que el sistema debería apoyar. Luego se definieron las siguientes estrategias:

- se estableció que resolvería ecuaciones o sistemas de ecuaciones diferenciales parciales (en adelante E.D.P.) en 2-D mediante elementos finitos triangulares. Esta decisión permite trabajar con dominios generales, obtener una buena precisión en la aproximación y simplicidad en el tratamiento.
- se estableció que debería utilizar métodos adaptativos y permitir la evaluación de estimadores de error para estos métodos.
- se estableció que se debería desarrollar una aplicación útil para su uso en Ingeniería, que dispusiera de interfaces amigables para el usuario.
- se estableció que debería soportar en una segunda etapa la inclusión de métodos de resolución multimallas.
- se estableció que la aplicación debía ser modular, para facilitar la incorporación de nuevas funcionalidades.
- se decidió realizar el desarrollo de la aplicación en el *lenguaje C*, desarrollándolo para sistemas UNIX $\text{\textcircled{R}}$ ¹, y que las interfaces gráficas se desarrollarían en el sistema *X-Window*².

2.2. Componentes o módulos

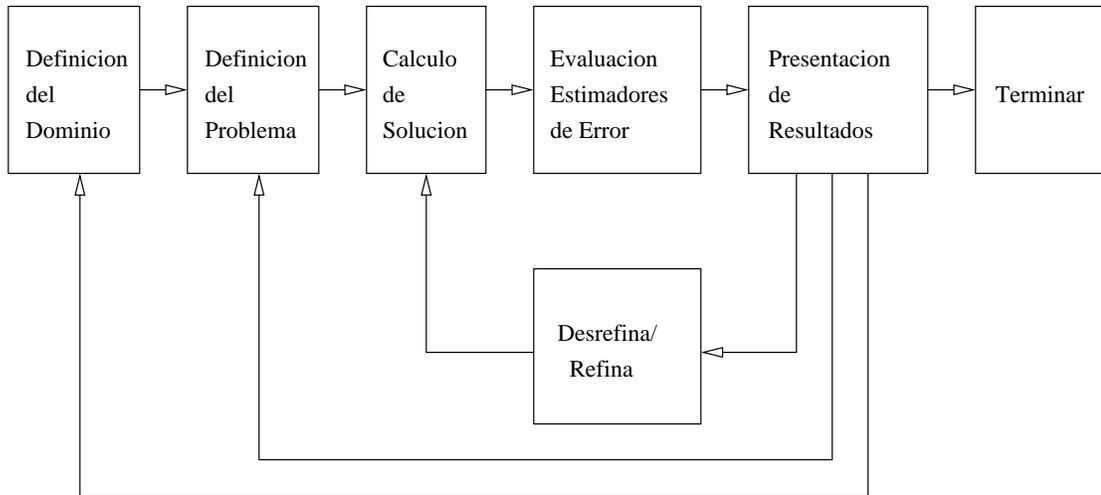
A continuación se procedió a la definición de los módulos que permitirían la construcción de la aplicación. Se definieron para ello los siguientes módulos:

1. Librería de rutinas para el refinamiento/desrefinamiento
2. Resolución de ecuaciones

¹UNIX es una marca comercial registrada de The Open Group en los Estados Unidos y otros países.

²X-Window systems, <http://www.x.org>

Figura 2.2.1: Esquema funcional de la aplicación



3. Control de la aplicación
4. Estimadores de error
5. Presentación de resultados

Basado en lo anterior se definió el esquema de operación de la aplicación. La figura 2.2.1 muestra un diagrama lógico que describe la relación entre las diferentes funcionalidades de la aplicación.

Finalmente se definieron las estructuras de datos, que se describe en el apéndice A, y el formato de los archivos, que se describe en el apéndice B.

Con posterioridad, dada la disponibilidad de nuevas tecnologías, se procedió a agregar una interfaz gráfica, que permitiera al usuario un mayor control sobre la operación.

2.2.1. Librería de rutinas de refinamiento/desrefinamiento

Este módulo corresponde al esqueleto del sistema que permite el refinamiento y desrefinamiento de triangulaciones sobre dominios generales, usando los algoritmos descritos en el capítulo 3. Para lograr una mejor aproximación en los problemas reales, los algoritmos se generalizaron para manejar triángulos de bordes curvos, pudiendo ser estos actualmente arcos de circunferencia o parábolas, aunque el diseño permite que en el futuro se agreguen nuevas opciones. Los triángulos de bordes curvos se usan solo en las fronteras de los dominios o en las interfaces entre materiales diferentes dentro del dominio.

Las librerías de refinamiento y desrefinamiento, corresponden a la implementación de las rutinas y métodos desarrollados por Rivara y Spencer[3]. Estas librerías contienen las rutinas requeridas para el refinamiento y desrefinamiento de mallas de triángulos conformes, definidas por medio de una estructura de datos denominada Estructura Molecular Compacta (EMC).

Todas las rutinas contenidas en las librerías fueron escritas en el *lenguaje C*.

Las rutinas se agrupan de acuerdo a su función, en dos librerías:

1. *libref.a* agrupa las rutinas usadas para el refinamiento de las mallas,
2. *libdes.a* agrupa las rutinas usadas para el desrefinamiento de las mallas.

2.2.2. Resolución de ecuaciones

Sobre el esqueleto que representa la EMC y las librerías de refinamiento y desrefinamiento se desarrollaron los programas que implementan el MEF, usando en esta etapa funciones de forma lineales, aunque el diseño permite incorporar posteriormente funciones de forma de mayor grado.

El módulo del MEF se ha diseñado de modo que la definición del problema sea realizada en forma externa por el usuario, sin necesidad de que éste sepa programar, o requiera recompilar parte de la herramienta para cada problema. Así el usuario puede definir externamente, como parámetros del sistema, todas las funciones que serán utilizadas, usando para ello el lenguaje que se describe en el apéndice C. En base a lo anterior el sistema construye la matriz de rigidez asociada al problema y al lado derecho. Para la resolución del sistema de ecuaciones se han utilizado librerías disponibles en el dominio público para la operación con matrices poco densas, que utilizan un método directo basado en la descomposición LU, que aprovecha las características de baja densidad de elementos de la matriz de rigidez, estas son la librería Sparse³ y la librería SuperLU⁴. Se dispone en forma opcional de un algoritmo de reenumeración de nodos para optimizar la resolución del problema. En el futuro podría incorporar el uso opcional de un método multimalla para la resolución del MEF.

Las rutinas desarrolladas para la resolución de las ecuaciones se agrupan de acuerdo a su función, en dos librerías:

1. *libmef.a* agrupa las rutinas usadas para construir la matriz de rigidez, el lado derecho del problema numérico que será resuelto y los indicadores del error.
2. *libmat.a* agrupa las rutinas que contienen herramientas de cálculo que son usadas en el resto de los módulos.

2.2.3. Control de la aplicación

La aplicación desarrollada originalmente para ser usada en modo terminal a través de la línea de comandos del shell, evolucionó, transformándose en una aplicación gráfica que permitió mejorar la usabilidad. El control de esta aplicación se realiza por medio de una interfaz gráfica. Esta interfaz fue desarrollada inicialmente usando las librerías de dominio público *XForms(The Forms Library)*⁵ que residen en la librería *libforms.a* del sistema. Estas librerías proveen funciones que permiten construir interfaces en Xwindows. Dado que dichas librerías dejaron de mantenerse, se reconstruyó la aplicación usando las librerías GTK+⁶ (GIMP Toolkit) y la herramienta para desarrollo rápido multiplataforma GLADE⁷

Este módulo permite las siguientes funcionalidades:

³Sparse 1.3 “A Sparse Linear Equation Solver”, Kenneth S. Kundert y Alberto Sangiovanni-Vincentelli, University of California, Berkeley. Estas librerías están disponibles a través de netlib (<http://cm.bell-labs.com/netlib/sparse/index.html>).

<http://www.cise.ufl.edu/research/sparse/umfpack/>

⁴SuperLU 3.0[9], “General purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines”, <http://crd.lbl.gov/~xiaoye/SuperLU/>

⁵XForms es Copyright (c) 1996-2000 by T.C. Zhao and Mark Overmars. (<http://www.nongnu.org/xforms/>)

⁶GTK+ is free software and part of the GNU Project. The licensing terms for GTK+, are GNU LGPL (<http://www.gtk.org/>)

⁷Glade is Free Software released under the GNU GPL License. (<http://glade.gnome.org/>)

- Definición de los dominios de los problemas y de la malla inicial.
- Definición de los problemas
- Control de la ejecución, permitiendo una operación semiautomática basada en procedimientos adaptativos de control de la calidad de la solución o procedimientos interactivos de refinamiento/desrefinamiento utilizando regiones previamente configuradas.

2.2.4. Estimadores de error

Una vez resuelto el sistema, se calculan los errores locales de aproximación, sobre cada elemento y se usa esta información tanto para decidir qué elementos de la malla actual deben modificarse, por refinación y/o desrefinación, como para calcular un estimador global del error que permita evaluar la precisión global de la solución actual. Si la solución actual no es aceptable se modifica la malla, para entonces proceder a reconstruir la matriz de rigidez y volver a empezar el ciclo adaptivo.

El sistema provee algunos estimadores de error que el usuario de la aplicación puede seleccionar. Se pueden definir nuevos estimadores para el control de la adaptatividad, pero para esto se requiere modificar una de las librerías del sistema.

2.2.5. Presentación de resultados

La presentación de resultados se realiza a través de:

1. la aplicación, por medio de paneles en los que se despliega una imagen gráfica de la malla de triángulos y que genera un archivo en postscript una versión para imprimir.
2. un programa de dominio público, 3DPlot⁸ que permite mostrar en 3 dimensiones las soluciones obtenidas, entregándole al usuario la posibilidad de realizar acciones sobre la imagen resultante, como son rotaciones o la impresión postscript de la imagen desplegada.

⁸Autor: Kenny K.H. Toh (ktoth@mascot.berkeley.edu)

Capítulo 3

Refinamiento/Desrefinamiento de mallas de triángulos

3.1. Estructura Molecular Compacta

La estructura molecular compacta (EMC) , es la estructura de datos usada para representar en forma compacta una secuencia de triangulaciones conformes anidadas. Esta estructura es capaz de soportar los algoritmos de refinación y desrefinación de las mismas.

La estructura de datos se componen de los siguientes objetos geométricos:

- Nodo, elemento básico de la triangulación. Contiene información sobre:
 - posición del nodo físico,
 - posición dentro de la malla, esto es, si está ubicado en el borde o al interior,
 - condiciones de borde que lo afectan,
 - tipo de material(para el caso de dominios no homogéneos),
 - edad relativa al resto de los triángulos,
 - pertenencia a lado curvo
- Molécula, asociada a un nodo está formada por el conjunto de sus nodos vecinos
- Triángulo, grupo de tres nodos vecinos que se definen en base a las moléculas
- Triangulación, grupo de triángulos generados en una operación de refinación o desrefinación y se obtienen por bisección de triángulos antiguos, creando nuevos nodos que se agregan a la antigua malla.
- Lado Curvo, polinomio que permite representar bordes o interfaces en dominios generales.

La EMC guarda la información de todas las mallas generadas, almacenando solo la modificación local de la malla actual con respecto a la anterior.

En la figura 3.1.2, de la página 10 se ilustra la estructura molecular asociada a la triangulación de la figura 3.1.1, de la página 10. Así, por ejemplo, la molécula asociada al nodo 2 es (3,5,7,1). Observe que la formación de las moléculas se hace en sentido positivo o antihorario.

Figura 3.1.1: Ejemplo de una Triangulación

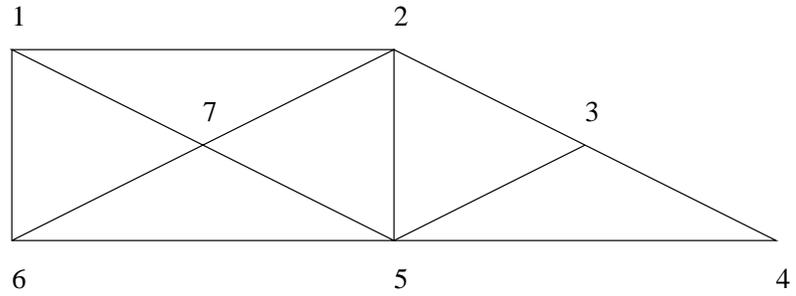
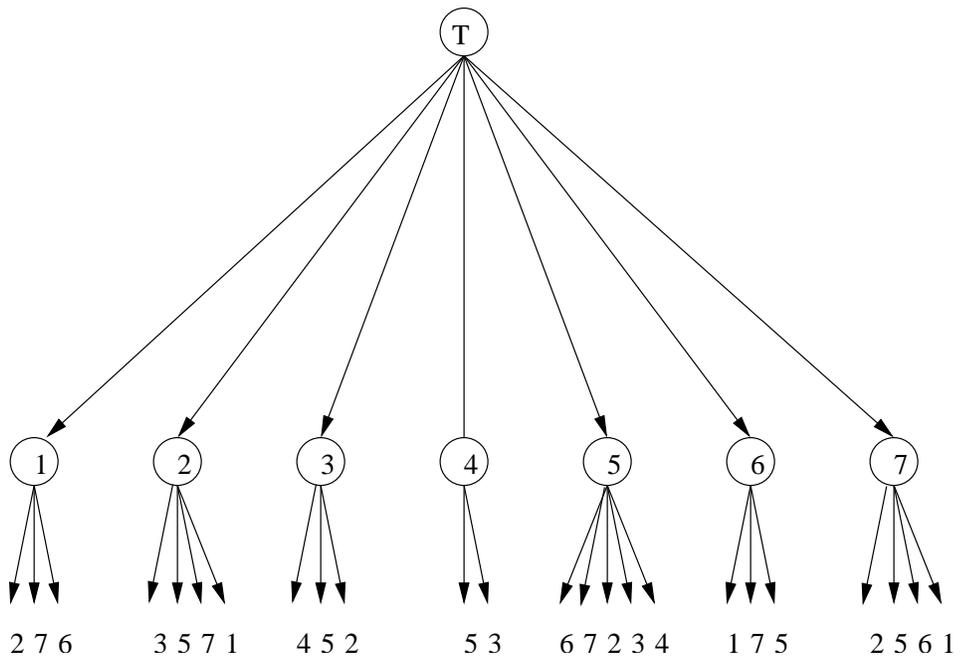


Figura 3.1.2: Representación de la triangulación conforme de la figura previa



3.2. Definiciones

Para usar las triangulaciones con el método de los elementos finitos se exigen las siguientes cualidades:

1. *Condición de Conformidad*, la intersección de cualquier par de triángulos adyacentes es un lado común o bien un vértice común.
2. *Condición de No Degeneración*, los ángulos internos de todos los triángulos están acotados inferiormente (por un valor $\alpha > 0$).
3. *Condición de Suavidad*, para cualquier par de triángulos no disjuntos t_1 y t_2 (cuyos respectivos diámetros son: h_1 y h_2) se cumple:

$$\frac{\min(h_1, h_2)}{\max(h_1, h_2)} > \delta \quad (3.2.1)$$

esta condición asegura las propiedades de convergencia del método de elementos finitos y permite el uso de discretizaciones irregulares para manejar ecuaciones parciales con soluciones singulares.

Las condiciones anteriores se pueden formalizar con la siguiente definición:

Definición 1 : T^* se dirá una familia (α_0, δ) admisible de triangulaciones irregulares, conformes y no degeneradas, si se cumplen las siguientes condiciones:

1. Para cualquier triangulación $\tau \in T^*$ y para cualquier par de triángulos adyacentes $t_1, t_2 \in \tau$, se cumpla la condición de suavidad 3.2.1 para la misma constante δ .
2. Si α_t es el ángulo interior mínimo del triángulo t , entonces $\min \alpha_t \geq \alpha_0 > 0$, sobre todos los triángulos de todas las triangulaciones $\tau \in T^*$.

Para dar soporte a métodos multimalla se requiere que la secuencia de triangulaciones sea anidada, esto es:

Definición 2 : Sean T_i y T_j dos triangulaciones pertenecientes a la secuencia de mallas, en que $(i > j) \Rightarrow (T_i \text{ es mas gruesa que } T_j)$. Sean t_i, t_j triángulos tales que $t_i \in T_i$ y $t_j \in T_j$. Se dice que la triangulación T_i y T_j son anidadas si:

$$\forall t_j \in T_j \quad \exists! t_i \in T_i \quad / \quad t_j \subseteq t_i$$

Para crear la secuencia de triangulaciones a partir de una triangulación inicial gruesa se requiere proceder a modificar la malla actual de modo que conserve las propiedades antes indicadas. Estos procedimientos se llamaran refinamiento y desrefinamiento.

Definición 3 : Se llama refinamiento a la operación que aplicada a una triangulación T_i genera una triangulación T_{i+1} que es más fina, esto es, si t_i y t_{i+1} son triángulos de las triangulaciones antes indicadas, entonces:

$$\forall t_i \in T_i \text{ no existe } t_{i+1} \text{ tal que } t_i \subset t_{i+1},$$

considerando a \subset como subconjunto disjunto

Definición 4 : Se llama desrefinamiento a la operación que aplicada a una triangulación T_i genera una triangulación T_{i+1} que es más gruesa, esto es, si t_i y t_{i+1} son triángulos de las triangulaciones antes indicadas, entonces:

$$\forall t_{i+1} \in T_{i+1} \text{ no existe } t_i \text{ tal que } t_{i+1} \subset t_i,$$

considerando a \subset como subconjunto disjunto

Definición 5 : Se llama segmento curvo S al conjunto de puntos definido por:

$$S = \{(x, y) / (x, y) = (a_x t^3 + b_x t^2 + c_x t + d_x, a_y t^3 + b_y t^2 + c_y t + d_y), t \in [0, 1]\}$$

Definición 6 : Se llamará triángulo curvo a aquel que posea un lado contenido en un segmento curvo S . El lado contenido se denominará lado curvo. Esto es:

$$\Delta_{ABC} \text{ curvo} \Leftrightarrow \exists \text{ lado } s_{AB} \in \Delta_{ABC} / s_{AB} \subseteq S, S \text{ segmento curvo}$$

3.3. Algoritmos de Refinamiento

Para realizar el refinamiento de una triangulación T se requiere:

1. Seleccionar los triángulos t a refinar.
2. Para cada triángulo t , realizar su división (o bisección).
3. Proceder a dejar la triangulación T conforme.

Para realizar la bisección se dispone de dos algoritmos, que fueron bautizados “algoritmo de 2 triángulos” y “algoritmo de 4 triángulos”. Estos algoritmos tienen la cualidad de que, en el caso de que la triangulación inicial cumpla con las condiciones indicadas en la definición 1 de la página 11, siempre producirán triangulaciones anidadas que pertenecen a la misma familia (α_0, δ) admisible de triangulaciones irregulares T^* , donde las constantes α_0 y δ que definen la familia solo dependen de la triangulación inicial T_0 .

El algoritmo de 2 triángulos, se basa en efectuar bisecciones por el punto medio del lado mas largo, a estas bisecciones las llamaremos “bisecciones simples”. El algoritmo de 4 triángulos consiste en que además de realizar la bisección por el punto medio del lado mas largo, se procede a realizar la bisección de los dos triángulos generados a partir de nodo creado en la bisección previa, a esta bisección la llamaremos “bisección generalizada”. En la figura 3.3.1 de la página 13 se muestran los dos algoritmos. En el caso de ambos algoritmos, luego de realizar las bisecciones se procede a hacer conforme la malla por los nodos creados. La figura 3.3.2 de la página 14 muestra un ejemplo de la propagación del refinamiento en una malla, para hacer la malla conforme. El desarrollo de estos algoritmos como la demostración de sus propiedades se puede ver en la referencia[3].

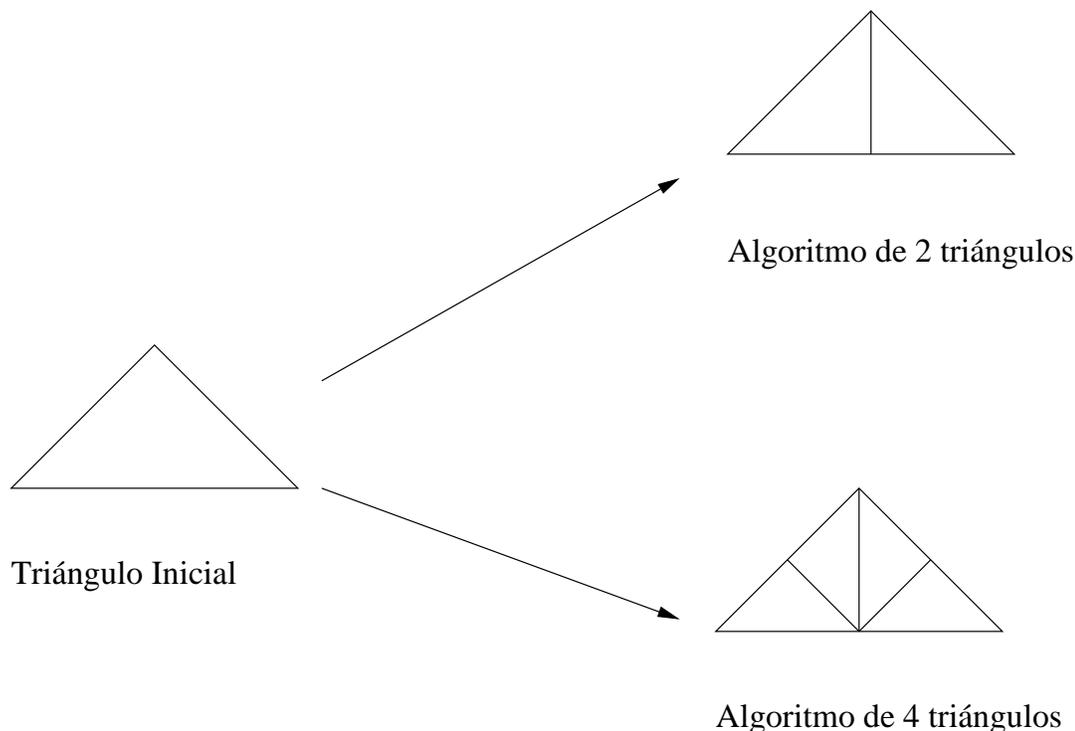


Figura 3.3.1: Algoritmos de refinamiento

3.4. Algoritmo de Desrefinamiento

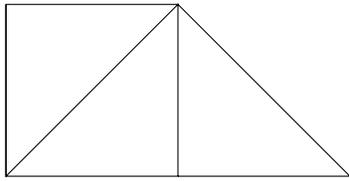
Consideremos una secuencia anidada de triangulaciones conformes $\{T_i, 0 \leq i \leq r\}$. Supongamos que se desea desrefinar un triángulo $t_i, i > 0$, que llamaremos i -nuevo dado que fue creado por refinamiento en la triangulación T_i . El desrefinamiento de t_i deberá revisar todas las triangulaciones $T_k, i \leq k \leq r$, en donde r es la triangulación mas fina del dominio.

El desrefinamiento sobre la secuencia T_i, T_{i+1}, \dots, T_r del triángulo t i -nuevo se desarrolla a continuación.

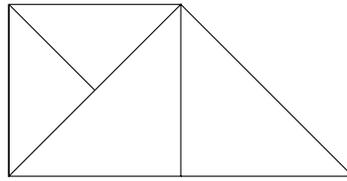
Sea $\tau = \{ \text{triángulos a ser eliminados por el desrefinamiento} \}$

1. Determinar el vértice n del triángulo t a eliminar de la malla (n es un nodo i -nuevo).
2. Determinar el triángulo $t^* \in \tau(T_{i-1})$ que incluye a t (los casos posibles se pueden observar en la figura 3.4.1 de la página 15, en donde $t^* = ABC$).
3. Según el origen del nodo tenemos dos situaciones:
 - a) Si es nodo creado por bisección simple de t^* figura 3.4.1(a) definir $\tau = \{t^*\}$
 - b) Si es nodo creado por bisección generalizada de t^* figura 3.4.1(b), definir $\tau = \{t'/t\}$ fue formado por la bisección generalizada de t^* que incluye a n
4. Definir $\tau = \tau \cup \{t' \in \tau(T_i) : t' \text{ es no conforme}\}$
5. Para cada $t' \in \tau$, eliminar todos los triángulos contenidos en t' de todas las mallas posteriores ($t' \in T_k, i \leq k \leq r$).

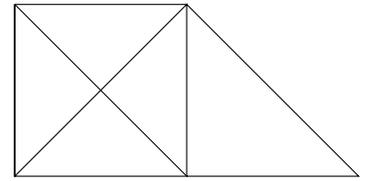
Refinamiento por algoritmo "2 Triángulos"



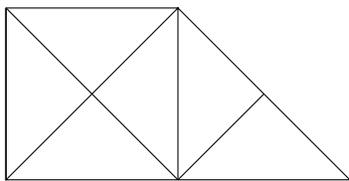
1. Triangulación inicial



2. Bisección

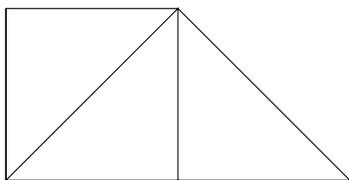


3. Dejo conforme

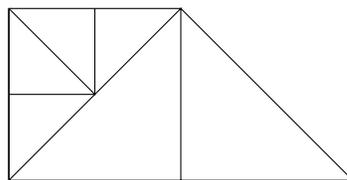


4. Bisección

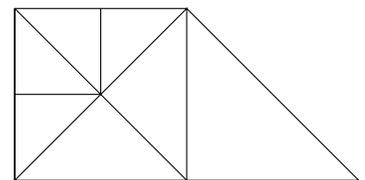
Refinamiento por algoritmo "4 Triángulos"



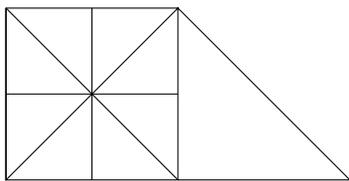
1. Triangulación inicial



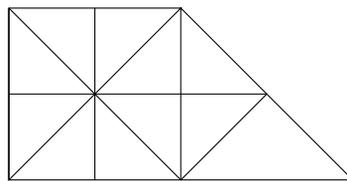
2. Bisección



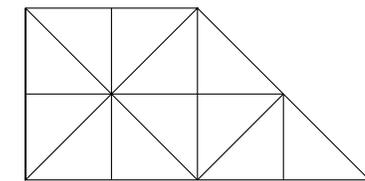
3. Dejo conforme



4. Bisección



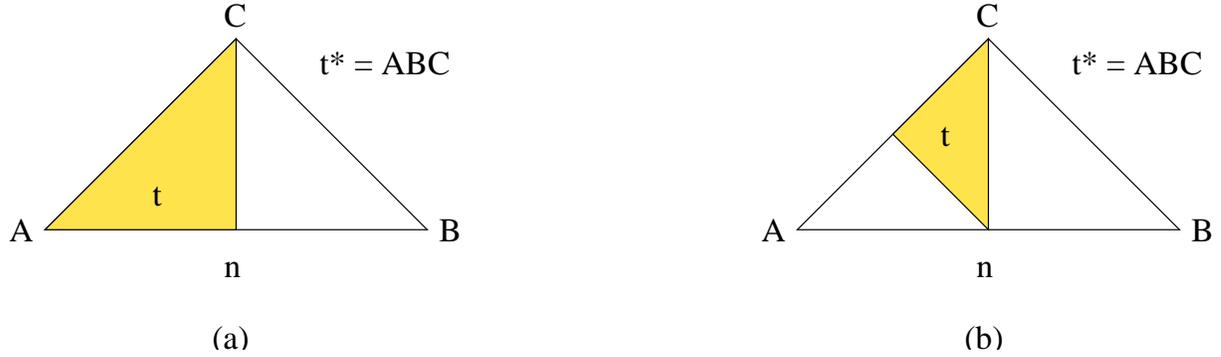
5. Dejo conforme



6. Bisección

Figura 3.3.2: Propagación del refinamiento

Figura 3.4.1: Triángulo a eliminar



La figura 3.4.2 de la página 16 muestra una secuencia de desrefinamiento de una triangulación, en la que se procede a eliminar el nodo n , y posteriormente los nodos más nuevos n_1 y n_2 , para poder así conservar la conformidad de la malla y evitar la existencia de elemento trapeczoidales.

El algoritmo de desrefinamiento, considera la propagación del mismo para asegurar la conformidad de la triangulación. El algoritmo de desrefinamiento no corresponde a la aplicación de la operación inversa de la operación de refinamiento, en particular esto casi siempre es cierto en el caso del algoritmo de 4 triángulos.

3.5. Lados curvos

Para poder resolver problemas sobre dominios irregulares, se ha escogido usar polinomios de grado 3 (cúbicos) para representar los lados curvos, ver definiciones 5 y 6 en la página 12. Los segmentos curvos estarán asociados a uno o más triángulos, de acuerdo a su definición inicial y a los procesos de refinamiento y desrefinamiento que se realizan durante la búsqueda de una solución al problema. La representación escogida permite aproximar por medio de polinomios de interpolación cualquier forma compleja o conjunto de punto, usando por ejemplo, splines cúbicos.

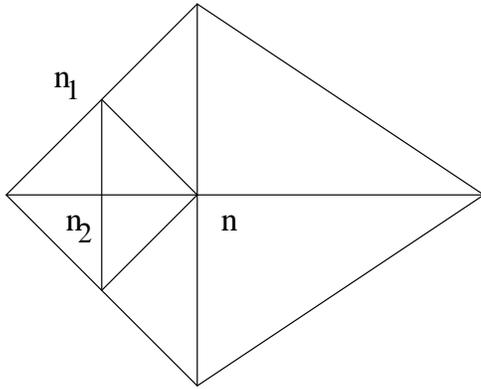
3.5.1. Bisección de un triángulo por su lado curvo

Para refinar un triángulo curvo por su lado curvo se ha definido el siguiente procedimiento (ver figura 3.5.1 en la página 17):

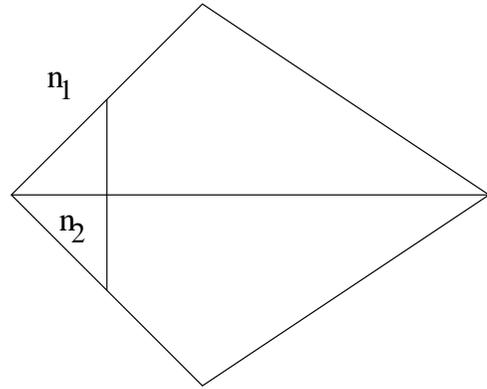
Sea s_{AB} el lado curvo de un triángulo \triangle_{ABC} , para encontrar el punto de bisección:

1. se encuentra el punto medio del segmento recto definido por los puntos A y B extremos del lado S_{AB}
2. se traza la recta perpendicular al punto medio del segmento \overline{AB}
3. se define como punto de bisección la intersección entre la recta perpendicular al punto medio del segmento \overline{AB} y el lado curvo s_{AB}

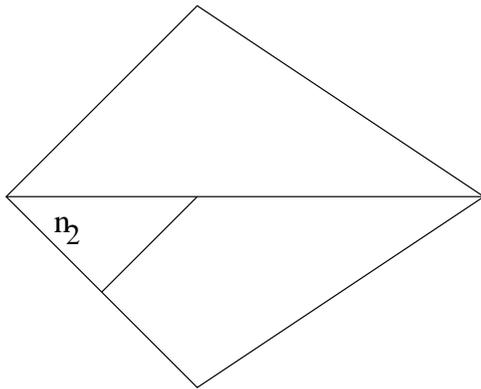
Figura 3.4.2: Desrefinamiento



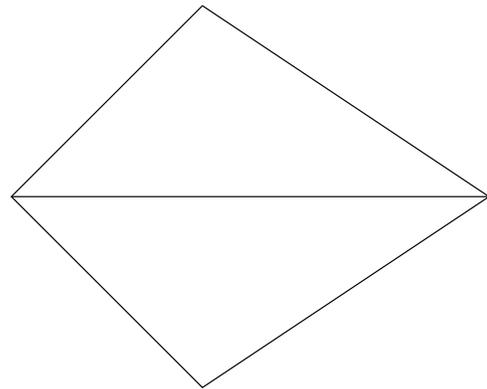
1. Triangulación Inicial



2. Elimino nodo n

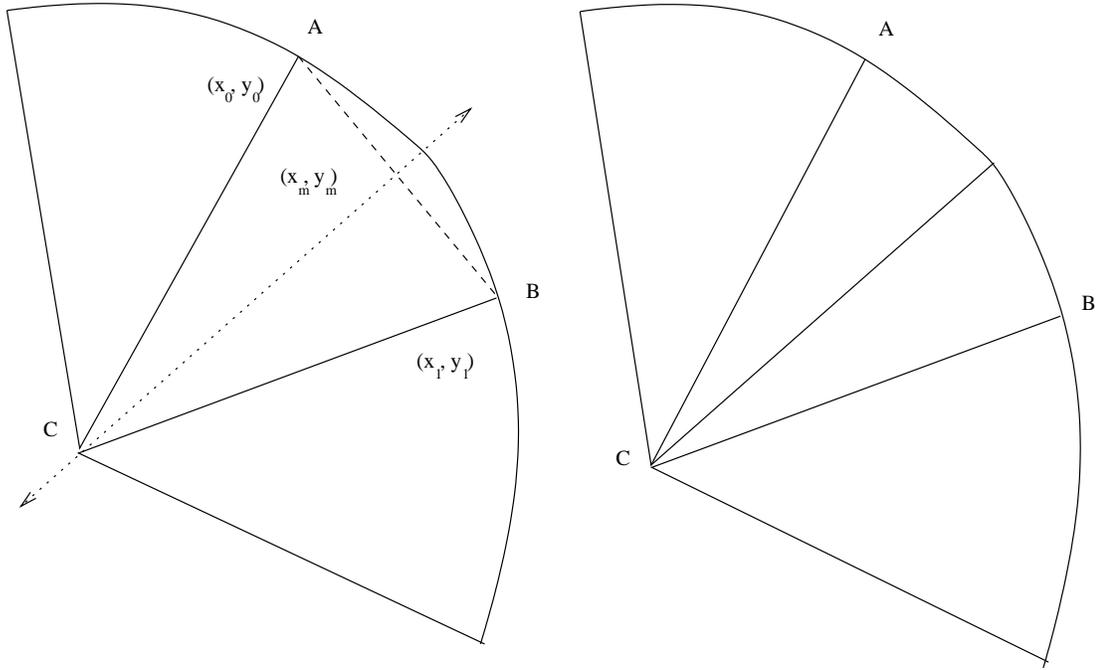


3. Elimino nodo n1



4. Elimino nodo n2

Figura 3.5.1: Refinamiento de un triángulo por el lado curvo



3.5.1.1. Expresión para hallar el punto medio del lado curvo (primer caso)

Sean $A = (x_0, y_0)$ y $B = (x_1, y_1)$ los extremos del lado curvo s_{AB} . Consideremos primero que $x_0 \neq x_1$ y que $y_0 \neq y_1$. El punto medio del segmento \overline{AB} es $(\frac{x_0+x_1}{2}, \frac{y_0+y_1}{2})$, la pendiente del segmento \overline{AB} sería $m = \frac{y_1-y_0}{x_1-x_0}$ por lo que la pendiente de la recta perpendicular será $m_{\perp} = -\frac{x_1-x_0}{y_1-y_0}$. Al conocer la pendiente de la recta perpendicular y un punto de esta (el punto medio de \overline{AB}) tenemos la ecuación de la recta perpendicular, esta es:

$$y = -\frac{x_1 - x_0}{y_1 - y_0}x + \frac{(y_1^2 - y_0^2) + (x_1^2 - x_0^2)}{2(y_1 - y_0)}$$

Si usamos la definición 5, tenemos que:

$$a_y t^3 + b_y t^2 + c_y t + d_y = -\frac{x_1 - x_0}{y_1 - y_0} (a_x t^3 + b_x t^2 + c_x t + d_x) + \frac{(y_1^2 - y_0^2) + (x_1^2 - x_0^2)}{2(y_1 - y_0)}$$

de donde obtenemos:

$$\begin{aligned} & 2(a_y(y_1 - y_0) + a_x(x_1 - x_0))t^3 + 2(b_y(y_1 - y_0) + b_x(x_1 - x_0))t^2 + \\ & + 2(c_y(y_1 - y_0) + c_x(x_1 - x_0))t + \\ & + 2(d_y(y_1 - y_0) + d_x(x_1 - x_0)) - ((y_1^2 - y_0^2) + (x_1^2 - x_0^2)) = 0 \end{aligned} \quad (3.5.1)$$

La ecuación 3.5.1 es de tercer grado en t , y en nuestro caso es resuelta por un método numérico iterativo, el método de Newton-Raphson.

3.5.1.2. Expresión para hallar el punto medio del lado curvo (segundo caso)

Consideremos el caso mostrado en la sección 3.5.1.1, pero en que tenemos que $x_0 = x_1$, entonces tenemos que el punto medio de \overline{AB} es $(x_0, \frac{y_0+y_1}{2})$, la ecuación del segmento \overline{AB} sería $x = x_0$, la recta perpendicular será $y = \frac{y_0+y_1}{2}$. Luego tenemos que para encontrar el punto medio del lado curvo debemos resolver:

$$a_y t^3 + b_y t^2 + c_y t + d_y = y = \frac{y_0 + y_1}{2}$$

$$2a_y t^3 + 2b_y t^2 + 2c_y t + 2d_y - (y_0 + y_1) = 0 \quad (3.5.2)$$

Se puede observar que la ecuación 3.5.2 es el caso particular de la ecuación 3.5.1 cuando $x_0 = x_1$.

3.5.1.3. Expresión para hallar el punto medio del lado curvo (tercer caso)

Consideremos el caso mostrado en la sección 3.5.1.1, pero en que tenemos que $y_0 = y_1$, entonces tenemos que el punto medio de \overline{AB} es $(\frac{x_0+x_1}{2}, y_0)$, la ecuación del segmento \overline{AB} sería $y = y_0$, la recta perpendicular será $x = \frac{x_0+x_1}{2}$. Luego tenemos que para encontrar el punto medio del lado curvo debemos resolver:

$$a_x t^3 + b_x t^2 + c_x t + d_x = x = \frac{x_0 + x_1}{2}$$

$$2a_x t^3 + 2b_x t^2 + 2c_x t + 2d_x - (x_0 + x_1) = 0 \quad (3.5.3)$$

Se puede observar que la ecuación 3.5.3 es el caso particular de la ecuación 3.5.1 cuando $y_0 = y_1$.

Es necesario destacar que en los desarrollos anteriores no se consideró el caso $x_0 = x_1$ e $y_0 = y_1$, pues no estaríamos en el caso de un triángulo.

3.5.2. Método de Newton-Raphson

Sea x_0 un valor aproximado a la raíz de $f(x) = 0$, sea h la diferencia entre el valor real de la raíz α y x_0 , esto es:

$$\alpha = x_0 + h$$

La expansión de Taylor de segundo grado en x_0 es:

$$f(\alpha) = f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(\xi)$$

en donde $\xi = x_0 + \theta h$, $0 < \theta < 1$, toma un valor entre α y x_0 .

Si ignoramos el término de segundo grado, y evaluamos $f(\alpha) = 0$, tenemos:

$$f(x_0) + hf'(x_0) \approx 0$$

de donde

$$h \approx -\frac{f(x_0)}{f'(x_0)}$$

de ahí que una mejor aproximación a la raíz de $f(x) = 0$, sea:

$$f(x_1) = x_0 - \frac{f(x_0)}{f'(x_0)}$$

lo cual se puede repetir sucesivamente hasta obtener una aproximación suficientemente buena de la raíz de $f(x) = 0$.

Obtenido el valor de t , se procede a obtener por medio de las ecuaciones del lado s_{AB} el punto de bisección.

Capítulo 4

Método de Elementos Finitos

4.1. Descripción

En lo que sigue se desarrollará el método de elementos finitos, para la aplicación considerada, destacando:

- Descripción de los problemas a resolver.
- Discretización del dominio,
- Sistema de ecuaciones para el caso de elementos triangulares lineales

El método se desarrollará en forma separada para los problemas elípticos y parabólicos. Es importante destacar que este trabajo no intenta realizar un completo tratamiento del método, más información con respecto del mismo se puede encontrar consultando [6], etc.

4.2. Problemas elípticos

4.2.1. Planteamiento

Sea Ω un dominio cerrado y acotado en \mathbb{R}^2 y $\Gamma = \partial\Omega$ su frontera. Sean Γ_E y Γ_N dos subconjuntos de Γ tales que:

$$\Gamma = \Gamma_E \cup \Gamma_N \quad y \quad \Gamma_E \cap \Gamma_N = \begin{cases} 0 & o, \\ un \ conjunto \ finito \ de \ puntos \end{cases}$$

El software esta diseñado para resolver problemas elípticos de la forma :

$$\begin{cases} -\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}) + \beta u = f & en \ \Omega \\ u = g_E & en \ \Gamma_E \\ \frac{\partial u}{\partial n} = g_N & en \ \Gamma_N \end{cases} \quad (4.2.1)$$

en donde:

- u, f, α_x, α_y y β son funciones de $\Omega \rightarrow \mathbb{R}$.
- g_E (condición de borde esencial, o de Dirichlet) es una función de $\Gamma_E \rightarrow \mathbb{R}$

- g_N (condición de borde natural o de Neumann) es una función de $\Gamma_{N \rightarrow \mathfrak{R}}$
- n es el vector unitario normal a la frontera del dominio.

4.2.2. Discretización del problema

Consideremos el espacio de Hilbert $H_0^1(\Omega)$ con el producto interno usual:

$$\forall (u, v) \in H \times H \quad \langle u, v \rangle = \int_{\Omega} (u(x, y)v(x, y) + \nabla u(x, y) \cdot \nabla v(x, y)) dx dy$$

Si multiplicamos la ecuación diferencial 4.2.1 por $v \in H$, obtenemos:

$$\left(-\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}) + \beta u\right)v = fv \quad (4.2.2)$$

Podemos luego a integrar la ecuación 4.2.2 sobre el dominio Ω , obtenemos:

$$\int_{\Omega} \left(-\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}) + \beta u\right)v d\Omega = \int_{\Omega} fv d\Omega \quad (4.2.3)$$

Dado que:

$$\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x})v = \frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}v) - \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \quad (4.2.4)$$

Obtenemos al introducir 4.2.4 en 4.2.3 :

$$\int_{\Omega} \left(-\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}v) + \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}v) + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega = \int_{\Omega} fv d\Omega \quad (4.2.5)$$

Si tomamos el primer y tercer termino de la integral del lado izquierdo tenemos que:

$$\int_{\Omega} \left[\frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}v) + \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}v)\right] d\Omega = \int_{\Omega} \nabla \cdot \begin{pmatrix} \alpha_x \frac{\partial u}{\partial x}v \\ \alpha_y \frac{\partial u}{\partial y}v \end{pmatrix} d\Omega \quad (4.2.6)$$

El teorema de Green en el plano dice que si $F : \Omega \cup \Gamma \subset \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$, $F \in H^1(\Omega)$, entonces:

$$\int_{\Omega} \nabla \cdot F d\Omega = \oint_{\Gamma} F \cdot n d\sigma \quad (4.2.7)$$

en donde Γ es la frontera de Ω , y $n = \begin{pmatrix} n_x \\ n_y \end{pmatrix}$ es la normal exterior unitaria a Γ .

Aplicando el teorema de Green 4.2.7 en la ecuación 4.2.5 obtenemos:

$$\int_{\Omega} \left(\alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega = \int_{\Omega} fv d\Omega + \oint_{\Gamma} \left(\alpha_x \frac{\partial u}{\partial x}vn_x + \alpha_y \frac{\partial u}{\partial y}vn_y\right) d\sigma$$

De lo anterior podemos concluir que si u es una solución de 4.2.1, entonces u también es solución de la formulación débil (variacional):

$$\text{Hallar } u \in H \quad \text{tal que } \forall v \in H \quad a(u, v) = l(v) \quad (4.2.8)$$

en donde:

$$a(u, v) = \int_{\Omega} (\alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv) d\Omega$$

$$l(v) = \int_{\Omega} f v d\Omega + \oint_{\Gamma} (\alpha_x \frac{\partial u}{\partial x} v n_x + \alpha_y \frac{\partial u}{\partial y} v n_y) d\sigma$$

$a(u, v)$ es una forma bilineal y $l(v)$ es lineal. También se cumple que a y l son continuas, y el problema está bien planteado si $\alpha_x, \alpha_y, \beta, f \in L^2(\Omega)$.

La formulación variacional del problema 4.2.8 que acabamos de presentar es el punto de partida del método de elementos finitos. El método consiste en encontrar una solución aproximada en un subespacio de dimensión finita.

Supongamos que $\Omega \subset \mathfrak{R}$ es discretizado en un conjunto de n_t elementos triangulares t_i , para $i = 1, \dots, n_t$, que forma una triangulación T_h con n_v vértices y n_l lados L_j , para $j = 1, \dots, n_l$, que forman la discretización de la frontera Γ . Así:

$$\Omega = \bigcup_{t_i \in T_h} t_i = \bigcup_{i=1}^{n_t} t_i, \quad \Gamma = \bigcup_{j=1}^{n_l} L_j$$

Luego, si K es un triángulo $t_i \in T_h$ cualquiera, podemos plantear el problema variacional 4.2.8 como:

$$\begin{cases} \text{Encontrar } u_h \in V_h \\ \int_{\Omega} (\alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv) d\Omega = \int_{\Omega} f v d\Omega + \oint_{\Gamma} (\alpha_x \frac{\partial u}{\partial x} v n_x + \alpha_y \frac{\partial u}{\partial y} v n_y) d\sigma \quad \forall v \in V_h \end{cases}$$

en donde:

$$V_h = \{v; v \text{ es continua en } \Omega, v|_K \in P^1(K) \text{ y } v = g_E \text{ en } \Gamma_E\} \quad (4.2.9)$$

con $P^1(K)$ el espacio de los polinomios en K de grado menor o igual a 1; que es el espacio de las funciones lineales en K . V_h es un espacio de dimensiones finitas. Consideremos el espacio:

$$\tilde{V}_h = \{v; v \text{ es continua en } \Omega, v|_K \in P^1(K) \supseteq V_h,$$

se tiene que $\dim(\tilde{V}_h) = n_v$ y que $\{\phi_i\}_{i=1}^{n_v}$ definidas por:

$$\phi_i \in P^1(K) \text{ y } \phi_i(x_j) = \delta_{ij} \quad \forall i, j = 1, \dots, n_v \quad (4.2.10)$$

es una base de \tilde{V}_h . De ahí podemos escribir $\forall v \in \tilde{V}_h$

$$v(x) = \sum_{i=1}^{n_v} \eta_i \phi_i(x), \text{ en donde } \eta_i = v(x_i) \text{ para } x \in \Omega.$$

Hay que destacar que $\forall i = 1, \dots, n_v$, el soporte de la función ϕ_i se reduce al conjunto de triángulos de los cuales x_i es un vértice (el soporte de una función es el conjunto de puntos en donde la función es diferente de 0). Se puede notar que:

$$\sum_{j=1}^{n_v} \phi_j(x) = 1 \quad \forall x \in \Omega$$

Sea \tilde{U}^K una solución sobre un elemento K ,

$$\tilde{U}^K = \sum_{j=1}^{n_v} a_j \phi_j^K(x, y) \quad (4.2.11)$$

considerando las funciones de forma definidas en 4.2.10 base para el espacio V_h , tenemos

$$\begin{aligned} & \int_K (\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} + \beta \tilde{U}^K \phi_i^K) d\Omega = \\ & \int_K f \phi_i^K d\Omega + \oint_{\Gamma} (\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \phi_i^K n_x + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \phi_i^K n_y) d\sigma \quad i = 1, \dots, n_v \end{aligned} \quad (4.2.12)$$

podemos simplificar la expresión 4.2.12 si consideramos las componentes para los flujos en x y en y :

$$\tilde{\tau}_x^K = -\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \quad (4.2.13)$$

$$\tilde{\tau}_y^K = -\alpha_y \frac{\partial \tilde{U}^K}{\partial y} \quad (4.2.14)$$

luego como:

$$\tilde{\tau}_n^K = \tilde{\tau}_x^K n_x^K + \tilde{\tau}_y^K n_y^K = -\tilde{\tau}_{-n}^K$$

en que el subscriptor $-n$ indica que el flujo es hacia el interior de la frontera del elemento. Resumiendo tenemos:

$$\oint_{\Gamma} (\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \phi_i^K n_x + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \phi_i^K n_y) d\sigma = \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma$$

luego:

$$\begin{aligned} & \sum_{j=1}^{n_v} \left(\int_K \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega + \int_K \beta \phi_j^K \phi_i^K d\Omega \right) a_j = \\ & \int_K f \phi_i^K d\Omega + \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma \quad i = 1, \dots, n_v \end{aligned} \quad (4.2.15)$$

Matricialmente, estas ecuaciones para el elemento K serían:

$$\begin{bmatrix} A_{11}^K & \dots & A_{1n_v}^K \\ \vdots & \ddots & \vdots \\ A_{n_v 1}^K & \dots & A_{n_v n_v}^K \end{bmatrix} \cdot \begin{Bmatrix} a_1 \\ \vdots \\ a_{n_v} \end{Bmatrix} = \begin{Bmatrix} L_1 \\ \vdots \\ L_{n_v} \end{Bmatrix} \quad (4.2.16)$$

en donde:

$$A_{ij}^K = \int_{\Omega} \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega + \int_K \beta \phi_j^K \phi_i^K d\Omega \quad (4.2.17)$$

$$L_i^K = \int_K f \phi_i^K d\Omega + \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma \quad (4.2.18)$$

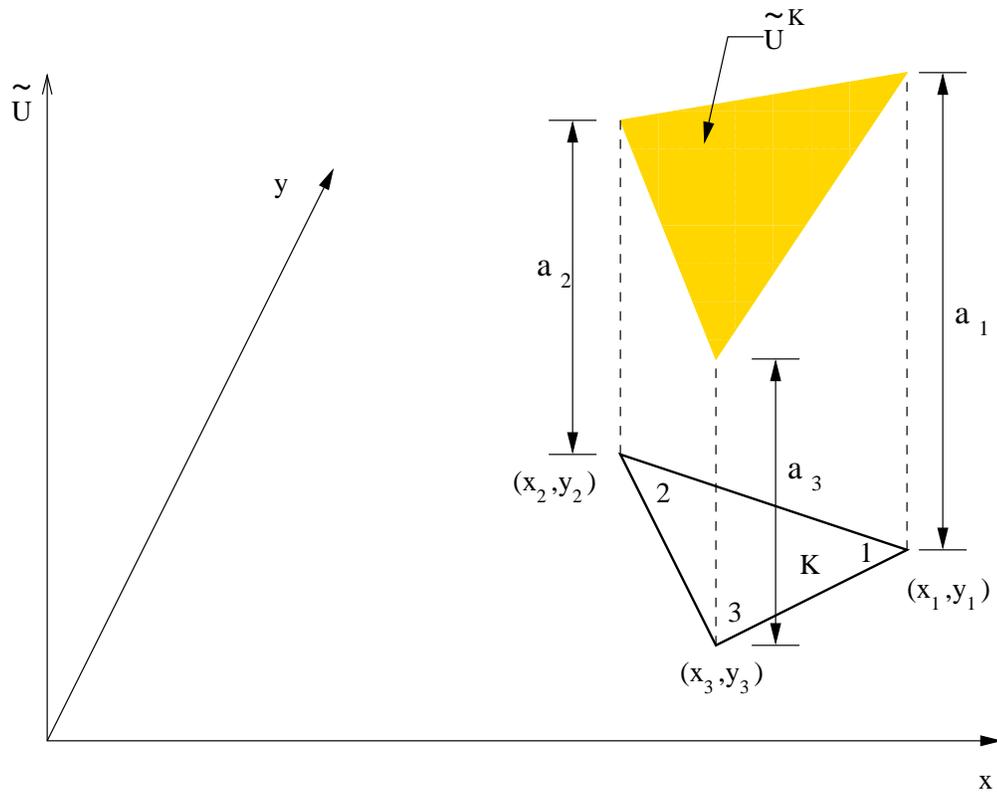


Figura 4.2.1: Elemento triangular lineal C^0 y su solución de prueba

la ecuación matricial 4.2.16 representa la discretización del problema variacional 4.2.9 para un elemento de la triangulación T_h .

Para resolver el problema 4.2.8 es necesario reunir las ecuaciones de cada elemento, en un gran sistema de ecuaciones, o lo que es lo mismo, una gran ecuación matricial. Esto lo haremos para un tipo de elemento específico.

4.2.3. Elementos triangulares lineales

Corresponden al más simple de todos los elementos que satisfacen las condiciones de convergencia. A continuación procederemos a:

- Desarrollar las funciones de forma
- Preparar las integrales para su evaluación numérica
- Preparar las expresiones para evaluar el flujo

4.2.3.1. Funciones de forma para Triángulos lineales C^0

La función solución de prueba del elemento K para el problema 4.2.9 es:

$$\tilde{U}^K = a + bx + cy \tag{4.2.19}$$

polinomio que podemos ver gráficamente en la figura 4.2.1 de la página 24.

necesitamos expresar \tilde{U}^K en función de los parámetros del elemento K . Dado que sabemos que:

$$\tilde{U}^K(x_i, y_i) = a_i \quad (4.2.20)$$

si aplicamos 4.2.20 en 4.2.19 tenemos el siguiente sistema de ecuaciones:

$$\left. \begin{aligned} a + bx_1 + cy_1 &= a_1 \\ a + bx_2 + cy_2 &= a_2 \\ a + bx_3 + cy_3 &= a_3 \end{aligned} \right\}$$

si despejamos los términos a, b y c en función de a_1, a_2 y a_3 obtenemos:

$$a = \frac{a_1(x_2y_3 - x_3y_2) + a_2(x_1y_1 - x_1y_3 + x_3y_1) + a_3(2x_1y_1 - x_1y_2 - x_2y_1)}{2\Delta}$$

$$b = \frac{a_1(y_2 - y_3) + a_2(y_3 - y_1) + a_3(y_2 - y_1)}{2\Delta}$$

$$c = \frac{a_1(x_3 - x_2) + a_2(x_3 - x_1) + a_3(x_2 - x_1)}{2\Delta}$$

En donde:

$$\Delta = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} [(x_2y_3 - x_3y_2) - (x_1y_3 - x_3y_1) + (x_1y_2 - x_2y_1)]$$

es decir, Δ es el área de elemento K .

Al introducir estos resultados en 4.2.19 y si reagrupamos términos para obtener la forma de interpolación presentada en 4.2.11, obtenemos que:

$$\phi_j^K(x, y) = \frac{a_j + b_jx + c_jy}{2\Delta} \quad \text{con } j = 1, 2, 3 \quad (4.2.21)$$

$$a_j = x_ky_l - x_ly_k$$

$$b_j = y_k - y_l$$

$$c_j = x_l - x_k$$

En 4.2.21 los subcriptores j, k, l toman los valores de 1,2,3 respectivamente para $\phi_1^K(x, y)$ y son permutados cíclicamente para $\phi_2^K(x, y)$ y $\phi_3^K(x, y)$.

Finalmente podemos expresar la función de prueba mostrada en 4.2.19, como:

$$\tilde{U}^K(x, y) = \sum_{i=1}^3 a_i \phi_i^K(x, y) \quad (4.2.22)$$

4.2.3.2. Evaluación numérica de las integrales

En lo que sigue desarrollaré la evaluación de cada una de los distintos tipos de integrales que componen la discretización 4.2.16.

Estas son:

1. Integrales de rigidez y de carga interior (página 26)
2. Integrales de flujo a través de los lados del elemento (página 27)
3. Integrales de flujo en el caso de cargas puntuales (página 28)
4. Integrales de flujo en el caso de lados internos (página 29)

Evaluación numérica de integrales de rigidez y de carga interior Para la evaluación numérica de las integrales presentes en 4.2.17 necesitamos aproximar las funciones α_x , α_y , β y f por una constante, válida para el elemento K . Para calcular dichas constantes aproximaremos los valores de las funciones respectivas por el valor que presenten en el centroide del elemento, así:

$$\begin{aligned}\alpha_x^K &= \alpha_x(x_c, y_c) \\ \alpha_y^K &= \alpha_y(x_c, y_c) \\ \beta^K &= \beta(x_c, y_c) \\ f^K &= f(x_c, y_c)\end{aligned}\tag{4.2.23}$$

con:

$$x_c = \frac{1}{3}(x_1 + x_2 + x_3) \quad y_c = \frac{1}{3}(y_1 + y_2 + y_3)$$

y x_i , y_i , $i = 1, 2, 3$, son las coordenadas de los nodos del elemento K .

Si consideramos primero las integrales de rigidez que contienen a α_x y a α_y en 4.2.17 y la función de forma descrita en 4.2.21, tenemos:

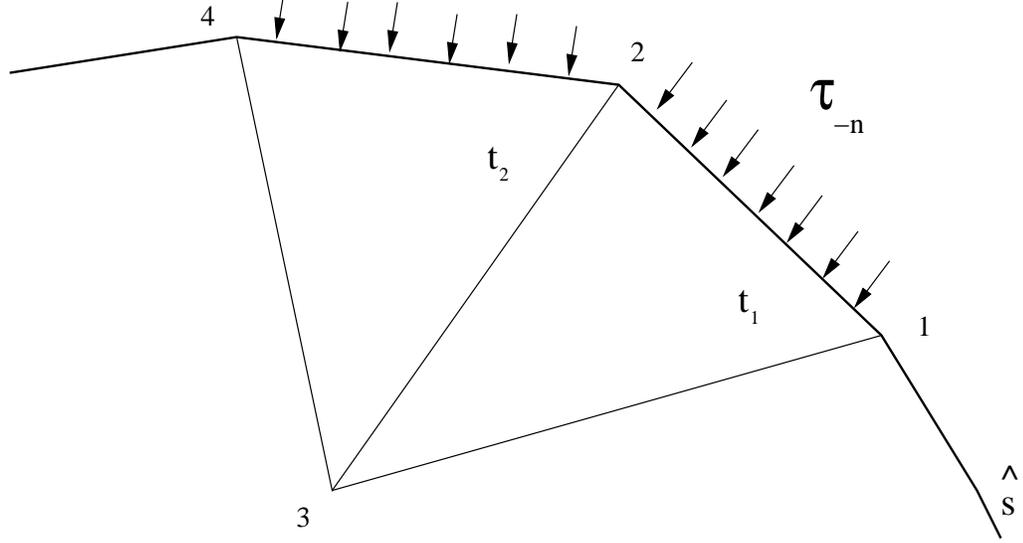
$$\begin{aligned}\int_K \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega &= \alpha_x^K \frac{b_j}{2\Delta} \frac{b_i}{2\Delta} \int_K d\Omega + \alpha_y^K \frac{c_j}{2\Delta} \frac{c_i}{2\Delta} \int_K d\Omega \\ &= \frac{\alpha_x^K}{4\Delta} b_j b_i + \frac{\alpha_y^K}{4\Delta} c_j c_i\end{aligned}\tag{4.2.24}$$

Si consideramos la integral de rigidez que contiene a β , dado que las funciones de forma $\phi_j^K(x, y)$ corresponden a las coordenadas de área ζ_j de un triángulo y como:

$$\int_K \zeta_1^l \zeta_2^m \zeta_3^n d\Omega = \frac{l!m!n!}{(l+m+n+2)!} 2\Delta$$

entonces:

Figura 4.2.2: Flujo a lo largo de la frontera de dos elementos con un lado interno común



$$\int_K \beta \phi_j^K \phi_i^K d\Omega = \beta^K \int_K \phi_j^K \phi_i^K d\Omega = \beta^K \int_K \zeta_j \zeta_i d\Omega = \begin{cases} \frac{\beta^K \Delta}{6} & i = j \\ \frac{\beta^K \Delta}{12} & i \neq j \end{cases} \quad (4.2.25)$$

Si consideramos la integral de carga, utilizando la aproximación de la función f a la constante f^K , y aprovechando la fórmula de integración basada en las coordenadas de área, tenemos que:

$$\int_K f \phi_i^K d\Omega = f^K \int_K \phi_i^K d\Omega = \frac{f^K \Delta}{3} \quad (4.2.26)$$

Evaluación numérica de integrales de flujo a través de los lados del elemento

Consideremos el caso de dos elementos t_1 y t_2 que compartan un lado común y tengan lados en la frontera de Ω , esto es,

$$\exists L_i \in t_j / L_i \text{ lado de } t_j, L_i \subset \Gamma; i = 1, 2, 3; j = 1, 2$$

Sea $\tau_{-n}(\hat{s})$ la componente normal interna del flujo aplicado en la frontera (sea \hat{s} la coordenada a lo largo de la frontera). Tenemos que $\tilde{\tau}_{-n}^{t_1} = \tau_{-n}(\hat{s})|_{t_1}$ y que $\tilde{\tau}_{-n}^{t_2} = \tau_{-n}(\hat{s})|_{t_2}$.

Para mayor claridad la figura 4.2.2 de la página 27 gráfica la situación indicada.

La integral del flujo para el elemento t_1 se puede descomponer como:

$$\oint_{t_1} \tilde{\tau}_{-n}^{t_1} \phi_i^{t_1} d\sigma = \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_i^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_i^{t_1} d\sigma + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_i^{t_1} d\sigma$$

en que los límites 1,2 y 3 corresponden a los nodos del elemento t_1 así, el vector del flujo lo podemos escribir como:

$$\begin{bmatrix} \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma \\ \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma \\ \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_3^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_3^{t_1} d\sigma + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_3^{t_1} d\sigma \end{bmatrix}$$

dado que las funciones de forma toman el valor 0 en el lado opuesto a su nodo la matriz anterior queda como:

$$\begin{bmatrix} \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma & & + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma \\ \int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma & + \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma & \\ & \int_2^3 \tilde{\tau}_{-n}^{t_1} \phi_3^{t_1} d\sigma & + \int_3^1 \tilde{\tau}_{-n}^{t_1} \phi_3^{t_1} d\sigma \end{bmatrix} \quad (4.2.27)$$

En que la primera columna corresponde a integrales de flujo externo y las restantes a integrales de flujos internos. Para evaluar el flujo en los lados de la frontera supondremos que podemos aproximar $\tilde{\tau}_{-n}^{t_1}$ como una constante τ^{t_1} que resulta de evaluar la primera función en el punto medio del lado, esto es:

$$\tau^{t_1} = \tilde{\tau}_{-n}^{t_1}(x_m, y_m), \quad \text{con } x_m = \frac{x_1 + x_2}{2} \quad e \quad y_m = \frac{y_1 + y_2}{2}$$

Si aprovechamos la siguiente fórmula de integración para lados de un triángulo:

$$\int_i^j \zeta_i^l \zeta_j^m d\sigma = \frac{l!m!}{(l+m+1)!} L_{ij}, \quad \text{con } L_{ij} = \int_i^j d\sigma$$

resumiendo:

$$\int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma = \frac{\tau^{t_1} L_{12}}{2} \quad (4.2.28)$$

$$\int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma = \frac{\tau^{t_1} L_{12}}{2} \quad (4.2.29)$$

Lo que en la práctica resulta en que, el flujo sobre el lado, se reparte en partes iguales sobre los nodos del mismo.

Evaluación numérica de integrales de flujo en el caso de cargas puntuales En el caso en que se presenten situaciones de flujos concentrados en algún punto de la frontera, como el que muestra la figura 4.2.3 de la página 29, entonces podemos representar, en el caso de que sea el nodo 1 el que concentre el flujo, la función del flujo como:

$$\tau_{-n}(\hat{s}) = \tau_1 \delta(\hat{s} - \hat{s}_1) \quad (4.2.30)$$

Si reemplazamos la expresión 4.2.30 en las integrales de la primera columna de la matriz 4.2.27 obtendremos:

$$\int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_1^{t_1} d\sigma = \tau_1 \quad (4.2.31)$$

$$\int_1^2 \tilde{\tau}_{-n}^{t_1} \phi_2^{t_1} d\sigma = 0 \quad (4.2.32)$$

que en la práctica aplica todo el flujo al nodo 1.

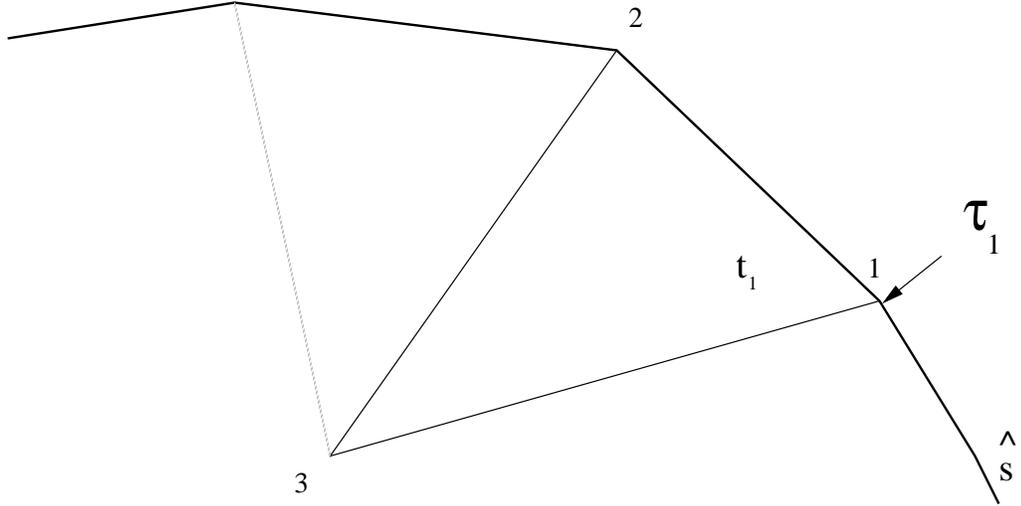


Figura 4.2.3: Flujo concentrado en un punto de la frontera

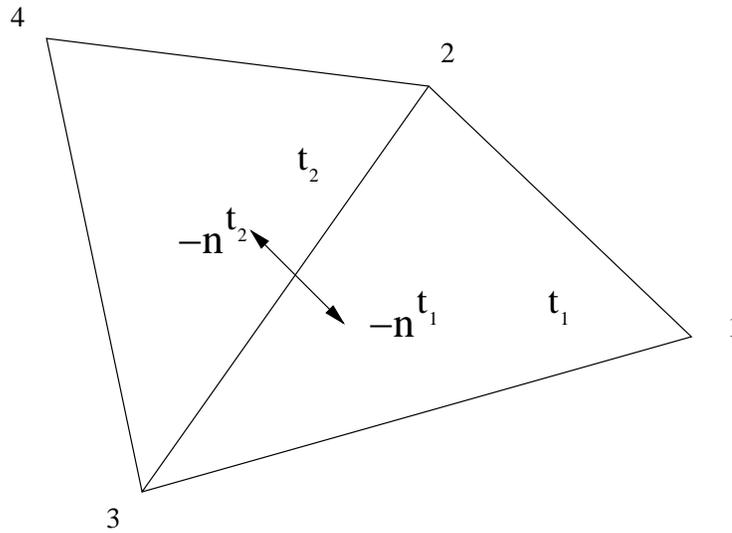


Figura 4.2.4: Flujo para un lado interno entre elementos

Evaluación numérica de integrales de flujo en el caso de lados internos Para evaluar el caso del flujo interno, consideremos la situación de dos elementos t_1 y t_2 vecinos, esto es que comparten un lado común, como se representa la figura 4.2.4 de la página 29.

El flujo en estos elementos queda representado por la ecuación 4.2.27 para el elemento t_1 y para el elemento t_2 , si consideramos a σ a la coordenada a lo largo del lado del elemento, tenemos que el vector del flujo queda presentado por:

$$\left[\begin{array}{l} \int_2^4 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_2^{t_2} d\sigma, \quad + \quad \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_2^{t_2} d\sigma, \\ \int_2^4 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_4^{t_2} d\sigma, \quad + \quad \int_4^3 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_3^{t_2} d\sigma, \quad + \quad \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_3^{t_2} d\sigma, \\ \int_2^4 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_4^{t_2} d\sigma, \quad + \quad \int_4^3 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_4^{t_2} d\sigma, \end{array} \right] \quad (4.2.33)$$

Si ensamblamos las ecuaciones 4.2.27 y las ecuaciones 4.2.33 obtenemos:

$$\left[\begin{array}{l} \int_1^2 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_1^{t_1} d\sigma + \int_3^1 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_1^{t_1} d\sigma \\ \int_1^2 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_2^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_2^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_2^{t_2} d\sigma, + \int_2^4 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_2^{t_2} d\sigma, \\ \int_3^1 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_3^{t_1} d\sigma + \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_3^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_3^{t_2} d\sigma, + \int_4^3 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_3^{t_2} d\sigma, \\ \int_2^4 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_4^{t_2} d\sigma, + \int_4^3 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_4^{t_2} d\sigma, \end{array} \right] \quad (4.2.34)$$

Pero dada la notación mostrada en la figura 4.2.4 de la página 29 tenemos que para el lado común L_{23} se cumple :

$$d\sigma = -d\sigma',$$

$$n^{t_1} = -n^{t_2}$$

$$\tilde{\tau}_{-n^{t_2}}^{t_2} = -\tilde{\tau}_{-n^{t_1}}^{t_2}$$

Ademas dada las condiciones de continuidad de las funciones de forma, tenemos que:

$$\phi_2^{t_1} |_{L_{23}} = \phi_2^{t_2} |_{L_{23}}$$

y

$$\phi_3^{t_1} |_{L_{23}} = \phi_3^{t_2} |_{L_{23}}$$

Considerando lo anterior en la ecuación 4.2.34 para aquellas partes que corresponde al lado común L_{23} , tenemos para las ecuaciones asociadas al nodo 2 que:

$$\begin{aligned} \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_2^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_2^{t_2} d\sigma, &= \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_2^{t_1} d\sigma + \int_3^2 (-\tilde{\tau}_{-n^{t_1}}^{t_2}) \phi_2^{t_1} (-d\sigma) \\ &= \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_2^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_1}}^{t_2} \phi_2^{t_1} d\sigma \\ &= \int_2^3 (\tilde{\tau}_{-n^{t_1}}^{t_1} - \tilde{\tau}_{-n^{t_1}}^{t_2}) \phi_2^{t_1} d\sigma \end{aligned} \quad (4.2.35)$$

y para las ecuaciones asociadas al nodo 3 que:

$$\begin{aligned} \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_3^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_2}}^{t_2} \phi_3^{t_2} d\sigma, &= \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_3^{t_1} d\sigma + \int_3^2 -(\tilde{\tau}_{-n^{t_1}}^{t_2}) \phi_3^{t_1} - (d\sigma) \\ &= \int_2^3 \tilde{\tau}_{-n^{t_1}}^{t_1} \phi_3^{t_1} d\sigma + \int_3^2 \tilde{\tau}_{-n^{t_1}}^{t_2} \phi_3^{t_1} d\sigma \\ &= \int_2^3 (\tilde{\tau}_{-n^{t_1}}^{t_1} - \tilde{\tau}_{-n^{t_1}}^{t_2}) \phi_3^{t_1} d\sigma \end{aligned} \quad (4.2.36)$$

Si la carga interior f no está concentrada a lo largo del lado L_{23} , o es continua a lo largo del lado, tendremos que:

$$\tilde{\tau}_{-n^{t_1}}^{t_1} - \tilde{\tau}_{-n^{t_1}}^{t_2} = 0 \quad (4.2.37)$$

si la carga interior f se concentra a lo largo del lado L_{23} entonces tendremos que:

$$\tilde{\tau}_{-n^{t_1}}^{t_1} - \tilde{\tau}_{-n^{t_1}}^{t_2} = \tau_{23} \quad (4.2.38)$$

en que τ_{23} corresponderá a una condición de borde del problema.

4.2.3.3. Expresión para el flujo

De las ecuaciones 4.2.13 y 4.2.14 prepararé las expresiones del flujo, así si t_1 es un elemento tal que $t_1 \subseteq \Omega$, tenemos que:

$$\begin{aligned}\tilde{\tau}_x^{t_1}(x, y) &= -\alpha_x(x, y) \frac{\partial \tilde{U}^{t_1}(x, y)}{\partial x} \\ \tilde{\tau}_y^{t_1}(x, y) &= -\alpha_y(x, y) \frac{\partial \tilde{U}^{t_1}(x, y)}{\partial y}\end{aligned}\tag{4.2.39}$$

Usando la ecuación 4.2.23 tenemos que:

$$\begin{aligned}\frac{\partial \tilde{U}^{t_1}(x, y)}{\partial x} &= \sum_{i=1}^3 a_i \frac{\partial \phi_i^{t_1}(x, y)}{\partial x} = \sum_{i=1}^3 a_i \frac{b_i}{2\Delta} \\ \frac{\partial \tilde{U}^{t_1}(x, y)}{\partial y} &= \sum_{i=1}^3 a_i \frac{\partial \phi_i^{t_1}(x, y)}{\partial y} = \sum_{i=1}^3 a_i \frac{c_i}{2\Delta}\end{aligned}\tag{4.2.40}$$

luego

$$\begin{aligned}\tilde{\tau}_x^{t_1}(x, y) &= -\alpha_x(x, y) \sum_{i=1}^3 a_i \frac{b_i}{2\Delta} \\ \tilde{\tau}_y^{t_1}(x, y) &= -\alpha_y(x, y) \sum_{i=1}^3 a_i \frac{c_i}{2\Delta}\end{aligned}\tag{4.2.41}$$

Ahora bien, para estimar el valor del flujo en el elemento t_1 , lo aproximaremos por el valor de las funciones del flujo en el centroide del triángulo. Así el valor de flujo óptimo a través del elemento será:

$$\begin{aligned}\tilde{\tau}_x^{t_1} &= -\alpha_x(x_c, y_c) \sum_{i=1}^3 a_i \frac{b_i}{2\Delta} \\ \tilde{\tau}_y^{t_1} &= -\alpha_y(x_c, y_c) \sum_{i=1}^3 a_i \frac{c_i}{2\Delta}\end{aligned}\tag{4.2.42}$$

Para calcular el valor del flujo en los nodos, dado que el flujo a través de los elementos es constante, obtendremos el flujo en el nodo como el promedio de los flujos de los elementos adyacentes a él. En el caso del ejemplo mostrado por la figura 4.2.5 de la página 32. tendremos que:

$$\begin{aligned}\tilde{\tau}_x|_5 &= \sum_{i=1}^4 \tilde{\tau}_x^{t_i} \\ \tilde{\tau}_y|_5 &= \sum_{i=1}^4 \tilde{\tau}_y^{t_i}\end{aligned}$$

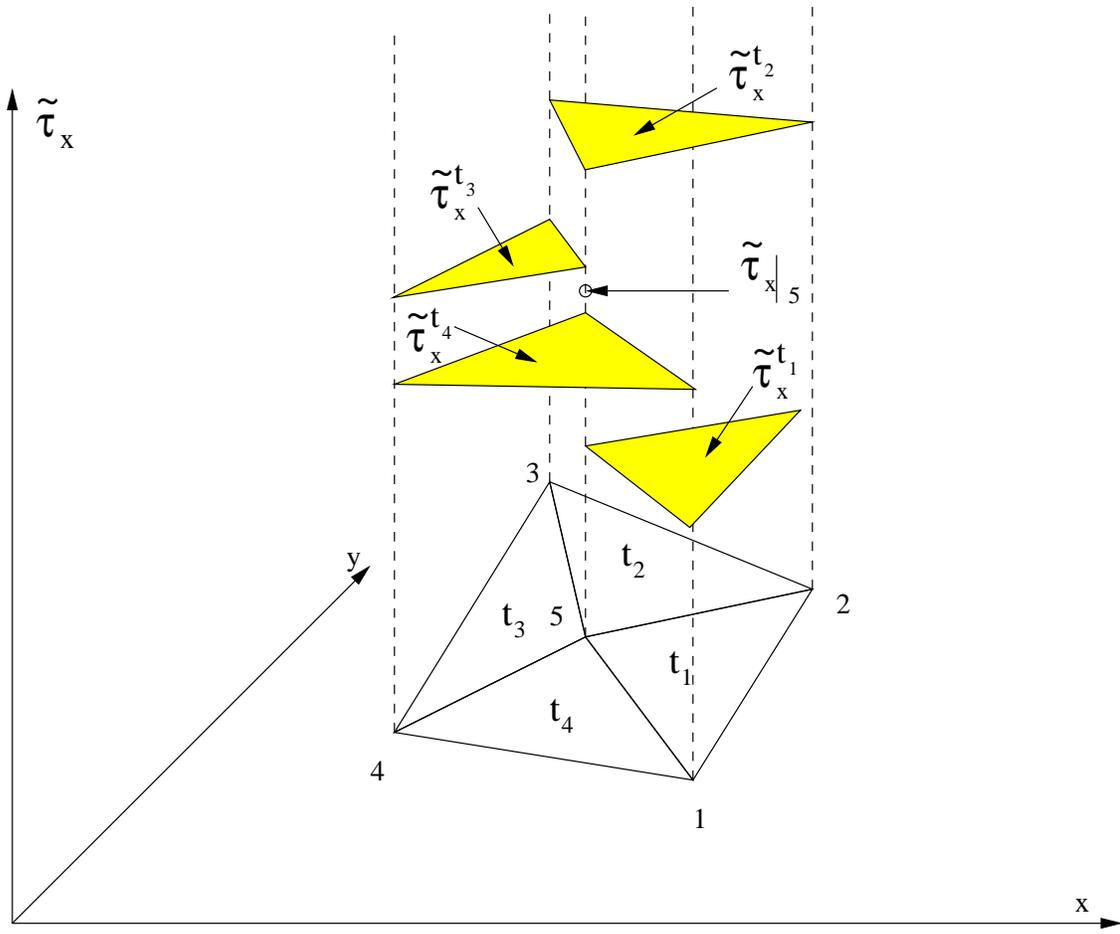


Figura 4.2.5: Ejemplo de flujo nodal

4.2.4. Resolución numérica

Para encontrar la solución al problema 4.2.8 de la página 21 es necesario ensamblar las matrices de cada elemento indicada en 4.2.16 de la página 23. Para ello se evalúan para cada elemento las ecuaciones de la matriz de rigidez 4.2.24, 4.2.25 y del lado derecho 4.2.26. De acuerdo a las condiciones de borde del problema se evaluarán las expresiones para el flujo de cada elemento (ecuaciones 4.2.28, 4.2.29, 4.2.31, 4.2.32, 4.2.37 y 4.2.38), añadiéndolos al lado derecho de la ecuación. Una vez evaluados todos los elementos, se procede a resolver numéricamente el sistema de ecuaciones. Obtenido el resultado, se procederá a evaluar los indicadores del error para cada elemento, que serán usados para determinar cual o cuales de ellos deberán ser incluidos en el refinamiento de la malla, para una nueva aplicación del método.

4.3. Problemas parabólicos

4.3.1. Planteamiento

Sea Ω un dominio cerrado y acotado en \mathfrak{R}^2 y $\Gamma = \partial\Omega$ su frontera. Sean Γ_E y Γ_N dos subconjuntos de Γ tales que:

$$\Gamma = \Gamma_E \cup \Gamma_N \quad y \quad \Gamma_E \cap \Gamma_N = \begin{cases} 0 & o, \\ un \text{ conjunto finito de puntos} \end{cases}$$

sea $T = [t_0, t_f) \subset \mathfrak{R}^+$, con $t_0 \geq 0$ y $t > 0$.

El software esta diseñado para resolver problemas parabólicos de la forma :

$$\begin{cases} \gamma_t \frac{\partial U}{\partial t} - \frac{\partial}{\partial x}(\alpha_x \frac{\partial U}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial U}{\partial y}) + \beta U = f & en (\Omega, T) \\ U = g_E & en (\Gamma_E, T) \\ \frac{\partial U}{\partial n} = g_N & en (\Gamma_N, T) \\ U = u_0 & en t = t_0 \end{cases} \quad (4.3.1)$$

en donde:

- $U, f, \gamma_t, \alpha_x, \alpha_y$ y β son funciones de $(\Omega, t) \rightarrow (\mathfrak{R}, T)$
- u_0 es una función de $\Omega \rightarrow \mathfrak{R}$
- g_E (condición de borde esencial, o de Dirichlet) es una función de $(\Gamma_E, t) \rightarrow (\mathfrak{R}, T)$
- g_N (condición de borde natural o de Neumann) es una función de $(\Gamma_N, t) \rightarrow (\mathfrak{R}, T)$
- n es el vector unitario normal a la frontera del dominio.

4.3.2. Discretización del problema

Consideremos el espacio de Hilbert $H_0^1(\Omega)$ con el producto interno usual:

$$\forall (u, v) \in H \times H \quad \langle u, v \rangle = \int_{\Omega} (u(x, y)v(x, y) + \nabla u(x, y) \cdot \nabla v(x, y)) dx dy$$

Si multiplicamos la ecuación diferencial 4.3.1 por $v \in H$, obtenemos:

$$\left(\gamma_t \frac{\partial U}{\partial t} - \frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}) + \beta u\right)v = fv \quad (4.3.2)$$

Podemos luego a integrar la ecuación 4.3.2 sobre el dominio Ω , obtenemos:

$$\int_{\Omega} \left(\gamma_t \frac{\partial U}{\partial t} - \frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y}) + \beta u\right)v d\Omega = \int_{\Omega} fv d\Omega \quad (4.3.3)$$

Si introducimos 4.2.4 de la página 21 en 4.3.3 :

$$\int_{\Omega} \left(\gamma_t \frac{\partial U}{\partial t} v - \frac{\partial}{\partial x}(\alpha_x \frac{\partial u}{\partial x} v) + \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} - \frac{\partial}{\partial y}(\alpha_y \frac{\partial u}{\partial y} v) + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega = \int_{\Omega} fv d\Omega \quad (4.3.4)$$

Aplicando el teorema de Green 4.2.7 en la ecuación 4.3.4 obtenemos:

$$\int_{\Omega} \left(\gamma_t \frac{\partial U}{\partial t} v + \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega = \int_{\Omega} fv d\Omega + \oint_{\Gamma} \left(\alpha_x \frac{\partial u}{\partial x} v n_x + \alpha_y \frac{\partial u}{\partial y} v n_y\right) d\sigma$$

De lo anterior podemos concluir que si u es una solución de 4.3.1, entonces u también es solución de la formulación débil (variacional):

$$\text{Hallar } u \in H \quad \text{tal que } \forall v \in H \quad a(u, v) = l(v) \quad (4.3.5)$$

en donde:

$$a(u, v) = \int_{\Omega} \left(\gamma_t \frac{\partial U}{\partial t} v + \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega$$

$$l(v) = \int_{\Omega} fv d\Omega + \oint_{\Gamma} \left(\alpha_x \frac{\partial u}{\partial x} v n_x + \alpha_y \frac{\partial u}{\partial y} v n_y\right) d\sigma$$

$a(u, v)$ es una forma bilineal y $l(v)$ es lineal. También se cumple que a y l son continuas, y el problema está bien planteado si $\alpha_x, \alpha_y, \beta, f \in L^2(\Omega)$.

La formulación variacional del problema 4.3.5 que acabamos de presentar es el punto de partida del método de elementos finitos. El método consiste en encontrar una solución aproximada en un subespacio de dimensión finita.

Supongamos que $\Omega \subset \mathfrak{R}$ es discretizado en un conjunto de n_t elementos triangulares t_i , para $i = 1, \dots, n_t$, que forma una triangulación T_h con n_v vértices y n_l lados L_j , para $j = 1, \dots, n_l$, que forman la discretización de la frontera Γ . Así:

$$\Omega = \bigcup_{t_i \in T_h} t_i = \bigcup_{i=1}^{n_t} t_i, \quad \Gamma = \bigcup_{j=1}^{n_l} L_j$$

Luego, si K es un triángulo $t_i \in T_h$ cualquiera, podemos plantear el problema variacional 4.3.5 como:

$$\left\{ \begin{array}{l} \text{Encontrar } u_h \in V_h \\ \int_{\Omega} \left(\gamma_t \frac{\partial U}{\partial t} v + \alpha_x \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \alpha_y \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \beta uv\right) d\Omega = \\ \int_{\Omega} fv d\Omega + \oint_{\Gamma} \left(\alpha_x \frac{\partial u}{\partial x} v n_x + \alpha_y \frac{\partial u}{\partial y} v n_y\right) d\sigma \quad \forall v \in V_h \end{array} \right.$$

en donde:

$$V_h = \{v; v \text{ es continua en } \Omega, v|_K \in P^1(K) \text{ y } v = g_E \text{ en } \Gamma_E\} \quad (4.3.6)$$

con $P^1(K)$ el espacio de los polinomios en K de grado menor o igual a 1; que es el espacio de las funciones lineales en K . V_h es un espacio de dimensiones finitas. Consideremos el espacio:

$$\tilde{V}_h = \{v; v \text{ es continua en } \Omega, v|_K \in P^1(K) \supseteq V_h,$$

se tiene que $\dim(\tilde{V}_h) = n_v$ y que $\{\phi_i\}_{i=1}^{n_v}$ definidas por:

$$\phi_i \in P^1(K) \text{ y } \phi_i(x_j) = \delta_{ij} \forall i, j = 1, \dots, n_v \quad (4.3.7)$$

es una base de \tilde{V}_h . De ahí podemos escribir $\forall v \in \tilde{V}_h$

$$v(x) = \sum_{i=1}^{n_v} \eta_i \phi_i(x), \text{ en donde } \eta_i = v(x_i) \text{ para } x \in \Omega.$$

Hay que destacar que $\forall i = 1, \dots, n_v$, el soporte de la función ϕ_i se reduce al conjunto de triángulos de los cuales x_i es un vértice (el soporte de una función es el conjunto de puntos en donde la función es diferente de 0). Se puede notar que:

$$\sum_{j=1}^{n_v} \phi_j(x) = 1 \forall x \in \Omega$$

Sea \tilde{U}^K una solución sobre un elemento K ,

$$\tilde{U}^K = \sum_{j=1}^{n_v} a_j(t) \phi_j^K(x, y) \quad (4.3.8)$$

considerando las funciones de forma definidas en 4.3.7 base para el espacio V_h , tenemos

$$\begin{aligned} & \int_K (\gamma_t \frac{\partial \tilde{U}^K}{\partial t} \phi_i^K + \alpha_x \frac{\partial \tilde{U}^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} + \beta \tilde{U}^K \phi_i^K) d\Omega = \\ & \int_{\Omega} f \phi_i^K d\Omega + \int_{\Gamma} (\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \phi_i^K n_x + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \phi_i^K n_y) d\sigma \quad i = 1, \dots, n_v \end{aligned} \quad (4.3.9)$$

podemos simplificar la expresión 4.3.9 si consideramos las componentes para los flujos en x y en y :

$$\tilde{\tau}_x^K = -\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \quad (4.3.10)$$

$$\tilde{\tau}_y^K = -\alpha_y \frac{\partial \tilde{U}^K}{\partial y} \quad (4.3.11)$$

luego como:

$$\tilde{\tau}_n^K = \tilde{\tau}_x^K n_x^K + \tilde{\tau}_y^K n_y^K = -\tilde{\tau}_{-n}^K$$

en que el subscritor $-n$ indica que el flujo es hacia el interior de la frontera del elemento. Resumiendo tenemos:

$$\oint_{\Gamma} \left(\alpha_x \frac{\partial \tilde{U}^K}{\partial x} \phi_i^K n_x + \alpha_y \frac{\partial \tilde{U}^K}{\partial y} \phi_i^K n_y \right) d\sigma = \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma$$

luego:

$$\begin{aligned} & \sum_{j=1}^{n_v} \left(\int_K \gamma_t \phi_j^K \phi_i^K d\Omega \right) \frac{da_j(t)}{dt} + \\ & \sum_{j=1}^{n_v} \left(\int_K \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega + \int_K \beta \phi_j^K \phi_i^K d\Omega \right) a_j(t) = \\ & \int_K f \phi_i^K d\Omega + \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma \quad i = 1, \dots, n_v \end{aligned} \quad (4.3.12)$$

Matricialmente, estas ecuaciones para el elemento K serían:

$$\begin{bmatrix} G_{11}^K & \cdots & G_{1n_v}^K \\ \vdots & \ddots & \vdots \\ G_{n_v 1}^K & \cdots & G_{n_v n_v}^K \end{bmatrix} \cdot \begin{Bmatrix} \frac{da_1}{dt} \\ \vdots \\ \frac{da_{n_v}}{dt} \end{Bmatrix} + \begin{bmatrix} A_{11}^K & \cdots & A_{1n_v}^K \\ \vdots & \ddots & \vdots \\ A_{n_v 1}^K & \cdots & A_{n_v n_v}^K \end{bmatrix} \cdot \begin{Bmatrix} a_1 \\ \vdots \\ a_{n_v} \end{Bmatrix} = \begin{Bmatrix} L_1 \\ \vdots \\ L_{n_v} \end{Bmatrix} \quad (4.3.13)$$

en donde:

$$G_{ij}^K = \int_K \gamma_t \phi_j^K \phi_i^K d\Omega \quad (4.3.14)$$

$$A_{ij}^K = \int_{\Omega} \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega + \int_K \beta \phi_j^K \phi_i^K d\Omega \quad (4.3.15)$$

$$L_i^K = \int_K f \phi_i^K d\Omega + \oint_K \tilde{\tau}_{-n}^K \phi_i^K d\sigma \quad (4.3.16)$$

la ecuación matricial 4.3.13 representa la discretización del problema variacional 4.3.6 para un elemento de la triangulación T_h .

Para resolver el problema 4.3.5 es necesario reunir las ecuaciones de cada elemento, en un gran sistema de ecuaciones, o lo que es lo mismo, una gran ecuación matricial. Esto lo haremos para un tipo de elemento específico.

4.3.3. Elementos triangulares lineales

Corresponden al más simple de todos los elementos que satisfacen las condiciones de convergencia. A continuación procederemos a:

- Desarrollar las funciones de forma
- Preparar las integrales para su evaluación numérica
- Preparar las expresiones para evaluar el flujo

4.3.3.1. Funciones de forma para Triángulos lineales C^0

La función solución de prueba del elemento K para el problema 4.3.6, se obtiene de una manera similar a la realizada en la sección 4.2.3.1 de la página 24, y esta es:

$$\tilde{U}^K(x, y, t) = \sum_{i=1}^3 a_i(t) \phi_i^K(x, y) \quad (4.3.17)$$

4.3.3.2. Evaluación numérica de las integrales

En lo que sigue desarrollaré la evaluación de cada una de los distintos tipos de integrales que componen la discretización 4.3.13.

Estas son:

1. Integrales de rigidez y de carga interior
2. Integrales de flujo a través de los lados del elemento¹
3. Integrales de flujo en el caso de cargas puntuales²
4. Integrales de flujo en el caso de lados internos³

Evaluación numérica de integrales de rigidez y de carga interior Para la evaluación numérica de las integrales presentes en 4.3.14, 4.3.15 y 4.3.16 necesitamos aproximar las funciones γ_t , α_x , α_y , β y f por una constante, válida para el elemento K . Para calcular dichas constantes aproximaremos los valores de las funciones respectivas por el valor que presenten en el centroide del elemento, así:

$$\begin{aligned} \gamma_t^K &= \gamma_t(x_c, y_c) \\ \alpha_x^K &= \alpha_x(x_c, y_c) \\ \alpha_y^K &= \alpha_y(x_c, y_c) \\ \beta^K &= \beta(x_c, y_c) \\ f^K &= f(x_c, y_c) \end{aligned} \quad (4.3.18)$$

con:

$$x_c = \frac{1}{3}(x_1 + x_2 + x_3) \quad y_c = \frac{1}{3}(y_1 + y_2 + y_3)$$

y x_i , y_i , $i = 1, 2, 3$, son las coordenadas de los nodos del elemento K .

Si consideramos primero las integrales de rigidez que contienen a α_x y a α_y en 4.3.15 y la función de forma descrita en 4.2.21, de la página 25, tenemos:

¹Este caso es idéntico al desarrollado en 4.2.3.2 de la página 27

²Este caso es idéntico al desarrollado en 4.2.3.2 de la página 28

³Este caso es idéntico al desarrollado en 4.2.3.2 de la página 29

$$\begin{aligned}
\int_K \alpha_x \frac{\partial \phi_j^K}{\partial x} \frac{\partial \phi_i^K}{\partial x} d\Omega + \int_K \alpha_y \frac{\partial \phi_j^K}{\partial y} \frac{\partial \phi_i^K}{\partial y} d\Omega &= \alpha_x^K \frac{b_j}{2\Delta} \frac{b_i}{2\Delta} \int_K d\Omega + \alpha_y^K \frac{c_j}{2\Delta} \frac{c_i}{2\Delta} \int_K d\Omega \\
&= \frac{\alpha_x^K}{4\Delta} b_j b_i + \frac{\alpha_y^K}{4\Delta} c_j c_i
\end{aligned} \tag{4.3.19}$$

Si consideramos la integral de rigidez que contiene a β , dado que las funciones de forma $\phi_j^K(x, y)$ corresponden a las coordenadas de área ζ_j de un triángulo y como:

$$\int_K \zeta_1^l \zeta_2^m \zeta_3^n d\Omega = \frac{l!m!n!}{(l+m+n+2)!} 2\Delta$$

entonces:

$$\int_K \beta \phi_j^K \phi_i^K d\Omega = \beta^K \int_K \phi_j^K \phi_i^K d\Omega = \beta^K \int_K \zeta_j \zeta_i d\Omega = \begin{cases} \frac{\beta^K \Delta}{6} & i = j \\ \frac{\beta^K \Delta}{12} & i \neq j \end{cases} \tag{4.3.20}$$

Se puede apreciar que la matriz G es similar a la evaluación a la realizada en el cálculo de las integrales de β . De ahí tenemos que:

$$\int_K \gamma_t \phi_j^K \phi_i^K d\Omega = \gamma_t^K \int_K \phi_j^K \phi_i^K d\Omega = \gamma_t^K \int_K \zeta_j \zeta_i d\Omega = \begin{cases} \frac{\gamma_t^K \Delta}{6} & i = j \\ \frac{\gamma_t^K \Delta}{12} & i \neq j \end{cases} \tag{4.3.21}$$

Si consideramos la integral de carga, utilizando la aproximación de la función f a una constante f^K definida en 4.3.18, y aprovechando la fórmula de integración basada en las coordenadas de área, tenemos que:

$$\int_K f \phi_i^K d\Omega = f^K \int_K \phi_i^K d\Omega = \frac{f^K \Delta}{3} \tag{4.3.22}$$

Evaluación numérica de integrales de flujo La evaluación numérica de las condiciones de borde del problema sigue el mismo desarrollo que lo revisado en el caso elíptico, ver páginas 29 y siguientes. Obteniendo como resultado las ecuaciones 4.2.28, 4.2.29, 4.2.31, 4.2.32, 4.2.37 y 4.2.38.

4.3.3.3. Expresión para el flujo

De las ecuaciones 4.3.10 y 4.3.11 prepararé las expresiones del flujo, así si t_1 es un elemento tal que $t_1 \subseteq \Omega$, tenemos que:

$$\begin{aligned}
\tilde{\tau}_x^{t_1}(x, y, t) &= -\alpha_x(x, y, t) \frac{\partial \tilde{U}^{t_1}(x, y, t)}{\partial x} \\
\tilde{\tau}_y^{t_1}(x, y, t) &= -\alpha_y(x, y, t) \frac{\partial \tilde{U}^{t_1}(x, y, t)}{\partial y}
\end{aligned} \tag{4.3.23}$$

Si siguiendo un desarrollo similar al mostrado en la sección 4.2.3.3 de la página 31 obtenemos que para estimar el valor del flujo en el elemento t_1 , lo aproximaremos por el valor de las funciones del flujo en el centroide del triángulo. Así el valor de flujo óptimo a través del elemento será:

$$\begin{aligned}\tilde{\tau}_x^{t_1}(t) &= -\alpha_x(x_c, y_c, t) \sum_{i=1}^3 a_i(t) \frac{b_i}{2\Delta} \\ \tilde{\tau}_y^{t_1}(t) &= -\alpha_y(x_c, y_c, t) \sum_{i=1}^3 a_i(t) \frac{c_i}{2\Delta}\end{aligned}\tag{4.3.24}$$

Para calcular el valor del flujo en los nodos, el procedimiento es similar al mostrado en la sección 4.2.3.3, puesto que dado que el flujo a través de los elementos es constante, obtendremos el flujo en el nodo como el promedio de los flujos de los elementos adyacentes a él.

4.3.4. Resolución numérica

Para encontrar la solución al problema 4.3.5 de la página 34 es necesario ensamblar las matrices de cada elemento indicada en 4.3.13 de la página 36. Para ello se evalúan para cada elemento las ecuaciones de la matriz de rigidez 4.3.19, 4.3.20, 4.3.21, del lado derecho 4.3.22 y de las condiciones de borde (página 38). , y se deben añadir al lado derecho de la ecuación. Una vez evaluados todos los elementos, tenemos que la ecuación 4.3.13 corresponde a una EDO (Ecuación Diferencial Ordinaria), en el tiempo. Para resolver el problema para un tiempo $t = n + 1$, dado que ya lo tenemos resuelto para un tiempo $t = n$ (inicialmente $t = t_0$, condición inicial del problema) procedemos con una aproximación implícita con peso variable para el tiempo, de modo que en la ecuación inicial:

$$G \frac{\partial a}{\partial t} + Aa = L$$

aproximaremos en un paso de tiempo t por un parámetro θ / $\theta \in [0, 1]$, de modo que tendremos:

$$G \left\{ \frac{\partial a}{\partial t} \right\}_\theta + A \{a\}_\theta = \{L\}_\theta$$

en que θ controla la estabilidad de la solución y en donde:

$$\theta = \frac{t - t_{n-1}}{\Delta t} \quad \text{con } \Delta t = t_n - t_{n-1}$$

Así las aproximaciones son:

$$\{a\}_\theta \simeq (1 - \theta)\{a\}_{n-1} + \theta\{a\}_n$$

$$\{L\}_\theta \simeq (1 - \theta)\{L\}_{n-1} + \theta\{L\}_n$$

Derivando obtenemos:

$$\left\{ \frac{\partial a}{\partial t} \right\}_\theta \simeq -\frac{\partial \theta}{\partial t} \{a\}_{n-1} + \frac{\partial \theta}{\partial t} \{a\}_n = \frac{\{a\}_n - \{a\}_{n-1}}{\Delta t}$$

luego,

$$G \frac{\{a\}_n - \{a\}_{n-1}}{\Delta t} + A((1-\theta)\{a\}_{n-1} + \theta\{a\}_n) = (1-\theta)\{L\}_{n-1} + \theta\{L\}_n$$

así,

$$\left(\frac{1}{\Delta t} [G] + \theta [A] \right) \{a\}_n = (1-\theta)\{[L]\}_{n-1} + \theta\{[L]\}_n + \left(\frac{1}{\Delta t} [G] + (1-\theta)[A] \right) \{a\}_{n-1}$$

es la ecuación que resolveremos numéricamente. En este caso se trata de resolver un sistema de la forma:

$$[\mathcal{A}] \{a\}_n = \{f\}$$

En donde:

$$[\mathcal{A}] = \frac{1}{\Delta t} [G] + \theta [A]$$

y

$$\{f\} = (1-\theta)\{[L]\}_{n-1} + \theta\{[L]\}_n + \left(\frac{1}{\Delta t} [G] - (1-\theta)[A] \right) \{a\}_{n-1}$$

Obtenido el resultado, se procederá a evaluar los indicadores del error para cada elemento, que serán usados para determinar cual o cuales de ellos deberán ser incluidos en el refinamiento de la malla, para una nueva aplicación del método.

4.4. Integración de elementos triangulares curvos.

Los bordes curvos se pueden definir por medio de dos mecanismos, circunferencias y splines cúbicas. Las circunferencias, se definen como:

$$(x, y) \in s_{curvo} \text{ si, } (x - a)^2 + (y - b)^2 = r^2$$

en que (a, b) corresponde al centro de la circunferencia y r corresponde al radio de la misma.

Las splines se definen por la mejor spline que representa el borde curvo que se desea representar, para ello los bordes se representan por polinomios de tercer grado en x e y , en que:

$$(x, y) \in s_{curvo} \text{ si, } \left| \begin{array}{l} a_x t^3 + b_x t^2 + c_x t + d_x = 0 \\ a_y t^3 + b_y t^2 + c_y t + d_y = 0 \end{array} \right|, t \in [0, 1]$$

en que los extremos del arco corresponden a los valores extremos de t .

Capítulo 5

Estimadores del Error

5.1. Descripción

Para revisar la calidad de la solución y poder efectuar un adecuado control adaptivo de la malla a usar, se necesita descubrir cuales de los elementos de la malla requieren ser modificado, esto es ser refinados o desrefinados. Para identificar los elementos, definiré un indicador del error cometido, que permita discriminar la continuidad del procedimiento de refinamiento o desrefinamiento aplicado.

El indicador del error en los elementos de la malla se construye en base a estimadores del error local cometido en la solución. Estos estimadores a posteriori del error, permiten hallar una solución óptima de acuerdo al criterio usado para definir el estimador. Para ello se modifica (refina/desrefina), la malla usada para resolver el problema planteado.

Este trabajo utiliza estimadores locales del error cometido, los que se miden sobre cada uno de los elementos de la malla y no sobre el dominio en su conjunto. De esta forma se realizan los refinamientos o desrefinamientos en las zonas en que dicho error estimado sea mayor, para así mejorar la solución obtenida

A continuación se presentaran el indicador y los estimadores usados.

5.2. Indicador del error

5.2.1. Definición

Sea $I(t)$ el indicador del error local cometido en el elemento, tal que:

$$I : \{t \in T\} \rightarrow [0, 1]$$

si Ω es el dominio del problema, ε el estimador del error local cometido, T la triangulación definida sobre el dominio y t un elemento de la triangulación. Defino:

$$\varepsilon_m = \max_{t \in T} \varepsilon(t)$$

entonces el Indicador del error sobre los elementos se define como:

$$I(t) = \left\| \frac{\varepsilon(t)}{\varepsilon_m} \right\|, \quad \forall t \in T$$

5.2.2. Aplicación

El indicador del error local cometido se usará para guiar el proceso de modificación de la malla. Para ello se definen valores límite que indicarán que durante el proceso de refinamiento si

$$I(t) > \text{limiteSup} \quad \Rightarrow \quad t \text{ es refinado}$$

y para los desrefinamientos:

$$I(t) < \text{limiteInf} \quad \Rightarrow \quad t \text{ es desrefinado}$$

5.3. Estimadores del Error

Los estimadores del error local cometido son funciones tales que:

$$\varepsilon(t) : \{t \in T\} \longrightarrow R^+$$

En este trabajo se compararon tres estimadores del error diferentes, tanto en su formulación, como en su espíritu o justificación, y se presentan a continuación.

5.3.1. Estimador basado en la solución del problema.

Este estimador se basa en la solución del problema.

Sea u_r la solución al problema, u_e la solución estimada, definimos al estimador como:

$$\varepsilon(t) = \sum_{v_i \in t} \frac{\|u_r(v_i) - u_e(v_i)\|}{\|u_r(v_i)\|}$$

5.3.2. Estimador basado en la solución obtenida

Este estimador se basa en la solución obtenida. Para ello se estima el error cometido de la forma siguiente :

$$\epsilon_t = \sqrt{A_t [\alpha \sum_{i=1}^3 J_s^2 + \beta L_t]}$$

en donde A_t es el área del triángulo t , J_s es una función relacionada con el salto de la derivada normal de la solución en curso de elementos finitos a lo largo del lado s , α y β son constantes y:

$$L_t = \int_t (\beta U - f)^2$$

5.3.3. Estimador basado en la suavidad de la solución obtenida

Este estimador se basa en la suavidad de la solución encontrada, busca disminuir la diferencia en la pendiente entre los triángulos con lados comunes. Este estimador se define en la forma siguiente:

Sea $t \in T$ un elemento de la triangulación definida sobre el dominio Ω .

Sean a, b, c los vértices del elemento.

Sean $T_a(t) = \{t_a \in T \mid t_a \cap t = \overline{xy}, x, y \in \{a, b, c\} \quad x \neq y\}$ los elementos que son contiguos al elemento t y que compartan un lado con este.

Entonces definiremos el estimador como:

$$\varepsilon(t) = \sum_{t_a \in T_a(t)} pendiente(t, t_a)$$

en que:

$$pendiente(t, v) = \begin{cases} 0 & \text{si } t \cap v \neq \emptyset \\ 2 & \text{si } \cos(v, t) < \cos(\text{Angmax}) \end{cases}, \quad t \in T, v \in T_a(t)$$

Capítulo 6

Problemas de prueba

Para ilustrar el uso y evaluar el comportamiento del sistema, se presentan cinco problemas de prueba con solución conocida. Se utilizó para estimar el error una medida de la suavidad de la solución obtenida entre elementos vecinos, y así poder construir la solución numérica adaptativa. La solución final de cada problema mostrado, se ajusta a la solución real de los problemas planteados.

6.1. Problema 1

6.1.1. Descripción

En el primer ejemplo se trata de resolver el problema :

$$\begin{aligned} -\Delta u &= 18\pi^2 \sin(3\pi x) \sin(3\pi y), & \text{en } \Omega &= [0, 1] \times [0, 1], \\ u &= 0 & \text{en } \Omega \end{aligned}$$

cuya solución exacta es la función:

$$u = \sin(3\pi x) \sin(3\pi y).$$

6.1.2. Resultados

Las figuras 6.1.1 a 6.1.4, muestran la secuencia de mallas del refinamiento 4, 12 y 23, y gráficas de la solución hasta la de última malla, en base a un estimador basado en la suavidad de la solución. La tabla 6.1, muestra información estadística sobre la ejecución de la aplicación durante la resolución de este problema.

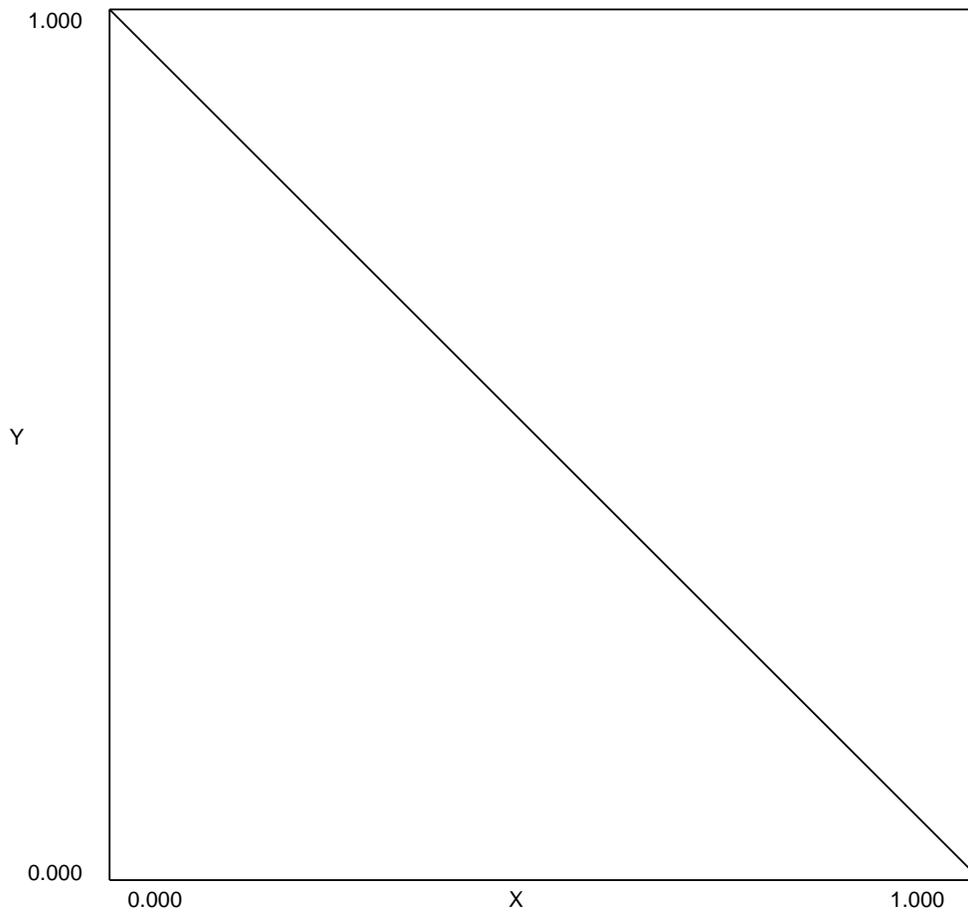


Figura 6.1.1: Malla inicial, problema 1

Iteración	Cantidad de nodos	Errores			Valor máximo	Duración real[seg]
		Máximo	Medio	Cuadrático medio		
4	21	0.465927	0.088748	0.203347	0.707107	1.326260
5	41	0.199285	0.038885	0.072926	0.853553	1.185568
6	56	0.183081	0.025302	0.061483	0.653281	1.020600
7	116	0.191647	0.030523	0.057093	0.961940	0.612462
8	211	0.090063	0.015184	0.028647	0.980785	0.411733
9	378	0.068391	0.008637	0.017500	0.990393	0.264394
10	667	0.035864	0.005945	0.011155	0.995185	0.674895
11	1133	0.021737	0.003780	0.007334	0.997592	2.616514
12	1785	0.015628	0.002364	0.005083	0.998795	3.022343
13	2605	0.013014	0.001625	0.003965	0.999398	5.031190
14	3288	0.011971	0.000780	0.002697	0.999699	8.706475
15	4042	0.011820	0.000605	0.002398	0.999849	13.559902
16	4640	0.011252	0.000334	0.001763	0.999925	15.316691

Cuadro 6.1: Estadísticas ejecución problema 1

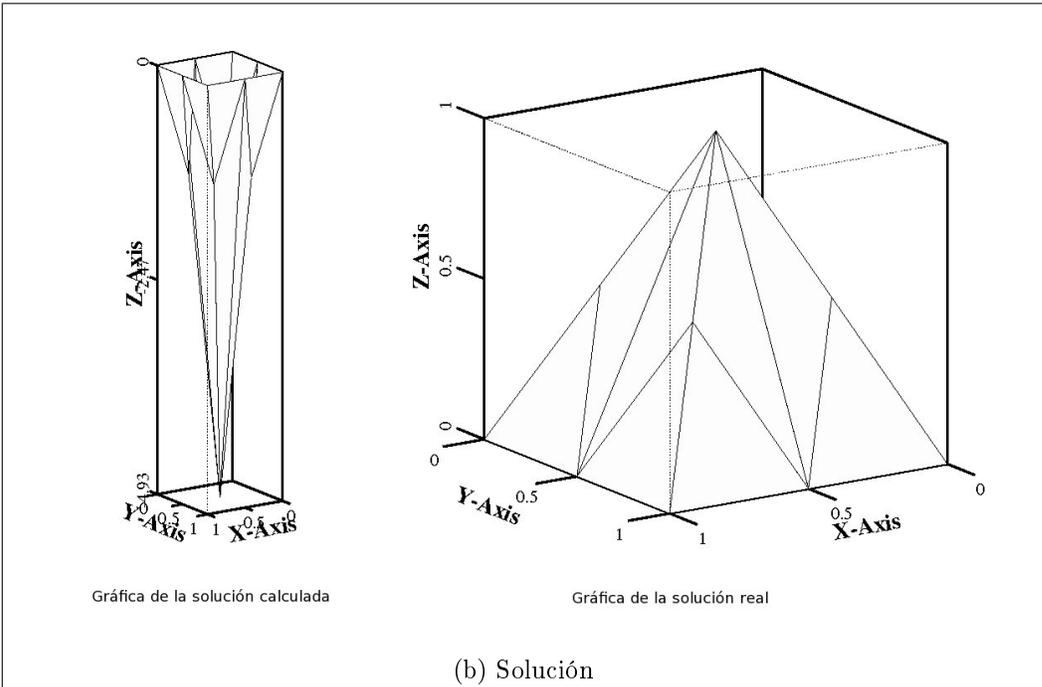
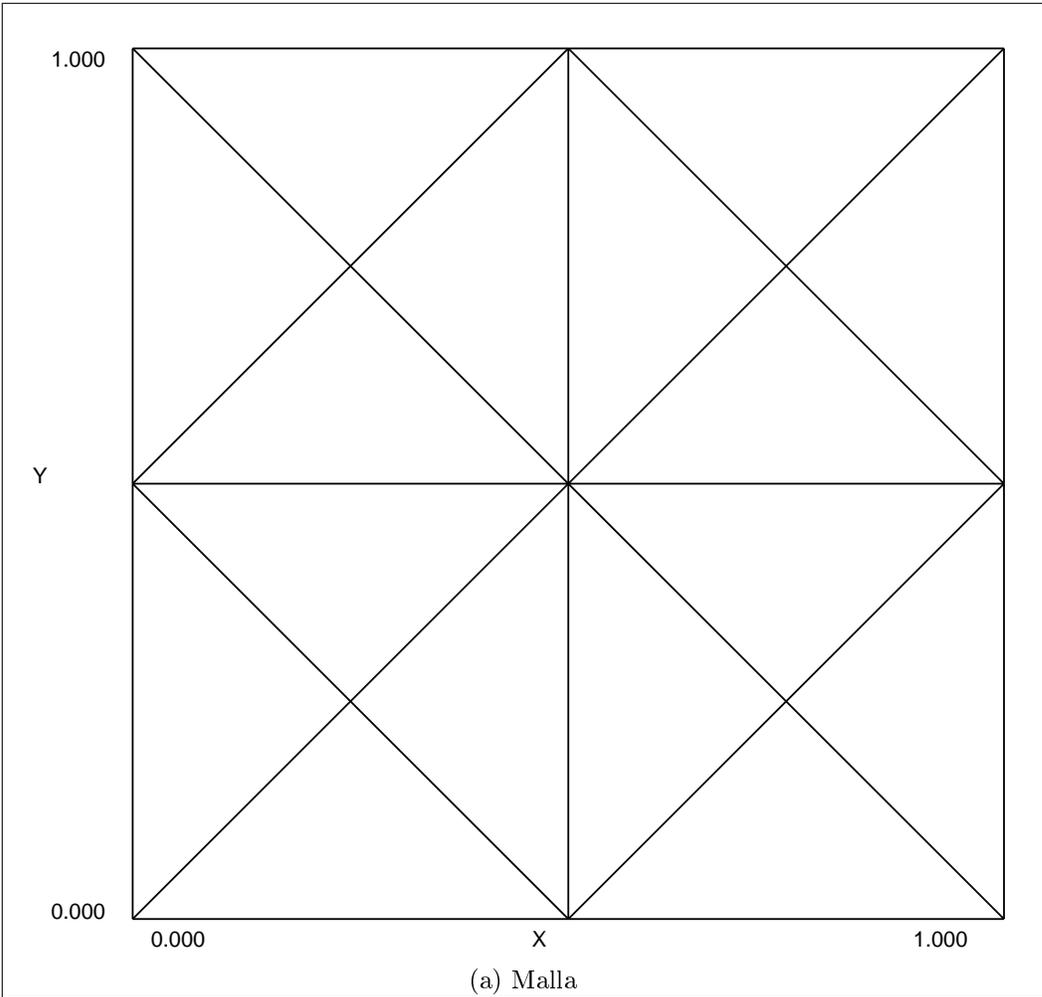


Figura 6.1.2: Iteración 4, problema 1

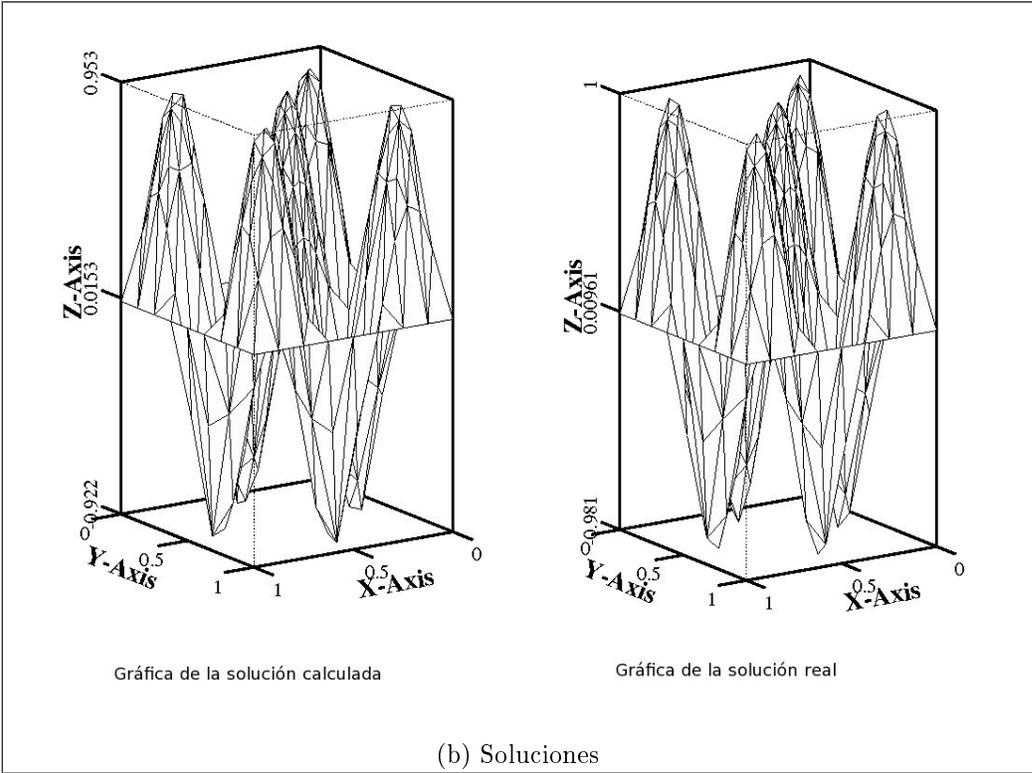
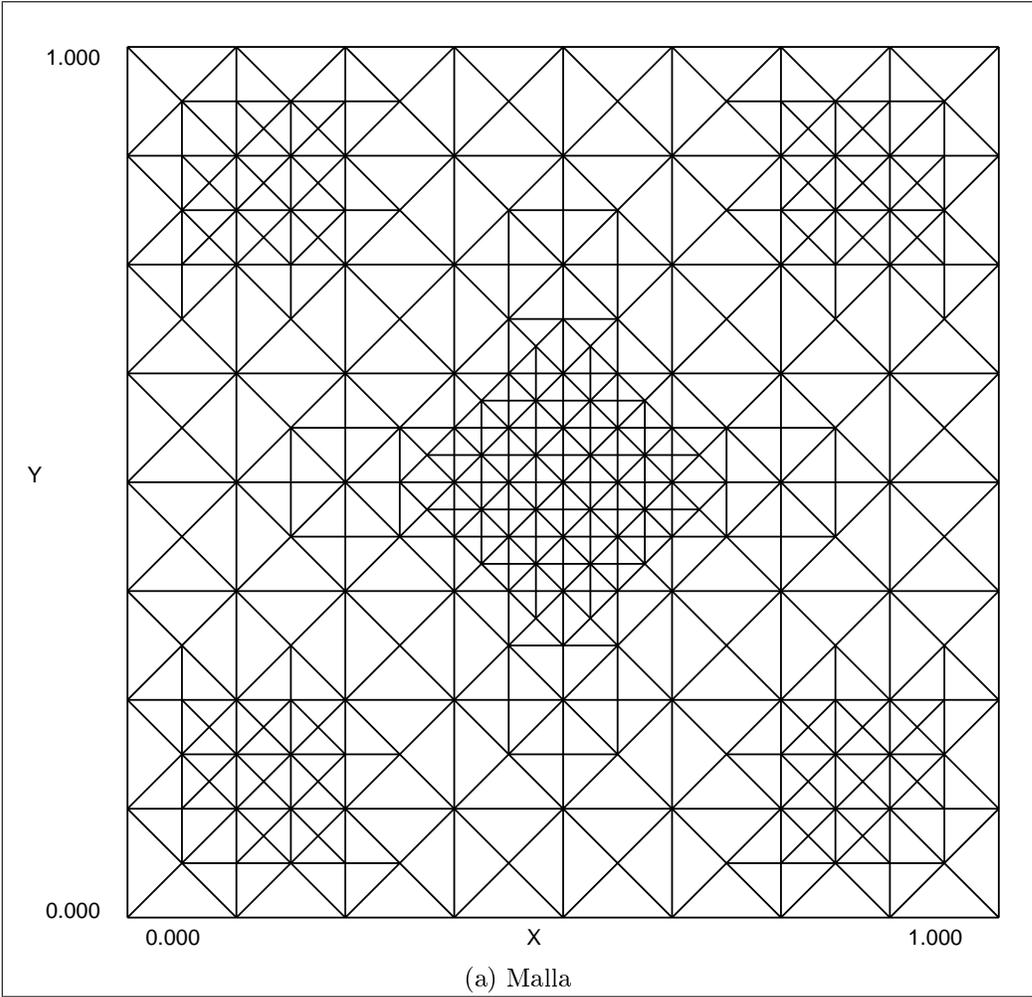


Figura 6.1.3: Iteración 12, problema 1

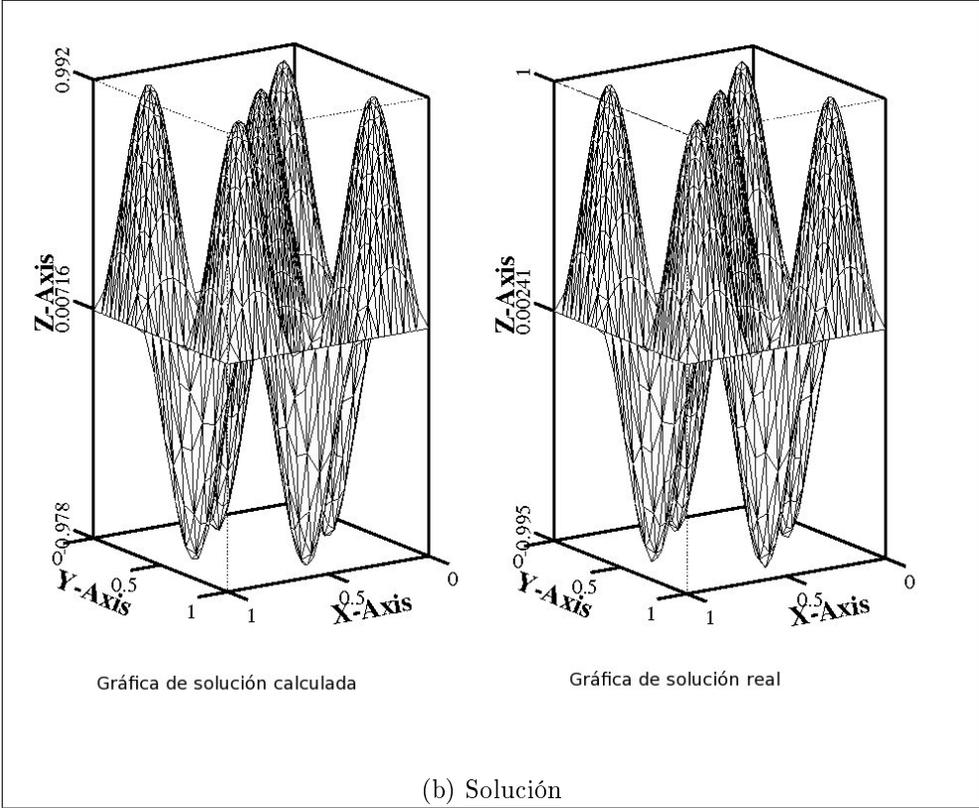
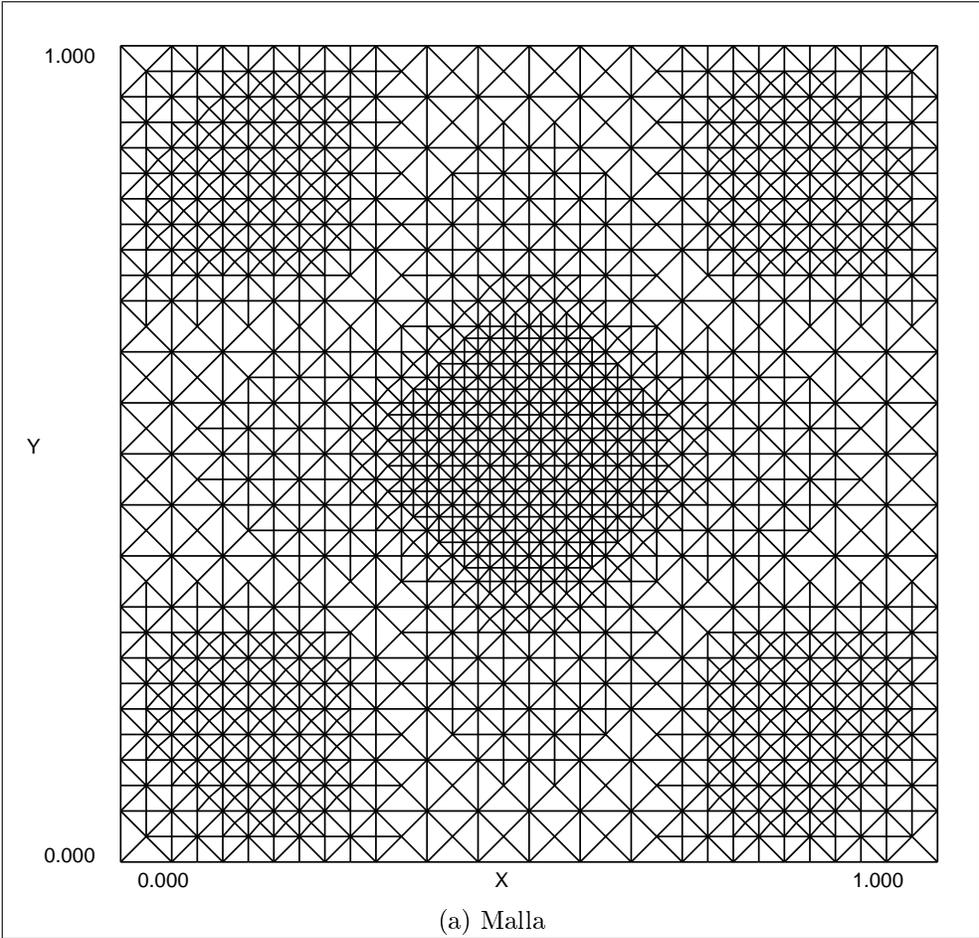


Figura 6.1.4: Iteración 23, problema 1

6.2. Problema 2

6.2.1. Descripción

En el segundo ejemplo se trata de resolver el problema :

$$\begin{aligned} -\Delta u &= f, & \text{en } \Omega &= [0, 1] \times [0, 1], \\ u &= 0 & \text{en } \partial\Omega & \end{aligned}$$

donde la solución exacta es la función:

$$u = x(x - 1)y(y - 1)e^{-100((x-0,5)^2+(y-0,117)^2)}$$

La Figura muestra la secuencia de mallas del refinamiento, y gráficas de la solución encontrada en las mallas 5 y 12, con vistas desde dos perspectivas diferentes para apreciar mejor la forma de la solución.

6.2.2. Resultados

Las figuras 6.2.1 a 6.2.3, muestran la secuencia de mallas del refinamiento, y gráficas de la solución encontrada para las mallas, en base a un estimador basado en el error real. La tabla 6.2, muestra información estadística sobre la ejecución de la aplicación durante la resolución de este problema.

Iteración	Cantidad de nodos	Errores			Valor máximo	Duración real[seg]
		Máximo	Medio	Cuadrático medio		
4	21	0.000749	0.000107	0.000268	0.007993	0.044970
5	31	0.008699	0.001458	0.003236	0.005339	1.460423
6	50	0.005200	0.000680	0.001527	0.027169	0.831821
7	94	0.003366	0.000153	0.000548	0.015432	0.349835
8	163	0.002610	0.000073	0.000305	0.023169	1.863678
9	291	0.002530	0.000096	0.000285	0.025524	1.395070
10	546	0.001213	0.000027	0.000112	0.028253	0.961774
11	921	0.000202	0.000007	0.000017	0.003628	1.231631
12	1693	0.000179	0.000009	0.000018	0.003642	3.344112

Cuadro 6.2: Estadísticas ejecución problema 2

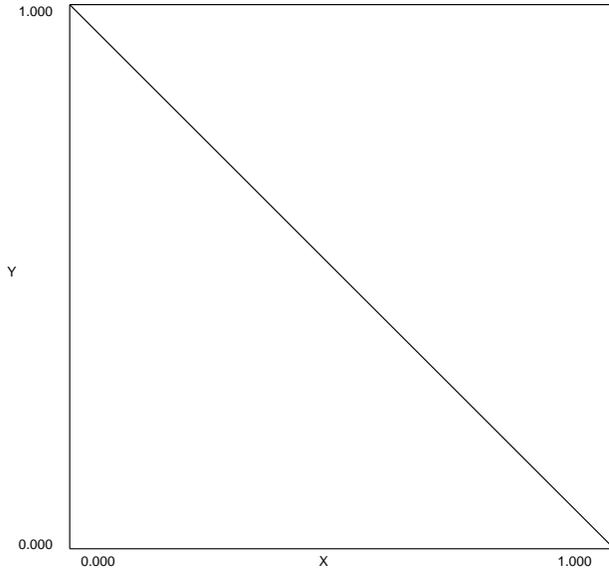


Figura 6.2.1: Malla inicial, problema 2

6.3. Problema 3

6.3.1. Descripción

En el tercer ejemplo se trata de resolver el problema :

$$\begin{aligned} -\Delta u &= f, & \text{en } \Omega &= \{(x, y)/0,5 \leq x^2 + y^2 \leq 1\}, \\ u &= 0 & \text{en } \partial\Omega_e &= \{(x, y)/x^2 + y^2 = 1\}, \\ u &= \cos\left(\frac{\pi}{8}\right) & \text{en } \partial\Omega_i &= \{(x, y)/x^2 + y^2 = 0,25\}, \end{aligned}$$

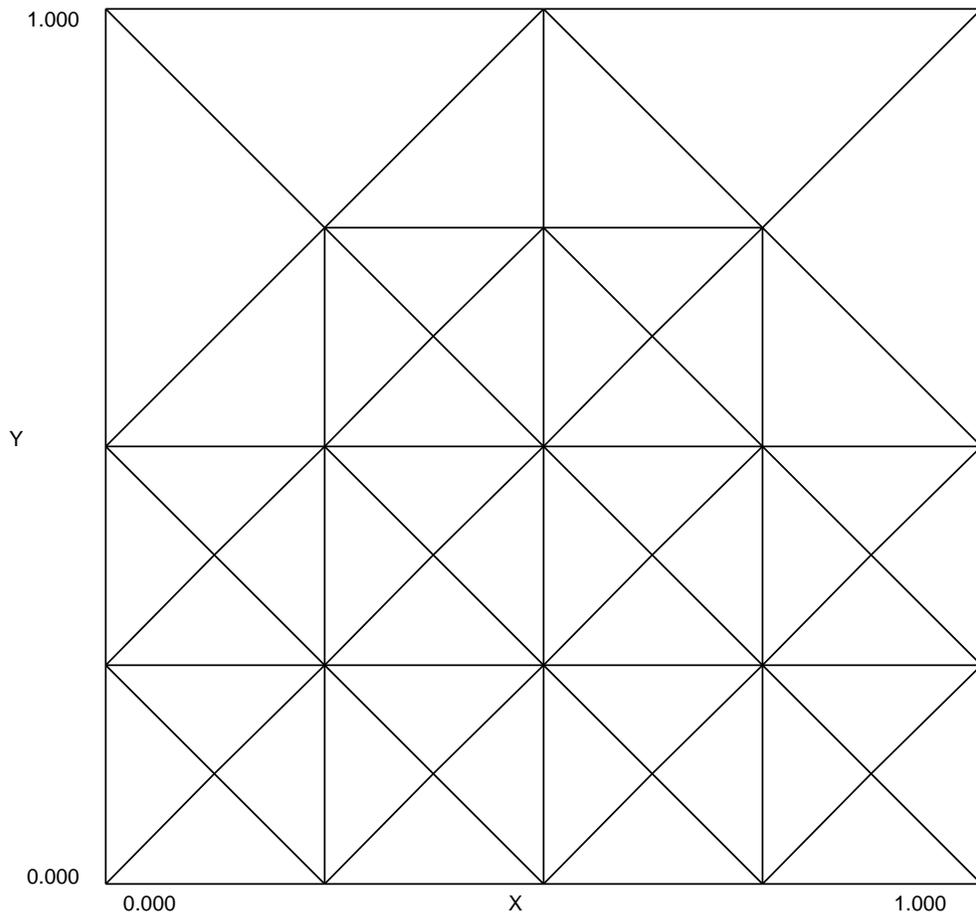
donde la solución exacta es la función:

$$u = \cos\left(\frac{\pi(x^2+y^2)}{2}\right)$$

La Figura muestra la secuencia de mallas del refinamiento, y gráficas de la solución encontrada en las mallas 2 y 9, con vistas desde dos perspectivas diferentes para apreciar mejor la forma de la solución.

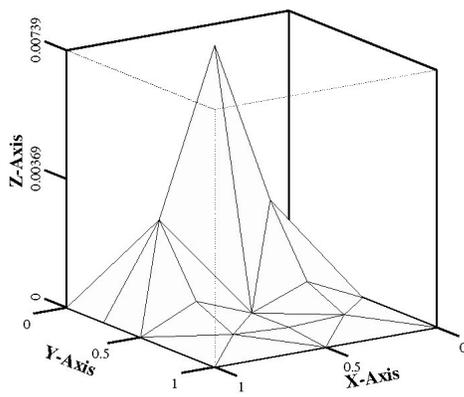
6.3.2. Resultados

Las figuras 6.2.1 a 6.2.3, muestran las mallas inicial, 2 y 9 del refinamiento, y gráficas de la solución encontrada para las mallas, en base a un estimador basado en el error estimado. La tabla 6.3, muestra información estadística sobre la ejecución de la aplicación durante la resolución de este problema.

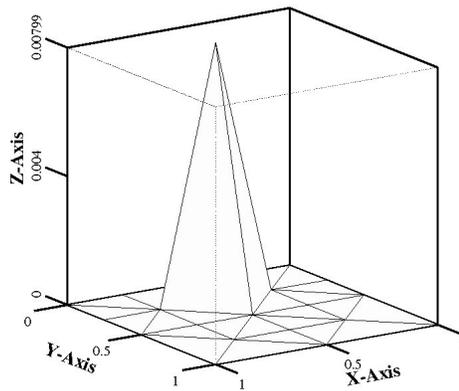


(a) Malla

Solucion calculada (iteracion 5)

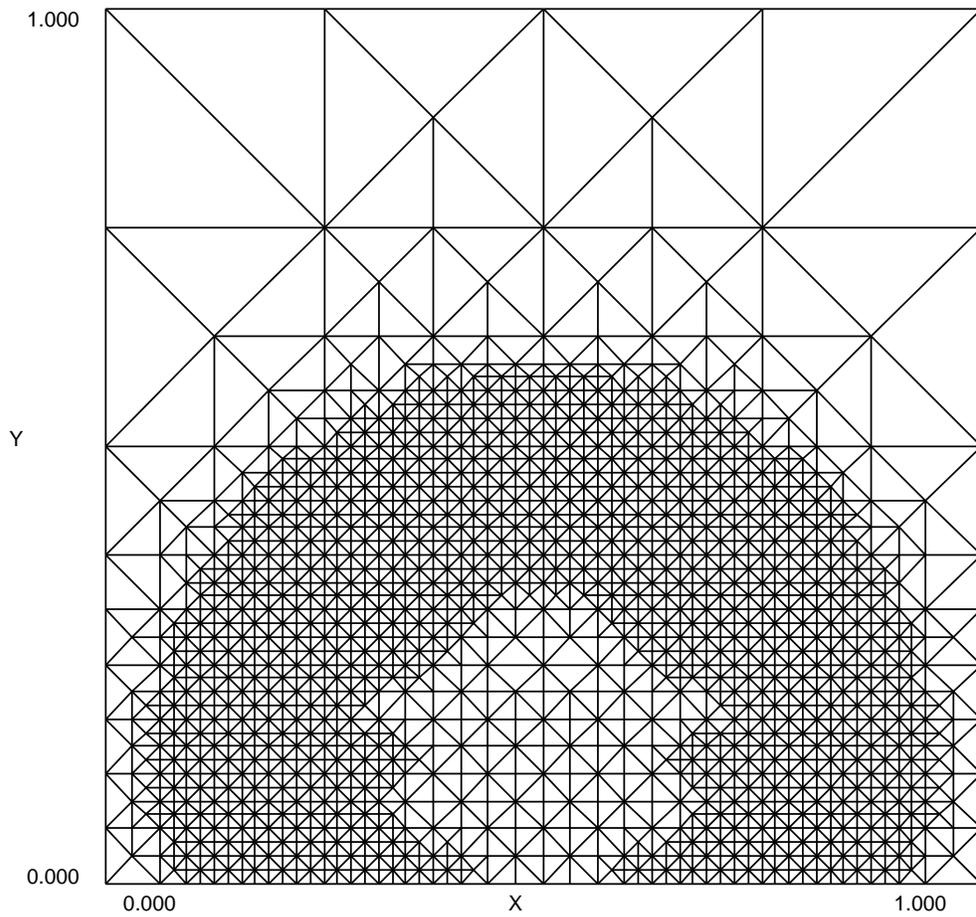


Solucion Real (iteracion 5)

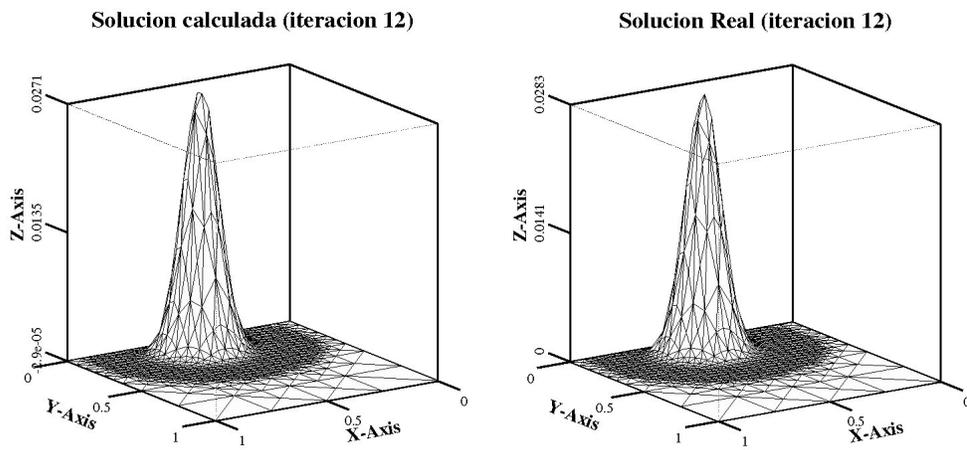


(b) Solución

Figura 6.2.2: Iteración 5, problema 2



(a) Malla



(b) Solución

Figura 6.2.3: Iteración 12, problema 2

Iteración	Cantidad de nodos	Errores			Valor máximo	Duración real [seg]
		Máximo	Medio	Cuadrático medio		
3	160	0.123725	0.038974	0.062351	0.923880	0.897689
4	316	0.133322	0.039409	0.064218	0.901320	1.042831
5	5,473	0.126765	0.023785	0.047932	0.923880	2.087441
6	6,540	0.126138	0.007097	0.027095	0.915940	1.181438
7	7,544	0.026406	0.000096	0.001588	0.923880	1.159591
8	8,547	0.030284	0.000093	0.001437	0.918207	0.485014
9	9,549	0.013936	0.000051	0.000837	0.915516	0.230722

Cuadro 6.3: Estadísticas ejecución problema 3

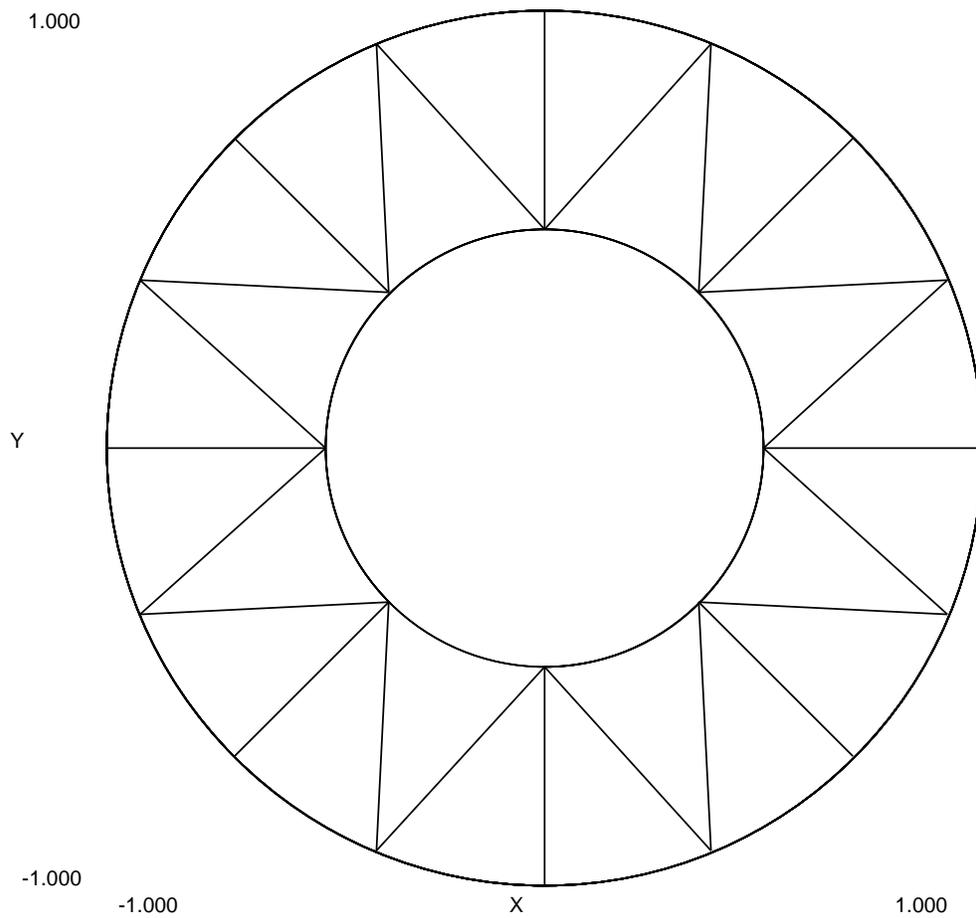
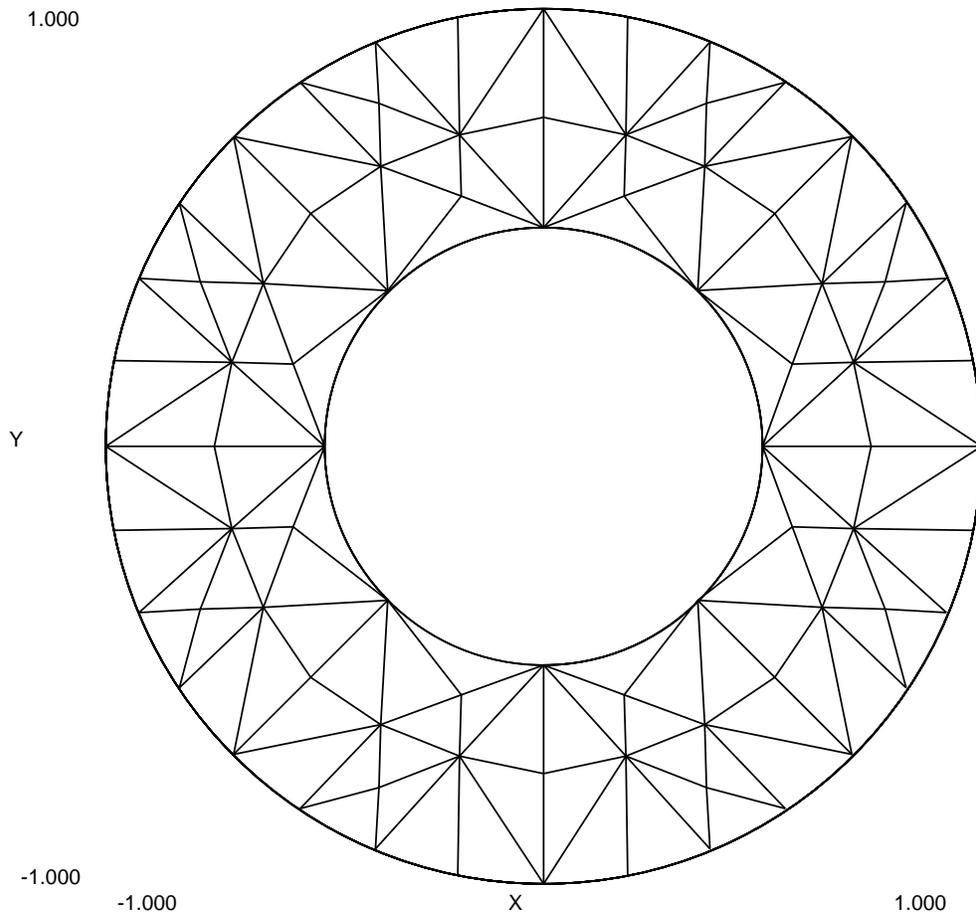
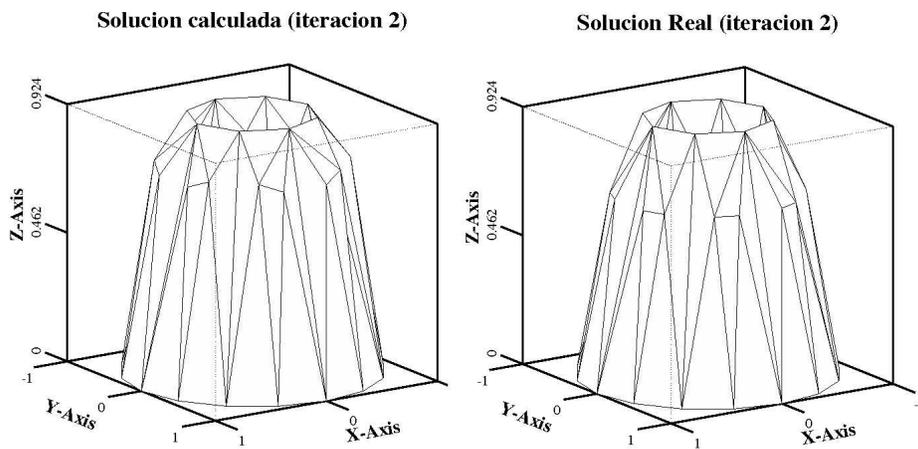


Figura 6.3.1: Malla inicial, problema 3

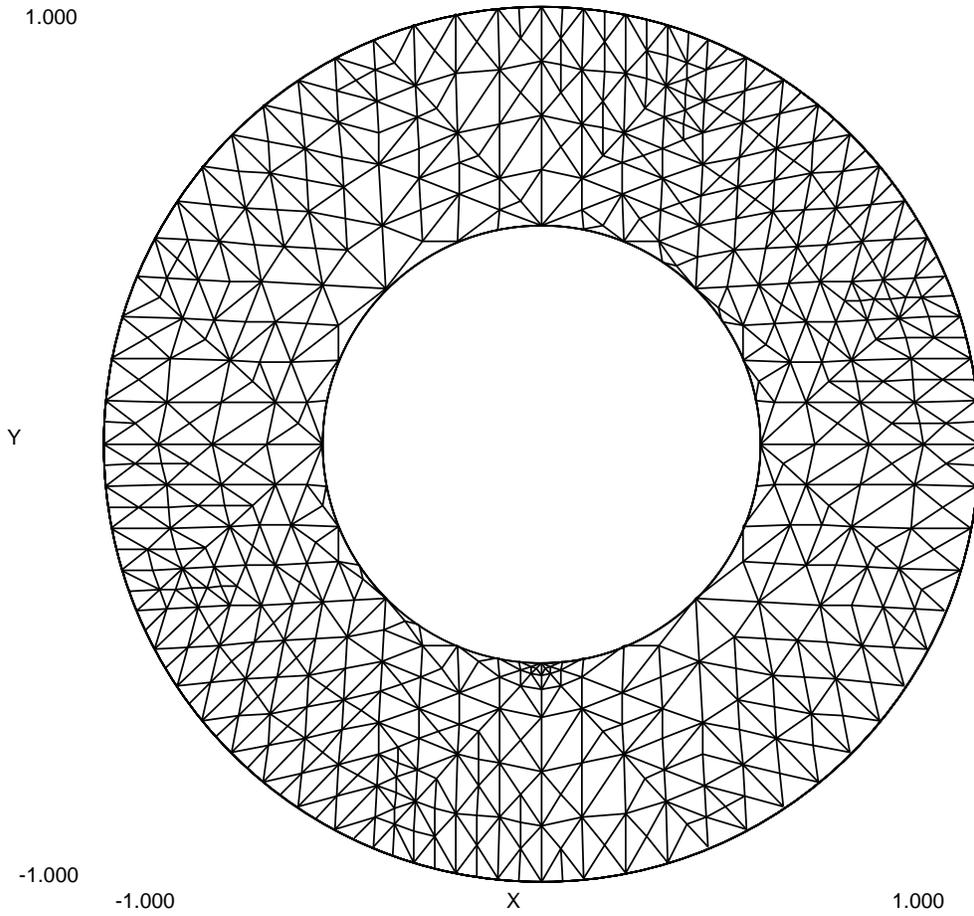


(a) Malla

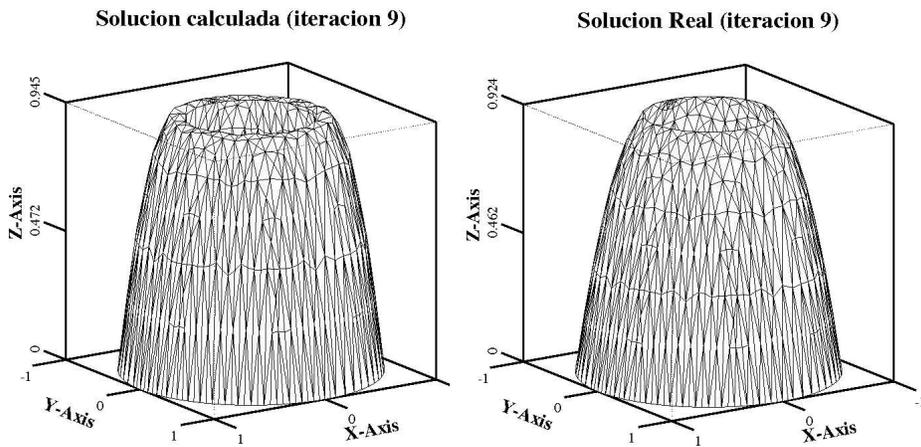


(b) Solución

Figura 6.3.2: Iteración 2, problema 3



(a) Malla inicial



(b) Solución

Figura 6.3.3: Iteración 9, problema 3

6.4. Problema 4

6.4.1. Descripción

En el cuarto ejemplo se trata de resolver el problema :

$$\begin{aligned} \Delta u &= 0, & \text{en } D \\ u(x, y) &= 4 & \text{en } L6 \cup L7 \\ \frac{\partial u}{\partial n}(x, y) &= x & \text{en } L2 \cup L4 \\ \frac{\partial u}{\partial n}(x, y) &= y & \text{en } L5 \\ \frac{\partial u}{\partial n}(x, y) &= \frac{x+y}{\sqrt{2}}y & \text{en } L1 \cup L3 \end{aligned}$$

donde la solución exacta es la función:

$$u(x, y) = (x * y) + 4$$

La figura 6.4.1 muestra el dominio D y los bordes L1 a L7 en que se aplican las condiciones de borde.

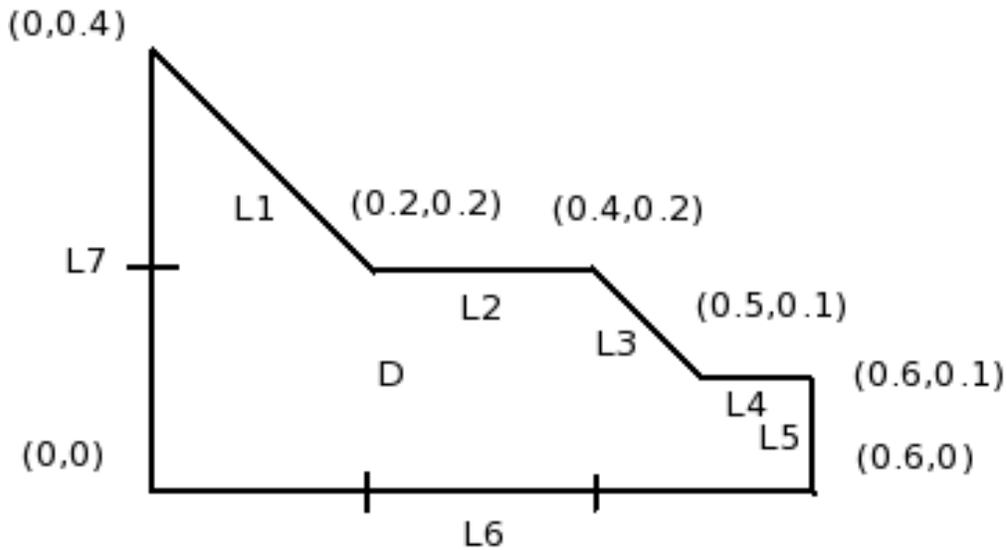
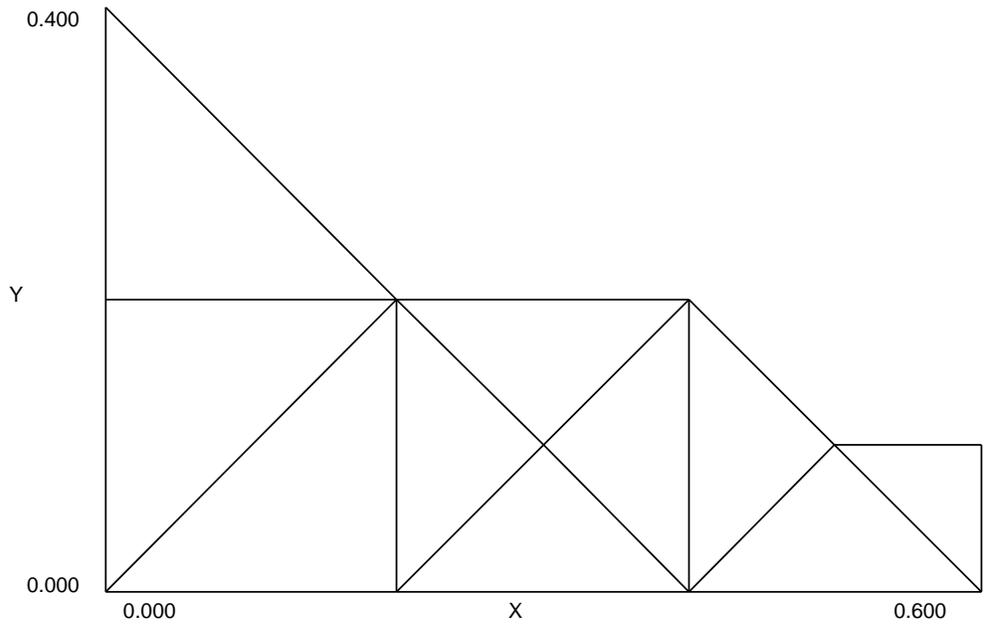
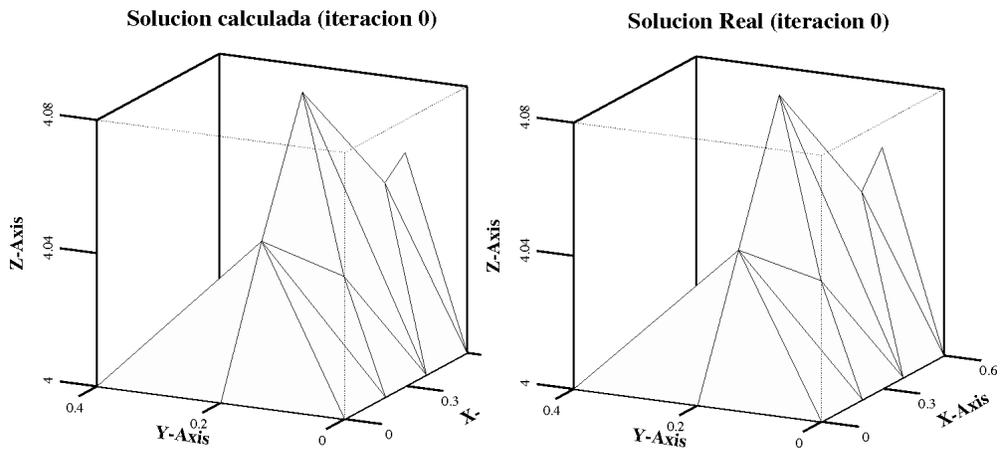


Figura 6.4.1: Dominio D

La figura 6.4.2 muestra la malla inicial y la solución del problema.



(a) Malla inicial



(b) Solución

Figura 6.4.2: Problema 4

Capítulo 7

Aplicación

7.1. Introducción

La aplicación desarrollada opera sobre equipos Linux, utilizando la interfaz gráfica XWindows. Como fue desarrollada sobre GLADE, debiera ser sencillo portarla a otras plataformas.

La aplicación permite crear, consultar y/o modificar los problemas y sus mallas asociadas, controlar el refinamiento, desrefinamiento de las mallas y la aplicación del método de elementos finitos sobre un problema.

La interacción del usuario con la aplicación se realiza principalmente a través de:

- Ventana principal o de control
- Ventana de consulta/creación/modificación del problema
- Ventana de consulta/creación/modificación de la malla
- Ventana de presentación de soluciones (3DPlot)
- Otras ventanas

Cada ventana tiene asociadas menús de control y botones para la interacción del usuario.

7.2. Ventana principal

La ventana principal presenta un menú superior que está separado de acuerdo a las funciones principales de la aplicación. Estos son:

1. Problema
2. Malla
3. Refina
4. Desrefina
5. M.E.F.
6. Spline

7. Ayuda

Además presenta un panel para el despliegue del dominio y su malla asociada, y un segundo panel destinado a presentar información de la aplicación.

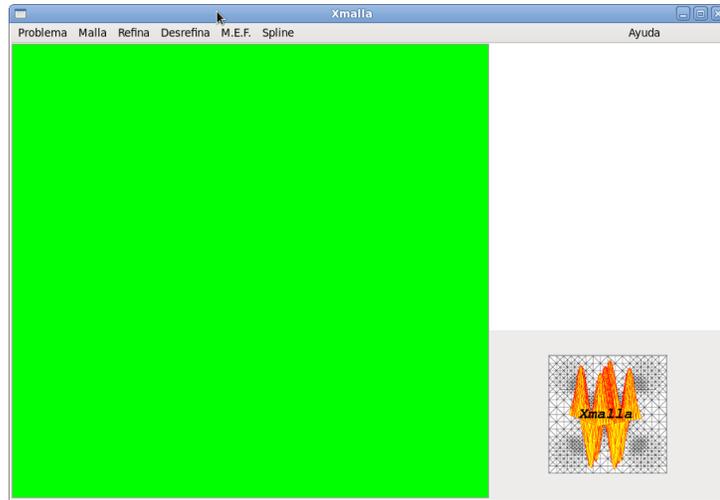


Figura 7.2.1: Ventana principal aplicación gxmalla

El ítem problema dispone de las siguientes opciones:

1. Nuevo
2. Abrir
3. Guardar
4. Guardar como
5. Salir

7.3. Ventana de consulta/creación/modificación del problema

La ventana del problema permite consultar, crear o modificar la especificación de un problema. Allí se aprecian los principales parámetros de control del problema, como la descripción de la ecuación diferencial como también de las regiones de refinamiento y parámetros que definirán como operará la aplicación al resolver el problema.

7.4. Ventana de consulta/creación/modificación de la malla

La ventana de la malla permite consultar, crear o modificar la especificación de una triangulación sobre el dominio especificado. En la práctica, el dominio está representado por la malla de triángulos. A través de esta interfaz, se puede acceder a la información de:

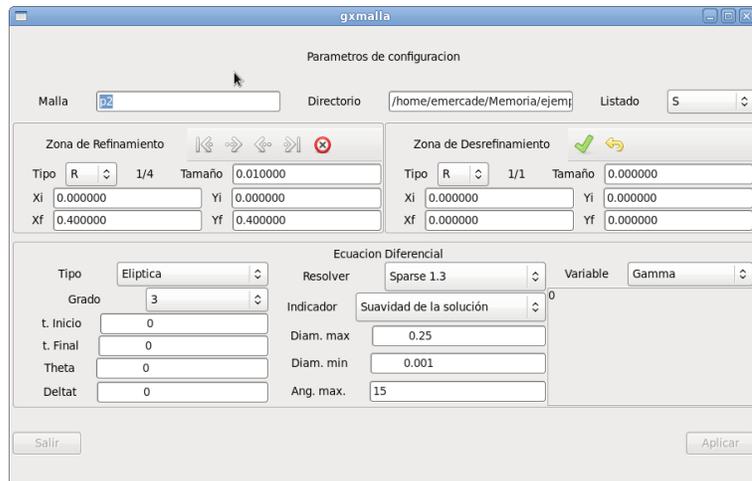


Figura 7.3.1: Ventana de visualización de un problema

1. los nodos,
2. las triangulaciones
3. los bordes curvos (splines)
4. las condiciones de borde
5. las coordenadas del dominio

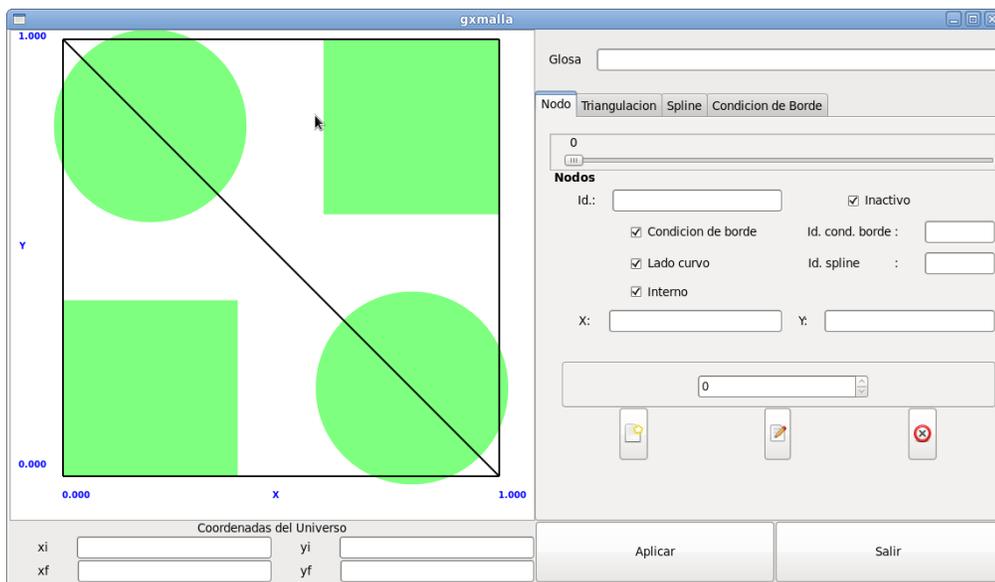


Figura 7.4.1: Ventana de visualización de una malla

7.5. Ventana de presentación de soluciones (3DPlot)

Esta ventana permite observar la solución en 3 dimensiones, permitiendo al usuario modificar la vista al rotar en cualquiera de los ejes principales la figura que representa la solución.

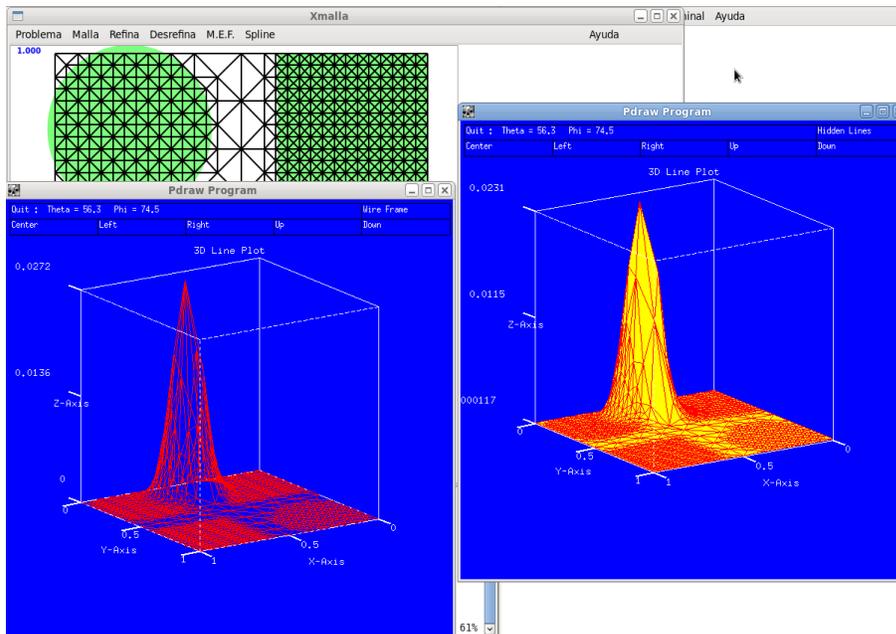


Figura 7.5.1: Presentación de la solución

7.6. Otras ventanas

El sistema dispone de otras ventanas de interacción con el usuario como son las que permiten escoger los archivos de problema, la ventana “A cerca de” o las ventanas de ayuda.

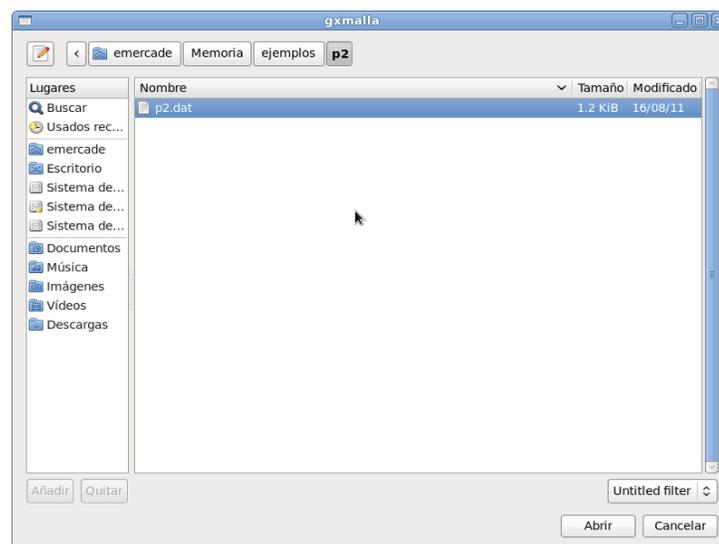


Figura 7.6.1: Abrir archivo



Figura 7.6.2: Acerca de

7.7. Ejemplo de definición de un problema

A modo de ejemplo se mostrará la secuencia de acciones para definir el problema siguiente:

$$\begin{array}{ll}
 -\Delta u = 0 & \text{en } \Omega = [0, 1] \times [0, 1], \\
 u = 4 & \text{en } \partial\Omega, \text{ si } x = 0 \\
 u = 4 & \text{en } \partial\Omega, \text{ si } y = 0 \\
 \frac{\partial u}{\partial n}(x, y) = x & \text{en } \partial\Omega, \text{ si } y = 1 \\
 \frac{\partial u}{\partial n}(x, y) = y & \text{en } \partial\Omega, \text{ si } x = 1 \\
 u = 5 & \text{en } \partial\Omega, \text{ si } y = 1, x = 1
 \end{array}$$

cuya solución exacta es la función:

$$u = (x * y) + 4$$

Estas acciones son:

1. Crear la malla que define el problema, para ello se debe proceder a:
 - a) Crear los nodos de la malla (solo coordenadas)
 - b) Crear la triangulación
 - c) Crea las moléculas
 - d) Definir los vecinos de los nodos para cada molécula
 - e) Crear los lados curvos si son requeridos
 - f) Crear las condiciones de borde
 - g) Asignar los lados curvos y las condiciones de borde a los nodos
2. Definir el problema asociado

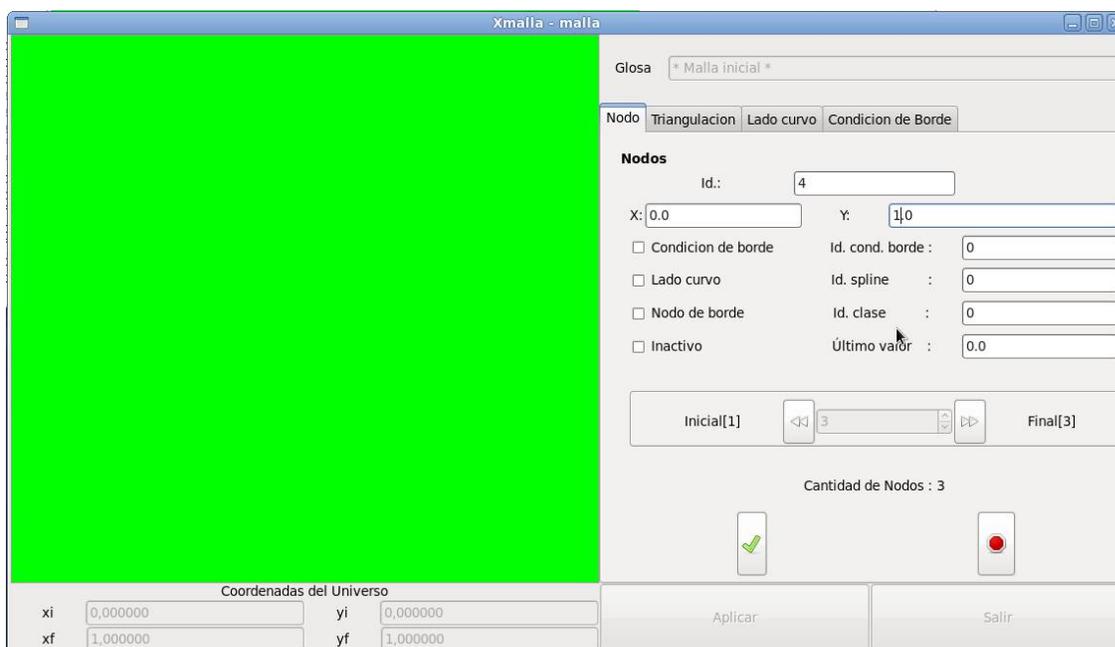


Figura 7.7.1: Creación de nodos de la malla inicial

7.7.1. Creación de la malla inicial

Ejecutar el programa y en el menú “Malla” ejecutar la opción “Nuevo”. En la pestaña “Nodo”, agregar los nodos, como se aprecia en la figura

En la pestaña “Triangulacion”, proceder a crear la triangulación, indicando como primer nodo el 1. Luego proceder a crear cada una de las moléculas asociadas a los nodos, asignando a cada una de ella los vecinos a ese nodo. Las moléculas se definen siguiendo el orden de creación de los nodos. Los vecinos del nodo en la molécula se definen siguiente el sentido positivo desde un nodo que esté en el borde, considerando todos los nodos que compartan un lado de un triángulo con el nodo de la molécula. La figura muestra la pestaña de “Triangulación”, en donde se definen la triangulación, las moléculas y los vecinos en las moléculas.

En la pestaña “Lado curvo”, se puede proceder a crear los lados curvos.

En la pestaña “Condicion de Borde”, proceder a crear las condiciones de borde, como se aprecia en la figura. Para esto se define el tipo de la condición de borde (Esencial o Natural), luego los nodos de los extremos de la condición (usando los con identificador de menor valor como nodo inicial) y se completa la expresión aritmética que define la condición.

En la pestaña “Nodo”, se procede a asigna a cada nodo las condiciones de borde y se modifica su tipo, indicando que el nodo modificado es un nodo de borde y tiene una condición de borde, como se muestra en la figura.

Una vez completado este ingreso, se procede a aplicar los cambios realizados y a salir de la ventana de modificación de la malla. En el menú “Malla”, se debe grabar la malla en un archivo con la opción de “Grabar como”. Esto creará el archivo de extensión “.trg” que contiene la malla definida.

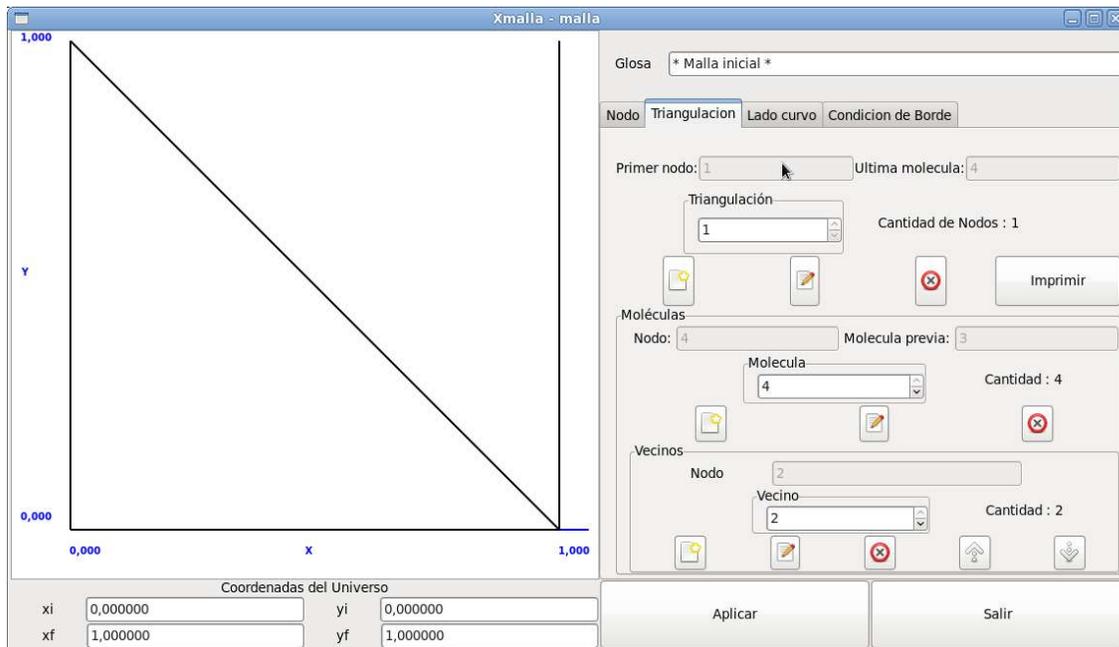


Figura 7.7.2: Creación de la triangulación, las moléculas y los vecinos

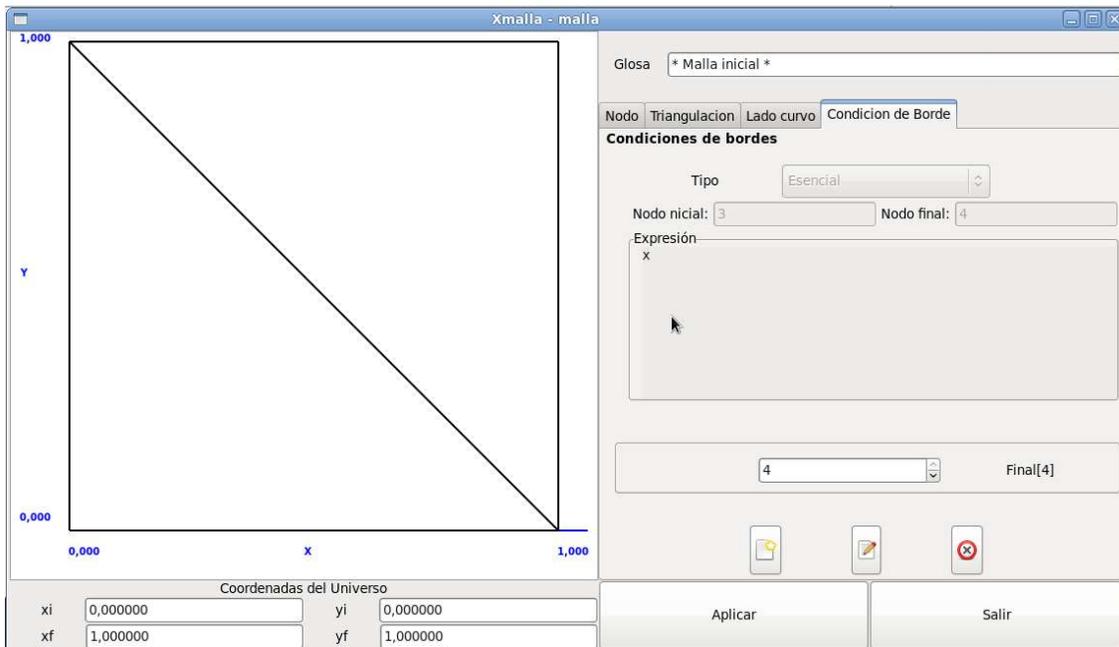


Figura 7.7.3: Creación de las condiciones de borde

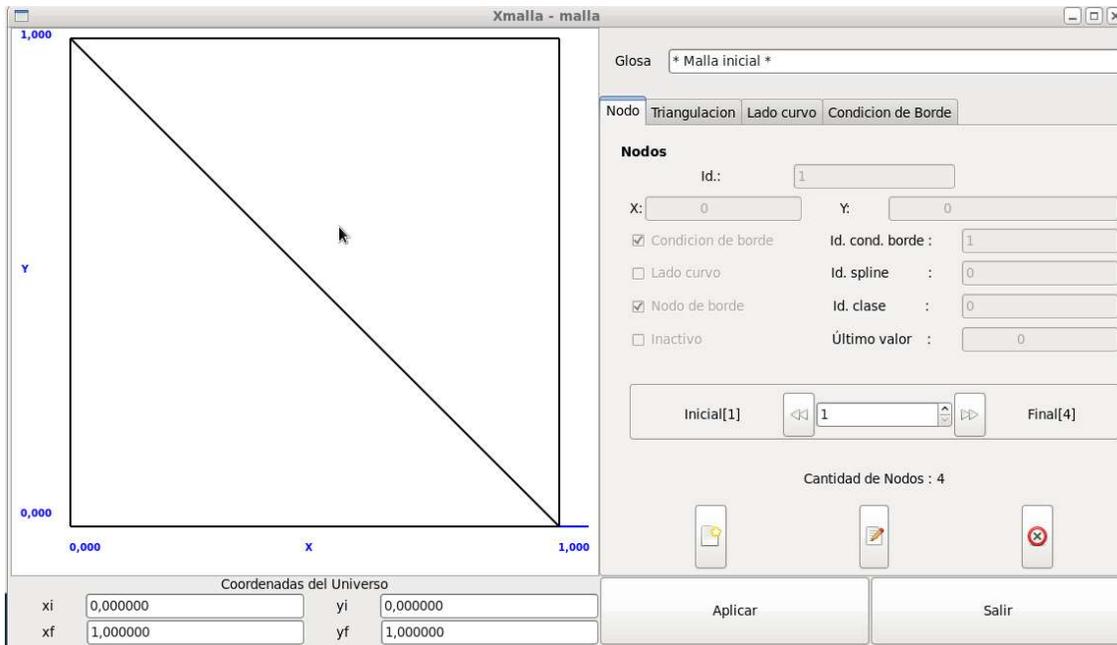


Figura 7.7.4: Modificación de los nodos de la malla

7.7.2. Definición del problema

En el menú “Problema”, se usa la opción “Nuevo”, para proceder a definir el problema. Para ello se debe indicar el nombre del archivo con la malla previamente definida. Se pueden definir áreas de refinamiento y desrefinamiento de ser necesarias y completar la definición del problema. En este caso se deben definir los diámetros máximos y mínimos de los triángulos de la malla y las variables :

- Gamma, parámetro de la ecuación diferencial que acompaña a $\partial U/\partial t$ (ver ecuación 4.3.1) en los problemas parabólicos
- Alfa X, parámetro de la ecuación diferencial que acompaña a $\partial U/\partial x$ (ver ecuaciones 4.2.1 y 4.3.1) en los problemas elípticos y parabólicos
- Alfa Y, parámetro de la ecuación diferencial que acompaña a $\partial U/\partial y$ (ver ecuaciones 4.2.1 y 4.3.1) en los problemas elípticos y parabólicos
- Beta, parámetro de la ecuación diferencial que acompaña a U (ver ecuaciones 4.2.1 y 4.3.1) en los problemas elípticos y parabólicos
- RHS, lado derecho de las ecuaciones diferenciales (ver ecuaciones 4.2.1 y 4.3.1)
- $u(x,y,0)$, solución de la ecuación en el tiempo $t = 0$
- $u(x,y,t)$, solución de la ecuación en cualquier tiempo
- $du(x,y,t)/dx$, derivada parcial de la solución con respecto a x
- $du(x,y,t)/dy$, derivada parcial de la solución con respecto a y

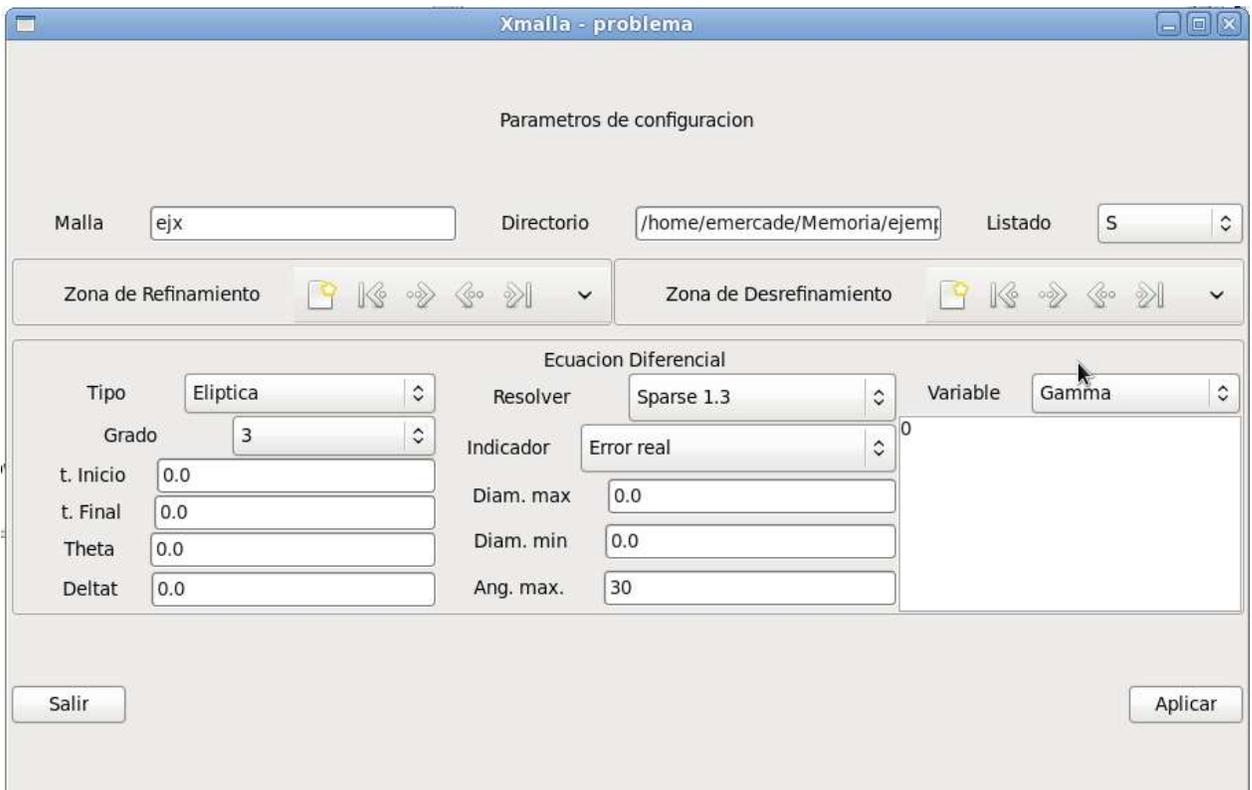


Figura 7.7.5: Definición del problema

Estos parámetros se seleccionan mediante un “combobox” que permite ir ingresando en la vista de texto su valor, como se aprecia en la figura

Entonces basta con aplicar los cambios realizados, salir de la edición y grabar en un archivo nuevo la definición realizada, para poder reutilizar posteriormente.

Capítulo 8

Conclusiones

El sistema construido, constituye un software general, flexible y fácil de usar para resolver EDP sobre regiones bidimensionales generales. Proporciona un ambiente amistoso de experimentación de métodos numéricos, que por ejemplo, permite comparar y evaluar diferentes indicadores del error y distintos criterios de adaptividad.

Los indicadores del error utilizados tuvieron buenos resultados en cuanto a la calidad de la aproximación de la solución que se buscaba. El software permite que se agreguen de una forma sencilla nuevos estimadores que permitan un mejor funcionamiento del proceso adaptivo.

La estructura de datos con que fue diseñado mantiene información suficiente para servir de base a algoritmos que usen métodos de multimallas; por lo que permitiría evaluar la utilización práctica de estos métodos.

El conjunto de problemas que permite resolver es suficientemente amplio como para que se pueda transformar en una herramienta útil para la solución de problemas reales. Además la posibilidad de definir dominios complejos e irregulares amplían su campo de acción.

Como una forma de mejorar la aplicación, se podrían:

- integrar a la misma, la generación automática de una primera malla de buena calidad adaptada a la geometría[7, 8],
- mecanismos mas amigables para ingresar la información referida tanto a la malla inicial como para la definición del problema
- mecanismo de definición de nuevos indicadores del error que no requieran reconstruir la aplicación.
- soportar elementos triangulares de segundo, tercer grado o superior.

Bibliografía

- [1] M.C.Rivara. *Algorithms for refining triangular grids suitable for adaptative & multigrid techniques*. International Journal for Numerical Methods in Engineering. Vol 20, 745-756 (1984).
- [2] M.C.Rivara. *Design & Data Structura of fully adaptative, multigrid, finite element method* ACM Trans. Math. Software. Vol. 10 242-264 (1984)
- [3] I.Spencer *Algoritmos de Refinamiento y Desrefinamiento para secuencias anidadas de triangulaciones conformes* Memoria de Ingeniero Civil Matemático (1987)
- [4] M.C.Rivara *Selective refinement/desrefinement algorithms for sequences of nested triangulations*. International Journal for Numerical Methods in Engineering. Vol 28, 2889-2906 (1989).
- [5] E. Mercader y M.C.Rivara *Herramienta para resolver EDP en 2-D mediante métodos adaptativos y multimallas*. Numerical Methods in Engineering and Applied Sciences, Part I,543-551(1992)
- [6] David S. Burnett, Finite Element Analysis, from concepts to applications. Addison-Wesley (1987)
- [7] Rivara (1997)
- [8] Rivara, Hitschfeld, Swipson (2002)
- [9] James W. Demmel and Stanley C. Eisenstat and John R. Gilbert and Xiaoye S. Li and Joseph W. H. Liu, A supernodal approach to sparse partial pivoting, SIAM J. Matrix Analysis and Applications, Vol 20, numero 3, 720-755 (1999).

Apéndice A

Estructura de datos

Algunas de las estructuras de datos que se utilizan en el programa corresponden a las de las definiciones de nodo, lado, triángulo, conexión, molécula, triangulación, región y expresión.

A.1. Nodo

El nodo es la unidad mínima de la triangulación y con ella se forman los lados, los triángulos, las posiciones y las moléculas.

```
/*
Definicion de Nodo: que incluye identificacion, coordenadas,
valor obtenido en la iteración previa, tipo,
molecula mas joven en que esta definido, y punteros a nodo previo
siguiente.
*/
struct molecule_info;
typedef struct node_info {
int id;
double coord[ 2 ];
double valor_pre;
struct tipo_info tipo;
struct molecule_info *mas_joven;
struct node_info *siguiente, *previo;
} *node;
```

En que el tipo está dado por la estructura siguiente:

```
/*
Definicion de Tipo: incluye las definiciones de las propiedades
de un nodo, como son posicion, forma del lado, referencia al
lado, referencia a las características del nodo.
```

```

*/
typedef struct tipo_info {
char tipo;
/* 0b0000nslp
n : cond. de borde natural (0 no 1 si)
s : status (0 activo 1 borrado)
l : lado (0 recto 1 curvo)
p : posicion (0 interior 1 borde) */
short cborde;
short spline;
short clase;
} _tipo_info;

```

A.2. Lado

Dos nodos contiguos forman el lado de un triángulo.

```

/*
Definicion Lado : figura compuesta por dos nodos.
*/
typedef node side[ 2 ];

```

A.3. Triángulo

Formado por los tres nodos contiguos que definen los vértices del triángulo.

```

/*
Definicion Triangulos : figura compuesta por tres nodos.
*/
typedef node triangle[ 3 ];

```

A.4. Conexión

Conexión, estructura que representa una lista de nodos asociados. Esta lista permitirá crear las moléculas.

```

/*
Definicion de Conexiones : que incluye nodo y siguiente conexion.
*/
typedef struct position_info {
node n;

```

```

struct position_info *nxt;
} *position;

```

A.5. Molécula

Molécula, estructura que representa a un nodo y a todos sus vecinos definidos por medio de una conexión.

```

/*
Definicion de Molecula : que incluye el nodo alrededor del cual
define la molecula, la lista de nodos de la misma, referencias a
las moleculas asociadas al nodo inmediatamente mas joven en la
misma triangulacion y a la de la version inmediatamente anterior
de dicho nodo.
*/
typedef struct molecule_info {
node nd;
struct molecule_info *old_t, *old_n;
position fstp;
} *molecule;

```

A.6. Triangulación

Triangulación, estructura formada por un nodo inicial y sus moléculas

```

/*
Definicion de Triangulacion : que incluye una identificacion, un
primer nodo de la misma, la ultima molecula y enlaces a las trian-
gulaciones previas y siguientes.
*/
typedef struct triangulation_info {
int id;
node primer_nodo;
struct triangulation_info *siguiente, *previo;
molecule ultima_molecula;
} *triangulation;

```

A.7. Región

La región permite definir áreas de refinamiento o desrefinamiento.

```
/*
Definicion de la estructura que permite manejar las regiones para,
refinar y desrefinar.
La region de refinamiento y la de desrefinamiento se logran con la
union de todas las regiones.
*/
typedef struct region_info{
char tipo;
/* c : circulo
r : rectangulo */
double p[4];
/* Si es circulo : (p[0],p[1]) centro
p[2] radio
Si es rectangulo : (p[0],p[1]) esq. sup izq
(p[2],p[3]) esq. inf der */
} region;
```

A.8. Expresión

Las expresiones permiten a la herramienta utilizar las expresiones algebraicas con que los usuarios definen el problema y sus condiciones de borde.

```
/*
Definicion de la estructura que permitira definir
funciones
*/
typedef struct def_expr {
char numarg; /* 0 operando, >0 operador */
union {
double (*oper)();
double val;
} u;
} *expresion;
typedef struct arb_eva {
char operacion;
double valor;
```

```
struct arb_eva *ramizq, *ramder;  
} *farbol;
```

Apéndice B

Formato de los archivos de datos

Los datos para el programa se entregan en dos archivos diferentes. Un archivo, contiene la descripción del problema a resolver (.dat), el otro la descripción de la malla a utilizar (.trg). Estos archivos son archivos de texto, en los que cada línea contiene información específica y su formato depende del tipo de información contenida.

B.1. Estructura del archivo de descripción del problema a resolver

Las tablas B.1 en la página 75, B.2 de la página 76 y B.3 de la página 77 describen la estructura del archivo de descripción del problema a resolver, indicando el significado de cada línea del archivo.

La tabla B.1, describe las primeras 6 líneas del archivo que definen parámetros generales sobre la forma de resolver el problema, esto es tipo de librería sparse a utilizar, tipo de indicador del error evaluar en los procesos de refinación, información sobre la cantidad de regiones de refinamiento y desrefinamiento, tipo de problema y parámetros asociados a estos.

La tabla B.2, describe las líneas siguientes, que incluyen parámetros de forma de los triángulos y descripción de las regiones de refinamiento y desrefinamiento, mas parámetros asociados al refinamiento y desrefinamiento.

La tabla B.3, describe las últimas líneas del archivo en las que se especifica la información de la ecuación diferencial a resolver.

B.2. Estructura del archivo de descripción de la malla a utilizar

Las tablas B.4 en la página 79, B.5 de la página 80 y B.6 de la página 81 describen la estructura del archivo de descripción de la malla a utilizar al resolver el problema, indicando el significado de cada línea del archivo.

La tabla B.4 contiene la descripción de los nodos de la malla y la información sobre la cantidad de estos y cuantos participan en la malla inicial.

La tabla B.5 contiene la descripción de las triangulaciones que forman la malla, comenzando con la cantidad de triangulación y luego para cada triangulación, la identificación de la triangulación, la cantidad de nodos de esta y la descripción de las moléculas asociadas a

Línea	Descripción del campo	Formato															
1	Nombre del archivo de descripción de la malla (sin la extensión '.trg')	Texto															
2	Tipo de librería sparse a utilizar: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Valor</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Sparse 1.3</td> </tr> <tr> <td>2</td> <td>Super LU 2.0</td> </tr> </tbody> </table>	Valor	Descripción	1	Sparse 1.3	2	Super LU 2.0	Numérico (1 dígito)									
Valor	Descripción																
1	Sparse 1.3																
2	Super LU 2.0																
3	Indicador: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Valor</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Error real</td> </tr> <tr> <td>2</td> <td>Error estimado</td> </tr> <tr> <td>3</td> <td>Suavidad de la solución</td> </tr> </tbody> </table>	Valor	Descripción	1	Error real	2	Error estimado	3	Suavidad de la solución	Numérico (1 dígito)							
Valor	Descripción																
1	Error real																
2	Error estimado																
3	Suavidad de la solución																
4	Genera archivo con información de depuración adicional: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Valor</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>S</td> <td>si se genera</td> </tr> <tr> <td>N</td> <td>no se genera</td> </tr> </tbody> </table>	Valor	Descripción	S	si se genera	N	no se genera	Texto (1 carácter)									
Valor	Descripción																
S	si se genera																
N	no se genera																
5	Otros parámetros: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>nref</td> <td>$0 \leq nref \leq 100$</td> <td>Número de regiones de refinamiento</td> </tr> <tr> <td>ndes</td> <td>$0 \leq ndes \leq 100$</td> <td>Número de regiones de desrefinamiento</td> </tr> <tr> <td>Tipo de problema</td> <td>1, 2</td> <td>1: Elíptico 2: Parabólico</td> </tr> <tr> <td>Grado</td> <td>3</td> <td>3: Triángulos lineales</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	nref	$0 \leq nref \leq 100$	Número de regiones de refinamiento	ndes	$0 \leq ndes \leq 100$	Número de regiones de desrefinamiento	Tipo de problema	1, 2	1: Elíptico 2: Parabólico	Grado	3	3: Triángulos lineales	Numérico múltiple (4 números enteros separados por espacios)
Parámetro	Valores	Descripción															
nref	$0 \leq nref \leq 100$	Número de regiones de refinamiento															
ndes	$0 \leq ndes \leq 100$	Número de regiones de desrefinamiento															
Tipo de problema	1, 2	1: Elíptico 2: Parabólico															
Grado	3	3: Triángulos lineales															
6	Parámetros de tiempo en problemas parabólicos: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>ti</td> <td>$ti \geq 0,0$</td> <td>Tiempo inicial del problema planteado</td> </tr> <tr> <td>tf</td> <td>$tf \geq 0,0$</td> <td>Tiempo final del problema planteado</td> </tr> <tr> <td>θ</td> <td>$0,0 \leq \theta \leq 1,0$</td> <td>Tasa</td> </tr> <tr> <td>dt</td> <td>$dt \geq 0,0$</td> <td>Incrementos del tiempo en cada iteración</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	ti	$ti \geq 0,0$	Tiempo inicial del problema planteado	tf	$tf \geq 0,0$	Tiempo final del problema planteado	θ	$0,0 \leq \theta \leq 1,0$	Tasa	dt	$dt \geq 0,0$	Incrementos del tiempo en cada iteración	Numérico múltiple (4 valores de punto flotante separados por espacios)
Parámetro	Valores	Descripción															
ti	$ti \geq 0,0$	Tiempo inicial del problema planteado															
tf	$tf \geq 0,0$	Tiempo final del problema planteado															
θ	$0,0 \leq \theta \leq 1,0$	Tasa															
dt	$dt \geq 0,0$	Incrementos del tiempo en cada iteración															

Cuadro B.1: Archivo de descripción del problema (1/3)

Línea	Descripción del campo	Formato																		
7	Parámetros de forma de los triángulos en la malla: <table border="1"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>dmax</td> <td>$dmax \geq 0,0$</td> <td>Diámetro máximo de los triángulos</td> </tr> <tr> <td>dmin</td> <td>$dmin \geq 0,0$</td> <td>Diámetro mínimo de los triángulos</td> </tr> <tr> <td>α</td> <td>$0,0 \leq \alpha \leq 1,0$</td> <td>Valores de ángulos interiores</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	dmax	$dmax \geq 0,0$	Diámetro máximo de los triángulos	dmin	$dmin \geq 0,0$	Diámetro mínimo de los triángulos	α	$0,0 \leq \alpha \leq 1,0$	Valores de ángulos interiores	Numérico múltiple (3 valores de punto flotante separados por espacios)						
Parámetro	Valores	Descripción																		
dmax	$dmax \geq 0,0$	Diámetro máximo de los triángulos																		
dmin	$dmin \geq 0,0$	Diámetro mínimo de los triángulos																		
α	$0,0 \leq \alpha \leq 1,0$	Valores de ángulos interiores																		
8 - n n=8+nref	Descripción de la regiones de refinamiento: <table border="1"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>tipo</td> <td>C, R</td> <td>Círculo, Rectángulo</td> </tr> <tr> <td>x_c, x_i</td> <td>$\{x_c, x_i\} \in \mathbb{R}, x_i \leq x_f$</td> <td>Coordenada horizontal del centro del círculo (tipo C) u horizontal inferior (tipo R)</td> </tr> <tr> <td>y_c, y_i</td> <td>$\{y_c, y_i\} \in \mathbb{R}, y_i \leq y_f$</td> <td>Coordenada horizontal del centro del círculo (tipo C) o vertical inferior (tipo R)</td> </tr> <tr> <td>r, x_f</td> <td>$\{r, y_f\} \in \mathbb{R}, x_i \leq x_f$</td> <td>Radio del círculo (tipo C) o coordenada horizontal superior (tipo R)</td> </tr> <tr> <td>$0,0, y_f$</td> <td>$\{y_f\} \in \mathbb{R}, y_i \leq y_f$</td> <td>0.0 (tipo C) o coordenada vertical superior (tipo R)</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	tipo	C, R	Círculo, Rectángulo	x_c, x_i	$\{x_c, x_i\} \in \mathbb{R}, x_i \leq x_f$	Coordenada horizontal del centro del círculo (tipo C) u horizontal inferior (tipo R)	y_c, y_i	$\{y_c, y_i\} \in \mathbb{R}, y_i \leq y_f$	Coordenada horizontal del centro del círculo (tipo C) o vertical inferior (tipo R)	r, x_f	$\{r, y_f\} \in \mathbb{R}, x_i \leq x_f$	Radio del círculo (tipo C) o coordenada horizontal superior (tipo R)	$0,0, y_f$	$\{y_f\} \in \mathbb{R}, y_i \leq y_f$	0.0 (tipo C) o coordenada vertical superior (tipo R)	Texto (1 carácter) + Numérico múltiple (4 valores de punto flotante separados por espacios)
Parámetro	Valores	Descripción																		
tipo	C, R	Círculo, Rectángulo																		
x_c, x_i	$\{x_c, x_i\} \in \mathbb{R}, x_i \leq x_f$	Coordenada horizontal del centro del círculo (tipo C) u horizontal inferior (tipo R)																		
y_c, y_i	$\{y_c, y_i\} \in \mathbb{R}, y_i \leq y_f$	Coordenada horizontal del centro del círculo (tipo C) o vertical inferior (tipo R)																		
r, x_f	$\{r, y_f\} \in \mathbb{R}, x_i \leq x_f$	Radio del círculo (tipo C) o coordenada horizontal superior (tipo R)																		
$0,0, y_f$	$\{y_f\} \in \mathbb{R}, y_i \leq y_f$	0.0 (tipo C) o coordenada vertical superior (tipo R)																		
(n+1) - m m=n+1+ndes	Descripción de la regiones de desrefinamiento: Idéntica a las regiones de refinamiento (líneas previas).	Ídem.																		
(m+1) - o o=m+1+nref	Diámetro mínimo de los triángulos en las áreas de refinamiento correspondientes	Numérico																		
(o+1) - p p=o+1+dref	Diámetro mínimo de los triángulos en las áreas de desrefinamiento correspondientes	Numérico																		

Cuadro B.2: Archivo de descripción del problema (2/3)

Línea	Descripción del campo	Formato
p+1	Constante (0)	Numérico (dígito 0)
p+2	γ_t : Constante para los problemas parabólicos (ver 4.3.1)	Texto (finaliza ^L)
p+3	α_x : Constante para los problemas elípticos y parabólicos (ver ecuaciones 4.2.1 y 4.3.1)	Texto (finaliza ^L)
p+4	α_y : Constante para los problemas elípticos y parabólicos (ver ecuaciones 4.2.1 y 4.3.1)	Texto (finaliza ^L)
p+5	β : Constante para los problemas elípticos y parabólicos (ver ecuaciones 4.2.1 y 4.3.1)	Texto (finaliza ^L)
p+6	f : RHS para los problemas elípticos y parabólicos (ver ecuaciones 4.2.1 y 4.3.1)	Texto (finaliza ^L)
p+7	$U(x, y, t_o)$: Solución para los problemas parabólicos en tiempo inicial	Texto (finaliza ^L)
p+8	$U(x, y, t)$: Solución para los problemas elípticos y parabólicos	Texto (finaliza ^L)
p+9	$\partial U(x, y, t)/\partial x$: Derivada parcial para los problemas elípticos y parabólicos	Texto (finaliza ^L)
p+10	$\partial U(x, y, t)/\partial y$: Derivada parcial para los problemas elípticos y parabólicos	Texto (finaliza ^L)
p+11	Constante (vacía)	Texto (finaliza ^L)

Cuadro B.3: Archivo de descripción del problema (3/3)

cada nodo. La cantidad de información dependerá de la cantidad de triangulaciones en la malla. Las triangulaciones se van describiendo por “edad”, desde la inicial o mas gruesa hasta la última o mas fina.

La tabla B.6 contiene la descripción de los lados curvos que dan forma al dominio y a las condiciones de borde asociadas. Comienza con la cantidad de lados curvos y condiciones de borde, para luego describirlas. Los lados curvos pueden ser splines cúbicas o arcos de círculo. Las condiciones de borde pueden ser naturales o esenciales.

Linea	Descripción del campo	Formato																																				
1	Descripción del problema	Texto																																				
2	Parámetros de la malla: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>totnod</td> <td>$0 \leq \text{totnod}$</td> <td>Cantidad total de nodos</td> </tr> <tr> <td>ulnodi</td> <td>$0 \leq \text{ulnodi} \leq \text{totnod}$</td> <td>Último nodo de la malla inicial</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	totnod	$0 \leq \text{totnod}$	Cantidad total de nodos	ulnodi	$0 \leq \text{ulnodi} \leq \text{totnod}$	Último nodo de la malla inicial	Numérico múltiple (2 valores enteros separados por espacio)																											
Parámetro	Valores	Descripción																																				
totnod	$0 \leq \text{totnod}$	Cantidad total de nodos																																				
ulnodi	$0 \leq \text{ulnodi} \leq \text{totnod}$	Último nodo de la malla inicial																																				
3-totnod+2	Descripción de los nodos: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>$1 \leq \text{id}$</td> <td>Identificación del nodo</td> </tr> <tr> <td>tipo</td> <td>$0 \leq \text{tipo} \leq 15$</td> <td> Características del nodo (binario): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>bit</th> <th>descripción</th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>0: Uso futuro</td> </tr> <tr> <td>4</td> <td>1: Tiene condición de borde natural 0: No tiene condición de borde natural</td> </tr> <tr> <td>5</td> <td>1: activo 0: borrado</td> </tr> <tr> <td>6</td> <td>1: lado curvo 0: lado recto</td> </tr> <tr> <td>7</td> <td>1: nodo borde 0: nodo interno</td> </tr> </tbody> </table> <p>Ejemplo: $9 \rightarrow 0x08$ (condición de borde natural) + $0x01$ (ubicado en el borde)</p> </td> </tr> <tr> <td>cborde</td> <td>$0 \leq \text{cborde}$</td> <td>Identificación de la condición de borde (0 sin condición de borde)</td> </tr> <tr> <td>spline</td> <td>$0 \leq \text{spline}$</td> <td>Identificación de la spline asociada a borde curvo(0 sin spline)</td> </tr> <tr> <td>clase</td> <td>$0 \leq \text{clase}$</td> <td>Clase del nodo (0: nodo pertenece a la interfase entre 2 medios)</td> </tr> <tr> <td>x</td> <td>$x \in \mathbb{R}$</td> <td>Coordenada del nodo en eje X</td> </tr> <tr> <td>y</td> <td>$x \in \mathbb{R}$</td> <td>cadena de caracteres, entre Coordenada del nodo en eje Y</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	id	$1 \leq \text{id}$	Identificación del nodo	tipo	$0 \leq \text{tipo} \leq 15$	Características del nodo (binario): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>bit</th> <th>descripción</th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>0: Uso futuro</td> </tr> <tr> <td>4</td> <td>1: Tiene condición de borde natural 0: No tiene condición de borde natural</td> </tr> <tr> <td>5</td> <td>1: activo 0: borrado</td> </tr> <tr> <td>6</td> <td>1: lado curvo 0: lado recto</td> </tr> <tr> <td>7</td> <td>1: nodo borde 0: nodo interno</td> </tr> </tbody> </table> <p>Ejemplo: $9 \rightarrow 0x08$ (condición de borde natural) + $0x01$ (ubicado en el borde)</p>	bit	descripción	0-3	0: Uso futuro	4	1: Tiene condición de borde natural 0: No tiene condición de borde natural	5	1: activo 0: borrado	6	1: lado curvo 0: lado recto	7	1: nodo borde 0: nodo interno	cborde	$0 \leq \text{cborde}$	Identificación de la condición de borde (0 sin condición de borde)	spline	$0 \leq \text{spline}$	Identificación de la spline asociada a borde curvo(0 sin spline)	clase	$0 \leq \text{clase}$	Clase del nodo (0: nodo pertenece a la interfase entre 2 medios)	x	$x \in \mathbb{R}$	Coordenada del nodo en eje X	y	$x \in \mathbb{R}$	cadena de caracteres, entre Coordenada del nodo en eje Y	
Parámetro	Valores	Descripción																																				
id	$1 \leq \text{id}$	Identificación del nodo																																				
tipo	$0 \leq \text{tipo} \leq 15$	Características del nodo (binario): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>bit</th> <th>descripción</th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>0: Uso futuro</td> </tr> <tr> <td>4</td> <td>1: Tiene condición de borde natural 0: No tiene condición de borde natural</td> </tr> <tr> <td>5</td> <td>1: activo 0: borrado</td> </tr> <tr> <td>6</td> <td>1: lado curvo 0: lado recto</td> </tr> <tr> <td>7</td> <td>1: nodo borde 0: nodo interno</td> </tr> </tbody> </table> <p>Ejemplo: $9 \rightarrow 0x08$ (condición de borde natural) + $0x01$ (ubicado en el borde)</p>	bit	descripción	0-3	0: Uso futuro	4	1: Tiene condición de borde natural 0: No tiene condición de borde natural	5	1: activo 0: borrado	6	1: lado curvo 0: lado recto	7	1: nodo borde 0: nodo interno																								
bit	descripción																																					
0-3	0: Uso futuro																																					
4	1: Tiene condición de borde natural 0: No tiene condición de borde natural																																					
5	1: activo 0: borrado																																					
6	1: lado curvo 0: lado recto																																					
7	1: nodo borde 0: nodo interno																																					
cborde	$0 \leq \text{cborde}$	Identificación de la condición de borde (0 sin condición de borde)																																				
spline	$0 \leq \text{spline}$	Identificación de la spline asociada a borde curvo(0 sin spline)																																				
clase	$0 \leq \text{clase}$	Clase del nodo (0: nodo pertenece a la interfase entre 2 medios)																																				
x	$x \in \mathbb{R}$	Coordenada del nodo en eje X																																				
y	$x \in \mathbb{R}$	cadena de caracteres, entre Coordenada del nodo en eje Y																																				

Cuadro B.4: Archivo de descripción de la malla (1/3)

Linea	Descripción del campo	Formato																		
totnod+3	Cantidad de triangulaciones	Numérico																		
totnod+4	Parámetros de la primera triangulación: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>$1 \leq id$</td> <td>Identificación de la triangulación</td> </tr> <tr> <td>numnod</td> <td>$0 \leq numod \leq totnod$</td> <td>Cantidad de nodos de la triangulación</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	id	$1 \leq id$	Identificación de la triangulación	numnod	$0 \leq numod \leq totnod$	Cantidad de nodos de la triangulación	Numérico múltiple (2 valores enteros separados por espacio)									
Parámetro	Valores	Descripción																		
id	$1 \leq id$	Identificación de la triangulación																		
numnod	$0 \leq numod \leq totnod$	Cantidad de nodos de la triangulación																		
totnod+4- totnod+ numnod+4	Descripción de las moléculas de la primera triangulación: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo</td> </tr> <tr> <td>cantidad</td> <td>$0 \leq tipo \leq 24$</td> <td>Cantidad de nodos vecinos que forman parte de la molécula</td> </tr> <tr> <td>idVecino</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo vecino</td> </tr> <tr> <td>idVecino</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo vecino</td> </tr> <tr> <td>... (ídem)</td> <td>... (ídem)</td> <td>... (ídem)</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	id	$1 \leq numnod$	Identificación del nodo	cantidad	$0 \leq tipo \leq 24$	Cantidad de nodos vecinos que forman parte de la molécula	idVecino	$1 \leq numnod$	Identificación del nodo vecino	idVecino	$1 \leq numnod$	Identificación del nodo vecino	... (ídem)	... (ídem)	... (ídem)	Numérico múltiple (2 valores enteros separados por espacio)
Parámetro	Valores	Descripción																		
id	$1 \leq numnod$	Identificación del nodo																		
cantidad	$0 \leq tipo \leq 24$	Cantidad de nodos vecinos que forman parte de la molécula																		
idVecino	$1 \leq numnod$	Identificación del nodo vecino																		
idVecino	$1 \leq numnod$	Identificación del nodo vecino																		
... (ídem)	... (ídem)	... (ídem)																		
linea siguiente	Parámetros de las siguientes triangulaciones (si existen): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>$1 \leq id$</td> <td>Identificación de la triangulación</td> </tr> <tr> <td>numnod</td> <td>$0 \leq numod \leq totnod$</td> <td>Cantidad de nodos de la triangulación</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	id	$1 \leq id$	Identificación de la triangulación	numnod	$0 \leq numod \leq totnod$	Cantidad de nodos de la triangulación	Numérico múltiple (2 valores enteros separados por espacio)									
Parámetro	Valores	Descripción																		
id	$1 \leq id$	Identificación de la triangulación																		
numnod	$0 \leq numod \leq totnod$	Cantidad de nodos de la triangulación																		
linea siguiente	Descripción de las moléculas de las siguientes triangulaciones (si existen): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Parámetro</th> <th>Valores</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo</td> </tr> <tr> <td>cantidad</td> <td>$0 \leq tipo \leq 24$</td> <td>Cantidad de nodos vecinos que forman parte de la molécula</td> </tr> <tr> <td>idVecino</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo vecino</td> </tr> <tr> <td>idVecino</td> <td>$1 \leq numnod$</td> <td>Identificación del nodo vecino</td> </tr> <tr> <td>... (ídem)</td> <td>... (ídem)</td> <td>... (ídem)</td> </tr> </tbody> </table>	Parámetro	Valores	Descripción	id	$1 \leq numnod$	Identificación del nodo	cantidad	$0 \leq tipo \leq 24$	Cantidad de nodos vecinos que forman parte de la molécula	idVecino	$1 \leq numnod$	Identificación del nodo vecino	idVecino	$1 \leq numnod$	Identificación del nodo vecino	... (ídem)	... (ídem)	... (ídem)	Numérico múltiple (2 valores enteros separados por espacio)
Parámetro	Valores	Descripción																		
id	$1 \leq numnod$	Identificación del nodo																		
cantidad	$0 \leq tipo \leq 24$	Cantidad de nodos vecinos que forman parte de la molécula																		
idVecino	$1 \leq numnod$	Identificación del nodo vecino																		
idVecino	$1 \leq numnod$	Identificación del nodo vecino																		
... (ídem)	... (ídem)	... (ídem)																		

Cuadro B.5: Archivo de descripción de la malla (2/3)

Linea	Descripción del campo			Formato
n	Parámetros de la primera triangulación:			Numérico múltiple (2 valores enteros separados por espacio)
	Parámetro	Valores	Descripción	
	numspl	$0 \leq \text{numspl}$	Cantidad de lados curvos	
	numcbd	$0 \leq \text{numcbd}$	Cantidad de condiciones de borde	
n+1- n+numspl+1	Descripción de los bordes curvos:			Numérico múltiple (valores enteros y reales separados por espacio)
	Parámetro	Valores	Descripción	
	tipo	s,c	Tipo del lado curvo s:spline c: círculo	
	id_inicio	$1 \leq \text{numnod}$	Identificación del nodo de inicio del lado curvo	
	id_final	$1 \leq \text{numnod}$	Identificación del nodo de fin del lado curvo	
parámetros	reales	3 valores reales que definen una círculo (coordenada x e y del centro del arco y el radio del mismo, o 8 valores reales que definen las coordenadas x e y de los puntos de control de la spline.		
n+numspl+1- n+numspl+ numcbd+1	Descripción de las condiciones de borde:			Numérico múltiple (2 valores enteros separados por espacio)
	Parámetro	Valores	Descripción	
	tipo	e,n	Tipo de la condición de borde: e: esencial n:natural	
	id_inicio	$1 \leq \text{numnod}$	Identificación del nodo de inicio de la condición de borde	
	id_final	$1 \leq \text{numnod}$	Identificación del nodo de inicio de la condición de borde	
expresión	cadena de caracteres, entre comillas	Expresión que contiene la ecuación que permite evaluar la condición de borde en un punto.		

Cuadro B.6: Archivo de descripción de la malla (3/3)

Apéndice C

Expresiones aritméticas

Las expresiones aritméticas, siguen las reglas de sintaxis, para su validación y evaluación, que se muestran en la cuadro siguiente, expresado de acuerdo a la notación de gramática BNF:

```
<expr> ::= "(" <expr> ")" |
          <expr> "+" <expr> |
          "-" <expr> |
          <expr> "-" <expr> |
          <expr> "*" <expr> |
          <expr> "/" <expr> |
          "sin (" <expr> ")" |
          "cos (" <expr> ")" |
          "tan (" <expr> ")" |
          "exp (" <expr> ")" |
          "log (" <expr> ")" |
          "pow (" <expr> "," <expr> ")" |
          "sqr (" <expr> ")" |
          "x" | "y" | "t" |
          "PI" |
          <cons>
<cons> ::= <natural> | <decimal>
<natural> ::= digito | <natural>
<digito> ::= "0" | "1" | "2" | "3" | "4" |
            "5" | "6" | "7" | "8" | "9"
<decimal> ::= <natural> "." |
             [ <natural> ] "." <natural>
```

El significado de las representaciones antes indicadas se muestra en la tabla C.1 en la página siguiente.

El orden de precedencia en los operadores, en caso de no haber paréntesis que definan la secuencia de operaciones es :

1. “-” unuario
2. “*” y “/”
3. “+” y “-”

Clase	Expresión	Descripción
expr	(expr)	Agrupación
	expr + expr	Operación suma
	- expr	Operación opuesto (inverso para suma)
	expr - expr	Operación resta
	expr * expr	Operación multiplicación
	expr / expr	Operación división
	sin(expr)	Función seno
	cos(expr)	Función coseno
	tan(expr)	Función tangente
	exp (expr)	Función exponencial
	log (expr)	Función logaritmo
	pow (expr , expr)	Función potencia
	sqrt(expr)	Función raíz cuadrada
	x	Variable largo (eje X)
	y	Variable ancho (eje Y)
	z	Variable alto (eje Z)
	t	Variable de tiempo
	pi	M_PI = 3.14159265358979323846
	cons	Constante
cons	[0-9]+	Número entero
	[0-9]+ "." [0-9]*	Número decimal, con 0 o más decimales
	[0-9]* "." [0-9]+	Número decimal, con 1 o más decimales y con 0 o mas dígitos enteros

Cuadro C.1: Descripción de los elementos de la gramática de expresiones