
INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



Revisión e implementación del método de elementos finitos.

TESIS

que para obtener el título de

Licenciado en Matemáticas Aplicadas

presenta

Miguel Angel Escalante Serrato

MÉXICO, D.F.

2013

Asesor: Juan Carlos Aguilar Villegas

Revisor: José Luis Farah Ibáñez

Con fundamento en los artículos 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada “**Revisión e implementación del método de elementos finitos.**”, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Baillères Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación.

Miguel Angel Escalante Serrato

Fecha

Firma

Revisión e implementación del método de elementos finitos

Miguel Angel Escalante Serrato

A mi Chango, Jefecita y Caranala.

AGRADECIMIENTOS

A mis padres ya que sin ellos, nada de esto pudo haber sucedido. A mi hermana por ser mi compañera incondicional.

A mi asesor Juan Carlos Aguilar por su paciencia y apoyo durante todo este tiempo.

A mis sinodales José Luis Farah, José Luis Morales y Zeferino Parada por tomarse el tiempo de leer este documento y por sus valiosas recomendaciones.

Al Animalito Salvaje, por las sesiones cafeteras y correcciones de todo tipo.

Al profesor Morones por sus comentarios desde España.

A mis amigos que de una u otra manera, alentado o ayudado a este trabajo. Rafa, Irving, Stefania, Alfredo, Maite, Andy, Marina, René.

A Anita por todo el apoyo, cariño y tiempo.

Prefacio

Un problema bastante común en diversas áreas científicas es el de resolver ecuaciones diferenciales parciales, es un problema tanto interesante como relevante ya que es muy usado, por ejemplo, en diversas áreas para modelar la reacción de distintos materiales a diversas condiciones y situaciones. Existen diversos métodos para resolver este tipo de ecuaciones; dependiendo del problema se debe de analizar qué herramienta es la más adecuada para el trabajo. Aquellos problemas en donde una malla no es lo suficientemente robusta para describir el dominio, es el tipo de problemas en los que el método de los elementos finitos se especializa. En particular en esta tesis se implementa un método para resolver un subconjunto de la amplia familia de ecuaciones diferenciales parciales, las elípticas. Es un trabajo que nace de la curiosidad de lo que hay más allá de la solución exacta en pizarrón o papel y lápiz; ¿qué pasa cuando el papel y el lápiz no pueden resolver el problema? ¿Qué pasa cuando se tiene que usar el poder de cómputo? Con la guía y apoyo de mi asesor fui respondiendo estas preguntas para este problema y espero sirva para que el lector encuentre una vela para buscar el camino a la respuesta que busca.

Índice general

1. Introducción	1
2. Conceptos	5
2.1. Espacios de Hilbert	5
2.2. Integración por partes	9
2.3. Derivada débil de funciones en $\mathcal{L}^2(I)$	10
2.4. Derivada débil en $\mathcal{L}^2(\Omega)$	13
2.5. Espacios $\mathcal{L}^2(\Omega), H^1(\Omega), H_0^1(\Omega)$	13
2.6. Formulación variacional	14
2.7. Ejemplos	18
3. Método de elementos finitos	20
3.1. Representación de funciones	20

<i>ÍNDICE GENERAL</i>	II
3.2. Nodos	21
3.3. Funciones nodales	24
3.4. Triangulación	26
3.5. Construcción del sistema de ecuaciones	29
3.6. Manejo de datos	31
3.7. Número de condición de la matriz A	32
3.8. Cuadraturas y cálculo de integrales.	34
3.9. Resolución del sistema de ecuaciones: gradiente conjugado	36
3.10. Error de aproximación	36
4. Dimensión 1	37
4.1. Ecuaciones diferenciales no lineales.	42
4.1.1. Método de Newton	43
5. Resultados numéricos	45
6. Conclusiones	50
A. Algoritmo de Delaunay	52
B. Método del gradiente conjugado.	55
C. Prueba del lema 3.1	58
D. Código	63
D.1. Main	63

ÍNDICE GENERAL

III

D.2. Triangulo.m	64
D.3. Vecinos.m	66
D.4. Construo.m	67
D.5. gradp.m	72
D.6. area.m	72
D.7. gradconj.m	73

CAPÍTULO 1

Introducción

El modelado es una de las herramientas más poderosas de la matemática actual. Muchos modelos en diversas áreas científicas tienen la forma de ecuaciones diferenciales o integrales. La capacidad de cómputo ha crecido considerablemente desde los inicios del cálculo numérico. Gracias a esto es posible modelar y resolver problemas cada vez más grandes y complejos, por lo que la necesidad de métodos eficientes para el cómputo se hace evidente. En este trabajo estudiamos un problema que tiene relevancia en toda clase de problemas físicos: ecuaciones diferenciales parciales. Dentro de las aplicaciones, ingeniería estructural, resistencia de materiales, mecánica de fluidos, ingeniería nuclear, electromagnetismo, propagación de onda, etc.

La resolución de ecuaciones diferenciales parciales es parte de nuestra vida cotidiana aunque no lo veamos explícitamente; se usa en una gran variedad de modelos para diseñar objetos que tenemos hoy en día, desde piezas de autos hasta aparatos más pequeños como los celulares. El Método de Elementos Finitos (MEF) es una herramienta que ayuda a resolver las ecuaciones necesarias.

El MEF es una técnica general que resuelve numéricamente ecuaciones diferen-

ciales parciales; el método que se programó para este trabajo, resuelve ecuaciones diferenciales parciales elípticas, aunque puede ser adaptado para otro tipo de problemas, ecuaciones parabólicas por ejemplo.

Uno de los primeros antecedentes del MEF, vino con un trabajo en 1909 por W. Ritz donde desarrolla un método de aproximación para problemas de mecánica en sólidos deformables. Más tarde, en 1943 R. Courant expandió el trabajo hecho por Ritz al introducir funciones lineales sobre dominios triangulares, sin embargo el desarrollo se retoma hasta años más tarde; el MEF como se plantea en este trabajo fue desarrollado por ingenieros de la universidad de Berkeley, a finales de los años 50, principios de los años 60, y en particular se generó como una técnica particular para resolver algunos problemas de estructuras en ingeniería. A mediados de los años 60, conforme se estudiaba, fue resultando evidente el hecho que es en realidad una técnica general para resolver ecuaciones diferenciales parciales, esto con base en la teoría variacional planteada a principios de siglo. Conforme fue avanzando el tiempo el MEF fue generalizado de tal manera que resolviese todo tipo de ecuaciones diferenciales e integrales.

Actualmente el MEF es usado en los programas de Diseño Asistido por Computadora (CAD por sus siglas en inglés). Los CAD pueden simular una gama amplia de condiciones como temperatura, elasticidad, materiales, fuerza aplicada y otros más, haciéndolos una herramienta clave para cualquier diseño

El MEF es un método muy usado por la simplicidad teórica así como la gran cantidad de problemas que puede atacar, todo esto sin demasiada complicación algorítmica, dejando así un método elegante y poderoso.

A continuación introduciremos ejemplos de los problemas que se pueden resolver con el MEF.

Ejemplo 1.1. Encontrar u tal que:

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= 0 \quad \text{en } \Gamma, \end{aligned} \tag{1.1}$$

donde

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2},$$

Ω es el dominio de la función u , el cual es abierto y acotado, con frontera Γ de clase C^1 por trozos.

Este es el ejemplo más sencillo y conocido de las ecuaciones diferenciales parciales elípticas y podemos ampliar la ecuación 1.1, cambiando la condición de frontera y hacerla no homogénea:

Ejemplo 1.2. Encontrar u tal que:

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= g \quad \text{en } \Gamma, \end{aligned}$$

donde g es una función de clase C^1 .

Además podemos explorar maneras de resolver ecuaciones con condiciones de frontera más complicadas:

Ejemplo 1.3. Encontrar u tal que:

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= g_1 \quad \text{en } \Gamma_1, \text{long}(\Gamma_1) > 0, \\ \frac{\partial u}{\partial \eta} &= g_2 \quad \text{en } \Gamma_2, \Gamma = \Gamma_1 \cup \Gamma_2, \end{aligned}$$

donde $\frac{\partial u}{\partial \eta}$ es la derivada direccional de u en la dirección η , que es un vector normal unitario a la curva cerrada Γ y que apunta hacia “afuera”.

Varios métodos para aproximar la solución de estas ecuaciones tienen la misma idea: discretizar reduciendo el problema de encontrar la función u , al de encontrar una serie de coeficientes; esto para encontrar una representación de u de tal manera que:

$$u(x) \approx \sum_{i=1}^n \alpha_i \phi_i(x),$$

donde $\{x_i\}_{i=1}^n$ son los puntos donde nos interesa conocer el valor de la función, $\{\phi_i\}_{i=1}^n$ son funciones base que se usarán para aproximar u , n es el tamaño de la base que se usará y finalmente α_i son los coeficientes asociados con las funciones ϕ_i .

El MEF se caracteriza por la forma en que se particiona el dominio (triángulos o cuadriláteros en el caso de dos dimensiones) y porque el tipo de funciones base que usa son polinomiales por trozos y de soporte compacto.

Se hizo una implementación de este método en MATLAB, resuelve ecuaciones diferenciales parciales no homogéneas, y puede ser extendido a resolver otro tipo de ecuaciones, como parabólicas o con condiciones de frontera como las planteadas en el ejemplo 1.3, e inclusive con distintas funciones base, como cuadráticas o cúbicas.

Este trabajo se estructuró de la siguiente manera, en el capítulo 2 mencionaremos algunos conceptos teóricos necesarios para la resolución del problema, el capítulo 3 aborda conceptos más particulares del MEF así como algunas características del mismo, en el capítulo 4 se menciona la implementación del método en una dimensión, con generalizaciones a otros problemas, el capítulo 5 ilustra los resultados del método programado para funciones en dos dimensiones.

CAPÍTULO 2

Conceptos

En este capítulo, daremos la teoría necesaria para trabajar con el método de elementos finitos. Primero necesitamos retomar algunos conceptos de álgebra lineal y análisis funcional.

2.1. Espacios de Hilbert

Definición 2.1. Un *espacio vectorial* V sobre \mathbb{R} consiste en un conjunto donde están definidas dos operaciones $+: V \times V \rightarrow V$ y $\cdot: V \times \mathbb{R} \rightarrow V$, con las siguientes propiedades:

- $\forall x, y \in V \ x + y = y + x$.
- $\forall x, y, z \in V \ (x + y) + z = x + (y + z)$.
- Existe un elemento en V llamado 0 tal que $x + 0 = x \ \forall x \in V$.

- $\forall x \in V \exists y \text{ tal que } x + y = 0.$
- $\forall x \in V 1x = x.$
- $\forall \alpha, \beta \in \mathbb{R}, x \in V (\alpha\beta)x = \alpha(\beta x) = \beta(\alpha x).$
- $\forall x, y \in V \alpha \in \mathbb{R} \alpha(x + y) = \alpha x + \alpha y.$
- $\forall \alpha, \beta \in \mathbb{R}, x \in V (\alpha + \beta)x = \alpha x + \beta x.$

Definición 2.2. Sea H un espacio vectorial, decimos que $\langle \cdot, \cdot \rangle : H \times H \longrightarrow \mathbb{R}$ es una *forma bilineal* si es lineal en cada componente. Es decir que para todos $\alpha, \beta \in \mathbb{R}, v, w, z \in H$ tenemos lo que sigue:

$$\langle \alpha v + \beta w, z \rangle = \alpha \langle v, z \rangle + \beta \langle w, z \rangle,$$

$$\langle z, \alpha v + \beta w \rangle = \alpha \langle z, v \rangle + \beta \langle z, w \rangle.$$

Definición 2.3. Sea H un espacio vectorial sobre \mathbb{R} . Un *producto interior* en H es una forma bilineal $\langle \cdot, \cdot \rangle_H : H \times H \longrightarrow \mathbb{R}$ que satisface lo siguiente:

1. $\langle u, v \rangle_H = \langle v, u \rangle_H \quad \forall u, v \in H$
2. $\langle v, v \rangle_H \geq 0 \quad \forall v \in H$
3. $\langle v, v \rangle_H = 0 \iff v = 0$

Un espacio vectorial con producto interior tiene una norma inducida por ese producto interior, misma que está dada por:

$$\|v\|_H = \sqrt{\langle v, v \rangle_H}.$$

Teorema 2.1. (*Desigualdad de Cauchy-Bunyakovsky-Schwarz (CBS)*) La siguiente desigualdad se cumple para todo espacio vectorial con norma inducida por el producto interior:

$$|\langle u, v \rangle_H| \leq \|u\|_H \|v\|_H.$$

Definición 2.4. Sea H un espacio vectorial con norma $\|\cdot\|_H$. Se dice que una sucesión $\{v_i\}_{i \in \mathbb{N}}$ en H , es de *Cauchy*, si se cumple que $\|v_i - v_j\|_H \rightarrow 0$ cuando $i, j \rightarrow \infty$.

Definición 2.5. Un espacio vectorial H con norma $\|\cdot\|_H$ es *completo* \iff Para cualquier sucesión de Cauchy $\{v_i\}_{i \in \mathbb{N}}$ en H , existe $v \in H$ tal que $\|v_i - v\|_H \rightarrow 0$ cuando $i \rightarrow \infty$.

Definición 2.6. Un *espacio de Hilbert* es un espacio vectorial con producto interior y que es completo con respecto a la norma inducida $\|\cdot\|_H$.

Definición 2.7. Sea H espacio de Hilbert con producto interior $\langle \cdot, \cdot \rangle_H$. Se dice que $u, v \in H$ son *ortogonales* si $\langle u, v \rangle_H = 0$.

Teorema 2.2. (De la proyección ortogonal para espacios de dimensión finita)

Si H es un espacio de Hilbert, $H' \subset H$ un subespacio vectorial de dimensión finita, y si $u \in H$, $w \in H'$ satisfacen que:

$$\langle u - w, v \rangle_H = 0 \quad \forall v \in H',$$

entonces

$$\|u - w\|_H \leq \|u - v\|_H \quad \forall v \in H'.$$

El siguiente es un ejemplo de espacio de Hilbert.

Sea $I = [0, 1] \subset \mathbb{R}$ y

$$\mathcal{L}^2(I) = \left\{ \text{funciones } v : [0, 1] \rightarrow \mathbb{R} \mid \int_0^1 |v(x)|^2 dx < \infty \right\}^1.$$

El espacio de funciones $\mathcal{L}^2(I)$ con el producto interior dado por:

$$\langle u, v \rangle_{\mathcal{L}^2} = \int_0^1 u(x)v(x) dx \quad \forall u, v \in \mathcal{L}^2(I),$$

es un espacio de Hilbert.

En este caso la desigualdad de CBS es:

$$|\langle u, v \rangle|^2 \leq \|u\|_{\mathcal{L}^2}^2 \|v\|_{\mathcal{L}^2}^2,$$

¹En realidad el espacio $\mathcal{L}^2(I)$ está formado por clases de equivalencia: dos funciones g_1 y g_2 están en la misma clase de equivalencia si g_1 y g_2 son iguales excepto posiblemente en un conjunto de medida cero.

$$\text{donde } \|u\|_{\mathcal{L}^2}^2 = \int_0^1 |u(x)|^2 dx.$$

Los siguientes espacios, son interesantes por su relación con $\mathcal{L}^2(I)$.

$$C^\infty(I) = \{ \text{funciones } v : [0, 1] \rightarrow \mathbb{R} \mid v \text{ tiene derivadas de todos los órdenes} \}$$

$$C_0^\infty(I) = \{ \text{funciones } v : [0, 1] \rightarrow \mathbb{R} \mid v \text{ es de clase } C^\infty \text{ y con soporte } \subset (0, 1) \}$$

El espacio $C_0^\infty(I)$ es denso en $\mathcal{L}^2(I)$. Lo que quiere decir que

$$\forall u \in \mathcal{L}^2(I) \text{ y } \varepsilon > 0 \exists v \in C_0^\infty(I) \text{ tal que } \|u - v\|_{\mathcal{L}^2} < \varepsilon.$$

Teorema 2.3. Si $g_1, g_2 \in \mathcal{L}^2(I)$ y:

$$\int_0^1 g_1(x)v(x) dx = \int_0^1 g_2(x)v(x) dx \quad \forall v \in C_0^\infty(I), \text{ entonces } g_1 = g_2.$$

La igualdad $g_1 = g_2$ es en el sentido de las funciones en \mathcal{L}^2 es decir, g_1 y g_2 difieren en un conjunto de medida cero².

Para facilidad de prueba, probemos antes el siguiente lema:

Lema 2.4. Si existe $f \in \mathcal{L}^2(I)$ tal que:

$$\int_0^1 f(x)v(x) dx = 0 \quad \forall v \in C_0^\infty,$$

entonces $f = 0$.

Demostración. Sea $f \in \mathcal{L}^2(I)$, tal que

$$\int_0^1 f(x)v(x) dx = 0 \quad \forall v \in C_0^\infty,$$

$$0 \leq \|f\|_{\mathcal{L}^2}^2 = \left| 0 - \int_0^1 f^2(x) dx \right| = \left| \int_0^1 f(x)v(x) dx - \int_0^1 f^2(x) dx \right|, \quad \forall v \in C_0^\infty.$$

Sabemos que el espacio C_0^∞ es denso en $\mathcal{L}^2(I)$ y por lo tanto, para cada $\varepsilon > 0$, existe $v_\varepsilon \in C_0^\infty$ tal que $\|f - v_\varepsilon\|_{\mathcal{L}^2(I)} < \varepsilon$, $\forall \varepsilon > 0$.

²Para mayor referencia al concepto de medida, consultar [6]

Por lo tanto, para cualquier $\varepsilon > 0$

$$\begin{aligned} \|f\|_{\mathcal{L}^2}^2 &= \left| \int_0^1 f(x)v_\varepsilon(x) \, dx - \int_0^1 f^2(x) \, dx \right| = \left| \int_0^1 f(x)[v_\varepsilon(x) - f(x)] \, dx \right| \\ &\leq \int_0^1 |f(x)[v_\varepsilon(x) - f(x)]| \, dx \leq \|f\|_{\mathcal{L}^2(I)} \|v_\varepsilon - f\|_{\mathcal{L}^2(I)} \end{aligned}$$

De lo cual, $\|f\|_{\mathcal{L}^2} \leq \|v_\varepsilon - f\|_{\mathcal{L}^2} \leq \varepsilon \, \forall \varepsilon > 0$. Por lo tanto, $f = 0$. \square

Demostración. (del teorema 2.3) Tenemos

$$\begin{aligned} \int_0^1 g_1(x)v(x) \, dx &= \int_0^1 g_2(x)v(x) \, dx \quad \forall v \in C_0^\infty(I) \\ \Rightarrow \int_0^1 g_1(x)v(x) \, dx - \int_0^1 g_2(x)v(x) \, dx &= 0 \\ \Rightarrow \int_0^1 (g_1(x) - g_2(x))v(x) \, dx &= 0 \\ \Rightarrow g_1 - g_2 &= 0 \\ \Rightarrow g_1 &= g_2. \end{aligned}$$

\square

2.2. Integración por partes

Recordemos la fórmula de integración por partes de Cálculo:

$$\int_0^1 u(x)v'(x) \, dx = u(1)v(1) - u(0)v(0) - \int_0^1 v(x)u'(x) \, dx,$$

con u y v de clase C^1 en $[0, 1]$.

Por lo tanto si $u, v : [0, 1] \rightarrow \mathbb{R}$ son funciones de clase C^1 y si $v \in C_0^\infty(I)$:

$$\int_0^1 u(x)v'(x) \, dx = - \int_0^1 u'(x)v(x) \, dx \quad \forall v \in C_0^\infty(I).$$

Esto gracias a que $v(0) = v(1) = 0$.

2.3. Derivada débil de funciones en $\mathcal{L}^2(I)$

Sea $u \in \mathcal{L}^2(I)$ no necesariamente diferenciable. Se dice que u tiene una **derivada débil** en $\mathcal{L}^2(I)$ si existe $g \in \mathcal{L}^2(I)$ tal que:

$$\int_0^1 u(x)v'(x) dx = - \int_0^1 g(x)v(x)dx, \quad \forall v \in C_0^\infty(I).$$

En este caso se dice que g es una derivada débil de u .

Se tiene que si u es de clase C^1 , entonces existe la derivada débil de u y además coincide con la derivada tradicional u' . Esto porque la derivada débil (si existe) es única, como se muestra en el siguiente resultado.

Teorema 2.5. (Unicidad de la derivada débil) Sea $u \in \mathcal{L}^2(I)$ tal que u tiene una derivada débil en $\mathcal{L}^2(I)$. Si g_1 y $g_2 \in \mathcal{L}^2(I)$ son derivadas débiles de u , entonces $g_1 = g_2$.

Demostración. Sean $u \in \mathcal{L}^2(I)$, y $g_1, g_2 \in \mathcal{L}^2(I)$ derivadas débiles de u .

$$\begin{aligned} \int_0^1 u(x)v'(x) dx &= - \int_0^1 g_1(x)v(x) dx \quad \forall v \in C_0^\infty(I), \\ \int_0^1 u(x)v'(x) dx &= - \int_0^1 g_2(x)v(x) dx \quad \forall v \in C_0^\infty(I), \\ \Rightarrow \int_0^1 g_1(x)v(x) dx &= \int_0^1 g_2(x)v(x) dx \quad \forall v \in C_0^\infty(I), \end{aligned}$$

y por el Teorema 2.3 tenemos que $g_1 = g_2$. □

Si $u \in \mathcal{L}^2(I)$ es continua en $[0, 1]$ y lineal por trozos, entonces u tiene una derivada débil, la cual es una función constante por trozos. Pero no cualquier función tiene una derivada débil, como se ilustra en el siguiente ejemplo: Si $u \in \mathcal{L}^2(I)$ tiene una discontinuidad de salto, entonces u no tiene derivada débil en $\mathcal{L}^2(I)$. Veamos el siguiente ejemplo:

Sean $a, b \in \mathbb{R}, a \neq b, x_0 \in (0, 1)$.

$$\text{Sea } u(x) = \begin{cases} a & \text{si } 0 \leq x \leq x_0 \\ b & \text{si } x_0 < x \leq 1 \end{cases},$$

Para probar que u no tiene derivada débil en $\mathcal{L}^2(I)$, denotemos primero:

$$u(x_o^+) = \lim_{x \rightarrow x_o^+} u(x) = b,$$

$$u(x_o^-) = \lim_{x \rightarrow x_o^-} u(x) = a.$$

Ahora tomemos $v \in C_0^\infty(I)$ e integremos sobre I .

$$\begin{aligned} \int_0^1 u(x)v'(x) \, dx &= \int_0^{x_0} u(x)v'(x) \, dx + \int_{x_0}^1 u(x)v'(x) \, dx \\ &= v(x_0^-)u(x_0^-) - v(0)u(0) - \int_0^{x_0} u'(x)v(x) \, dx \\ &\quad + v(1)u(1) - v(x_0^+)u(x_0^+) - \int_{x_0}^1 u'(x)v(x) \, dx \\ &= 0 + av(x_0) - 0 - 0 - bv(x_0) + 0 \\ &= (a - b) \cdot v(x_0) \\ \therefore \int_0^1 u(x)v'(x) &= (a - b) \cdot v(x_0), \end{aligned}$$

y deberíamos encontrar una g de tal manera que

$$\int_0^1 u(x)v'(x) = - \int_0^1 g(x)v(x) \, dx = (a - b) \cdot v(x_0), \quad \forall v \in C_0^\infty(I).$$

Pero no existe $g \in \mathcal{L}^2(I)$ con la propiedad anterior. Un múltiplo de la Delta de Dirac $((a - b) \cdot \delta_{x_0})$ satisface la propiedad anterior; de hecho g es un múltiplo constante de la Delta de Dirac. Por lo tanto, $u(x)$ no tiene derivada débil en $\mathcal{L}^2(I)$.

Queda demostrar que

$$\delta_{x_0} \notin \mathcal{L}^2(I). \quad (2.1)$$

Pero antes se muestra un resultado más sencillo que ayudará a demostrar lo anterior.

Sea $f \in \mathcal{L}^2(I)$ con $\|f\|_{\mathcal{L}^2} = c$, por simplicidad en este caso $I = [-1, 1]$. Se tiene que:

$$\int_{-1}^1 |f(x)|^2 \, dx = c^2,$$

sea $\lambda > 1$; se define la función

$$f'_\lambda(x) = f(\lambda x), \quad (2.2)$$

su norma calculada como sigue:

$$\|f'_\lambda\| = \int_{-1}^1 |f(\lambda x)|^2 dx$$

con el cambio de variable $u = \lambda x, du = \lambda dx$ queda lo siguiente:

$$\int_{-\lambda}^{\lambda} |f(u)|^2 \frac{1}{\lambda} du = \int_{-1}^1 |f(u)|^2 \frac{1}{\lambda} du = \frac{1}{\lambda} c^2$$

por lo tanto tenemos que, si definimos ahora la función $f_\lambda(x) = \sqrt{\lambda} f(\lambda x)$, $\|f_\lambda\| = \|f\| = c$.

Demostración. (De (2.1)) Por simplicidad suponemos que $x_0 = 0$, y que el intervalo sigue siendo $I = [-1, 1]$. Para demostrar que delta no pertenece a $\mathcal{L}^2(I)$, suponemos lo contrario, i.e. $\delta_0 \in \mathcal{L}^2(I)$, y se llegará a una contradicción.

Sea entonces la función $f \in C^\infty(I)$, definida como sigue:

$$f(x) = \begin{cases} e^{\frac{-1}{1-x^2}} & \text{si } |x| < 1 \\ 0 & \text{en otro caso} \end{cases}, \quad (2.3)$$

sea k tal que $\|f\| = k$.

Si $\delta_0 \in \mathcal{L}^2(I) \Rightarrow \exists M \in \mathbb{R}^+$ tal que:

$$\int_{-1}^1 \delta_{x_0}^2(x) dx = \|\delta_0\|_{\mathcal{L}^2} \leq M.$$

Ahora tomamos la transformación (2.2) de la función (2.3) con $\lambda = (M+1)^2 e^2 k^2$. Podemos tomar $M > 1$ tal que $\lambda > 1$. En este caso $f_\lambda \in C_0^\infty(I)$, por lo que:

$$\int_{-1}^1 \delta_0 f_\lambda(x) dx = f_\lambda(0).$$

Sabemos por la desigualdad CBS:

$$\left| \int_{-1}^1 \delta_0 f_\lambda(x) dx \right| \leq \|\delta_{x_0}\|_{\mathcal{L}^2} \|f_\lambda\|_{\mathcal{L}^2}$$

$$\begin{aligned}
&\Rightarrow |f_\lambda(0)| \leq \|\delta_0\|_{\mathcal{L}^2} k \\
&\Rightarrow |\sqrt{\lambda} e^{-1}| \leq \|\delta_0\|_{\mathcal{L}^2} k \\
&\Rightarrow (M+1)k e^{-1} e^1 \leq \|\delta_0\|_{\mathcal{L}^2} k \\
&\Rightarrow (M+1) \leq M.
\end{aligned}$$

Por ende se concluye que δ_0 no puede pertenecer a $\mathcal{L}^2(I)$. □

2.4. Derivada débil en $\mathcal{L}^2(\Omega)$

Sea $u \in \mathcal{L}^2(\Omega)$, ($u = u(x_1, x_2)$), se dice que $g \in \mathcal{L}^2(\Omega)$ es una derivada débil de u con respecto a la variable x_i si:

$$\iint_{\Omega} u \frac{\partial v}{\partial x_i} dx dy = - \iint_{\Omega} g v dx dy \quad \forall v \in C_0^\infty(\Omega).$$

Se puede probar de forma similar al caso univariado, que si u tiene derivada débil respecto a x_i , entonces la derivada débil es única.

2.5. Espacios $\mathcal{L}^2(\Omega), H^1(\Omega), H_0^1(\Omega)$

Sea Ω un subconjunto abierto y acotado de \mathbb{R}^2 . Los siguientes son ejemplos de espacios de Hilbert.

$$a) \quad \mathcal{L}^2(\Omega) = \left\{ v : \Omega \rightarrow \mathbb{R} \mid \iint_{\Omega} v^2 dx dy < \infty \right\}.$$

El producto interior y norma que asignamos a este espacio son:

$$\langle u, v \rangle_{\mathcal{L}^2} = \iint_{\Omega} uv dx dy,$$

$$\|v\|_{\mathcal{L}^2}^2 = \iint_{\Omega} v^2 dx dy.$$

$$b) H^1(\Omega) = \left\{ v \in \mathcal{L}^2(\Omega) \mid \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in \mathcal{L}^2(\Omega) \right\}.$$

Las derivadas referidas en este espacio son en el sentido de las derivadas débiles. El producto interior y norma que asignamos son:

$$\langle u, v \rangle_{H^1} = \iint_{\Omega} (uv + \nabla u \cdot \nabla v) \, dx \, dy,$$

$$\|v\|_{H^1}^2 = \iint_{\Omega} (v^2 + \nabla v \cdot \nabla v) \, dx \, dy = \iint_{\Omega} (v^2 + |\nabla v|^2) \, dx \, dy.$$

El espacio H^1 contiene las funciones continuas y lineales por trozos.

$$c) H_0^1(\Omega) = \left\{ v \in H^1(\Omega) \mid \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in \mathcal{L}^2(\Omega), v = 0 \text{ sobre } \Gamma = \partial\Omega \right\}.$$

El producto interior y norma que asignamos a este espacio son los mismos que tenemos para $H^1(\Omega)$.

2.6. Formulación variacional

Un problema variacional es un problema que optimiza sobre funcionales continuos³ definidos sobre un espacio funcional. Resolver una ecuación diferencial del estilo:

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= 0 \quad \text{en } \Gamma = \partial\Omega. \end{aligned} \tag{2.4}$$

Es equivalente a resolver el siguiente problema variacional.

Encontrar $u \in H_0^1(\Omega)$ tal que:

$$a(u, v) = \langle f, v \rangle \quad \forall v \in H_0^1(\Omega), \tag{2.5}$$

donde

$$a(u, v) = \iint_{\Omega} \nabla u \cdot \nabla v \, dx \, dy = \iint_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \, dx \, dy,$$

³Para mayor referencia a análisis funcional y a problemas variacionales, revisar [10] y [5]

y

$$\langle f, v \rangle = \iint_{\Omega} f v \, dx \, dy.$$

Para probar esta equivalencia entre nuestros problemas, enunciaremos el teorema de la divergencia.

Teorema 2.6. (De la Divergencia⁴) Sea $A : \Omega \rightarrow \mathbb{R}^2$ una función con valores vectoriales $A = (A_1, A_2)$ y sea $\eta = (\eta_1, \eta_2)$ el vector unitario perpendicular a la curva Γ (frontera de Ω) que apunta “hacia afuera”. Entonces tenemos:

$$\iint_{\Omega} \operatorname{div}(A) \, dx \, dy = \int_{\Gamma} A \cdot \eta \, ds. \quad (2.6)$$

Donde $\operatorname{div}(A)$ denota la divergencia de la función A y se define como sigue.

$$\operatorname{div}(A) = \frac{\partial A_1}{\partial x_1} + \frac{\partial A_2}{\partial x_2}.$$

Con el teorema de la divergencia deduciremos una fórmula de Green.

Sean $v, w : \Omega \rightarrow \mathbb{R}$, v de clase C^1 , w de clase C^2 , y sean $A_{\alpha}, A_{\beta} : \Omega \rightarrow \mathbb{R}^2$ definidas como sigue: $A_{\alpha} = (vw, 0)$, $A_{\beta} = (0, vw)$. Sabemos que:

$$\begin{aligned} \operatorname{div}(A_{\alpha}) &= \frac{\partial vw}{\partial x_1} + 0 = \frac{\partial v}{\partial x_1} w + \frac{\partial w}{\partial x_1} v \\ \Rightarrow \iint_{\Omega} \frac{\partial v}{\partial x_1} w + \frac{\partial w}{\partial x_1} v \, dx \, dy &= \int_{\Gamma} vw \eta_1 + 0 \eta_2 \, ds, \end{aligned}$$

y análogamente para A_{β} :

$$\iint_{\Omega} \frac{\partial v}{\partial x_2} w + \frac{\partial w}{\partial x_2} v \, dx \, dy = \int_{\Gamma} vw \eta_2 + 0 \eta_1 \, ds.$$

Ahora, si en lugar de w usamos $\frac{\partial w}{\partial x_1}$ y $\frac{\partial w}{\partial x_2}$ respectivamente:

$$\iint_{\Omega} \frac{\partial v}{\partial x_1} \frac{\partial w}{\partial x_1} \, dx \, dy + \iint_{\Omega} v \frac{\partial^2 w}{\partial x_1^2} \, dx \, dy = \int_{\Gamma} v \frac{\partial w}{\partial x_1} \eta_1 \, ds,$$

⁴Para mayor referencia ver [1].

$$\iint_{\Omega} \frac{\partial v}{\partial x_2} \frac{\partial w}{\partial x_2} dx dy + \iint_{\Omega} v \frac{\partial^2 w}{\partial x_2^2} dx dy = \int_{\Gamma} v \frac{\partial w}{\partial x_2} \eta_2 ds.$$

Sumando ambas ecuaciones se obtiene la siguiente fórmula de Green:

$$\iint_{\Omega} \nabla v \cdot \nabla w dx dy = \int_{\Gamma} v \frac{\partial w}{\partial \eta} ds - \iint_{\Omega} v \Delta w dx dy$$

$$\therefore \iint_{\Omega} v \Delta w dx dy = \int_{\Gamma} v \frac{\partial w}{\partial \eta} ds - \iint_{\Omega} \nabla v \cdot \nabla w dx dy, \quad (2.7)$$

$$\frac{\partial w}{\partial \eta} = \frac{\partial w}{\partial x_1} \eta_1 + \frac{\partial w}{\partial x_2} \eta_2,$$

donde η es la derivada de w en dirección η .

Veamos ahora cómo usar la identidad de green (2.7) para dar una formulación variacional a la ecuación (2.4).

Sean $u \in H^1(\Omega)$ solución a la ecuación (2.4), y $v \in H_0^1(\Omega)$. Como u satisface (2.4), $-\Delta u = f$ y además v se anula en Γ , por (2.7) tenemos lo siguiente:

$$\iint_{\Omega} f v dx dy = - \iint_{\Omega} \Delta u v dx dy = \int_{\Gamma} \frac{\partial u}{\partial \eta} v ds - \iint_{\Omega} \nabla v \cdot \nabla u dx dy = a(u, v).$$

Por lo tanto tenemos que si u es solución de 2.4 entonces es solución de 2.5. Falta ver que si es solución de 2.5 entonces también será solución de 2.4.

Tomemos una función $u \in H_0^1(\Omega)$, y por definición de a tenemos que:

$$\iint_{\Omega} f v dx dy = \iint_{\Omega} \nabla u \cdot \nabla v dx dy.$$

Además, gracias a la fórmula de Green y suponiendo que u es de clase C^2 se sigue que:

$$\begin{aligned} \iint_{\Omega} \nabla u \cdot \nabla v dx dy &= \int_{\Gamma} \frac{\partial u}{\partial \eta} v ds - \iint_{\Omega} v \Delta u dx dy, \\ \Rightarrow \iint_{\Omega} f v dx dy &= - \iint_{\Omega} v \Delta u dx dy, \\ \Rightarrow \iint_{\Omega} f v + \Delta u v dx dy &= 0, \end{aligned}$$

$$\begin{aligned}
\Rightarrow \iint_{\Omega} (f + \Delta u) v \, dx \, dy &= 0 \quad \forall v \in H_0^1(\Omega), \\
\Rightarrow f + \Delta u &= 0, \\
\Rightarrow -\Delta u &= f,
\end{aligned}$$

en el sentido de las funciones en $\mathcal{L}^2(\Omega)$. La igualdad anterior es cierta puntualmente si $u \in C^2$ y f es continua.

Estudiaremos algunas características de $a(\cdot, \cdot)$ con el siguiente teorema:

Teorema 2.7. *Sea $v \in H_0^1(\Omega)$, entonces existe una constante $\gamma > 0$ tal que:*

$$\gamma \|v\|_{H_0^1}^2 \leq \iint_{\Omega} |\nabla v|^2 \, dx \, dy = a(v, v), \quad \forall v \in H_0^1(\Omega),$$

Demostración. Para demostrar el teorema, enunciaremos un lema del capítulo 7 en el libro [7].

Lema 2.8. *Existe una constante $C > 0$, tal que*

$$\|v\|_{\mathcal{L}^2}^2 \leq C \iint_{\Omega} |\nabla v|^2 \, dx \, dy, \quad \forall v \in H_0^1(\Omega).$$

Con el lema 2.8, tomando $\gamma = \frac{1}{C+1}$, y tomando en cuenta que:

$$\|v\|_{H_0^1}^2 = \|v\|_{\mathcal{L}^2}^2 + \iint_{\Omega} |\nabla v|^2 \, dx \, dy,$$

tenemos que:

$$\begin{aligned}
\gamma \|v\|_{H_0^1}^2 &= \gamma \left(\|v\|_{\mathcal{L}^2}^2 + \iint_{\Omega} |\nabla v|^2 \, dx \, dy \right) \\
&\leq \gamma \left(C \iint_{\Omega} |\nabla v|^2 \, dx \, dy + \iint_{\Omega} |\nabla v|^2 \, dx \, dy \right) \\
&= \iint_{\Omega} |\nabla v|^2 \, dx \, dy = a(v, v).
\end{aligned}$$

□

Para ver más a detalle la formulación variacional ver [3] y [9]

2.7. Ejemplos

A continuación presentaremos algunos ejemplos de la transformación de un problema de ecuaciones diferenciales parciales a uno de forma variacional.

Ejemplo 2.1.

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= g \quad \text{en } \Gamma, \end{aligned}$$

Tomamos de una función $v \in H_0^1(\Omega)$, y la multiplicamos por la primera ecuación e integramos sobre Ω :

$$\begin{aligned} \iint_{\Omega} -v \Delta u \, dx \, dy &= \iint_{\Omega} f v \, dx \, dy \\ \Rightarrow \iint_{\Omega} f v \, dx \, dy &= - \int_{\Gamma} \frac{\partial g}{\partial \eta} v \, ds + \iint_{\Omega} \nabla v \cdot \nabla u \, dx \, dy. \end{aligned}$$

Dado que $v = 0$ sobre Γ , problema variacional es encontrar $u \in H^1(\Omega)$ con $u|_{\Gamma} = g$, tal que:

$$\iint_{\Omega} \nabla u \cdot \nabla v \, dx \, dy = \iint_{\Omega} f v \, dx \, dy, \quad \text{para toda } v \in H_0^1(\Omega).$$

Ejemplo 2.2.

$$\begin{aligned} -\Delta u &= f \quad \text{en } \Omega \subset \mathbb{R}^2, \\ u &= g_1 \quad \text{en } \Gamma_1, \text{long}(\Gamma_1) > 0, \\ \frac{\partial u}{\partial \eta} &= g_2 \quad \text{en } \Gamma_2, \Gamma = \Gamma_1 \cup \Gamma_2. \end{aligned}$$

Para pasar este problema a forma variacional, tomamos de una función $v \in H^1(\Omega)$, y la multiplicamos por la primera ecuación e integramos sobre Ω :

$$\begin{aligned} \iint_{\Omega} -v \Delta u \, dx \, dy &= \iint_{\Omega} f v \, dx \, dy = - \int_{\Gamma} \frac{\partial u}{\partial \eta} v \, ds + \iint_{\Omega} \nabla v \cdot \nabla u \, dx \, dy. \\ \Rightarrow \iint_{\Omega} f v \, dx \, dy &= - \int_{\Gamma_1} \frac{\partial g_1}{\partial \eta} v \, ds - \int_{\Gamma_2} g_2 v \, ds + \iint_{\Omega} \nabla v \cdot \nabla u \, dx \, dy. \end{aligned}$$

Por lo tanto el problema variacional es encontrar

$$u \in V = \{w | w \in H^1, w|_{\Gamma_1} = g_1\}$$

de tal manera que

$$\iint_{\Omega} \nabla u \cdot \nabla v \, dx \, dy = \iint_{\Omega} f v \, dx \, dy + \int_{\Gamma_2} g_2 v \, ds, \quad \forall v \in H^1(\Omega) \text{ con } v = 0 \text{ en } \Gamma_1,$$

para cualquier $v \in H^1(\Omega)$ y tal que $v = 0$ en Γ_1 . Como podemos ver en ambos problemas, el lado derecho depende únicamente de la función v y de otras funciones fijas. mientras que el lado izquierdo, tenemos un funcional que depende de u y de v . Por lo que un cambio en las condiciones de frontera no afecta en gran medida a nuestro algoritmo.

CAPÍTULO 3

Método de elementos finitos

En este capítulo revisaremos las partes principales para la implementación numérica del MEF: triangulación del dominio, clasificación de nodos, funciones nodales, manejo de información, el sistema de ecuaciones a resolver; así como algunos otros conceptos y procedimientos.

3.1. Representación de funciones

La solución que buscamos está en un espacio funcional de dimensión infinita, por lo que definimos un subespacio de dimensión finita que dé una buena aproximación de la función que se busca u , a este procedimiento se le llama discretización de la solución. Se podrá representar a u , como una suma de funciones que serán base del subespacio:

$$u(x) \approx \hat{u}(x) = \sum_{i=1}^n \alpha_i \phi_i(x)$$

Cabe mencionar que la elección de las funciones base es muy importante para el desarrollo, implementación y precisión del método.

3.2. Nodos

$\Omega \subset \mathbb{R}^2$, es un conjunto cualquiera. Como ya mencionamos, no podemos representar el espacio completo en una computadora. Lo que si podemos hacer es elegir un conjunto finito de elementos del espacio, con la finalidad de obtener una buena aproximación del mismo. Por simplicidad y para este trabajo, suponemos que Ω es un polígono.

Definición 3.1. Un **nodo** es un elemento cualquiera a del conjunto Ω . Llamamos a un nodo **interior** si $a \in \Omega \setminus \Gamma$. Por otro lado se le llamará **nodo frontera** si $a \in \Gamma$.

Supongamos ahora que tenemos un conjunto de nodos, digamos $A = \{a_i\}_{i=1}^n$; a un triángulo t lo podemos representar como un subconjunto de tres elementos $t = \{a_1, a_2, a_3\} \subset A$.

Definición 3.2. Llamaremos **triangulación** a un conjunto de triángulos $\{t_i\}_{i=1}^m$ de tal manera que el área cubierta por ellos sea Ω (i.e. $\bigcup_{i=1}^m t_i = \Omega$). Además debemos pedir que entre dos triángulos distintos cualesquiera, su intersección $t_i \cap t_j$ sea una de las siguientes:

- \emptyset (conjunto vacío).
- Un lado en común de ambos triángulos.
- Un vértice en común.

A continuación un ejemplo para clarificar este concepto.

Ejemplo 3.1.

Supongamos que el dominio Ω es el de la figura 3.1. Si se eligen los nodos como en la figura 3.2, una triangulación válida posible se muestra en la figura 3.3.

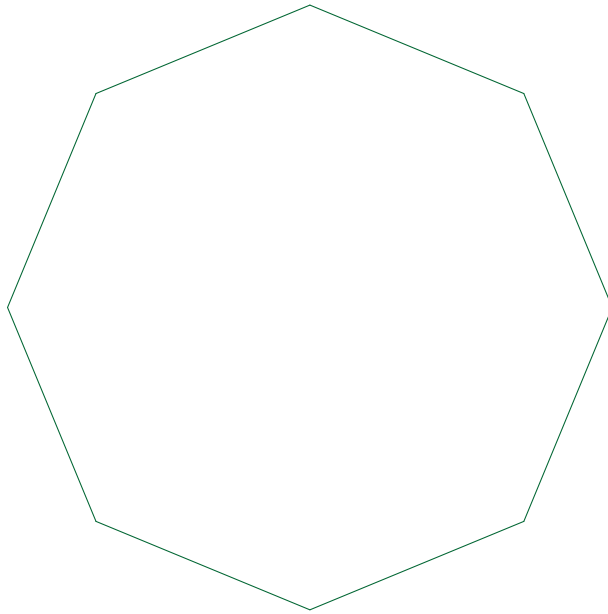


Figura 3.1: Ω

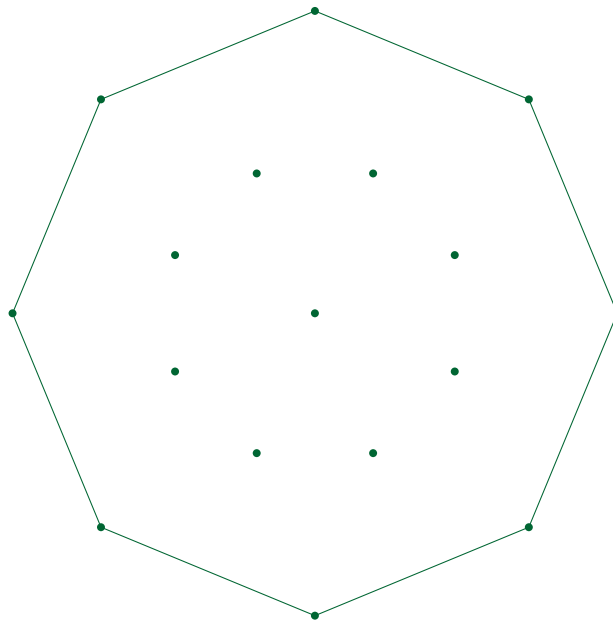


Figura 3.2: Nodos.

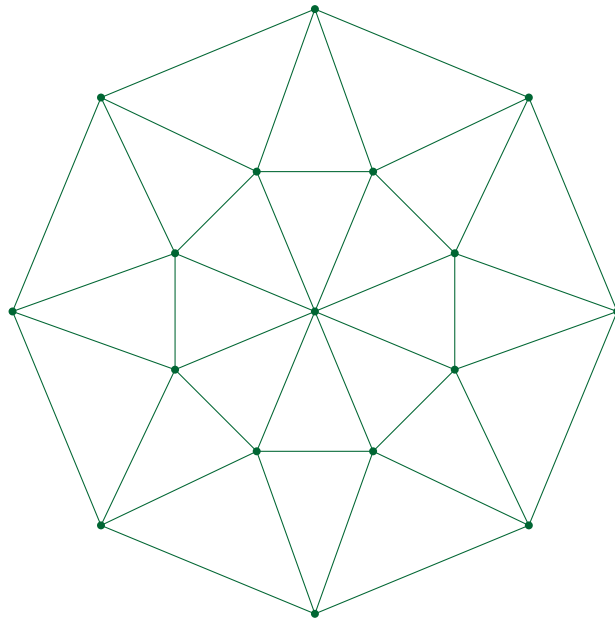


Figura 3.3: Triangulación válida.

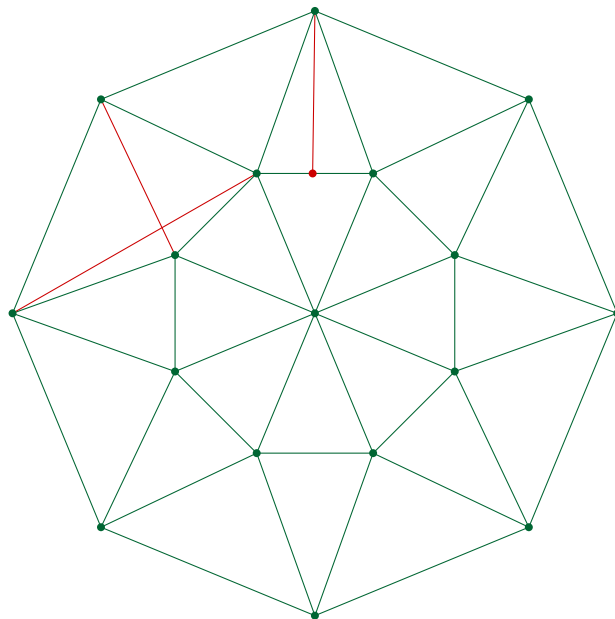


Figura 3.4: Triangulación inválida.

En la figura 3.4 se ilustra una triangulación inválida. Al remover un lado y agregar otro se obtiene una triangulación válida en la figura 3.5.

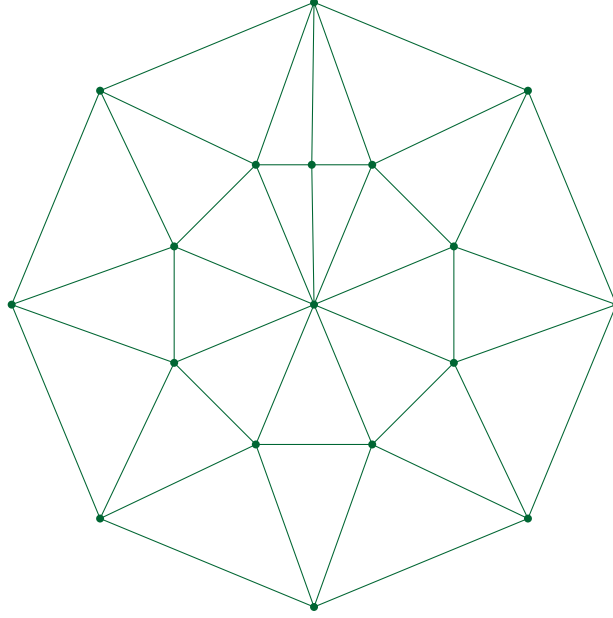


Figura 3.5: Triangulación Válida.

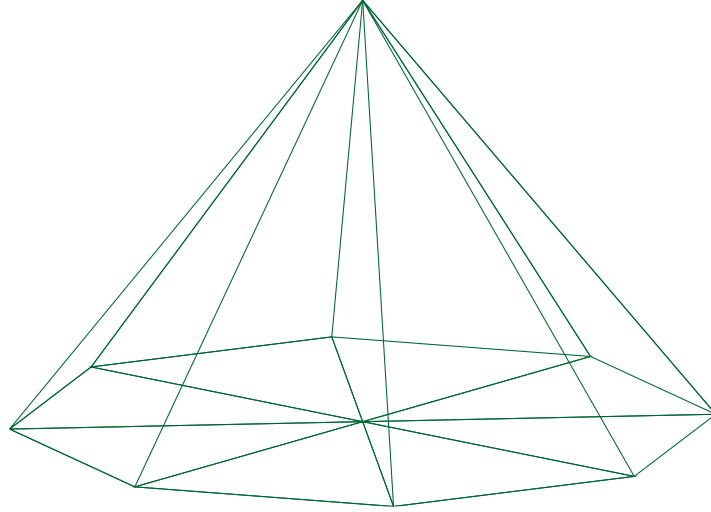
Existen varios algoritmos para generar triangulaciones. En este trabajo usamos el algoritmo de Delaunay¹, pero con una modificación, misma que mencionaremos más adelante. Por el momento basta con la definición para poder introducir las funciones $\{\phi_1, \dots, \phi_n\}$.

3.3. Funciones nodales

Dado el i -ésimo nodo a_i , la función nodal asociada $\phi_i : \Omega \rightarrow \mathbb{R}$, es una función lineal en cada triángulo y está determinada por

$$\phi_i(a_j) = \delta_{i,j} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}.$$

¹Para mayor detalle de la teoría asociada con el algoritmo de triangulación Delaunay, ver [2]

Figura 3.6: Ejemplo de una función ϕ

Por lo tanto el soporte de ϕ_i está determinado por los triángulos que tienen a a_i como vértice.

En resumen, para cada nodo, tenemos una función nodal que depende de los triángulos a los que pertenezca el nodo; en cada triángulo la gráfica función nodal será un plano distinto. Por otro lado, cada triángulo deberá tener asociados tres planos de tres distintas funciones nodales. Una manera muy visual de imaginar las funciones nodales es verlas cada una como una pirámide, centrada en el nodo de altura uno y que cae a cero en los nodos vecinos.

Al espacio generado por las funciones nodales ϕ_i restringidas al triángulo t le llamaremos $P_1(t)$, que en este caso son planos.

Notar que dos funciones nodales distintas: $\phi_i, \phi_j, i \neq j$, tienen la propiedad de que la intersección de sus soportes o bien es vacío o está formado por uno o dos triángulos.

3.4. Triangulación

Para obtener la triangulación necesaria, usamos el algoritmo *Delaunay* [2]. El algoritmo recibe como entrada los dos vectores:

$$x = (x_1, x_2, \dots, x_N)$$

$$y = (y_1, y_2, \dots, y_N)$$

con N abscisas y N ordenadas, donde el i -ésimo nodo es la pareja (x_i, y_i) . La función da como resultado ternas de números enteros, cada número correspondiente a un número de nodo. El número entero simboliza la posición que ocupa el nodo en el vector que damos. La salida que se obtiene es de la forma:

Triángulo	a_i	a_j	a_k
t_1	2	1	3
t_2	4	2	1
t_3	1	5	2
\vdots	\vdots	\vdots	\vdots

Una particularidad importante a notar del algoritmo Delaunay, es que supone que nuestra región es convexa, y en caso de que no lo sea, podría dar triángulos fuera del dominio, por ejemplo si tenemos el siguiente dominio:

$$\Omega = \left\{ (x, y) \mid \frac{1}{4} \leq x^2 + y^2 \leq 1 \right\}$$

que representa un anillo acotado por los círculos concéntricos de radio $\frac{1}{2}$ y 1 respectivamente. El algoritmo Delaunay, nos dará triángulos dentro del círculo más pequeño, y por ende fuera del dominio como se ilustra en la figura 3.7.

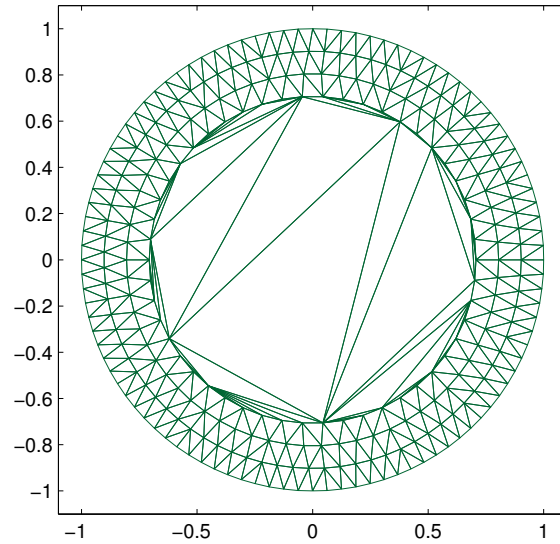


Figura 3.7: Triángulos fuera del dominio.

Para resolver este problema, simplemente eliminamos los triángulos que están fuera del dominio. ¿Cómo determinamos si un triángulo está fuera del dominio? Calculamos el centroide del triángulo. Dado un triángulo $t_1 = (a_i, a_j, a_k)$, calculamos el punto $\frac{a_i + a_j + a_k}{3}$ y vemos si queda fuera del dominio, en cuyo caso eliminamos el triángulo.

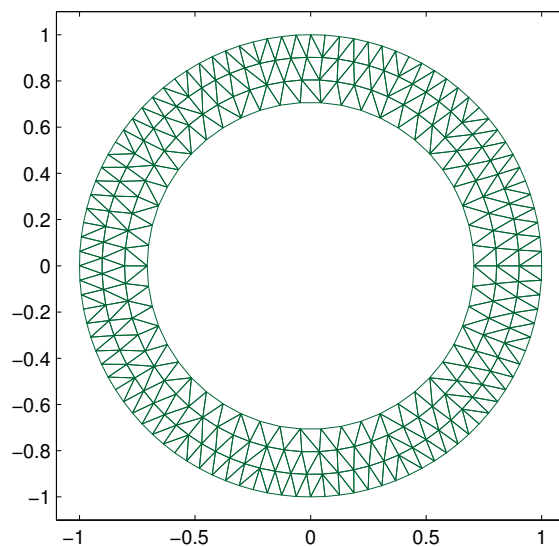
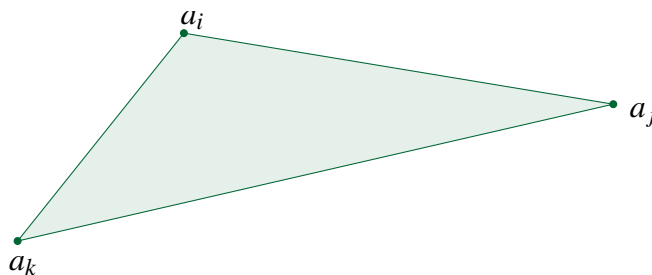


Figura 3.8: Corrección de la triangulación.

Para llevar a cabo los cálculos subsecuentes requerimos tener información acerca de los triángulos y los nodos. Para cada nodo, nos interesa saber quienes son sus “vecinos”. Dos nodos distintos a_i, a_j son vecinos si a_i y a_j son nodos pertenecientes a un mismo triángulo. Por ejemplo:

Figura 3.9: Triángulo t_1 .

Si tenemos el triángulo $t_1 = (a_i, a_j, a_k)$ los vecinos de a_i en el triángulo t_1 serían a_j y a_k . Algo que podría pasar es que en otro triángulo vuelvan a ser vecinos

una pareja de nodos $t_2 = (a_i, a_k, a_l)$. En este caso a_i y a_k vuelven a ser vecinos, pero no es el mismo triángulo. Es importante en la implementación computacional del MEF guardar para cada nodo la información de sus nodos vecinos y de los triángulos a los que pertenecen.

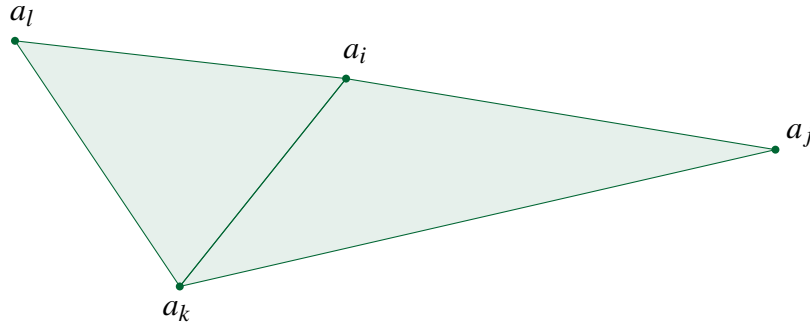


Figura 3.10: Triángulos t_1 y t_2 .

Una manera de obtener dicha información es recorrer todos los nodos y encontrar a qué triángulos pertenece cada uno. Esto resulta ser muy ineficiente ya que se visitan todos los triángulos tres veces, una por cada nodo. Una manera más eficiente de encontrar los triángulos a los que pertenece un nodo dado consiste en recorrer la matriz de triángulos que nos da el algoritmo de Delaunay y de guardar la información para cada nodo en una otra matriz; se guarda el nodo vecino y el triángulo al que pertenece en común. La matriz que se guarda es similar a la que se presenta a continuación:

Nodo	Vecino 1	Triángulo 1	Vecino 2	Triángulo 2	...
a_1	a_j	t_1	a_k	t_2	...
a_2	a_l	t_3	a_m	t_4	...
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots

3.5. Construcción del sistema de ecuaciones

Hasta ahora sólo hemos mostrado las equivalencias entre los problemas variacionales y las ecuaciones diferenciales; no hemos expuesto como se resolverán, ni

el problema que le plantearemos a la máquina para que resuelva. En esta parte hablaremos precisamente de ese paso, y además de una dificultad relacionada con el número de condición de una matriz. Recordemos el problema variacional que queremos resolver:

Encontrar $u \in H_0^1(\Omega)$, de tal manera que:

$$a(u, v) = \iint_{\Omega} \nabla u \cdot \nabla v \, dx \, dy = \iint_{\Omega} f v \, dx \, dy, \forall v \in H_0^1.$$

Sean $\{x_i\}_{i=1}^n$ nodos interiores en Ω , y $\{\phi_i\}_{i=1}^n$, funciones nodales como las descritas en 3.3. Sea V_h el espacio generado por $\{\phi_i\}_{i=1}^n$. El problema que vamos a resolver numéricamente es el siguiente:

Encontrar $\hat{u} \in V_h \subset H_0^1(\Omega)$, de tal manera que:

$$\iint_{\Omega} \nabla \hat{u} \cdot \nabla v \, dx \, dy = \iint_{\Omega} f v \, dx \, dy, \forall v \in V_h$$

Por lo cual basta encontrar los coeficientes $\{\alpha_i\}_{i=1}^n$ de tal manera que:

$$\begin{aligned} \iint_{\Omega} \nabla \left(\sum_{i=1}^n \alpha_i \phi_i \right) \cdot \nabla \phi_j \, dx \, dy &= \iint_{\Omega} f \phi_j \, dx \, dy, \text{ para } j = 1 \dots n \\ \Rightarrow \sum_{i=1}^n \alpha_i \iint_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \, dy &= \iint_{\Omega} f \phi_j \, dx \, dy, \text{ para } j = 1 \dots n \\ \Rightarrow \sum_{i=1}^n \alpha_i a(\phi_i, \phi_j) &= \iint_{\Omega} f \phi_j \, dx \, dy, \text{ para } j = 1 \dots n \end{aligned}$$

Podemos escribir la igualdad anterior como un sistema lineal de ecuaciones algebraicas:

$$A \xi = b, \tag{3.1}$$

donde:

$$A_{i,j} = a(\phi_i, \phi_j), \xi_i = \alpha_i \text{ y } b_i = \iint_{\Omega} f \phi_i \, dx \, dy.$$

Para calcular las entradas de la matriz A del sistema se tiene lo siguiente:

$$a(\phi_i, \phi_j) = \iint_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \, dy = \sum_{t \in T_i \cap T_j} \iint_{T_t} \nabla \phi_i \cdot \nabla \phi_j \, dx \, dy,$$

donde T_i es el conjunto de triángulos a los que el nodo x_i pertenece, por lo tanto la matriz A es rala. Más adelante se proporciona una manera de calcular la matriz A del sistema.

3.6. Manejo de datos

Como ya mencionamos, se requiere resolver un sistema de ecuaciones:

$$A\xi = b. \quad (3.2)$$

La matriz A puede ser muy grande y tomar mucho espacio guardarla. Por ejemplo, para un cuadrado con 10×10 nodos necesitamos guardar una matriz de 100×100 . Para un cuadrado con 100×100 nodos, se necesita una matriz de $10^4 \times 10^4$ lo cual requeriría un total de 10^8 entradas. Afortunadamente la matriz A es rara, por lo que no es necesario almacenar cada entrada.

Al analizar la estructura de la matriz A notamos en primer lugar que la matriz $A_{i,j} = a(\phi_i, \phi_j)$ es simétrica:

$$a(\phi_i, \phi_j) = \iint_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \, dy = \iint_{\Omega} \nabla \phi_j \cdot \nabla \phi_i \, dx \, dy = a(\phi_j, \phi_i).$$

En segundo lugar, por el soporte compacto de las funciones nodales, podemos ver que gran parte de las entradas de la matriz A son ceros. Para los dominios y triangulaciones probadas, hubo a lo más 15 vecinos en algún nodo. Por lo que en lugar de guardar toda la matriz, usamos una estructura de matriz rara; es decir, en lugar de guardarla completa, sólo guardamos las entradas distintas de cero, junto con indicadores del lugar que ocupan. Por ejemplo, si tenemos la siguiente matriz:

$$\begin{pmatrix} 0 & 0 & 1 \\ 3 & 0 & 0 \\ 0 & 5 & 0 \end{pmatrix}$$

Entonces sólo guardaríamos la información siguiente:

$$\begin{array}{ccc} a_{i,j} & i & j \\ 1 & 1 & 3 \\ 3 & 2 & 1 \\ 5 & 3 & 2 \end{array}.$$

En casos más grandes, y con más ceros dentro de la matriz (como es el caso de la matriz que estamos resolviendo) el ahorro de memoria ocupada para almacenar la matriz es considerable.

Con la estructura rala se pueden programar la mayoría de los métodos iterativos para aproximar la solución del sistema (3.2) en esta tesis usaremos el método del Gradiente Conjugado el cual describiremos más adelante.

3.7. Número de condición de la matriz A .

Es de nuestro interés conocer el número de condición $\kappa(A)$ de la matriz A , ya que de ello dependerá en gran medida la eficiencia del algoritmo. Si la matriz A del sistema (3.2) tiene un número de condición grande, al resolver numéricamente el sistema se puede producir pérdida de precisión. Además, un número de condición grande (y dependiendo de la distribución de los valores propios de A) puede producir que el método del gradiente conjugado requiera más iteraciones al aproximar la solución de (3.2) con una tolerancia dada al error de aproximación.

El número de condición dependerá de los triángulos que tomemos en el dominio. A continuación algunas definiciones que nos ayudarán a estimar el tamaño del número de condición de la matriz.

Supongamos que tenemos una triangulación $T = \{t_i\}$. Para cada triángulo t , llamaremos ρ_t al diámetro del círculo inscrito en t , por otro lado, llamaremos h_t , al lado más largo del triángulo t . Además sea $h = \max_{t \in T} \{h_t\}$. (Ver Figura 3.7)

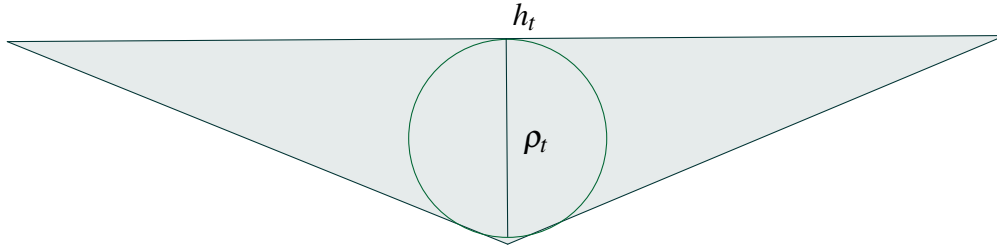


Figura 3.11: ρ_t y h_t .

Con estos parámetros, podemos hablar de familias de triangulaciones, $\{T_h\}$ es el conjunto de triangulaciones de tal manera que el lado más grande sea h . Suponemos que las triangulaciones T_h cumplen con lo siguiente: existen constantes

positivas β_1, β_2 , independientes de h , de tal manera que para todo triángulo $t \in T_h$, se cumpla que:

$$h_t \geq \beta_1 h, \quad (3.3)$$

$$\frac{\rho_t}{h_t} \geq \beta_2, \quad (3.4)$$

si cumple con las ecuaciones, se dice que la triangulación T_h , es cuasi-uniforme. Demostraremos que $\kappa(A) = O(h^{-2})$ para triangulaciones cuasi-uniformes. Para ello enunciaremos un lema:

Lema 3.1. ² *Existen constantes c y C , que solo dependen de β_i , tales que*

$$ch^2|\alpha|^2 \leq \|v\|^2 \leq Ch^2|\alpha|^2. \quad (3.5)$$

$$\iint_{\Omega} |\nabla v|^2 dx dy \leq Ch^{-2} \|v\|^2, \quad (3.6)$$

$$\forall v \in V_h, \alpha \in \mathbb{R}^n, v = \sum_{i=1}^n \alpha_i \phi_i$$

Donde $\|v\| = \|v\|_{\mathcal{L}^2(\Omega)}$ y $|\alpha|$ es la norma 2 de \mathbb{R}^n .

Para estimar el número de condición de la matriz con el lema 3.1, notemos que la matriz A es simétrica y positiva definida, por ello tenemos que

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \frac{\text{Valor propio más grande de } A}{\text{Valor propio más chico de } A},$$

gracias a ello, lo que se debe acotar son los valores propios de la matriz.

Notemos que si

$$v = \sum_{i=1}^n \alpha_i \phi_i,$$

entonces $a(v, v) = \alpha \cdot A\alpha$, donde

$$A_{i,j} = a(\phi_i, \phi_j).$$

Ahora por 3.5 y 3.6, se obtiene:

$$\frac{\alpha \cdot A\alpha}{|\alpha|^2} = \frac{a(v, v)}{|\alpha|^2} \leq Ch^{-2} \frac{\|v\|^2}{|\alpha|^2} \leq C^2 \quad \forall \alpha \in \mathbb{R}^n. \quad (3.7)$$

²Para una prueba del lema, 3.1 ver el Apéndice C.

Dado que:

$$\iint_{\Omega} v^2 + |\nabla v|^2 \, dx \, dy = \|v\|_{H^1}^2 \geq \|v\|^2 = \iint_{\Omega} v^2 \, dx \, dy,$$

y por el teorema 2.7, tenemos:

$$\frac{\alpha \cdot A\alpha}{|\alpha|^2} = \frac{a(v, v)}{|\alpha|^2} \geq \frac{\|v\|_{H^1}^2}{|\alpha|^2} \geq \gamma \frac{\|v\|^2}{|\alpha|^2} \geq C\gamma h^2 \quad \forall \alpha \in \mathbb{R}^n \setminus \{0\}. \quad (3.8)$$

Juntando los resultados 3.7 y 3.8, resulta:

$$C\gamma h^2 \leq \frac{\alpha \cdot A\alpha}{|\alpha|^2} \leq C^2.$$

Gracias a esto sabemos que los valores propios están acotados:

$$\lambda_{\min}(A) \geq C\gamma h^2,$$

$$\lambda_{\max}(A) \leq C^2.$$

Esto debido a que:

$$\lambda_{\max}(A) = \max_{\alpha \in \mathbb{R}^n \setminus \{0\}} \frac{\alpha \cdot A\alpha}{|\alpha|^2},$$

$$\lambda_{\min}(A) = \min_{\alpha \in \mathbb{R}^n \setminus \{0\}} \frac{\alpha \cdot A\alpha}{|\alpha|^2}.$$

Por lo tanto

$$\kappa(A) \leq C' h^{-2}.$$

3.8. Cuadraturas y cálculo de integrales.

Para obtener el lado derecho de la ecuación (3.1) debemos calcular las siguientes integrales:

$$\iint_{\Omega} f \phi_i \, dx \, dy \quad \text{para } i = 1 \dots n. \quad (3.9)$$

Con la elección de funciones nodales se tiene la ventaja de que el integrando vale cero en gran parte del dominio. La región donde no es cero está partido en triángulos; hay una forma sencilla de escribir la función ϕ_i en cada triángulo. Por otro lado necesitamos una manera de calcular (3.9) numéricamente.

Tomando en cuenta la notación de (3.3), debemos notar que (3.9) se puede escribir a su vez como:

$$\sum_{t \in T_i} \iint_t f \phi_i \, dx \, dy \text{ para } i = 1 \dots n. \quad (3.10)$$

Por lo tanto, el cálculo de (3.9) se reduce a calcular integrales en varios triángulos. El valor de la integral doble en cada triángulo t será aproximado por una cuadratura que requiere 7 evaluaciones del integrando. A continuación describiremos esta cuadratura (ver [7]).

Tomemos un triángulo $t = (a_1, a_2, a_3)$, llamaremos $b_1 = \frac{1}{2}(a_1 + a_2)$, $b_2 = \frac{1}{2}(a_2 + a_3)$, $b_3 = \frac{1}{2}(a_1 + a_3)$ los puntos medios entre los vértices del triángulo, y a su vez calculamos el centroide del triángulo: $a' = \frac{1}{3}(a_1 + a_2 + a_3)$, y a_t será el área del triángulo. Ahora sea g una función cualquiera integrable. La fórmula de la cuadratura que usaremos está dada por:

$$\int_t g \, dx \, dy \sim a_t \left[\sum_{i=1}^3 \left(\frac{g(a_i)}{20} + \frac{2g(b_i)}{15} \right) + \frac{9g(a')}{20} \right], \quad (3.11)$$

La fórmula de cuadratura (3.11) es exacta para polinomios de grado a lo más 3 (Ver [7]).

Hay otros métodos de integración, por ejemplo la regla de Simpson para integrar, el problema con este método en particular dentro del contexto de los elementos finitos es que se requiere calcular la integral sobre un triángulo, y la regla de Simpson es muy buena cuando se tienen dominios rectangulares, por lo que en este caso particular nos conviene más usar las cuadraturas.

3.9. Resolución del sistema de ecuaciones: gradiente conjugado

Para resolver el sistema de ecuaciones requerido, usaremos el método del gradiente conjugado. El método del gradiente conjugado, resuelve de manera general y numérica un sistema de ecuaciones lineales:

$$Ax = b, \quad (3.12)$$

donde la matriz A tiene que ser simétrica y positiva definida de tamaño n ; x, b vectores en \mathbb{R}^n . (Ver anexo B.)

3.10. Error de aproximación

Se puede probar que el error de aproximación a la función u tiene una cota, que de hecho es la siguiente(Ver [7]):

$$\|u - u_h\|_{\mathcal{L}^2(\Omega)} \leq Ch^2 \|u\|_{H^2(\Omega)},$$

para triangulaciones cuasi uniformes T_h y u_h la solución aproximada del MEF usando funciones nodales continuas y lineales por trozos.

Este error dice que entre más fina sea la partición que tomemos de Ω , mejor aproximación tendremos a u . Como ya demostramos arriba, el problema con hacer las particiones tan pequeñas es que el número de condición de la matriz usada crece, y se complica resolver el sistema lineal que necesitamos para obtener la solución. En respuesta a esto existen métodos de preconditionamiento³ de la matriz A . El preconditionamiento es un método para encontrar o aproximar la solución x de (3.12) resolviendo un sistema con una matriz cuyo número de condición sea menor que el de A .

Otra manera de resolver el sistema de ecuaciones es por medio de la factorización Cholesky, que a diferencia del método del gradiente conjugado, no es sensible al número de condición de la matriz, aunque por otro lado, puede ser un poco más lento. Ambos son usados para la resolución del sistema de ecuaciones en el MEF.

³Para mayor profundidad sobre el tema de preconditionamiento de la matriz, ver [7] y [8]

CAPÍTULO 4

Dimensión 1

En este capítulo implementaremos el MEF para problemas en dimensión 1, incluyendo una ecuación no lineal. Un ejemplo sencillo de implementación del método es el siguiente. Supongamos que tenemos la siguiente ecuación:

$$-u''(x) = f(x), \quad \forall x \in (0, 1) \quad (4.1)$$

$$u(0) = u(1) = 0. \quad (4.2)$$

Al tomar el intervalo $[0, 1]$ para simplificar los cálculos pero es sencillo extender el resultado a un intervalo general, de la forma $[a, b]$.

Tomemos una partición del intervalo $[0, 1]$ por simplicidad será una partición uniforme:

$$0 = x_0 < x_1 < \dots < x_{n+1} = 1,$$

tenemos que

$$\forall i \in \{1, \dots, n\}, x_i - x_{i-1} = h,$$

con h constante.

Sea V el espacio de funciones continuas y lineales a pedazos en el intervalo $[0, 1]$ y V_h el conjunto de funciones v que cumplen con lo siguiente:

- Son continuas
- En cada intervalo $\{I_i\}_{i=1}^n$ son lineales.
- $v(0) = v(1) = 0$.

Las funciones nodales ϕ_i , en este caso son funciones que pertenecen a V_h y que además cumplen con:

$$\phi_i(x_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}.$$

Para poder ilustrar mejor el concepto de las funciones nodales en dimensión 1, ver la gráfica de la figura 4.1.

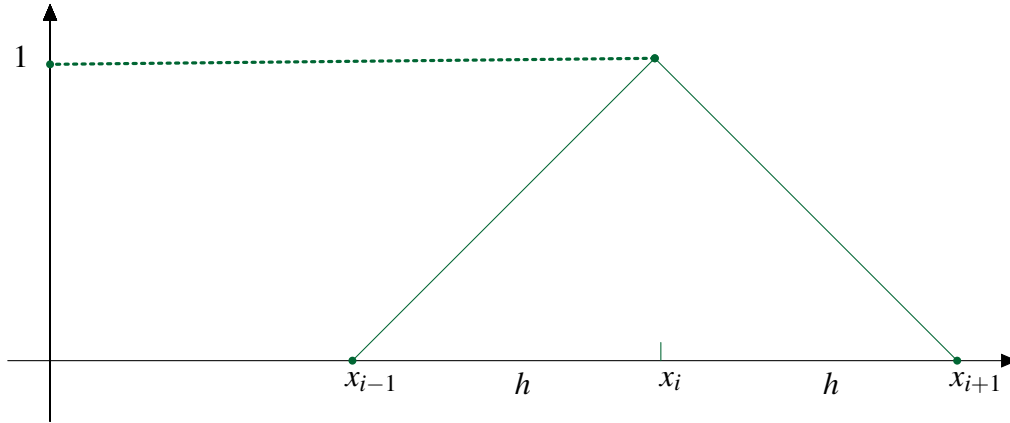


Figura 4.1: ϕ_i .

Para resolver la ecuación transformamos el problema (4.2) en uno variacional discretizado. El problema en forma variacional es encontrar $\hat{u} \in V_h$, de tal forma que:

$$\int_0^1 \hat{u}'(x) v'(x) \, dx = \int_0^1 f(x) v(x) \, dx, \forall v \in V_h,$$

que es equivalente a encontrar los coeficientes ξ_i de tal manera que:

$$\int_0^1 \sum_{i=1}^n \xi_i \phi_i'(x) \phi_j'(x) \, dx = \int_0^1 f(x) \phi_j(x) \, dx, \text{ para } j = 1 \dots n$$

$$\Rightarrow \sum_{i=1}^n \xi_i \int_0^1 \phi'_i(x) \phi'_j(x) dx = \int_0^1 f(x) \phi_j(x) dx, \text{ para } j = 1 \dots n.$$

Ahora estudiemos el término $\int_0^1 \phi'_i(x) \phi'_j(x) dx$. Podemos ver en la figura 4.1 que las funciones ϕ_i se anulan en gran parte del dominio, por lo mismo, la integral es sencilla de calcular:

$$\int_0^1 \phi'_i(x) \phi'_j(x) dx = 0 \text{ si } |i - j| > 1,$$

por otro lado tenemos tres casos distintos, $j = i - 1, 1, i + 1$:

$$\blacksquare j = i - 1$$

$$\begin{aligned} \int_0^1 \phi'_{i-1}(x) \phi'_i(x) dx &= \int_{x_{i-1}}^{x_i} \phi'_{i-1}(x) \phi'_i(x) dx \\ &= \int_{x_{i-1}}^{x_i} -\frac{1}{h} \frac{1}{h} = -\frac{(x_i - x_{i-1})}{h^2} dx = -\frac{1}{h}. \end{aligned}$$

$$\blacksquare j = i$$

$$\begin{aligned} \int_0^1 \phi'_i(x) \phi'_i(x) dx &= \int_{x_{i-1}}^{x_{i+1}} \phi_i'^2(x) dx \\ &= \int_{x_{i-1}}^{x_i} \phi_j'^2(x) dx + \int_{x_i}^{x_{i+1}} \phi_j'^2(x) dx \\ &= \int_{x_{i-1}}^{x_i} \frac{1}{h^2} dx + \int_{x_i}^{x_{i+1}} \frac{1}{h^2} dx \\ &= \frac{(x_i - x_{i-1})}{h^2} + \frac{(x_{i+1} - x_i)}{h^2} = \frac{2h}{h^2} = \frac{2}{h}. \end{aligned}$$

$$\blacksquare j = i + 1$$

$$\begin{aligned} \int_0^1 \phi'_{i+1}(x) \phi'_i(x) dx &= \int_{x_i}^{x_{i+1}} \phi'_{i+1}(x) \phi'_i(x) dx \\ &= \int_{x_i}^{x_{i+1}} -\frac{1}{h} \frac{1}{h} = -\frac{(x_{i+1} - x_i)}{h^2} dx = -\frac{1}{h}. \end{aligned}$$

Por lo que la matriz A queda como sigue:

$$A = \frac{1}{h} \begin{bmatrix} 2 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

El sistema de ecuaciones a resolver:

$$A\xi = b,$$

donde ξ_i es el vector de coeficientes a encontrar y

$$b_i = \int_0^1 f(x)\phi_i(x) dx$$

es el lado derecho del sistema. Ejemplificaremos el funcionamiento del algoritmo con un problema muy sencillo. Considerar el problema con valores en la frontera

$$\begin{aligned} -u''(x) &= \pi^2 \text{sen}(\pi x) \\ u(0) &= u(1) = 0 \end{aligned}$$

En este caso se puede calcular la solución exacta,

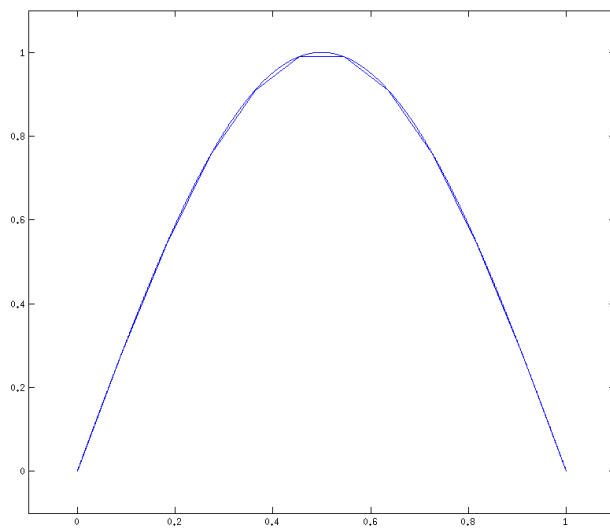
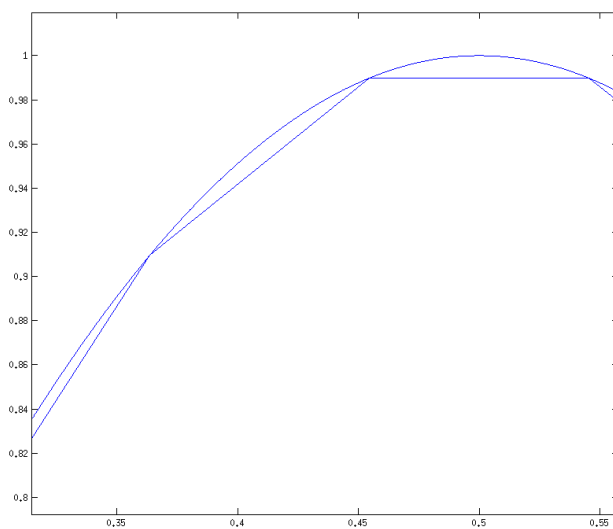
$$u(x) = \text{sen}(\pi x),$$

la cual usaremos para analizar el grado de precisión de la solución aproximada. Tomando 11 nodos se obtiene el siguiente sistema de ecuaciones:

$$10 \begin{bmatrix} 2 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \xi = b.$$

$$b_i = \int_0^1 \text{sen}(x)\phi_i(x) dx$$

La solución la podemos apreciar en la figura 4.2, y en el acercamiento de la figura 4.3 podemos apreciar el error asociado a la aproximación lineal de la función $\text{sen}(x)$.

Figura 4.2: $u(x)$ y $\hat{u}(x)$.Figura 4.3: $u(x)$ y $\hat{u}(x)$.

4.1. Ecuaciones diferenciales no lineales.

Para ilustrar un poco más la flexibilidad del MEF, desarrollaremos para el caso unidimensional una extensión del mismo para resolver ecuaciones diferenciales no lineales, en particular ecuaciones de la forma:

$$-u''(x) + g(u(x)) = f(x) \quad (4.3)$$

$$u(a) = 0, u(b) = 0 \quad (4.4)$$

De nuevo, para resolver (4.4), aproximaremos u , con una base de funciones como la que se ilustra en la figura 4.1, de tal manera que:

$$u(x) \approx \hat{u}(x) = \sum_{i=1}^n \alpha_i \phi_i(x).$$

Para aproximar la ecuación seguimos el mismo procedimiento que hemos seguido hasta el momento para discretizar las ecuaciones diferenciales: sea $\phi_j(x)$ un elemento de la base del espacio V_h . Tomamos la ecuación 4.4 y la multiplicamos por ϕ_j e integramos sobre el intervalo $[a, b]$ y nos queda lo siguiente:

$$\begin{aligned} - \int_a^b \hat{u}''(x) \phi_j(x) dx + \int_a^b g(\hat{u}(x)) \phi_j(x) dx &= \int_a^b f(x) \phi_j(x) dx \\ \Rightarrow \int_a^b \hat{u}'(x) \phi_j'(x) dx + \int_a^b g(\hat{u}(x)) \phi_j(x) dx &= \int_a^b f(x) \phi_j(x) dx \end{aligned} \quad (4.5)$$

En el caso lineal, la ecuación (4.5) daba lugar a un sistema lineal de ecuaciones. En este caso el sistema es no lineal.

Para resolver la ecuación (4.5) se plantea un sistema de ecuaciones no lineales. Tendremos una ecuación por cada ϕ_j en la base, por lo que tendremos n ecuaciones no lineales con n incógnitas: α_j . El sistema a resolver es el siguiente:

$$F(\alpha) = \mathbf{0} \quad (4.6)$$

Donde $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ y se puede expresar entrada a entrada como sigue:

$$F_j(\alpha) = \int_a^b \hat{u}'(x) \phi_j'(x) dx + \int_a^b g(\hat{u}(x)) \phi_j(x) dx - \int_a^b f(x) \phi_j(x) dx,$$

es claro que si $F(\alpha) = 0$ entonces:

$$\int_a^b \hat{u}'(x) \phi_j'(x) dx + \int_a^b g(\hat{u}(x)) \phi_j(x) dx = \int_a^b f(x) \phi_j(x) dx,$$

que al sustituir $\hat{u}(x)$ por $\sum_{i=1}^n \alpha_i \phi_i(x)$ queda como sigue:

$$\int_a^b \left(\sum_{i=1}^n \alpha_i \phi_i(x) \right)' \phi_j'(x) dx + \int_a^b g \left(\sum_{i=1}^n \alpha_i \phi_i(x) \right) \phi_j(x) dx = \int_a^b f(x) \phi_j(x) dx.$$

4.1.1. Método de Newton

El algoritmo de Newton¹ es un método iterativo, usado para aproximar una solución del sistema no lineal (4.6). El método parte de una aproximación inicial x_0 dada. Se calculan nuevas aproximaciones x_1, x_2, \dots con la fórmula

$$J_F(x_k)(x_{k+1} - x_k) = -F(x_k),$$

dada la aproximación x_k . Por lo que la aproximación x_{k+1} se obtiene resolviendo un sistema lineal de ecuaciones.

El algoritmo se detendrá cuando $F(x_k)$ sea menor a una tolerancia preestablecida, que en el caso del algoritmo programado es 10^{-9} .

Por otro lado para aproximar $J_F(x_k)$ se usa la siguiente fórmula:

$$(J_F(x_k))_{\cdot, i} \approx \frac{F(x_k + he_i) - F(x_k - he_i)}{2h},$$

donde $h \in \mathbb{R}$ es constante y pequeña, en este caso de orden 10^{-3} . Mientras que $e_i \in \mathbb{R}^n$ es el vector que tiene cero en todas las entradas excepto en la i -ésima donde vale 1.

En muchos casos el método de Newton nos ayuda a resolver la ecuación 4.6 a una velocidad de convergencia cuadrática. Para el caso de este trabajo el punto inicial se tomó con puntos sobre la línea que una $(a, u(a)), (b, u(b))$.

¹Ver [8]

Como hemos ejemplificado, el método de elementos finitos es muy flexible y aplicable a otro tipo de ecuaciones, variando tanto el espacio al que pertenece, como el tipo de problema.

Por ejemplo, podríamos usar en lugar de las bases lineales que hemos usado hasta ahora, bases cuadráticas, e inclusive con mayores grados de polinomio. Dependiendo del grado del polinomio en las bases dependerán las características que podremos aproximar de u . Por ejemplo, podríamos estimar $u'(x)$, si usásemos polinomios cúbicos.

CAPÍTULO 5

Resultados numéricos

En este capítulo presentaremos resultados numéricos de algunos ejemplos. La manera de medir el error será la siguiente:

$$\text{err}(\alpha) = \frac{\|\alpha - u(\hat{x})\|}{\|u(\hat{x})\|},$$

donde α es el vector de aproximaciones con el MEF en los nodos, $\hat{x} = (x_1, \dots, x_n)$ es el vector de nodos $\{x_1, \dots, x_n\}$ y $u(\hat{x}) = (u(x_1), u(x_2), \dots, u(x_n))$. En los ejemplos presentados conocemos la solución exacta al problema, sin embargo las pruebas nos servirán para poder observar el comportamiento del método. La máquina en la que se hicieron las pruebas tiene un procesador con 4 núcleos a 200 GHz, 8gb de memoria RAM, en un sistema operativo Ubuntu, con Matlab.

Dado el problema: Encontrar u tal que:

$$-\Delta u = f \quad \text{en } \Omega \subset \mathbb{R}^2, \tag{5.1}$$

$$u = g \quad \text{en } \Gamma = \partial\Omega. \tag{5.2}$$

Primer ejemplo:

$$\Omega = [0, 1] \times [0, 1],$$

$$g(x, y) = 0.$$

$$f(x, y) = -2\pi^2 \sin(\pi x) \sin(\pi y).$$

En este caso sabemos que la solución es:

$$u(x, y) = \sin(\pi x) \sin(\pi y),$$

este problema no explota realmente la fortaleza del método, el dominio es sencillo, pero es un buen inicio para ver cómo se comporta. Para resolverlo, tomamos 2,500 nodos dentro del cuadrado; en estos puntos es donde encontraremos el valor aproximado de la función u ($\alpha \in \mathbb{R}^{2500}$). En este ejemplo el error que se obtuvo fue del orden de 4.54×10^{-4} con un tiempo de 5.3 segundos. En el siguiente ejemplo se ilustra un caso en el que la solución no se anula en la frontera.

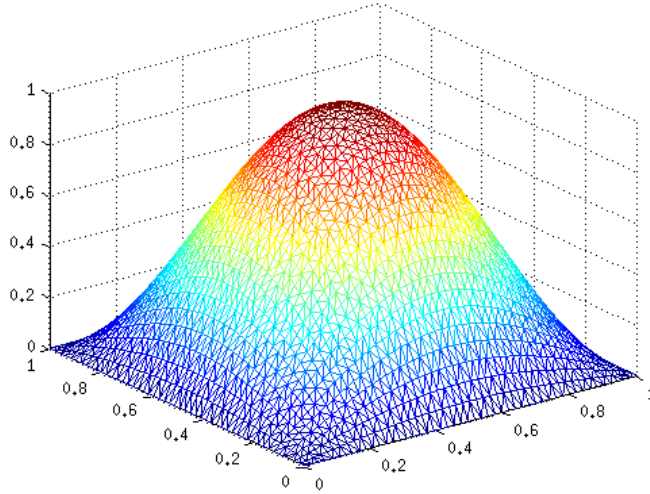


Figura 5.1: Gráfica de la solución exacta.

Las funciones para el siguiente problema son las mismas a excepción de g no es cero:

$$\Omega = [0, 1] \times [0, 1],$$

$$g(x, y) = 2x - y.$$

$$f(x, y) = -2\pi^2 \sin(\pi x) \sin(\pi y).$$

La solución al problema es:

$$u(x, y) = \sin(\pi x) \sin(\pi y) + 2x - y,$$

De nuevo, tomaremos 2,500 nodos dentro del cuadrado.

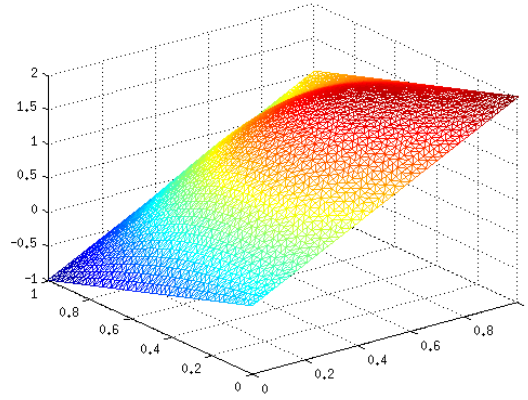
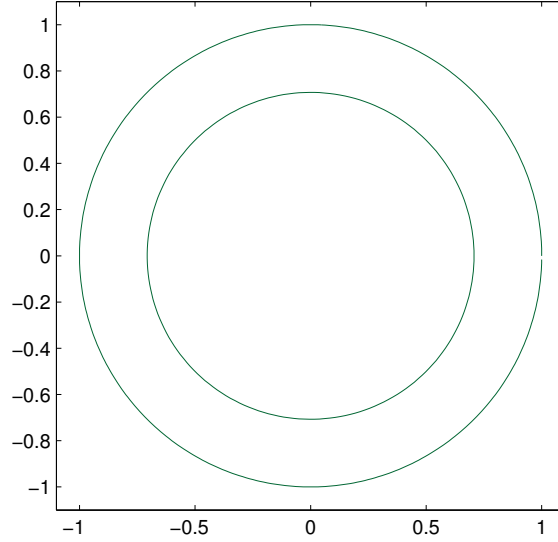


Figura 5.2: Gráfica de la solución exacta.

En este ejemplo el error es del orden de 1.94×10^{-4} de nuevo con un tiempo de 5.3 segundos que no es de sorprender que tengan el mismo tiempo ya que tienen el mismo número de variables.

En el siguiente ejemplo se usará un anillo como región donde se define la ecuación diferencial, a diferencia del cuadrado en los dos primeros ejemplos. en este caso la región está dada por:

$$\left\{ x \in \mathbb{R}^2 \mid \frac{1}{2} \leq \|x\|_2^2 \leq 1 \right\}.$$

Figura 5.3: Representación visual de Ω .

Las funciones f y g de la ecuación (5.1) son:

$$f(x,y) = 16\pi(x^2 + y^2) \sin\left(2\pi\left(x^2 + y^2 - \frac{1}{2}\right)\right) - 8\pi \cos\left(2\pi\left(x^2 + y^2 - \frac{1}{2}\right)\right) \quad (5.3)$$

$$g(x,y) = 2x - y.$$

De nuevo se conoce la solución al problema, para obtener el error de aproximación:

$$u(x,y) = \sin\left(2\pi\left(x^2 + y^2 - \frac{1}{2}\right)\right) + 2x - y.$$

Para este ejemplo, tomaremos 1,000 puntos dentro del anillo. El método no tiene mayor problema con el nuevo conjunto Ω y se adapta bien para conjuntos no tan sencillos. El error obtenido es del orden de 7.9×10^{-4} con un tiempo de procesamiento de 3.4 segundos. La figura 5.5, muestra cómo se ve la solución de la ecuación; mientras que la figura 5.4 muestra la triangulación dada para el dominio.

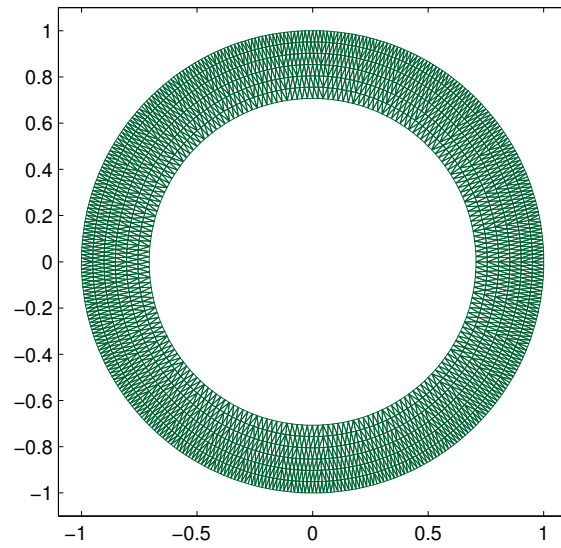


Figura 5.4: Triangulación de el dominio.

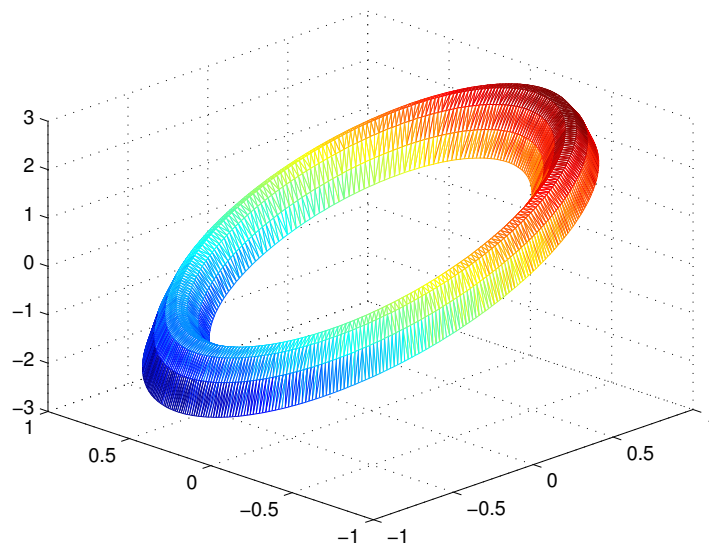


Figura 5.5: Gráfica de la solución.

CAPÍTULO 6

Conclusiones

Un problema de aplicación requiere muchas herramientas para ser resuelto, una de ellas es la creatividad para poder conjuntar las diferentes disciplinas necesarias para resolverlo. El MEF es una gran representación de esta creatividad, ya que se logra hacer un puente entre distintas áreas de las matemáticas. Al hacer esta conjunción se puede obtener una solución tanto útil como práctica.

En principio el problema de la resolución de ecuaciones diferenciales parciales aquí tratado, es encontrar una función, un elemento que “vive” en un espacio de dimensión infinita. Gracias a la formulación variacional se puede hacer una discretización, que reduce el problema en otro de dimensión finita. Con la discretización se hace posible el poder manejar el problema dentro de una computadora con lo que se incrementa la capacidad de resolución a problemas mucho más grandes y complejos.

El manejo de cualquier problema matemático en una computadora implica ciertas limitaciones, algunas pueden ser superadas estudiando y explotando las características de cada problema. En el caso presentado, se solucionó el manejo de matrices muy grandes aprovechando la estructura de las mismas: se almacenó de una manera diferente y eficiente por ser éstas ralas. Asimismo, se procuró utilizar métodos eficientes en el cálculo de integrales que, aunado a la estructuración en el dominio de las funciones utilizadas, permiten realizar un cálculo muy bueno incluso frente a dominios irregulares.

El dominio Ω de la función es una característica fundamental tanto en el problema como en la solución. El MEF triangula el dominio, haciendo más flexible la aproximación del mismo, dando así una mejor representación. El tipo de triangulación usada en esta tesis es cuasi uniforme, pero para algunos dominios más complicados se pueden usar triangulaciones localmente cuasi-uniformes.

El MEF implementado puede ser ampliado de varias maneras. Es posible cambiar la base de funciones $\{\phi_i\}$ a una que contenga funciones más generales, por ejemplo, aumentando el grado del polinomio usado. Esto afecta de forma directa el cálculo de las integrales y la precisión, ganando así un menor error de aproximación. En algunas aplicaciones será necesario representar dominios de mayor dimensión, de tal manera que $\Omega \subset \mathbb{R}^3$. El MEF implementado puede adaptarse utilizando polígonos en el espacio en lugar de triángulos. Esto significa aumentar la complejidad de cálculo y el tamaño del problema, ya que se tienen que tomar en cuenta conceptos de volumen y orientación, pero con el MEF es posible. Además, como ya se mencionó, se puede generalizar el tipo de ecuaciones resueltas a parabólicas o como se mencionó en el capítulo 4, a ecuaciones diferenciales no lineales.

El tener un algoritmo implementando el MEF brinda la posibilidad de resolver una enorme gama de problemas en muchas disciplinas distintas. El dinamismo que posee el método es enorme. Este trabajo brinda ejemplos para una y dos dimensiones con un error de aproximación bastante bueno. La disminución del error, la mejora en el cálculo y la extensión a otras dimensiones o bases serían elementos fácilmente implementables siguiendo los detalles teóricos y técnicos aquí expuestos.

APÉNDICE A

Algoritmo de Delaunay

Para encontrar las triangulaciones necesarias para el desarrollo de este trabajo, como ya mencionamos, se usó el algoritmo de Delaunay. Dicho algoritmo cuenta con ciertas características que son muy deseables para el desarrollo del MEF; tiene también algunas desventajas, mismas que han sido resueltas dentro del algoritmo hecho.

La característica principal que ayuda al desarrollo del MEF es que el algoritmo de Delaunay, maximiza el mínimo de los ángulos de todos los triángulos en la triangulación. Lo que esto garantiza es que el número de condición de la matriz A que usamos en el MEF, tendrá el número de condición máximo posible para un conjunto de puntos en Ω . Una desventaja del algoritmo de Delaunay es que no puede tomar en cuenta dominios como el que presentamos en esta tesis para el ejemplo (5.3) donde Ω no es convexo, pero esta limitante la saltamos como explicamos en la sección (3.4).

Definición A.1 (Condición de Delaunay.). Se dice que una triangulación cumple con la **condición de Delaunay** si ninguno de los circuncírculos de los triángulos pertenecientes a la misma, contiene otros nodos en el interior.

La triangulación final debe cumplir la **condición de Delaunay**. Un ejemplo con cuatro nodos que ilustra la condición de Delaunay:

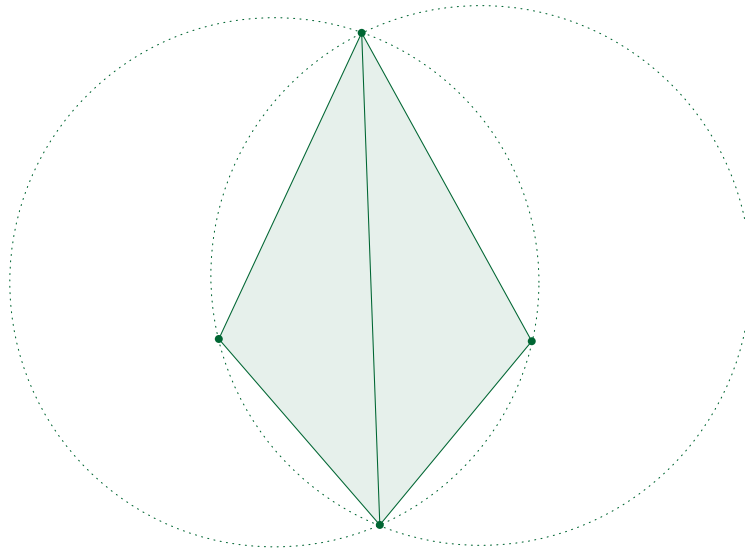


Figura A.1: Triangulación de cuatro nodos que no cumple la condición de Delaunay.

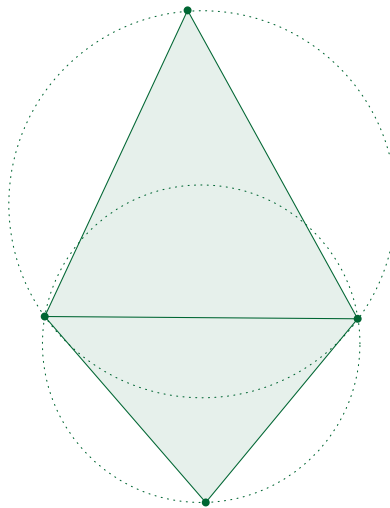


Figura A.2: Triangulación de cuatro nodos que cumple la condición de Delaunay.

Con estas dos figuras ilustramos cómo se cumple la condición de Delaunay, pero a la vez ilustramos un concepto que se refiere a la reflexión, y con ello también un algoritmo sencillo para generar una triangulación con las características deseadas. Por supuesto hay algoritmos más complicados y eficientes, pero esto queda fuera del alcance de este documento.

La manera más directa para generar una triangulación es agregar un vértice a la vez y retriangular las partes cercanas a este nodo, luego para cada triángulo que no cumpla con la condición de delaunay reflejarlo. Así sucesivamente hasta tener la triangulación completa. Para ahondar más en los algoritmos y en la teoría relacionada con la triangulación Delaunay, ver [2] y [4].

APÉNDICE B

Método del gradiente conjugado.

El método del gradiente conjugado, resuelve un sistema de ecuaciones lineales:

$$Ax = b, \quad (B.1)$$

con A simétrica y positiva definida, dando una serie de “pasos”, acercándose en cada uno a la solución. Para resolver el sistema (B.1) se plantea el siguiente problema de minimización:

$$\phi(x) = \frac{1}{2}x^T Ax - b^T x.$$

Como la matriz A es simétrica y positiva definida, sabemos que $x^T Ax > 0 \quad \forall x \in \mathbb{R}^n$. El siguiente problema es equivalente a nuestro problema original B.1: Encontrar $x \in \mathbb{R}^n$ tal que:

$$\phi(x) = \min_{z \in \mathbb{R}^n} \phi(z) \quad (B.2)$$

Sabemos que al derivar nos queda lo siguiente:

$$\nabla \phi(x) = Ax - b = r(x)$$

sabemos por condición de suficiencia que si $\nabla \phi(x') = 0$ entonces x' es un mínimo local, despejando, tenemos que $Ax' = b$ y por el hecho de que A es positiva definida

se puede probar que el mínimo de ϕ se detienen en $x = x'$. Resumiendo, para resolver el sistema de ecuaciones $Ax = b$ podemos resolver equivalentemente el problema B.2.

El método del gradiente conjugado, genera una base *conjugada* de \mathbb{R}^n con respecto a la matriz simétrica positiva definida A . Por vectores conjugados con respecto a una matriz A entenderemos lo siguiente.

Definición B.1. Sean $\{p_1, \dots, p_n\}$ vectores en \mathbb{R}^n distintos de cero, decimos que este conjunto de vectores son conjugados con respecto a la matriz simétrica positiva definida A si:

$$p_i^T A p_j = 0 \quad \text{para todo } i \neq j. \quad (\text{B.3})$$

Consecuencia de esta propiedad es la ortogonalidad de los vectores:

Teorema B.1. Si el conjunto $\{p_1, \dots, p_n\}$ es conjugado con respecto a la matriz A , simétrica y positiva definida, entonces el conjunto es linealmente independiente.

Demostración. Tomemos una combinación lineal de los vectores y la igualamos a cero:

$$\begin{aligned} & \sum_{i=1}^n \alpha_i p_i &= 0 \\ \Rightarrow & \sum_{i=1}^n \alpha_i p_i \cdot A &= 0 \cdot A = 0 \\ \Rightarrow & \sum_{i=1}^n \alpha_i p_i^T \cdot A^T &= 0 \\ \Rightarrow & \left(\sum_{i=1}^n \alpha_i p_i^T \cdot A^T \right) \cdot \sum_{i=1}^n \alpha_i p_i &= 0 \\ \Rightarrow & \sum_{i=1}^n (\alpha_i^2) (p_i^T \cdot A^T \cdot p_i) + \sum_{i \neq j} (\alpha_i \alpha_j) \cdot (p_i \cdot A \cdot p_j) &= 0 \end{aligned}$$

Ahora, sabemos que como A es simétrica y positiva definida, tenemos que $p_i^T \cdot A^T \cdot p_i > 0$. Además por que es un conjunto conjugado con respecto a la matriz A , tenemos también que $p_i \cdot A \cdot p_j = 0$ si $i \neq j$. Por lo tanto tenemos lo siguiente:

$$\sum_{i=1}^n (\alpha_i^2) = 0 \Rightarrow \alpha_i = 0 \quad \forall i = 1 \dots n$$

Por lo tanto, los vectores son linealmente independientes. □

Presento a continuación el algoritmo del gradiente conjugado programado para esta tesis:

Algoritmo 1 Método del gradiente conjugado.

1: Dado x_0 punto inicial para el método.

2: $r_0 \leftarrow Ax_0 - b$.

3: $p_0 \leftarrow -r_0$.

4: $k \leftarrow 0$.

5: **while** $r_k > 10^{-10}$ **do**

6: $\alpha_k \leftarrow \frac{r'_k r_k}{p'_k A p_k}$.

7: $x_{k+1} \leftarrow x_k + \alpha_k p_k$.

8: $r_{k+1} \leftarrow r_k + \alpha_k A p_k$.

9: $\beta_{k+1} \leftarrow \frac{r'_{k+1} r_{k+1}}{r'_k r_k}$.

10: $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$

11: $k \leftarrow k + 1$.

12: **end while**

13: **Return** x_{k+1}

El algoritmo 1 converge a la solución de nuestro sistema de ecuaciones en un número acotado de pasos; el siguiente teorema del libro [8] nos dice que son a los más n :

Teorema B.2. *Para cualquier $x_0 \in \mathbb{R}^n$, la secuencia generada por el algoritmo 1 converge a la solución x^* del sistema B.1 lineal en a lo más n pasos.*

APÉNDICE C

Prueba del lema 3.1

Para demostrar 3.1, probaremos que para cada triángulo $t \in T$, con vértices a_i y $v \in P_1(t)$, se cumple lo siguiente:

$$ch_t^2 \sum_{i=1}^3 |v(a_i)|^2 \leq \|v\|_{\mathcal{L}^2(t)}^2 \leq Ch_t^2 \sum_{i=1}^3 |v(a_i)|^2 \quad (\text{C.1})$$

$$\iint_t |\nabla v|^2 \, dx_1 dx_2 \leq Ch^{-2} \iint_t |v|^2 \, dx_1 dx_2 \quad (\text{C.2})$$

El lema resulta al sumar sobre todos los triángulos $t \in T$, que resulta claro para la desigualdad C.2, pero no necesariamente tan claro para C.1:

$$ch^2 |\alpha|^2 = \beta_1 h^2 \sum_{i=1}^n \alpha_i^2 \leq \beta_1 h^2 c \sum_{i=1}^n n_i \alpha_i^2 \leq \sum_{t \in T} ch_t^2 \sum_{i=1}^3 |v(a_i)|^2 \alpha_i^2,$$

donde $n_i \geq 1$ es el número de triángulos a los que pertenece el nodo i , mientras que también podemos acotar:

$$\sum_{t \in T} Ch_t^2 \sum_{i=1}^3 |v(a_i)|^2 \leq Ch^2 \sum_{i=1}^n n_i \alpha_i^2 \leq Ch^2 n \sum_{i=1}^n \alpha_i^2 = Ch^2 n |\alpha|^2.$$

Por lo que bastará probar C.1 y C.2, para ello demostraremos que se cumplen en un triángulo de “referencia”, luego generalizando el resultado, para cualquier triángulo dado.

Sea $t' = \{(0,0), (0,1), (1,0)\}$ nuestro triángulo de referencia, y $\{\lambda_i\}_{i=1}^3$ base de $P_1(t')$. Tenemos además que para cada λ_i , podemos encontrar un índice j de tal manera que $\phi_j|_{t'} = \lambda_i$. Tenemos que cualquier función lineal dentro del triángulo, puede ser descrita como combinación lineal de la base presentada:

$$\hat{v} = \sum_{i=1}^3 \hat{\alpha}_i \hat{\lambda}_i(\hat{x}_i), \quad \hat{x} \in t'.$$

Para encontrar las cotas deseadas, debemos definir algunas funciones auxiliares:

$$f_1(\hat{\alpha}) = \iint_{t'} |\nabla \hat{v}|^2 d\hat{x}_1 d\hat{x}_2, \quad (\text{C.3})$$

$$f_2(\hat{\alpha}) = \iint_{t'} \hat{v}^2 d\hat{x}_1 d\hat{x}_2, \quad (\text{C.4})$$

$$f_3(\hat{\alpha}) = \sum_{i=1}^3 \hat{\alpha}_i^2, \quad (\text{C.5})$$

$$f_4(\hat{\alpha}) = \frac{f_1(\hat{\alpha})}{f_2(\hat{\alpha})}, \quad (\text{C.6})$$

$$f_5(\hat{\alpha}) = \frac{f_2(\hat{\alpha})}{f_3(\hat{\alpha})} \quad (\text{C.7})$$

con $\hat{\alpha} = (\hat{\alpha}_1, \hat{\alpha}_2, \hat{\alpha}_3)$. Ahora notemos que f_4 y f_5 son homogéneas de grado 0, que implica:

$$f_4(\beta \hat{\alpha}) = f_4(\hat{\alpha}), \quad f_5(\beta \hat{\alpha}) = f_5(\hat{\alpha})$$

Con lo que basta probar C.1 y C.2 para $\hat{\alpha} \in B = \{\hat{\alpha} \mid |\hat{\alpha}| \leq 1\}$, que es un conjunto cerrado y acotado.

Ahora sea $\hat{\alpha} \in B$, como f_4 y f_5 son funciones continuas, por lo cual ambas alcanzan su máximo y su mínimo sobre B , lo que implica que:

$$f_4(\hat{\alpha}) \leq C_1,$$

$$c \leq f_5(\hat{\alpha}) \leq C_2,$$

y si $C = \max(C_1, C_2)$, queda finalmente:

$$\begin{aligned} \iint_t |\nabla \hat{v}|^2 d\hat{x}_1 d\hat{x}_2 &\leq C \iint_t \hat{v}^2 d\hat{x}_1 d\hat{x}_2 \\ c \sum_{i=1}^3 \hat{\alpha}_i^2 &\leq \iint_t \hat{v}^2 d\hat{x}_1 d\hat{x}_2 \leq C \sum_{i=1}^3 \hat{\alpha}_i^2. \end{aligned}$$

Esto demuestra C.1 y C.2 para el caso del triángulo t' .

Ahora probaremos el resultado para un triángulo general $t = (a^1, a^2, a^3)$, para simplificar el álgebra utilizada y sin pérdida de generalidad, supondremos que el triángulo está anclado en el origen, i.e. $t = (0, a^2, a^3)$.

Definimos una transformación que nos mapea el triángulo t' , en el triángulo t :

$$\begin{aligned} x &= F(\hat{x}) = a^2 \hat{x}_1 + a^3 \hat{x}_2, \quad \forall \hat{x} = (\hat{x}_1, \hat{x}_2) \in t' \\ x_1 &= a_1^2 \hat{x}_1 + a_1^3 \hat{x}_2, \\ x_2 &= a_2^2 \hat{x}_1 + a_2^3 \hat{x}_2, \end{aligned}$$

donde $x = (x_1, x_2)$, $a^2 = (a_1^2, a_2^2)$, $a^3 = (a_1^3, a_2^3)$. Ahora tomemos $v \in P_1(t)$ y definimos:

$$v(x) = v(F(\hat{x})) = \hat{v}(\hat{x}), x \in t, \hat{x} \in t',$$

El Jacobiano de la transformación F , se define como sigue:

$$J_F = \begin{pmatrix} \frac{\partial x_1}{\partial \hat{x}_1} & \frac{\partial x_1}{\partial \hat{x}_2} \\ \frac{\partial x_2}{\partial \hat{x}_1} & \frac{\partial x_2}{\partial \hat{x}_2} \end{pmatrix} = \begin{pmatrix} a_1^2 & a_1^3 \\ a_2^2 & a_2^3 \end{pmatrix}.$$

Notemos que:

$$|\det(J_F)| = |a_1^2 a_2^3 - a_1^3 a_2^2| = 2\text{área}(t);$$

gracias al teorema de cambio de variable[1] tenemos lo siguiente:

$$\iint_t v(x) dx_1 dx_2 = \iint_{t'} v(F(x)) |\det(J_F)| d\hat{x}_1 d\hat{x}_2$$

Y de igual manera:

$$\iint_{t'} \hat{v}(\hat{x}) d\hat{x}_1 d\hat{x}_2 = \iint_t v(F^{-1}(x)) |\det(J_{F^{-1}})| dx_1 dx_2$$

y como

$$\begin{aligned} \left| \det(J_{F^{-1}}) \right| &= \frac{1}{\left| \det(J_F) \right|} = \frac{1}{2\text{área}(t)} \\ \Rightarrow \iint_{t'} \hat{v}(\hat{x}) \, d\hat{x}_1 d\hat{x}_2 &= \iint_t v(F^{-1}(x)) \frac{1}{\left| \det(J_F) \right|} \, dx_1 dx_2. \end{aligned}$$

Por hipótesis tenemos que lo siguiente:

$$\frac{\rho_t}{h_t} \geq \beta_2,$$

donde ρ_t es el radio del círculo inscrito en el triángulo t (ver figura 3.7. Es claro que $\rho_t \leq v_t$, donde v_t es la altura del triángulo t , por lo tanto:

$$\begin{aligned} \rho_t h_t &\leq 2\text{área}(t) \leq 2h_t^2 \\ \Rightarrow \frac{1}{2\text{área}(t)} &\leq \frac{1}{\rho_t h_t} \leq \frac{1}{h_t^2 \beta_2} \end{aligned}$$

Con lo anterior podemos ver lo siguiente:

$$\begin{aligned} \iint_t v^2(x) \, dx_1 dx_2 &= \iint_{t'} v^2(F(\hat{x})) \left| \det(J_F) \right| \, d\hat{x}_1 d\hat{x}_2 \\ &= \iint_{t'} \hat{v}^2(\hat{x}) 2\text{área}(t) \, d\hat{x}_1 d\hat{x}_2 \\ &\leq 2h_t^2 \iint_{t'} \hat{v}^2(\hat{x}) \, d\hat{x}_1 d\hat{x}_2 \end{aligned}$$

Y por otro lado, se tiene:

$$\begin{aligned} \iint_{t'} \hat{v}^2(\hat{x}) \, d\hat{x}_1 d\hat{x}_2 &= \iint_t \hat{v}^2(F^{-1}(\hat{x})) \left| \det(J_{F^{-1}}) \right| \, dx_1 dx_2 \\ &= \iint_t v^2(x) \frac{1}{2\text{área}(t)} \, dx_1 dx_2 \\ &\leq \frac{1}{h_t^2 \beta_2} \iint_t v^2(x) \, dx_1 dx_2 \end{aligned}$$

$$\therefore h_t^2 \beta_2 \iint_{t'} \hat{v}^2(\hat{x}) \, d\hat{x}_1 d\hat{x}_2 \leq \iint_t v^2(x) \, dx_1 dx_2 \leq 2h_t^2 \iint_{t'} \hat{v}^2(\hat{x}) \, d\hat{x}_1 d\hat{x}_2$$

Con lo que tenemos que se cumple la desigualdad C.1.

Sólo queda probar C.2, y para ello notemos que:

$$\begin{aligned}
 \frac{\partial \hat{v}}{\partial \hat{x}_1} &= \frac{\partial v}{\partial x_1} \frac{\partial x_1}{\partial \hat{x}_1} + \frac{\partial v}{\partial x_2} \frac{\partial x_2}{\partial \hat{x}_1} \\
 \frac{\partial \hat{v}}{\partial \hat{x}_2} &= \frac{\partial v}{\partial x_1} \frac{\partial x_1}{\partial \hat{x}_2} + \frac{\partial v}{\partial x_2} \frac{\partial x_2}{\partial \hat{x}_2} \\
 &\Rightarrow \nabla \hat{v} = J_F^T \nabla v \\
 &\Rightarrow \nabla v = J_F^{-T} \nabla \hat{v} \\
 &\Rightarrow |\nabla v|^2 = |J_F^{-T} \nabla \hat{v}|^2 \leq \|J_F^{-T}\|_{fro}^2 |\nabla \hat{v}|^2 \leq \frac{h_t^{-2}}{4\beta_2^2} |\nabla \hat{v}|^2
 \end{aligned}$$

Y con esto nos quedan las siguientes desigualdades:

$$\begin{aligned}
 \iint_t |\nabla v|^2 dx_1 dx_2 &\leq \iint_{t'} |\nabla \hat{v}|^2 h_t^{-2} h_t^2 \xi_1 d\hat{x}_1 d\hat{x}_2 \\
 &\leq \xi_1 C \iint_{t'} \hat{v}^2 \\
 &\leq \xi_2 h^{-2} C \iint_t v(x) dx_1 dx_2,
 \end{aligned}$$

lo cual prueba el lema 3.1

APÉNDICE D

Código

D.1. Main

```
1 %% Fijar Parametros
2 %% Definicion de las funciones de la ecuacion.
3 f = 'fn';
4 g = 'g5';
5 %% Numero de puntos sobre el circulo.
6 fineza = 200;
7 nc = 5;
8 tic
9 %% La funcion triangulo nos da la triangulacion y
10 %% otros datos necesarios.
11 [x,y,tri2,ni,nf,mm] = triangulo(fineza,nc);
12 %% En este caso conocemos la solucion de la ecuacion
13 %% usaremos z para medir el comportamiento del algoritmo
14 z = u5(x,y);
15 %% Procesar datos
16 tn = length(x);
```



```

17 [corte,m] = size(ni);
18 %% Vecinos nos da los vecinos de cada nodo, triangulos internos
19 %% triangulos frontera
20 [veci,trif,trin] = vecinos(tn,tri2,mm,corte);
21 %% Construir el sistema de ecuaciones.
22 [A,ig,b] = constru(f,g,tri2,veci,trif,trin,corte,x,y);
23 %% Resolver el sistema de ecuaciones.
24 x0 = ones(corte,1)*1.25;
25 %% En este punto es donde se podria cambiar el metodo de solucion
26 %% del sistema lineal.
27 [alpha,k] = gradconj(A,x0,b);
28 toc
29 %% Presentar resultados.
30 norm(z(1:corte)-alpha)/norm(z)
31 frontera = feval(g,x(corte+1:tn),y(corte+1:tn));
32 close all
33 %% Graficar los resultados.
34 close all
35 figure
36 figure(1)
37 trimesh(tri2,x,y,[alpha;frontera])
38 figure(2)
39 trimesh(tri2,x,y,z)
40 figure(3)
41 triplot(tri2,x,y,'Color',[0,0.4,0.2])
42 axis equal
43 axis([-1.1 1.1 -1.1 1.1])

```

D.2. Triangulo.m

```

1  %%% Funcion que calcula en funcion de el numero de circulos
2  %%% y del numero de puntos sobre los circulos la triangulacion
3  %%% los nodos, cuales son nodos interiores, cuales son nodos
4  %%% frontera, y el numero de triangulos finales de la
5  %%% triangulacion.
6  function [x,y,tri2,ni,nf,mm]=triangulo(fineza,nc)
7  t=linspace(0,1,fineza+1);
8  x=[];
9  y=[];
10 %% Creamos los nodos en la frontera, tanto interna como externa.
11 t=t(1:fineza);
12 cx1=sqrt(0.5)*cos(2*pi*t);

```

```

13 cyl=sqrt(0.5)*sin(2*pi*t);
14 t=linspace(0,1,2*fineza+1);
15 t=t(1:2*fineza);
16 cx4=cos(2*pi*t);
17 cy4=sin(2*pi*t);
18 nf=[cx1,cx4;cy1,cy4]';
19
20 %% Vamos incrementando poco a poco el radio de los circulos
21 %% para generar los puntos interiores del conjunto de interes.
22
23 for i = 1 : nc
24     cambio=floor(fineza*(1+i/(nc+1)));
25     t=linspace(0,1,cambio+1);
26     t=t(1:cambio);
27     cxi=((sqrt(0.5)+(1-sqrt(0.5))*i/(nc+1)))*cos(2*pi*t);
28     cyi=((sqrt(0.5)+(1-sqrt(0.5))*i/(nc+1)))*sin(2*pi*t);
29     x=[x,cxi];
30     y=[y,cyi];
31 end
32 %% Pegamos los nodos en un mismo vector.
33 ni=[x;y]';
34 x=[x,cx1,cx4]';
35 y=[y,cy1,cy4]';
36
37 %% Aqui se genera la triangulacion con el algoritmo delaunay(x,y)
38 %% funcion de Matlab.
39 tri=delaunay(x,y);
40 [m,n]=size(tri);
41
42 indices=ones(m,1);
43 %% Vamos triangulo por triangulo buscando si estan dentro
44 %% o fuera del dominio, si estan fuera, los sacamos de la
45 %% triangulacion.
46 for k=1:m
47     centro_x=(x(tri(k,1))+x(tri(k,2))+x(tri(k,3)))/3;
48     centro_y=(y(tri(k,1))+y(tri(k,2))+y(tri(k,3)))/3;
49     r2= centro_x^2 + centro_y^2;
50     if r2<0.5 || r2>1
51         indices(k)=0;
52     end
53 end
54 mm=sum(indices);
55 tri2=zeros(mm,3);
56 cont=0;
57 for k=1:m

```

```

58     if indices(k) == 1
59         cont = cont + 1;
60         tri2(cont, :) = tri(k, :);
61     end
62 end

```

D.3. Vecinos.m

```

1  %%% Funcion que para cada nodo encuentra todos los
2  %%% vecinos y a los triangulos por el cual es vecino
3  %%% del nodo en cuestion.
4
5  function [veci, trif, trin] = vecinos(tn, tri2, mm, corte)
6
7  trin = zeros(tn, 10);
8  veci = zeros(corte, 2 * 16);
9  trif = zeros(corte, 8);
10 % En los for's se itera sobre todos los nodos y
11 % encontramos los vecinos y de que triangulos
12 % son vecinos. Ademas de clasificar a los
13 % triangulos en fronterizos e interiores.
14 for i = 1:mm
15     for j = 1:3
16         l = 1;
17         while trin(tri2(i, j), l) ~= 0
18             l = l + 1;
19         end
20         trin(tri2(i, j), l) = i;
21     end
22 end
23
24 for i = 1:corte
25     l = 1;
26     while trin(i, l) ~= 0
27         for j = 1:3
28             s = 1;
29             if tri2(trin(i, l), j) < corte + 1
30
31                 if tri2(trin(i, l), j) ~= i
32                     while veci(i, s) ~= 0
33                         s = s + 1;
34                     end

```

```

35         %% Sacar el vecino.
36         veci(i,s)=tri2(trin(i,l),j);
37         %% El numero de triangulo que
38         %% esta.
39         veci(i,s+1)=trin(i,l);
40     end
41 else
42     while trif(i,s)~=0 && trif(i,s)~=trin(i,l)
43         s=s+1;
44     end
45     %% El numero de triangulo que esta.
46     trif(i,s)=trin(i,l);
47 end
48 end
49 l=l+1;
50 end
51 end

```

D.4. Constr.m

```

1  %% Constr es una fucion que construye la matriz A y el
2  %% vector b para ponerlo en el sistema lineal de ecuaciones.
3  %% Recibe todos los nodos, triangulos, vecinos f y g como
4  %% funciones. (nombres de funciones)
5
6  function [A,ig,b]=constru(fname,...
7      gname, ...
8      tri2, ...
9      veci, ...
10     trif, ...
11     trin, ...
12     corte, ...
13     x, ...
14     y)
15     A=zeros(corte,30);
16     t=[0,0,0];
17     ig=zeros(corte,1);
18     b=zeros(corte,1);
19     for i =1: corte
20         l=1;
21         while l<32 && veci(i,l)~=0
22             %% Aqui sacamos el triangulo veci(i,l)

```

```

23     %% es el vecino en el l+1 esta el triangulo.
24     t(1:3)=tri2(veci(i,l+1),:);
25     %% Encontramos donde esta el que queremos.
26     %% Para el nodo i encontramos la pocision
27     %% sobre el triangulo.
28     [x0,loc]=min(abs(t-i));
29     %% Construimos el primer punto (con altura 1)
30     %% para poder calcular la mult de los grads.
31     x0=[x(t(loc)),y(t(loc)),1]';
32     %% Para poder sacar el area del triangulo.
33     xtr0=[x(t(loc)),y(t(loc))]'';
34     %% el switch es en funcion de en que parte
35     %% del triangulo esta el nodo donde estamos.
36     switch loc
37     case 1
38         x1=[x(t(2)),y(t(2)),0]';
39         x2=[x(t(3)),y(t(3)),0]';
40         xtr1=[x(t(2)),y(t(2))]'';
41         xtr2=[x(t(3)),y(t(3))]'';
42     case 2
43         x1=[x(t(1)),y(t(1)),0]';
44         x2=[x(t(3)),y(t(3)),0]';
45         xtr1=[x(t(1)),y(t(1))]'';
46         xtr2=[x(t(3)),y(t(3))]'';
47
48     case 3
49         x1=[x(t(2)),y(t(2)),0]';
50         x2=[x(t(1)),y(t(1)),0]';
51         xtr1=[x(t(2)),y(t(2))]'';
52         xtr2=[x(t(1)),y(t(1))]'';
53     end
54     %% Debemos construir los triangulos y
55     %% la funcion para el vecino,
56     [x00,loc]=min(abs(t-veci(i,l)));
57     %% Sacamos con altura 1 el nodo.
58     x01=[x(t(loc)),y(t(loc)),1]';
59     switch loc
60     case 1
61         x11=[x(t(2)),y(t(2)),0]';
62         x21=[x(t(3)),y(t(3)),0]';
63     case 2
64         x11=[x(t(1)),y(t(1)),0]';
65         x21=[x(t(3)),y(t(3)),0]';
66
67     case 3

```

```

68             x11=[x(t(2)),y(t(2)),0]';
69             x21=[x(t(1)),y(t(1)),0]';
70         end
71         s=1;
72         %% Búsqueda para ver si ya habíamos
73         %% hecho la integral y si no, para
74         %% que encuentre el lugar para guardarlo.
75         while A(i,s)~=0 && A(i,s+1)~=veci(i,l)
76             s=s+1;
77         end
78         %% area del triangulo por el producto de
79         %% los gradientes.
80         A(i,s)=A(i,s)+gradp(x0,x1,x2)'* ...
81             gradp(x01,x11,x21)*...
82             area(xtr1,xtr2,xtr0);
83         A(i,s+1)=veci(i,l);
84         l=l+2;
85     end
86     l=1;
87
88     %% A(i,i) Tiene que calcularse aparte
89     %% por cuestiones de que se requieren
90     %% todos los triangulos
91     while l<32 && trin(i,l)~=0
92         t(1:3)=tri2(trin(i,l),:);
93         [x0,loc]=min(abs(t-i));
94         %% Construimos el primer punto (con altura 1)
95         %% para poder calcular la mult de los grads.
96         x0=[x(t(loc)),y(t(loc)),1]';
97         %% Para poder sacar el area del triangulo.
98         xtr0=[x(t(loc)),y(t(loc))]'';
99         %% el switch es en funcion de en que parte del
100         %% triangulo esta el nodo donde estamos.
101         switch loc
102             case 1
103                 x1=[x(t(2)),y(t(2)),0]';
104                 x2=[x(t(3)),y(t(3)),0]';
105                 xtr1=[x(t(2)),y(t(2))]'';
106                 xtr2=[x(t(3)),y(t(3))]'';
107             case 2
108                 x1=[x(t(1)),y(t(1)),0]';
109                 x2=[x(t(3)),y(t(3)),0]';
110                 xtr1=[x(t(1)),y(t(1))]'';
111                 xtr2=[x(t(3)),y(t(3))]'';
112

```

```

113         case 3
114             x1=[x(t(2)),y(t(2)),0]';
115             x2=[x(t(1)),y(t(1)),0]';
116             xtr1=[x(t(2)),y(t(2))]';
117             xtr2=[x(t(1)),y(t(1))]';
118         end
119
120         while A(i,s)~=0 && A(i,s+1)~= i
121             s=s+1;
122         end
123         A(i,s)=A(i,s)+ gradp(x0,x1,x2) '*gradp(x0,x1,x2)* ...
124             area(xtr1,xtr2,xtr0);
125         A(i,s+1)=i;
126         %% aqui se calcula ig que es el vector que se resta
127         %% al lado derecho para tomar en cuenta una condicion
128         %% de frontera no-homogenea
129         if l<32 && trif(i,l)~=0
130             %% Aqui guardo los tres triangulos con un nodo
131             %% frontera.
132             t(1:3)=tri2(trif(i,l),:);
133             %% Encontramos la loc donde esta el nodo dentro
134             %% del triangulo.
135             [x0,loc]=min(abs(t-i));
136             %% Construimos el primer punto (con altura 1)
137             %% para poder calcular la mult de los grads.
138             x0=[x(t(loc)),y(t(loc)),1]';
139             %% Para poder sacar el area del triangulo.
140             xtr0=[x(t(loc)),y(t(loc))]';
141             %% Practicamente analogo al bloque anterior
142             switch loc
143
144                 case 1
145                     x1=[x(t(2)),y(t(2)),0]';
146                     x2=[x(t(3)),y(t(3)),0]';
147                     xtr1=[x(t(2)),y(t(2))]';
148                     xtr2=[x(t(3)),y(t(3))]';
149                     x01=[x(t(1)),y(t(1)),0]';
150                     if t(2)>corte
151                         x11=[x(t(2)),y(t(2)), ...
152                             feval(gname,x(t(2)),y(t(2)))]';
153                     else
154                         x11=[x(t(2)),y(t(2)),0]';
155                     end
156                     if t(3)>corte
157                         x21=[x(t(3)),y(t(3)), ...

```

```

158         feval(gname,x(t(3)),y(t(3)))];';
159     else
160         x21=[x(t(3)),y(t(3)),0]';
161     end
162
163     case 2
164         x1=[x(t(1)),y(t(1)),0]';
165         x2=[x(t(3)),y(t(3)),0]';
166         xtr1=[x(t(1)),y(t(1))]';
167         xtr2=[x(t(3)),y(t(3))]';
168         x01=[x(t(2)),y(t(2)),0]';
169         if t(1)>corte
170             x11=[x(t(1)),y(t(1)),...
171                 feval(gname,x(t(1)),y(t(1)))];';
172         else
173             x11=[x(t(1)),y(t(1)),0]';
174         end
175         if t(3)>corte
176             x21=[x(t(3)),y(t(3)),...
177                 feval(gname,x(t(3)),y(t(3)))];';
178         else
179             x21=[x(t(3)),y(t(3)),0]';
180         end
181     case 3
182         x1=[x(t(2)),y(t(2)),0]';
183         x2=[x(t(1)),y(t(1)),0]';
184         xtr1=[x(t(2)),y(t(2))]';
185         xtr2=[x(t(1)),y(t(1))]';
186         x01=[x(t(3)),y(t(3)),0]';
187         if t(2)>corte
188             x11=[x(t(2)),y(t(2)),...
189                 feval(gname,x(t(2)),y(t(2)))];';
190         else
191             x11=[x(t(2)),y(t(2)),0]';
192         end
193         if t(1)>corte
194             x21=[x(t(1)),y(t(1)),...
195                 feval(gname,x(t(1)),y(t(1)))];';
196         else
197             x21=[x(t(1)),y(t(1)),0]';
198         end
199     end
200     ig(i)= ig(i)+ gradp(x0,x1,x2)'* ...
201         gradp(x21,x11,x01)* ...

```



```

203                                     area(xtr1,xtr2,xtr0);
204
205                                     end
206                                     l=l+1;
207     end
208     %% Calculo la integral de la multiplicacion por medio
209     %% de la cuadratura y se le resta ig.
210     b(i)=cuadratura(fname,i,trin(i,:),tri2,x,y)-ig(i);
211 end

```

D.5. gradp.m

```

1  %% Evalua el gradiente de la funcion para
2  %% los planos definidos en un triangulo.
3  %% x0 es el punto con altura 1 o f(nodo).
4  %% x1 y x2 son dos puntos sobre el plano
5  %% (con tercera coordenada cero).
6  function h=gradp(x1,x2,x0)
7      a=x1-x2;
8      b=x0-x2;
9      eta=[a(2).*b(3)-a(3).*b(2) ...
10          a(3).*b(1)-a(1).*b(3) ...
11          a(1).*b(2)-a(2).*b(1)];
12      h=-[eta(1)/eta(3);eta(2)/eta(3)];
13 end

```

D.6. area.m

```

1  %% funcion que recibe tres puntos y
2  %% calcula el area que delimitan.
3  %% x,y,z puntos en el plano.
4  %% a area total dada por los puntos.
5  function a=area(x,y,z)
6      ac=x-y;
7      ab=z-y;
8      nab=[ab(2),-ab(1)];
9      a=abs(nab*ac)/2;
10 end

```

D.7. gradconj.m

```
1 %% Funcion Gradiente conjugado, Recibe A matriz rala,  
2 %% x0 que es el punto inicial, y b el vector de la derecha.  
3 %% Todos vectores columna.Modificamos el metodo del  
4 %% gradiente conjugado para recibir una matriz con solo  
5 %% las entradas distintas de cero.  
6 function [xk,k]=gradconj(A,xk,b)  
7     rk = mult(A,xk) - b;  
8     pk = -rk;  
9     k = 0;  
10    while norm(rk) >1e-10  
11        v=mult(A,pk);  
12        alphak=(rk'*rk)/(pk'*v);  
13        xk1 = xk + alphak*pk ;  
14        rk1 = rk + alphak * v ;  
15        betak1 = (rk1'*rk1)/(rk'*rk);  
16        pk1=-rk1+betak1*pk;  
17        xk=xk1;  
18        rk=rk1;  
19        pk=pk1;  
20        k=k+1;  
21    end  
22 end
```

Bibliografía

- [1] Tom M. Apostol. *CALCULUS. Cálculo con funciones de varias variables y Álgebra Lineal, con aplicaciones a las ecuaciones diferenciales y las probabilidades*. Editorial Reverté, 1985.
- [2] M. Berg. *Computational geometry: algorithms and applications*. Springer-Verlag, 2008.
- [3] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2002.
- [4] B. J. Schachter D. T. Lee. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3), 1980.
- [5] A.N. Kolmogorov & S.V. Fomin. *Introductory Real Analysis*. Dover Publications, Inc., 1998.
- [6] Guillermo Grabinsky. *Teoría de la Medida*. Facultad de Ciencias, UNAM, 2009.
- [7] Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover, 2009.

- [8] J. Nocedal & S.J. Wright. *Numerical optimization*. Springer-Verlag, 1999.
- [9] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals: Its Basis and Fundamentals*. Elsevier Science, 2005.
- [10] José Ángel Canavati Ayub. *Introducción al Análisis Funcional*. Fondo de Cultura Económica, 1998.