

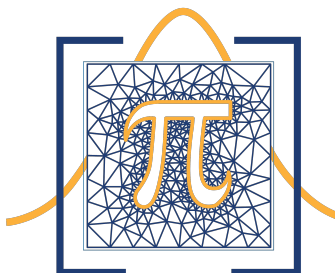
UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS
FACULTAD DE CIENCIAS
ESCUELA DE MATEMÁTICA Y CIENCIAS DE LA COMPUTACIÓN
MAESTRÍA EN MATEMÁTICA



UNAH
UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

Método de Elementos Finitos Multiescala Generalizado (GMsFEM) aplicado a flujo en medios porosos

Tesis presentada para optar al título de máster en matemática con
orientación en ingeniería matemática



Tesis presentada por: Yesy Karina Sarmiento Perdomo

Asesor tesis: Juan Galvis
Fecha de defensa: Junio de 2026

TEGUCIGALPA, HONDURAS 2026.

Método de Elementos Finitos Multiescala Generalizado (GMsFEM) aplicado a flujo en medios porosos

Presentada por: Yesy Karina Sarmiento Perdomo

**Tesis presentada para optar al título de máster en matemática con
orientación en ingeniería matemática**

Asesor: Juan Galvis
Universidad Nacional de Colombia, sede Bogotá

TERNA EXAMINADORA
Ph.D Sergio Alejandro Gómez Macías
Università degli Studi di Milano-Bicocca

Ph.D Luis Zerón
Universidad Nacional Autónoma de Honduras

Ph.D Mario Suazo Euceda
Universidad Nacional Autónoma de Honduras

TEGUCIGALPA M.D.C JUNIO DEL 2026.

Agradecimientos

A mi asesor, mis gatos y amistades.

Yesy Karina Sarmiento

Índice general

Agradecimientos	II
Índice de figuras	V
Índice de tablas	VI
Resumen	VII
Abstract	VIII
1. Introducción	1
1.1. Objetivos	4
1.1.1. Objetivo general	4
1.1.2. Objetivos específicos	4
2. Fundamentos	6
2.1. Espacios normados y espacios de Banach	6
2.2. Funcionales	7
2.3. Espacios con producto interno y espacios de Hilbert	9
2.3.1. Teorema de representación de Riesz	10
2.4. Espacios de funciones	10
2.5. Espacios de Sobolev	11
2.6. Formas bilineales	13
2.6.1. Formulación variacional	14
2.6.2. Principio de Rayleigh-Ritz	15
2.6.3. Principio de Galerkin	16
2.6.4. Equivalencia entre el problema variacional y el problema de minimización	17
2.6.5. Teorema de Lax-Milgram	20
3. Método de elementos finitos en una dimensión	21
3.1. Implementación didáctica en Julia para funciones base de grado 1	26
3.2. Implementación didáctica en Julia para funciones base de grado 2	32
4. Método de elementos finitos en dos dimensiones	39
4.1. Formulación fuerte	39
4.1.1. Formulación débil	39
4.1.2. Formulación de Galerkin	42
4.1.3. Estimación del error a posteriori	44
5. Uso de Gridap	47
5.1. Componentes principales	47
5.2. Ejemplo práctico con Gridap	48
5.2.1. Declaración del problema	49
5.2.2. Esquema numérico	49
5.2.3. Implementación con Gridap	51
5.2.3.1. Modelo discreto y carga de la malla	51
5.2.3.2. Espacios de funciones y condiciones de frontera	51
5.2.3.3. Formulación del problema y resolución	52

6. Método generalizado de elementos finitos multiescala (GMsFEM) con malla gruesa	54
6.1. Problema Fino	54
6.2. Formulación variacional y estabilidad	55
6.3. Construcción de funciones base multiescala	62
6.4. Teoría del error del método generalizado de elementos finitos multiescala (GMsFEM)	66
6.4.1. Espacio de Galerkin multiescala	66
6.4.2. Norma de energía	67
6.4.3. Estimación a priori del error	68
6.4.4. Estimación a posteriori del error	68
6.4.5. Discusión e interpretación	69
7. Experimentos numéricos	70
7.1. Caso de validación: medio poroso sintético	70
7.2. Caso de estudio: SPE10 — capa 85	73
7.2.1. Metodología de preparación del dato SPE10	73
7.2.2. Formulación del problema y parámetros numéricos	74
7.2.3. Resultados numéricos y análisis	74
8. Estudios adicionales: GMsFEM sin malla gruesa	78
8.1. Modelo matemático	78
8.2. Construcción de la escala gruesa sin malla	79
8.3. Cálculo de las funciones base multiescala	80
8.4. Formulación variacional en la escala gruesa	81
8.5. Recuperación de la solución a la escala fina	82
9. Conclusiones y trabajos futuros	84
Bibliografía	87
A. Implementaciones	90

Índice de figuras

3.1.	Comparación solución exacta y solución de elementos finitos en 3 elementos.	30
3.2.	Comparación solución exacta y solución de elementos finitos en 10 elementos.	31
3.3.	Comparación solución exacta y solución de elementos finitos en 3 elementos.	37
3.4.	Comparación solución exacta y solución de elementos finitos en 10 elementos.	37
5.1.	Dominio tridimensional con forma de bola Ω utilizado en la resolución de la ecuación de Poisson.	50
5.2.	Distribución de la solución numérica en una bola tridimensional.	53
6.1.	Vecindades del dominio D . Elaboración propia en Matlab.	63
7.1.	Distribución del campo de permeabilidad heterogéneo. Elaboración propia.	71
7.2.	Comparación de la solución a escala fina y la multiescala con $m = 12$. Ambas se comportan de manera similar, lo que demuestra que el método multiescala reproduce fielmente la solución fina.	72
7.3.	Comparación visual entre la solución fina y la solución reconstruida mediante GMSFEM para la capa 85 del modelo SPE10. Ambas imágenes utilizan la misma escala de color para resaltar las diferencias locales.	76
7.4.	Comportamiento del error relativo en función del número de bases adicionales m en cada vecindad. Elaboración propia.	77
8.1.	Construcción de la discretización meshfree utilizada en el método MFGMs-FEM. Se muestra la malla fina de referencia, la distribución heterogénea del coeficiente de permeabilidad y la nube de puntos empleada para definir los vecindarios locales.	79
8.2.	Construcción de funciones base multiescala en el método MFGMsFEM. Se muestra una función de partición de unidad asociada a un vecindario local y ejemplos de funciones base multiescala obtenidas mediante problemas espectrales locales.	81
8.3.	Comparación entre la solución multiescala obtenida mediante el método GMSFEM y la solución de referencia calculada sobre una malla fina. . . .	82

Índice de tablas

7.1. Errores relativos para diferentes números de bases adicionales en el caso sintético ($t = 0$, $N_x = 10$, $N_y = 10$, $n_x = n_y = 4$).	72
7.2. Errores relativos para diferentes números de bases adicionales ($t = 0$, $N_x = 6$, $N_y = 22$, $n_x = n_y = 10$).	76

Método de Elementos Finitos Multiescala Generalizado(GMsFEM) aplicado a flujo en medios porosos

Resumen

En esta tesis se estudia y aplica el Método Generalizado de Elementos Finitos Multiescala (GMsFEM) para la resolución de la ecuación de Darcy en medios porosos heterogéneos. Esta investigación se enmarca dentro de los métodos multiescala, los cuales permiten representar los efectos sobre la solución de heterogeneidades y variaciones locales del medio sin recurrir a la solución de sistemas lineales grandes generados por las discretizaciones finas globales los cuales requieren de un alto costo computacional para su solución.

En el marco del desarrollo teórico y computacional, se presentan fundamentos del método de elementos finitos clásico, entre los que se encuentran formulaciones variacionales, herramientas de análisis funcional e implementaciones en Julia para problemas en una y tres dimensiones. Luego, se desarrolla el método GMsFEM mediante la construcción de funciones base multiescala obtenidas a partir de problemas espectrales locales definidos sobre vecindades de una malla gruesa.

Para analizar el desempeño del método, se considera inicialmente un campo de permeabilidad heterogéneo sintético generado aleatoriamente, con el fin de estudiar la sensibilidad del error respecto al número de funciones base adicionales por vecindad. Posteriormente, el método se aplica al benchmark internacional SPE10, específicamente en la capa 85, caracterizada por presentar altos contrastes y heterogeneidades complejas en el coeficiente de permeabilidad. En ambos casos se considera la ecuación de Darcy estacionaria con condiciones de contorno mixtas y término fuente nulo.

Además, se presenta un estudio complementario de una variante del método GMsFEM sin malla, basada en la construcción de vecindades locales y funciones base multiescala a partir de nubes de puntos. Este enfoque permite extender la metodología multiescala a dominios donde la generación de mallas estructuradas resulta compleja.

Los resultados numéricos muestran que los errores relativos en norma L^2 y norma energía disminuyen al incrementar el número de funciones base adicionales, evidenciando la convergencia del método hacia la solución de referencia de malla fina. Se observa que el intervalo $m \in [12, 24]$ proporciona un balance adecuado entre precisión y costo computacional, mientras que valores mayores producen errores menores a expensas de mayores tiempos de ejecución.

Finalmente, la implementación desarrollada en Julia demuestra la eficiencia del lenguaje para la simulación numérica de problemas multiescala y verifica la capacidad del método GMsFEM para aproximar flujos en medios porosos altamente heterogéneos.

Palabras clave: GMsFEM, métodos multiescala, ecuación de Darcy, medios porosos, Julia, SPE10, simulación numérica.

Generalized Multiscale Finite Element Method (GMsFEM) applied to flow in porous media

Abstract

In this thesis, the Generalized Multiscale Finite Element Method (GMsFEM) is studied and applied to the numerical solution of the Darcy equation in heterogeneous porous media. This research is framed within multiscale methods, which allow the representation of the effects of heterogeneities and local variations of the medium on the solution without resorting to the solution of large linear systems generated by global fine-scale discretizations, whose resolution involves a high computational cost.

Within the theoretical and computational framework, the foundations of the classical finite element method are presented, including variational formulations, tools from functional analysis, and Julia implementations for one- and three-dimensional problems. Subsequently, the GMsFEM is developed through the construction of multiscale basis functions obtained from local spectral problems defined over neighborhoods of a coarse mesh.

To analyze the performance of the method, a synthetic heterogeneous permeability field generated randomly is initially considered in order to study the sensitivity of the error with respect to the number of additional basis functions per neighborhood. Afterwards, the method is applied to the international SPE10 benchmark, specifically to layer 85, which is characterized by high contrasts and complex heterogeneities in the permeability coefficient. In both cases, the stationary Darcy equation with mixed boundary conditions and zero source term is considered.

In addition, a complementary study of a meshfree variant of the GMsFEM is presented, based on the construction of local neighborhoods and multiscale basis functions from point clouds. This approach makes it possible to extend the multiscale methodology to domains where the generation of structured meshes is difficult.

The numerical results show that the relative errors in the L^2 norm and energy norm decrease as the number of additional basis functions increases, demonstrating the convergence of the method toward the fine-scale reference solution. It is observed that the interval $m \in [12, 24]$ provides a suitable balance between accuracy and computational cost, whereas larger values produce smaller errors at the expense of higher execution times.

Keywords: GMsFEM, multiscale methods, Darcy equation, porous media, Julia, SPE10, numerical simulation.

Capítulo 1

Introducción

La simulación de flujos en medios porosos heterogéneos constituye un problema fundamental en diversas áreas de la ingeniería y las ciencias aplicadas, tales como la hidrogeología, la ingeniería de yacimientos y la transferencia de calor. Estos fenómenos suelen modelarse mediante ecuaciones diferenciales parciales de tipo elíptico, en las cuales los coeficientes presentan variaciones significativas a lo largo de múltiples escalas espaciales. En particular, la ecuación de Darcy estacionaria puede formularse matemáticamente como una ecuación elíptica de difusión, se escribe como

$$-\nabla \cdot (\kappa(x) \nabla u(x)) = f(x) \quad \text{en } \Omega \subset \mathbb{R}^2,$$

$$u = g_D \text{ en } \Gamma_D, \quad (\kappa \nabla u) \cdot n = g_N \text{ en } \Gamma_N,$$

donde $\kappa(x)$ representa el coeficiente de permeabilidad del medio, u la presión del fluido, f un término fuente y las condiciones de contorno pueden ser de Dirichlet o Neumann. En medios altamente heterogéneos, $\kappa(x)$ varía drásticamente en el espacio, lo que induce oscilaciones locales en la solución u que no pueden resolverse si no se construyen mallas suficientemente finas [11, 12].

Los métodos estándar de elementos finitos (FEM) proporcionan una aproximación robusta para este tipo de problemas [18, 21]; sin embargo, su desempeño se deteriora significativamente en presencia de coeficientes con altos contrastes. En tales situaciones, es necesario recurrir a mallas localmente refinadas que permitan capturar adecuadamente las variaciones de $\kappa(x)$. Este refinamiento conduce a un incremento sustancial en el número de grados de libertad del sistema y en el número de condición de la matriz del sistema lineal resultante, y en consecuencia, a un aumento considerable del costo computacional [7].

Para superar esta limitación, se han desarrollado métodos multiescala que incorporan información local del medio en funciones base enriquecidas. El Método de Elementos Finitos

Multiescala (MsFEM) propone resolver problemas locales en cada celda de una malla gruesa con el fin de construir funciones base adaptadas al campo de permeabilidad [17].

Posteriormente, el Método Generalizado de Elementos Finitos Multiescala (GMsFEM) extiende esta idea mediante la construcción sistemática de funciones base enriquecidas a partir de problemas espectrales locales de la forma

$$A_i \phi = \lambda S_i \phi,$$

donde A_i y S_i representan, respectivamente, las matrices locales de rigidez y de masa asociadas a cada vecindad gruesa ω_i . La selección de los autovalores más pequeños λ y sus autofunciones asociadas ϕ permite capturar de manera eficiente las oscilaciones y variaciones de la solución inducidas por las heterogeneidades del medio poroso. De esta forma, se construye un espacio multiscale enriquecido que mejora significativamente el desempeño global del método [1, 8]. Recientemente, se desarrolló un trabajo derivado de esta tesis de maestría relacionado con métodos de elementos finitos y aproximaciones multiscale para ecuaciones diferenciales parciales, en el que se estudiaron el método de elementos finitos clásico, el método multiscale de elementos finitos y el método multiscale generalizado de elementos finitos. Dicho trabajo fue realizado en colaboración con Angie Daniela Forero Laiton y Yessica Vanessa Trujillo Ladino de la Universidad Nacional de Colombia, sede Bogotá, Colombia y fue aceptado para publicación en [20]. En esta tesis de maestría se estudian aspectos importantes del método de elementos finitos clásico, incluyendo su formulación teórica y su implementación computacional detallada con propósitos pedagógicos en Julia.

El enfoque principal de esta tesis de maestría se centra en los métodos multiscale, en particular en el método GMsFEM. Se analizan diferentes aspectos de este método en dos variantes: una formulación basada en malla gruesa, que será estudiada en detalle, y una formulación sin malla gruesa, que será abordada de manera más general. Para cada una de estas variantes se presentan aplicaciones representativas cuyos desarrollos completos se encuentran en trabajos previamente publicados. En particular, en [22], desarrollado por Djulustan Nikiforov, Leonardo A. Poveda, Dmitry Ammosov, Yesy Sarmiento y Juan Galvis, se estudia una formulación meshfree del método GMsFEM para problemas

parabólicos multiescala. Asimismo, como continuación de esta colaboración, en [23] se presenta una extensión tridimensional del enfoque meshfree para problemas multiescala de advección–difusión.

En el caso de la formulación con malla gruesa se presentarán aspectos de implementación y experimentos numéricos. Para ilustrar el comportamiento del método, se estudia su desempeño en un medio heterogéneo sintético generado de manera aleatoria mediante la función `rand`, definido como

$$\kappa(x_i, y_j) = 1 + 10^4 \xi_{ij}, \quad \xi_{ij} \sim \mathcal{U}(0, 1),$$

con el objetivo de analizar la sensibilidad del error con respecto al número de funciones base adicionales m . Posteriormente, el método se aplica al benchmark SPE10, capa 85, el cual constituye un estándar internacional para la validación de esquemas numéricos en simulaciones de flujo en medios porosos [7, 9].

El resto del trabajo está organizado de la siguiente manera. En el Capítulo 2 se presentan algunos fundamentos de análisis funcional y formulación variacional necesarios para el desarrollo teórico del trabajo. En el Capítulo 3 se estudia el método de elementos finitos en una dimensión, incluyendo implementaciones didácticas en Julia utilizando funciones base de grado 1 y grado 2.

En el Capítulo 4 se desarrolla el método de elementos finitos en dos dimensiones. Posteriormente, en el Capítulo 5 se introduce el uso de Gridap para la implementación computacional en Julia de un problema tridimensional.

En el Capítulo 6 se presenta el método generalizado de elementos finitos multiescala (GMsFEM) utilizando una malla gruesa estructurada, incluyendo la construcción de funciones base multiescala y la formulación variacional correspondiente. En el Capítulo 7 se presentan diversos experimentos numéricos y casos de validación.

Finalmente, en el Capítulo 8 se presentan estudios adicionales relacionados con una variante del método GMsFEM sin malla gruesa, basada en una construcción meshfree mediante vecindades locales y funciones base multiescala. Por último, se incluyen las conclusiones generales del trabajo, la bibliografía y un apéndice con algunas implementaciones

computacionales utilizadas durante el desarrollo de la tesis.

En este contexto se presentan las principales contribuciones de esta tesis:

- La formulación e implementación computacional del método GMsFEM para problemas de flujo en medios porosos heterogéneos utilizando Julia, así como el análisis numérico del comportamiento del error relativo respecto al número de funciones base multiescala adicionales por vecindad.
- La validación del método sobre medios heterogéneos sintéticos y sobre el benchmark internacional SPE10, caracterizado por altos contrastes de permeabilidad.
- El estudio comparativo entre soluciones de malla fina y aproximaciones multiescala, analizando la capacidad del método para reducir grados de libertad manteniendo una adecuada precisión numérica.
- La presentación de una variante meshfree del método GMsFEM, basada en vecindades locales y funciones base multiescala construidas a partir de nubes de puntos.

1.1. Objetivos

1.1.1. Objetivo general

Estudiar e implementar el Método Generalizado de Elementos Finitos Multiescala (GMsFEM) para la aproximación numérica de la ecuación de Darcy en medios porosos heterogéneos, analizando el comportamiento del error con respecto al enriquecimiento del espacio multiescala y validando su desempeño mediante problemas representativos con altos contrastes en el coeficiente de permeabilidad.

1.1.2. Objetivos específicos

1. Estudiar los fundamentos matemáticos del método de elementos finitos clásico, incluyendo su formulación variacional y las herramientas de análisis funcional necesarias para el desarrollo del método GMsFEM.

2. Construir las funciones base multiescala mediante la resolución de problemas espectrales locales asociados a una malla gruesa.
3. Implementar el método GMsFEM en el lenguaje de programación Julia para la resolución de la ecuación de Darcy en medios porosos heterogéneos.
4. Analizar la influencia del número de funciones base multiescala adicionales sobre la precisión de la aproximación y el costo computacional del método mediante el estudio de los errores relativos.
5. Validar el método mediante experimentos numéricos sobre un medio heterogéneo sintético y sobre el benchmark internacional SPE10.

Capítulo 2

Fundamentos

Antes de comenzar el estudio del problema principal de este trabajo, es necesario introducir algunas herramientas fundamentales del análisis funcional que permitirán establecer la formulación variacional de las ecuaciones diferenciales parciales, así como garantizar propiedades clave como existencia, unicidad y estabilidad de la solución. La presentación que sigue se basa principalmente en [5].

2.1. Espacios normados y espacios de Banach

En esta sección se introducen los conceptos básicos de espacios normados y espacios de Banach, fundamentales para el análisis funcional y el desarrollo teórico del presente trabajo.

Definición 2.1 (Espacio normado [19]). *Sea X un espacio vectorial (real o complejo). Una norma en X es una aplicación $\|\cdot\| : X \rightarrow \mathbb{R}$ tal que, para todo $x, y \in X$ y todo escalar α , se cumplen:*

$$(N1) \quad \|x\| \geq 0,$$

$$(N2) \quad \|x\| = 0 \iff x = 0,$$

$$(N3) \quad \|\alpha x\| = |\alpha| \|x\|,$$

$$(N4) \quad \|x + y\| \leq \|x\| + \|y\|.$$

El par $(X, \|\cdot\|)$ se denomina espacio normado.

Un concepto importante que se deriva del concepto anterior es el de completitud en espacios normados. Un espacio lineal normado $(X, \|\cdot\|)$ se denomina espacio de Banach si es completo respecto a la métrica inducida por la norma, es decir, si toda sucesión de Cauchy en X converge a un elemento de X , [5].

2.2. Funcionales

Con el objetivo de establecer el marco funcional adecuado para la formulación variacional de los problemas en derivadas parciales, se introduce a continuación un concepto fundamental del análisis funcional, tales como los funcionales lineales. Este concepto es esencial para el estudio de existencia y unicidad de soluciones en espacios de Sobolev [5, 18].

En este contexto, resulta natural considerar aplicaciones lineales definidas sobre estos espacios. Un funcional lineal sobre un espacio vectorial X es una aplicación

$$L : X \rightarrow \mathbb{R}$$

tal que satisface la propiedad de linealidad

$$L(u + av) = L(u) + aL(v), \quad \text{para todo } u, v \in X, \ a \in \mathbb{R}.$$

Los funcionales lineales juegan un papel central en la formulación débil de ecuaciones diferenciales, donde el término fuente de una ecuación diferencial se interpreta como un funcional actuando sobre funciones de prueba [18].

Un resultado fundamental en el análisis funcional establece la equivalencia entre continuidad y acotación de funcionales lineales en espacios normados [5].

Proposición 2.1 ([5]). *Sea L un funcional lineal definido sobre un espacio de Banach B . Entonces, L es continuo si y solo si es acotado, es decir, si existe una constante $C > 0$ tal que*

$$|L(v)| \leq C\|v\|_B, \quad \text{para todo } v \in B.$$

Demostración. Demostramos ambas implicaciones.

Supongamos primero que L es continuo. Procederemos por contradicción. Si L no fuera acotado, entonces para todo $n \in \mathbb{N}$ existiría $v_n \in B$ tal que

$$\frac{|L(v_n)|}{\|v_n\|_B} \geq n.$$

Definimos

$$w_n := \frac{v_n}{\|v_n\|_B},$$

de modo que $\|w_n\|_B = 1$ y

$$|L(w_n)| = \frac{|L(v_n)|}{\|v_n\|_B} \geq n.$$

Consideremos ahora la sucesión

$$z_n := \frac{1}{n}w_n.$$

Entonces

$$\|z_n\|_B = \frac{1}{n} \rightarrow 0,$$

lo que implica que $z_n \rightarrow 0$ en B . Por la continuidad de L , se sigue que

$$L(z_n) \rightarrow L(0) = 0.$$

Sin embargo,

$$|L(z_n)| = \frac{1}{n}|L(w_n)| \geq 1,$$

lo cual es una contradicción. Por lo tanto, L debe ser acotado.

Recíprocamente, supongamos que L es acotado. Entonces existe $C > 0$ tal que

$$|L(v)| \leq C\|v\|_B, \quad \text{para todo } v \in B.$$

Para cualesquiera $u, v \in B$, se tiene

$$|L(u) - L(v)| = |L(u - v)| \leq C\|u - v\|_B,$$

lo que muestra que L es Lipschitz continuo, y por tanto continuo.

Concluimos que L es continuo si y solo si es acotado.

□

2.3. Espacios con producto interno y espacios de Hilbert

Con el fin de caracterizar los funcionales lineales continuos en espacios de Hilbert, se introducen a continuación algunos conceptos fundamentales del análisis funcional, los cuales permiten establecer una conexión directa entre funcionales y elementos del espacio. La exposición sigue de manera estándar los textos [5, 18].

Sea X un espacio vectorial sobre $\mathbb{K}$ ($\mathbb{R}$ o $\mathbb{C}$). Un producto interno es una aplicación

$$(\cdot, \cdot) : X \times X \rightarrow \mathbb{K}$$

tal que para todo $x, y, z \in X$ y $\alpha \in \mathbb{K}$ se verifican las propiedades de linealidad en la primera variable, simetría conjugada y positividad, es decir,

- $(x + y, z) = (x, z) + (y, z)$,
- $(\alpha x, y) = \alpha(x, y)$,
- $(x, y) = \overline{(y, x)}$,
- $(x, x) \geq 0$ y $(x, x) = 0$ si y solo si $x = 0$.

Este producto interno induce de manera natural una norma en X , definida por

$$\|x\| = \sqrt{(x, x)}.$$

Un espacio de Hilbert es entonces un espacio vectorial dotado de un producto interno que es completo respecto a la norma inducida. Esta estructura es particularmente importante en el análisis variacional, ya que proporciona un marco adecuado para el estudio de problemas elípticos.

Por otra parte, si X y Y son espacios normados y $T : \mathcal{D}(T) \subset X \rightarrow Y$ es un operador lineal acotado, su norma se define como

$$\|T\| := \sup_{x \in \mathcal{D}(T), x \neq 0} \frac{\|Tx\|}{\|x\|}.$$

Este concepto permite medir el crecimiento de operadores lineales y será útil en el análisis de estabilidad de métodos numéricos.

En este contexto, resulta natural introducir el espacio dual. Dado un espacio normado X , el conjunto de todos los funcionales lineales acotados sobre X se denota por X' y se denomina el espacio dual de X . Este espacio se dota de la norma

$$\|f\| = \sup_{x \in X, x \neq 0} \frac{|f(x)|}{\|x\|} = \sup_{\|x\|=1} |f(x)|.$$

Esta definición es consistente con la noción de acotación introducida previamente y convierte a X' en un espacio normado.

2.3.1. Teorema de representación de Riesz

El siguiente resultado establece una caracterización fundamental de los funcionales lineales continuos en espacios de Hilbert.

Teorema 2.1 (Teorema de Representación de Riesz [5]). *Todo funcional lineal continuo L sobre un espacio de Hilbert H puede representarse de manera única como*

$$L(v) = (u, v), \tag{2.1}$$

para algún $u \in H$. Además, se tiene

$$\|L\|_{H'} = \|u\|_H. \tag{2.2}$$

2.4. Espacios de funciones

En esta sección se introducen los espacios funcionales que desempeñan un papel central en la formulación débil de problemas elípticos. En particular, se describen las funciones de prueba, los espacios de funciones integrables y, finalmente, los espacios de Sobolev.

Sea $u : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ una función continua. El soporte de u , denotado por $\text{supp}(u)$, se

define como el cierre del conjunto de puntos donde la función no es nula, es decir,

$$\text{supp}(u) := \overline{\{x \in \Omega : u(x) \neq 0\}}.$$

Se dice que u tiene soporte compacto en Ω si $\text{supp}(u)$ es un subconjunto compacto de Ω . Este concepto es fundamental para definir las funciones de prueba utilizadas en la formulación débil de problemas variacionales [5].

Las funciones de prueba se modelan mediante el espacio $C_0^\infty(\Omega)$, el cual consiste en funciones infinitamente diferenciables con soporte compacto contenido en Ω . Este espacio se utiliza ampliamente en el análisis de ecuaciones diferenciales parciales, ya que permite realizar integraciones por partes sin términos de frontera [5].

Por otra parte, para establecer un marco adecuado de integrabilidad, se introducen los espacios L^p . En particular, dado un conjunto medible $K \subset \mathbb{R}^n$, una función medible $f : K \rightarrow \mathbb{R}$ se dice integrable si

$$\int_K |f(x)| dx < \infty,$$

y el conjunto de todas estas funciones se denota por $L^1(K)$. Este concepto se generaliza a los espacios $L^p(\Omega)$, los cuales son fundamentales en el análisis funcional moderno [5].

En muchas aplicaciones es necesario trabajar con funciones que no son globalmente integrables, pero sí lo son en subconjuntos compactos. Esto conduce a la noción de funciones localmente integrables. Se dice que una función f pertenece a $L_{\text{loc}}^1(\Omega)$ si es integrable en todo subconjunto compacto de Ω . Este espacio es particularmente útil en la definición de derivadas débiles y en el estudio de soluciones generalizadas [5].

2.5. Espacios de Sobolev

A continuación se introduce el concepto central de esta sección.

Definición 2.2 (Espacios de Sobolev [5]). *Sea $\Omega \subset \mathbb{R}^d$ un conjunto abierto, $k \geq 1$ un*

entero positivo y $p \in [1, +\infty]$. Se define

$$W^{k,p}(\Omega) = \{f \in L^p(\Omega) : D^\alpha f \text{ existe en sentido débil y pertenece a } L^p(\Omega), \forall |\alpha| \leq k\}.$$

Para $1 \leq p < \infty$, la norma en $W^{k,p}(\Omega)$ está dada por

$$\|f\|_{k,p} = \left(\sum_{|\alpha| \leq k} \|D^\alpha f\|_p^p \right)^{1/p},$$

mientras que para $p = \infty$,

$$\|f\|_{k,\infty} = \max_{|\alpha| \leq k} \|D^\alpha f\|_\infty.$$

La seminorma asociada se define como

$$|f|_{k,p} = \left(\sum_{|\alpha|=k} \|D^\alpha f\|_p^p \right)^{1/p}, \quad 1 \leq p < \infty.$$

Los espacios de Sobolev permiten formular problemas en derivadas parciales en un sentido débil, lo cual es esencial para el desarrollo del método de elementos finitos. En particular, el caso $p = 2$ es de especial importancia, ya que el espacio $W^{k,2}(\Omega)$, denotado por $H^k(\Omega)$, posee estructura de espacio de Hilbert. En este caso, la norma proviene de un producto interno definido por

$$(u, v)_{H^k(\Omega)} = \sum_{|\alpha| \leq k} (D^\alpha u, D^\alpha v)_{L^2(\Omega)},$$

lo que induce la norma

$$\|u\|_{H^k(\Omega)} = \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|_{L^2(\Omega)}^2 \right)^{1/2}.$$

Esta estructura hilbertiana es fundamental en el análisis variacional, ya que permite aplicar herramientas como el teorema de Lax-Milgram para garantizar la existencia y unicidad de soluciones en formulaciones débiles [5, 18].

2.6. Formas bilineales

En esta sección se introducen las formas bilineales, las cuales serán utilizadas en la formulación variacional de los problemas considerados.

Definición 2.3 (Forma bilineal [2]). *Sea X un espacio vectorial sobre $\mathbb{R}$, una forma bilineal es una aplicación $a : X \times X \rightarrow \mathbb{R}$ que satisface lo siguiente:*

- $a(u_1 + u_2, v) = a(u_1, v) + a(u_2, v).$
- $a(u, v_1 + v_2) = a(u, v_1) + a(u, v_2).$
- $a(\alpha u, v) = \alpha a(u, v).$
- $a(u, \alpha v) = \alpha a(u, v).$

para todo $\alpha \in \mathbb{R}$, $u, v, u_1, u_2, v_1, v_2 \in X$.

A continuación se presenta un ejemplo de una forma bilineal.

Ejemplo 2.1. *El producto escalar de un espacio euclideo define una forma bilineal. Por ejemplo en $\mathbb{R}^2$, si consideramos dos vectores $u = (a, b)$ y $v = (c, d)$, su producto escalar está dado por $\langle u, v \rangle = ac + bd$, donde dicha forma es una forma bilineal.*

Definición 2.4 (Forma bilineal simétrica). *Una forma bilineal simétrica es aquella que cumple $a(u, v) = a(v, u)$ para todo $u, v \in V$.*

Se muestra ahora la forma de la matriz asociada a una forma bilineal.

Definición 2.5 (Matriz asociada [2]). *Toda forma bilineal en un espacio finito dimensional se puede expresar de manera matricial. Sea $\mathbb{R}$ un cuerpo y X un espacio vectorial, consideremos $a : X \times X \rightarrow \mathbb{R}$ una forma bilineal y considérese una base $B = \{e_1, e_2, \dots, e_n\}$ del espacio vectorial X de dimensión finita, se define la matriz asociada a la forma a respecto de la base B como:*

$$A = \begin{pmatrix} a(e_1, e_1) & a(e_1, e_2) & \cdots & a(e_1, e_n) \\ a(e_2, e_1) & a(e_2, e_2) & \cdots & a(e_2, e_n) \\ \vdots & \vdots & \ddots & \vdots \\ a(e_n, e_1) & a(e_n, e_2) & \cdots & a(e_n, e_n) \end{pmatrix}.$$

Proposición 2.2 ([2]). Sea $a : X \times X \rightarrow \mathbb{R}$ una forma bilineal y una base $B = \{e_1, e_2, \dots, e_n\}$ del espacio vectorial X , la imagen de a de los vectores u con coordenadas $(u_1, u_2, \dots, u_n)$ y v con coordenadas $(v_1, v_2, \dots, v_n)$ en la base B está dada por

$$a(u, v) = u^t \cdot A \cdot v = \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix} \cdot \begin{pmatrix} a(e_1, e_1) & a(e_1, e_2) & \dots & a(e_1, e_n) \\ a(e_2, e_1) & a(e_2, e_2) & \dots & a(e_2, e_n) \\ \vdots & \vdots & \ddots & \vdots \\ a(e_n, e_1) & a(e_n, e_2) & \dots & a(e_n, e_n) \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}.$$

Algunas consecuencias de la proposición anterior son las siguientes

- Una forma bilineal es simétrica si y solo si su matriz en una base cualquiera es simétrica.
- Si se fija la base B y la forma bilineal a , entonces la matriz A que hace $a(u, v) = u^t \cdot A \cdot v$ para todo u, v es única. Por último enunciamos el siguiente teorema.

Teorema 2.2 (Teorema Espectral Real). Supongamos que V es un espacio de producto interno real de dimensión finita y $T \in L(V)$. Entonces, V tiene una base ortonormal compuesta por autovectores de T si y sólo si T es autoadjunto.

La demostración de este hecho la puede encontrar en [2].

2.6.1. Formulaci3n variacional

La formulaci3n variacional constituye un marco fundamental en el estudio de ecuaciones diferenciales parciales, ya que permite reformular problemas en derivadas parciales en t3rminos de formas bilineales y funcionales lineales. En este contexto, la soluci3n del problema se busca en un espacio de Hilbert V , donde la ecuaci3n diferencial original se expresa en su forma d3bil mediante una relaci3n del tipo

$$a(u, v) = L(v), \text{ para todo } v \in V.$$

Este enfoque resulta especialmente útil en el análisis teórico y en el desarrollo de métodos numéricos, como el método de elementos finitos, al facilitar el manejo de condiciones de frontera y garantizar propiedades de existencia y unicidad bajo hipótesis apropiadas sobre la forma bilineal y el funcional lineal.

Definición 2.6. (*[5]*) Sea V un espacio de Hilbert con producto escalar $(\cdot, \cdot)$ y la correspondiente norma $\|\cdot\|_V$ (la norma V). Supongamos que $a(\cdot, \cdot)$ es una forma bilineal en $V \times V$ y L una forma lineal en V tal que:

1. $a(\cdot, \cdot)$ es simétrica,

2. $a(\cdot, \cdot)$ es continua, es decir, existe una constante $\gamma > 0$ tal que

$$|a(v, w)| \leq \gamma \|v\|_V \|w\|_V \quad \text{para todo } v, w \in V,$$

3. $a(\cdot, \cdot)$ es V -elíptica, es decir, existe una constante $\alpha > 0$ tal que

$$a(v, v) \geq \alpha \|v\|_V^2 \quad \text{para todo } v \in V.$$

4. L es continua, es decir, existe una constante $\Lambda > 0$ tal que

$$|L(v)| \leq \Lambda \|v\|_V \quad \text{para todo } v \in V.$$

2.6.2. Principio de Rayleigh-Ritz

El principio de Rayleigh–Ritz proporciona una formulación variacional del problema, en la cual la solución se caracteriza como el minimizador de un funcional de energía definido sobre un espacio de Hilbert. Bajo condiciones apropiadas, este enfoque es equivalente a la formulación variacional clásica y resulta útil para analizar y aproximar numéricamente soluciones de ecuaciones diferenciales parciales.

Definición 2.7 (Método de Rayleigh-Ritz [5]). Sea V un espacio de Hilbert real y sea $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ una forma bilineal simétrica, continua y coerciva. Sea $L \in V'$ donde

V' es el espacio dual de V .

Definimos el funcional $J : V \rightarrow \mathbb{R}$ por

$$J(v) = \frac{1}{2}a(v, v) - L(v), \text{ para todo } v \in V.$$

El principio de Rayleigh–Ritz consiste en encontrar $u \in V$ tal que

$$J(u) = \min_{v \in V} J(v).$$

A partir del principio de Rayleigh–Ritz, se introduce el principio de Galerkin como una formulación variacional equivalente bajo condiciones apropiadas.

2.6.3. Principio de Galerkin

El principio de Galerkin establece que la solución del problema variacional se caracteriza mediante una relación de ortogonalidad respecto al espacio de prueba. Este enfoque constituye la base de numerosos métodos numéricos, en particular del método de elementos finitos.

Definición 2.8 (Principio de Galerkin [5]). *Sea V un espacio de Hilbert real, sea $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ una forma bilineal continua y sea $L \in V'$ el dual de V . El problema variacional consiste en encontrar $u \in V$ tal que*

$$a(u, v) = L(v), \text{ para todo } v \in V.$$

En el caso en que la forma bilineal $a(\cdot, \cdot)$ es simétrica y definida positiva, se define la norma de energía en V por

$$\|v\|_a = \sqrt{a(v, v)}, \text{ para todo } v \in V.$$

Obsérvese que la expresión anterior define estrictamente una seminorma en espacios generales como $H^1(\Omega)$, ya que funciones constantes poseen gradiente nulo. Sin embargo, cuando se trabaja sobre el espacio $H_0^1(\Omega)$, dicha expresión constituye efectivamente una norma, debido a la desigualdad de Poincaré. En esta tesis se utilizará el término norma

de energía en aquellos contextos donde el espacio funcional considerado garantice esta propiedad. Además, si $a(\cdot, \cdot)$ es continua y coerciva (elíptica), entonces la norma de energía $\|\cdot\|_a$ es equivalente a la norma $\|\cdot\|_V$, es decir, existen constantes positivas c y C tales que

$$c\|v\|_V \leq \|v\|_a \leq C\|v\|_V, v \in V.$$

Cuando la forma bilineal $a(\cdot, \cdot)$ es simétrica, continua y V -elíptica, la solución aproximada obtenida mediante el método de Galerkin coincide con la solución que se obtiene minimizando un funcional de energía, es decir, mediante el método de Rayleigh-Ritz.

En este contexto, el principio de Rayleigh–Ritz permite reformular el problema de valores en la frontera como un problema variacional de minimización.

Las siguientes condiciones se imponen sobre la forma bilineal $a(\cdot, \cdot)$ y el funcional lineal L con el fin de garantizar que el problema variacional esté bien planteado.

Ahora si se considera el siguiente problema de minimización: Encontrar $u \in V$ tal que

$$J(u) = \min_{v \in V} J(v), \tag{2.3}$$

donde

$$J(v) = \frac{1}{2}a(v, v) - L(v),$$

y considérese también el siguiente problema variacional: Encontrar $u \in V$ tal que

$$a(u, v) = L(v) \quad \text{para todo } v \in V. \tag{2.4}$$

2.6.4. Equivalencia entre el problema variacional y el problema de minimización

Ahora, el siguiente teorema es crucial ya que establece la equivalencia entre (2.3) y (2.4).

Teorema 2.3. (*[18]*) *Los problemas (2.3) y (2.4) son equivalentes, es decir, $u \in V$ satisface (2.3) si y solo si satisface (2.4). Además, existe una solución única $u \in V$ de*

estos problemas y la siguiente estimación de estabilidad se mantiene

$$\|u\|_V \leq \frac{\Lambda}{\alpha}.$$

Demostración. Sea $v \in V$ y $\epsilon \in \mathbb{R}$ arbitrarios. Entonces

$$J(u) \leq J(u + \epsilon v) \quad \text{para todo } v \in V.$$

Usando la notación $g(\epsilon) = J(u + \epsilon v)$, $\epsilon \in \mathbb{R}$, tenemos

$$g(0) \leq g(\epsilon) \quad \text{para todo } \epsilon \in \mathbb{R},$$

de modo que g tiene un mínimo en $\epsilon = 0$. Por lo tanto $g'(0) = 0$ si la derivada $g'(\epsilon)$ existe en $\epsilon = 0$. Luego,

$$\begin{aligned} g(\epsilon) &= \frac{1}{2}a(u + \epsilon v, u + \epsilon v) - L(u + \epsilon v) \\ &= \frac{1}{2}a(u, u) + \epsilon a(u, v) + \frac{\epsilon^2}{2}a(v, v) - L(u) - \epsilon L(v) \\ &= \frac{1}{2}a(u, u) - L(u) + \epsilon[a(u, v) - L(v)] + \frac{\epsilon^2}{2}a(v, v), \end{aligned}$$

si usamos el hecho de que $a(\cdot, \cdot)$ es simétrica. Se deduce que

$$0 = g'(0) = a(u, v) - L(v).$$

Por tanto

$$L(v) = a(u, v).$$

Ahora probamos la implicación inversa: si $a(u, v) = L(v)$ para todo $v \in V$, entonces u minimiza $J(v)$. Recordemos que

$$J(v) = \frac{1}{2}a(v, v) - L(v).$$

Queremos mostrar que

$$J(u) \leq J(v) \quad \text{para todo } v \in V.$$

Consideremos la función $g(\epsilon) = J(u + \epsilon v)$ para $\epsilon \in \mathbb{R}$. Expandimos $g(\epsilon)$:

$$g(\epsilon) = J(u + \epsilon v) = \frac{1}{2}a(u + \epsilon v, u + \epsilon v) - L(u + \epsilon v).$$

Usando la bilinealidad de $a(\cdot, \cdot)$ y la linealidad de L , tenemos

$$g(\epsilon) = \frac{1}{2}a(u, u) + \epsilon a(u, v) + \frac{\epsilon^2}{2}a(v, v) - L(u) - \epsilon L(v).$$

Dado que $a(u, v) = L(v)$, podemos sustituir $L(v)$ en la expresión:

$$g(\epsilon) = \frac{1}{2}a(u, u) + \epsilon L(v) + \frac{\epsilon^2}{2}a(v, v) - L(u) - \epsilon L(v).$$

Observamos que los términos lineales en ϵ se cancelan, lo que resulta en

$$g(\epsilon) = \frac{1}{2}a(u, u) - L(u) + \frac{\epsilon^2}{2}a(v, v).$$

Luego, dado que $\epsilon^2 a(v, v) \geq 0$ para todo $\epsilon \in \mathbb{R}$ y $a(v, v) \geq 0$, se deduce que

$$g(\epsilon) \geq g(0) \quad \text{para todo } \epsilon \in \mathbb{R},$$

lo que implica que $g(0)$ es un mínimo. Como $g(0) = J(u)$ y $g(\epsilon) = J(u + \epsilon v)$, esto significa que

$$J(u) \leq J(u + \epsilon v) \quad \text{para todo } \epsilon \in \mathbb{R}.$$

En particular, para $\epsilon = 1$, tenemos

$$J(u) \leq J(u + v) \quad \text{para todo } v \in V.$$

Por lo tanto, u minimiza $J(v)$ sobre V , completando así la prueba. □

2.6.5. Teorema de Lax-Milgram

El teorema de Lax-Milgram proporciona condiciones suficientes para garantizar la existencia y unicidad de la solución del problema variacional.

Teorema 2.4 (Teorema de Lax-Milgram [5]). *Suponga que $a(.,.)$ es V -elíptica y acotada y que $L(v)$ también es acotada en $H_0^1(a, b)$. Entonces, el problema débil:*

Encontrar $u \in H_E^1(a, b)$ tal que

$$a(u, v) = L(v),$$

para toda función $v \in H_0^1(a, b)$ tiene solución única.

Capítulo 3

Método de elementos finitos en una dimensión

Para obtener la solución de elementos finitos de una ecuación diferencial se siguen los siguientes pasos:

1. Partimos de la formulación fuerte del problema, es decir, la ecuación diferencial parcial con condiciones de frontera y condición inicial en una geometría en la cual se quiere resolver.
2. Se construye la formulación débil la cual consiste en llevar a un problema integral nuestra ecuación diferencial parcial.
3. Realizamos la formulación de Galerkin del paso anterior, aquí se busca la solución en un espacio de dimensión finita.
4. Planteamos la formulación ahora como un sistema de ecuaciones.
5. Resolvemos el sistema de ecuaciones ya sea por método directo o métodos iterativos, esto va a depender del tipo de matriz de rigidez que se obtenga.

A continuación se aplicará cada uno de los pasos antes mencionados para encontrar la solución de la ecuación de la banda elástica de una dimensión espacial.

Formulación fuerte

Considere una cuerda elástica sujeta en los extremos a y b con carga f . Si ocurren algunos desplazamientos dados por $x \in [a, b]$, dicho desplazamiento satisface

$$\left\{ \begin{array}{l} -(\kappa(x)u'(x))' = f(x) \quad \text{para todo } x \in (a, b) \\ u(a) = 0, \quad u(b) = 0, \end{array} \right. \quad (3.1)$$

donde $\Omega = [a, b] \subset \mathbb{R}$ con condiciones de frontera tipo Dirichlet.

Formulación débil

Dada la formulación fuerte, construiremos la formulación débil realizando los siguientes pasos:

1. Multiplicar ambos lados de (3.1) por una función de prueba

$$v \in C_0^\infty = \{v \in C^\infty(a, b) \mid v(a) = 0, v(b) = 0\} \text{ con}$$

$$C^\infty = \{v : [a, b] \rightarrow \mathbb{R} \mid v \text{ y todas sus derivadas son continuas}\}.$$

Sea $v \in C_0^\infty$

$$-(\kappa(x)u'(x))'v(x) = f(x)v(x)$$

2. Integrar a ambos lados de la igualdad obtenida en el paso anterior.

$$-\int_a^b (\kappa(x)u'(x))'v(x) = \int_a^b f(x)v(x)$$

Ahora aplicamos integración por partes para reducir el orden de la ecuación

$$-(\kappa(x)u'(x))v(x)|_a^b - \int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x)$$

$$-[\kappa(a)u'(a)v(a) - \kappa(b)u'(b)v(b)] + \int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x)$$

$$-[\kappa(a)u'(a)v(a) - \kappa(b)u'(b)v(b)] + \int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x)$$

$$\int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x).$$

Por tanto la forma débil es:

Encontrar u tal que

$$\begin{cases} \int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x) \text{ para todo } v \in C_0^\infty(a, b) \\ u(a) = u(b) = 0 \end{cases} \quad (3.2)$$

Nota 1) Si u resuelve el problema fuerte entonces resuelve el problema débil.

2) Si u resuelve el problema débil no necesariamente resuelve el problema fuerte ya que la segunda derivada puede no existir.

3. Cambiar el espacio de la función de prueba $v \in C_0^\infty$ por V tal que $v \in V$.

Escoja una base $\{\varphi_1, \varphi_2, \dots, \varphi_N\}$ y genere el espacio

$$V = \text{span} \{\varphi_1, \varphi_2, \dots, \varphi_N\} = \{\alpha_1\varphi_1 + \alpha_2\varphi_2 + \dots + \alpha_N\varphi_N \mid \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_N \in \mathbb{R}\}.$$

Por lo que se ha llegado a la siguiente formulación de Galerkin:

Encontrar $u \in V$ tal que

$$\begin{cases} \int_a^b \kappa(x)u'(x)v'(x) = \int_a^b f(x)v(x) \text{ para todo } v \in V \\ u(a) = 0, u(b) = 0. \end{cases}$$

Por último construimos la formulación como un sistema de ecuaciones, para ello note

como $u \in V$, es de la forma $u = x_1\varphi_1 + x_2\varphi_2 + \dots + x_N\varphi_N$

y usando φ_i , con $i = 1, 2, \dots, N$ como funciones de prueba.

Ahora, sustituyendo u y v en la formulación de Galerkin

$$\int_a^b \kappa(x_1\varphi_1 + x_2\varphi_2 + \dots + x_N\varphi_N)' \varphi_i' = \int_a^b f \varphi_i, \quad i = 1, 2, \dots, N$$

$$x_1 \int_a^b \kappa \varphi_1' \varphi_i' + x_2 \int_a^b \kappa \varphi_2' \varphi_i' + \dots + x_N \int_a^b \kappa \varphi_N' \varphi_i' = \int_a^b f \varphi_i, \quad i = 1, 2, \dots, N$$

tomando

$$a_{i1} = \int_a^b \kappa \varphi_1' \varphi_i', \quad a_{i2} = \int_a^b \kappa \varphi_2' \varphi_i', \quad a_{iN} = \int_a^b \kappa \varphi_N' \varphi_i' \text{ y } b_i = \int_a^b f \varphi_i$$

$$\sum_{j=1}^N a_{ij}x_j = b_j, \quad i = 1, 2, \dots, N$$

Si $A = [a_{ij}]$ y $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$ entonces $Ax = b$.

Por lo que, hemos llegado a la formulación como un sistema de ecuaciones y ahora nuestro problema es:

Encontrar $x_1, x_2, \dots, x_N$ tal que $u_h = \sum x_i \varphi_i$ y

$$\begin{cases} Ax = b \\ a_{ij} = \int_a^b \kappa \varphi'_i \varphi'_j, b_i = \int_a^b f \varphi_i, u(a) = 0, u(b) = 0. \end{cases} \quad \text{La implementación para ele-}$$

mentos finitos lineales (usando funciones base lineales), las funciones base $\phi_i(x)$ y sus derivadas son fundamentales para construir las matrices de rigidez y los vectores de fuerza. Para un elemento finito lineal típico, las funciones base son:

$$\phi_1(x) = \frac{x_1 - x}{x_1 - x_0}, \quad \phi_2(x) = \frac{x - x_0}{x_1 - x_0}$$

y sus derivadas con respecto a x son:

$$\frac{\partial \phi_1}{\partial x} = -\frac{1}{x_1 - x_0}, \quad \frac{\partial \phi_2}{\partial x} = \frac{1}{x_1 - x_0}.$$

La matriz de rigidez local A_{loc} se calcula mediante la cuadratura de Gauss como sigue:

$$A_{\text{loc}}^{ij} = \int_{x_0}^{x_1} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} dx \approx \sum_{k=1}^n w_k \frac{d\phi_i}{dx}(\zeta_k) \frac{d\phi_j}{dx}(\zeta_k) |J|$$

donde w_k son los pesos de cuadratura, ζ_k son los puntos de cuadratura, y $|J|$ es el determinante del jacobiano de la transformación del intervalo de integración de $[-1, 1]$ a $[x_0, x_1]$. Para un elemento finito lineal, este determinante simplemente corresponde al tamaño del elemento dividido por 2, es decir, $|J| = \frac{x_1 - x_0}{2}$.

El vector de fuerza local $\mathbf{f}_{\text{loc}}$ se calcula utilizando la cuadratura de Gauss de la

siguiente manera:

$$f_{\text{loc},i} = \int_{x_0}^{x_1} f(x)\phi_i(x) dx \approx \sum_{k=1}^n w_k f(\zeta_k)\phi_i(\zeta_k)|J|$$

se aplica un enfoque similar al utilizado para la matriz de rigidez local.

Una vez calculadas las matrices de rigidez local $\mathbf{A}_{\text{loc}}$ y los vectores de fuerza local $\mathbf{f}_{\text{loc}}$ para todos los elementos, estos se ensamblan en la matriz de rigidez global $\mathbf{A}$ y el vector de fuerza global $\mathbf{b}$, respectivamente. Este proceso consiste en ir sumando a las entradas apropiadas de $\mathbf{A}$ y $\mathbf{b}$ la contribución de cada elemento, según los grados de libertad globales a los que esté conectado. El proceso de ensamblaje se puede expresar matemáticamente de la siguiente manera: Para cada elemento e , con grados de libertad globales dof_e , las entradas de la matriz de rigidez local $\mathbf{A}_{\text{loc}}^{(e)}$ y del vector de fuerza local $\mathbf{f}_{\text{loc}}^{(e)}$ se suman a la matriz de rigidez global $\mathbf{A}$ y al vector de fuerza global $\mathbf{b}$ de la manera siguiente:

$$\mathbf{A}[dof_e, dof_e] + = \mathbf{A}_{\text{loc}}^{(e)}$$

$$\mathbf{b}[dof_e] + = \mathbf{f}_{\text{loc}}^{(e)}$$

donde dof_e son los índices en $\mathbf{A}$ y $\mathbf{b}$ asociados a los grados de libertad del elemento e . Este proceso garantiza que la influencia de cada parte en el comportamiento del todo se vea reflejada.

Finalmente, con la matriz de rigidez global $\mathbf{A}$ y el vector de fuerza global $\mathbf{b}$ totalmente ensamblados, se resuelve el sistema de ecuaciones lineal:

$$\mathbf{Ax} = \mathbf{b}$$

donde $\mathbf{x}$ es el vector de desplazamientos nodales desconocidos en todo el dominio. La solución de este sistema nos da una aproximación numérica de los desplazamientos causados por las fuerzas aplicadas, resolviendo de esta manera el problema de elementos finitos.

3.1. Implementación didáctica en Julia para funciones base de grado 1

El primer paso para aplicar el Método de Elementos Finitos con funciones base de grado 1 es establecer el dominio y crear una malla de elementos finitos. En este caso tomamos un dominio unidimensional, discretizado en $N = 10$ elementos finitos. Los nodos de la malla se distribuyen de manera uniforme sobre el dominio. Los siguientes códigos en Julia ilustran cómo se define el dominio y se crea la malla:

Código: Cálculo de pesos de cuadratura

```
using StructArrays
using Plots
w_r = [5/9, 8/9, 5/9]
x_r = [-sqrt(3/5), 0.0, sqrt(3/5)]
```

Código: Estructura de datos para un elemento finito

```
mutable struct Element
    x0
    x1
    ω
    ζ
    φ1
    φ2
    ∂φ1
    ∂φ2
    dof
    Aloc
    floc
end
```

Código: Usamos N para el número de elementos.

```
N=10;
```

Código: Definimos la triangulación como un array de elementos.

```
Th=Array{Element}(undef,N);
```

Código: Calculamos la información necesaria para cada elemento.

```
for i=1:N
    x0=(i-1)/N
    x1=i/N
    h=x1-x0
     $\omega=0.5*(x1-x0)*\omega_r$ 
     $\zeta=[x0,x0,x0]+0.5*([1,1,1]+\zeta_r)*(h)$ 
     $\varphi1=([x1,x1,x1]-\zeta)/h$ 
    println( $\varphi1$ )
     $\varphi2=(\zeta-[x0,x0,x0])/h$ 
     $\partial\varphi1=[-1,-1,-1]/h$ 
     $\partial\varphi2=[1,1,1]/h$ 
    dof=[i,i+1]
    Th[i]=Element(x0,x1, $\omega,\zeta,\varphi1,\varphi2,\partial\varphi1,\partial\varphi2,dof,[],[]$ )
end
```

Una vez establecida la malla, el siguiente paso es ensamblar las matrices de rigidez y el vector de carga. Esta matriz y este vector son el sistema de ecuaciones discretizado que hay que resolver. En cada elemento se establece una matriz de rigidez local y un vector de carga local, los cuales luego se ensamblan en matrices y vectores globales que representan todo el sistema. A continuación, te muestro cómo hacerlo en Julia:

Código: Ahora calculamos las matrices locales.

```
for i=1:N
     $\varphi1=Th[i].\varphi1$ 
     $\varphi2=Th[i].\varphi2$ 
     $\partial\varphi1=Th[i].\partial\varphi1$ 
```

```

     $\partial\varphi_2$ =Th[i]. $\partial\varphi_2$ 
     $\zeta$ =Th[i]. $\zeta$ 
     $\omega$ =Th[i]. $\omega$ 
    coef= [1,1,1]+0*sin. ( $\zeta$ )
    a11= sum(coef.* $\omega$ .* $\partial\varphi_1$ .* $\partial\varphi_1$ )
    a12= sum(coef.* $\omega$ .* $\partial\varphi_1$ .* $\partial\varphi_2$ )
    a21= sum(coef.* $\omega$ .* $\partial\varphi_2$ .* $\partial\varphi_1$ )
    a22= sum(coef.* $\omega$ .* $\partial\varphi_2$ .* $\partial\varphi_2$ )

    Aloc=[a11 a12; a21 a22]
    Th[i].Aloc=Aloc

    if i == 1
        println("Matriz local del primer elemento (Aloc):")
        println(Th[i].Aloc)
    end

    if i == 2
        println("Matriz local del primer elemento (Aloc):")
        println(Th[i].Aloc)
    end

    #f=sin. ( $\zeta$ );
    f=[1,1,1]
    f1=sum( $\omega$ .*f.* $\varphi_1$ )
    f2=sum( $\omega$ .*f.* $\varphi_2$ )

    Th[i].floc=[f1,f2]

end

```

Código: Se ensamblan las matrices globales.

```
A=zeros(N+1,N+1);
```

```
b=zeros(N+1,1);
```

```
for k=1:N
    Aloc=Th[k].Aloc
    floc=Th[k].floc
    dof=Th[k].dof
    A[dof,dof]=A[dof,dof]+Aloc
    b[dof]= b[dof]+floc
    println(A[dof,dof])
    println(dof)
end
```

Posteriormente, se imponen las condiciones de frontera de Dirichlet. En este caso, consideramos condiciones de frontera de Dirichlet en los extremos del dominio. El siguiente fragmento de código muestra cómo se imponen estas condiciones en Julia:

Código: Cálculos relacionados al dato Dirichlet.

```
begin
    x_d=zeros(N+1,1)
    x_d[1]=0
    x_d[N+1]=1
    b_d=b-A*x_d
    println(b_d)
end;
```

Finalmente, se obtiene la solución aproximada resolviendo el sistema lineal de ecuaciones discretizado para obtener la solución aproximada en los nodos del dominio.

Código: Resolvemos el sistema lineal.

```
begin
    int=2:N
    x_sol=x_d;
    A_int=A[int,int]
    b_int=b_d[int]
```

```

    x_aux=A_int\b_int

    x_sol[int]=x_aux

    println(x_sol)

    println(size(x_sol))

end;

```

Código: Se ensambla la solución calculada.

```

t=0:(1/N):1

```

Visualización de la solución aproximada.

```

plot(t,x_sol,lw=5, title="sol. calculada", label="u_h(x)",
      linewidth=4, linecolor="red")

```

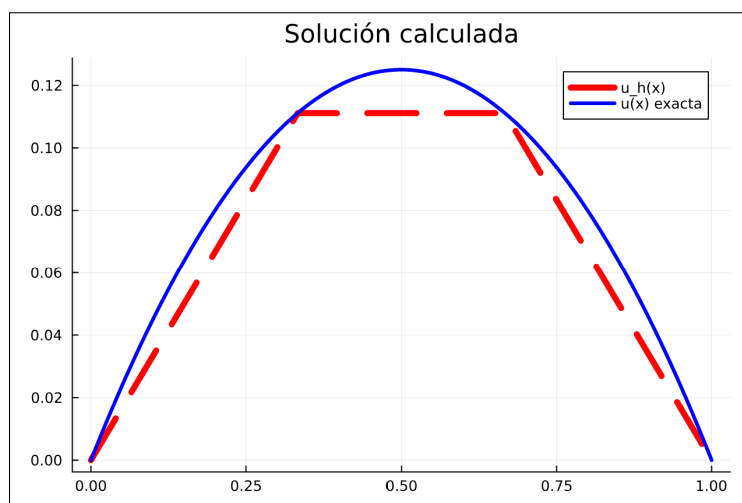


Figura 3.1: Comparación solución exacta y solución de elementos finitos en 3 elementos.

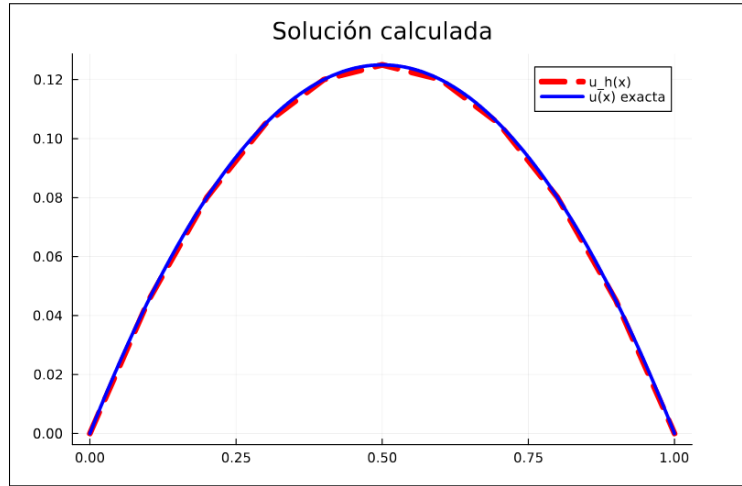


Figura 3.2: Comparación solución exacta y solución de elementos finitos en 10 elementos.

La solución aproximada se representa mediante una interpolación lineal por partes entre los nodos de la discretización.

A continuación se extiende la implementación al caso cuadrático. (usando funciones base de grado 2), es similar a la anterior. Las funciones base $\phi(x)$ y sus derivadas son fundamentales para construir las matrices de rigidez y los vectores de fuerza. Para un elemento finito cuadrático típico, las funciones base son:

$$\phi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}, \quad \phi_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}, \quad \phi_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

y sus derivadas con respecto a x son:

$$\frac{d\phi_1}{dx} = \frac{(x - x_3) + (x - x_2)}{(x_1 - x_2)(x_1 - x_3)}, \quad \frac{d\phi_2}{dx} = \frac{(x - x_3) + (x - x_1)}{(x_2 - x_1)(x_2 - x_3)}, \quad \frac{d\phi_3}{dx} = \frac{(x - x_2) + (x - x_1)}{(x_3 - x_1)(x_3 - x_2)}$$

Se sigue el mismo proceso anterior para las matrices y vectores locales, para el ensamblaje lo único que cambia es la definición de los grados de libertad dof , que ya lo veremos en la implementación en Julia.

3.2. Implementación didáctica en Julia para funciones base de grado 2

Para resolver por FEM, discretizamos el dominio en elementos finitos. Aquí se usa una malla uniforme de elementos cuadrados, la cuadratura gaussiana de 3 puntos, la cual nos da los pesos y puntos necesarios para calcular las integrales más adelante, y luego se crea una estructura `Element` para los elementos finitos.

Código: Declaración de paquetes

```
using StructArrays
using Plots
```

Código: Definición de pesos y puntos de cuadratura para la cuadratura Gaussiana de 3 puntos

```
ω_r = [0.555556, 0.888889, 0.555556]
ζ_r = [-0.774597, 0.0, 0.774597]
```

La siguiente estructura contiene las coordenadas de los nodos, los pesos y puntos de cuadratura, las funciones de forma y sus derivadas, y otras propiedades necesarias, también establecemos el número de elementos y generamos la malla como un arreglo de elementos `Element`.

Código: Definición de la estructura de datos para representar elementos

```
mutable struct Element
    x0
    x1
    x2
    ω
    ζ
    φ1
    φ2
    φ3
end
```

```

     $\partial\varphi_1$ 

     $\partial\varphi_2$ 

     $\partial\varphi_3$ 

    dof

    Aloc

    floc

end

```

Código: Número de elementos y definición de la triangulación como un arreglo de elementos

```

N = 3

Th = Array{Element}(undef, N);

```

En el siguiente código se calcula la información requerida para cada elemento, calculando las coordenadas de los nodos, puntos de cuadratura y funciones de forma y sus derivadas. También definimos los grados de libertad dof para cada elemento.

Código: Cálculo de la información necesaria para cada elemento

```

for i = 1:N

    x0 =(i-1)/ N

    x1=(2*i-1)/(2*N)

    x2 = i/N

    l1=x1-x0

    l2=x2-x1

    l3=x2-x0

     $\omega$  = 0.5 * l3*  $\omega_r$ 

     $\zeta$ =[x0,x0,x0]+ 0.5*([1,1,1]+ $\zeta_r$ ).*l3)

     $\varphi_1$  = ([x1,x1,x1].- $\zeta$ ) .* ([x2,x2,x2].- $\zeta$ ) ./ (l1*l3)

     $\varphi_2$  = ( $\zeta$  .- [x0,x0,x0]) .* ([x2,x2,x2].- $\zeta$ ) ./(l1*l2)

     $\varphi_3$  = ( $\zeta$  .- [x0,x0,x0]) .* ( $\zeta$  .- [x1,x1,x1]) ./ (l3*l2)

     $\partial\varphi_1$  = (-[x1,x1,x1] .- [x2,x2,x2]+2* $\zeta$ ) ./ (l1*l3)

     $\partial\varphi_2$  = (-2 *  $\zeta$  .+ [x0,x0,x0] .+ [x2,x2,x2]) ./(l1*l2)

     $\partial\varphi_3$  = (2 *  $\zeta$  .- [x0,x0,x0] .- [x1,x1,x1]) ./(l3*l2)

```

```
dof = [2*i-1, 2*i, 2*i+1]

Th[i] = Element(x0, x1,x2,  $\omega$ ,  $\zeta$ ,  $\varphi_1$ ,  $\varphi_2$ ,  $\varphi_3$ ,  $\partial\varphi_1$ ,  $\partial\varphi_2$ ,  $\partial\varphi_3$ , dof,
[], [])

println(dof)

end
```

Para cada elemento, calculamos las matrices locales utilizando las funciones de forma y sus derivadas.

Código: Cálculo de las matrices locales

```
for i = 1:N

     $\varphi_1$  = Th[i]. $\varphi_1$ 
     $\varphi_2$  = Th[i]. $\varphi_2$ 
     $\varphi_3$  = Th[i]. $\varphi_3$ 
     $\partial\varphi_1$  = Th[i]. $\partial\varphi_1$ 
     $\partial\varphi_2$  = Th[i]. $\partial\varphi_2$ 
     $\partial\varphi_3$  = Th[i]. $\partial\varphi_3$ 
     $\zeta$  = Th[i]. $\zeta$ 
     $\omega$  = Th[i]. $\omega$ 

    coef = [1, 1, 1] .+ 0 .* sin.( $\zeta$ )

    a11 = sum(coef .*  $\omega$  .*  $\partial\varphi_1$  .*  $\partial\varphi_1$ )
    a12 = sum(coef .*  $\omega$  .*  $\partial\varphi_1$  .*  $\partial\varphi_2$ )
    a13 = sum(coef .*  $\omega$  .*  $\partial\varphi_1$  .*  $\partial\varphi_3$ )
    a22 = sum(coef .*  $\omega$  .*  $\partial\varphi_2$  .*  $\partial\varphi_2$ )
    a23 = sum(coef .*  $\omega$  .*  $\partial\varphi_2$  .*  $\partial\varphi_3$ )
    a33 = sum(coef .*  $\omega$  .*  $\partial\varphi_3$  .*  $\partial\varphi_3$ )

    Aloc = [a11 a12 a13; a12 a22 a23; a13 a23 a33]

    Th[i].Aloc = Aloc

    println("Matriz local Aloc para el elemento $i:")
    println(Aloc)

f = [1, 1, 1]
```

```
f1 = sum(w .* f .* phi1)
f2 = sum(w .* f .* phi2)
f3 = sum(w .* f .* phi3)
Th[i].floc = [f1, f2, f3]
end
```

Luego procedemos al ensamblaje de las matrices globales.

Código: Ensamblaje de las matrices globales

```
A = zeros(2 * N + 1, 2 * N + 1);
b = zeros(2 * N + 1, 1);
for k = 1:N
    Aloc = Th[k].Aloc
    floc = Th[k].floc
    dof = Th[k].dof
    A[dof, dof] = A[dof, dof] .+ Aloc
    b[dof] = b[dof] .+ floc
end
```

Ahora realizamos los cálculos relacionados con las condiciones de Dirichlet.

Código: Cálculos relacionados con el dato Dirichlet

```
begin
x_d = zeros(2 * N + 1, 1)
x_d[1] = 0
x_d[2 * N + 1] = 0
b_d = b .- A * x_d
end
```

Resolvemos el sistema de ecuaciones lineales resultante para obtener la solución aproximada de la PDE en los nodos de la malla.

Código: Resolución del sistema lineal para los nodos interiores

```
begin
```

```
        int = 2:(2*N)
x_sol = x_d
A_int = A[int, int]
b_int = b_d[int]
x_aux = A_int \ b_int
x_sol[int] = x_aux
x_sol[1] = 0
x_sol[end] = 0
        size(x_sol)
end
```

Código: Creación del gráfico de la solución calculada y solución exacta.

```
begin
u_exact(x) = 0.5*x*(1 - x)
t = 0:(1/(2*N)):1
u_ex = u_exact.(t_f)
plot(t,x_sol,lw=5, title="Solucin calculada", label="u_h(x)", linewidth=4,
        linestyle=:dash,
        linecolor="red")

plot!(t_f, u_ex,
        lw=3,
        linecolor="blue",
        label="u(x) exacta")
end
```

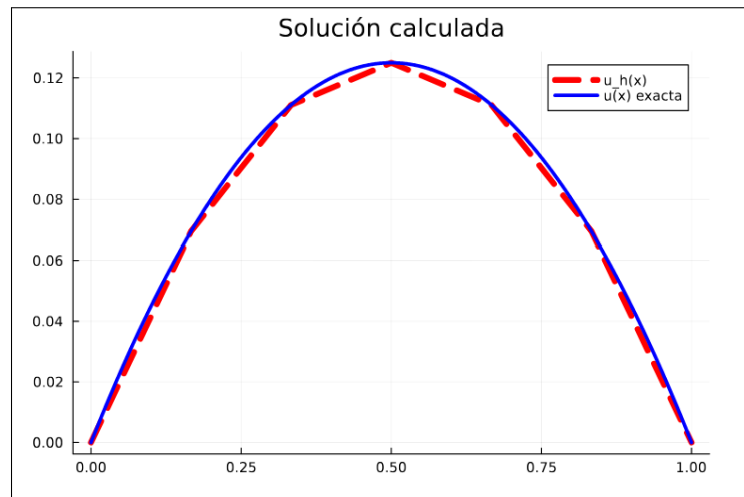


Figura 3.3: Comparación solución exacta y solución de elementos finitos en 3 elementos.

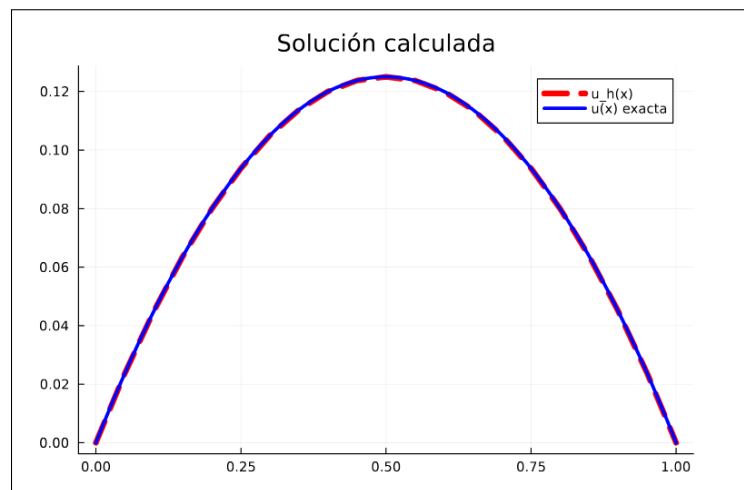


Figura 3.4: Comparación solución exacta y solución de elementos finitos en 10 elementos.

Aunque la aproximación se construye mediante funciones base polinomiales de grado dos, para la representación gráfica se utilizaron únicamente los valores aproximados en los vértices de la discretización. Por ello, la curva mostrada corresponde a una interpolación lineal por partes entre nodos consecutivos y no a la reconstrucción cuadrática completa de la solución aproximada.

En esta sección se desarrolló el procedimiento general del método de elementos finitos, desde la definición de las estructuras básicas hasta la resolución del sistema lineal. Se mostraron las etapas de cálculo de información local, ensamblaje de matrices globales y aplicación de condiciones de frontera. El proceso se implementó tanto

para funciones base de primer grado como de segundo grado, permitiendo comparar la precisión de ambas aproximaciones en la solución numérica del problema.

Capítulo 4

Método de elementos finitos en dos dimensiones

Se desarrolla el método de elementos finitos en dos dimensiones, partiendo de la formulación fuerte del problema.

4.1. Formulación fuerte

En esta sección se presenta el método de elementos finitos aplicado a ecuaciones diferenciales parciales en dos dimensiones, empleando su formulación tanto fuerte como débil. Se toma como modelo la ecuación de Poisson y se usan funciones base continuas por partes de primer grado para construir la discretización del dominio. Esta aproximación numérica permite abordar de forma sistemática y computacional la solución del problema. Para saber más del método en 2D y su fundamentación matemática, véase [21].

Considere la ecuación de Poisson en un dominio $\Omega \subset \mathbb{R}^2$, cuya formulación fuerte es:

$$\begin{cases} -\Delta u(x) = f(x), & \text{para } x \in \Omega, \\ u(x) = g(x), & \text{para } x \in \partial\Omega. \end{cases} \quad (4.1)$$

Aquí, $u(x)$ representa la solución desconocida, $f(x)$ es una función de carga, y $g(x)$ impone condiciones de Dirichlet en el borde del dominio $\partial\Omega$.

4.1.1. Formulación débil

Para obtener una formulación débil, multiplicamos ambos lados de la ecuación diferencial por una función de prueba $v \in C_0^\infty(\Omega)$, lo que nos lleva a la siguiente

identidad:

$$-\Delta u(x)v(x) = f(x)v(x). \quad (4.2)$$

Integrando en todo el dominio Ω , se obtiene:

$$-\int_{\Omega} \Delta u(x)v(x) dx = \int_{\Omega} f(x)v(x) dx. \quad (4.3)$$

Esta ecuación representa el primer paso en la construcción del método de elementos finitos, al reformular el problema en términos de integrales que pueden ser discretizadas sobre una base finita de funciones.

Al aplicar la fórmula de integración de Green en la integral del lado izquierdo de la igualdad obtenida anteriormente y considerando la condición $v(x) = 0$ en la frontera $\partial\Omega$, se obtiene

$$\begin{aligned} -\int_{\Omega} \Delta u(x)v(x) dx &= \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx - \int_{\partial\Omega} (\nabla u(x) \cdot \eta(x))v(x) dS, \\ &= \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx - 0, \\ &= \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx. \end{aligned}$$

Por lo tanto, la formulación débil del problema de Poisson se expresa como:

Calcular $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ tal que

$$\begin{cases} \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x)v(x) dx, & \text{para todo } v \in C_0^\infty(\Omega), \\ u(x) = g(x), & \text{para } x \in \partial\Omega. \end{cases}$$

A continuación se escogen los espacios apropiados V y U para la formulación de Galerkin. Considere el espacio de Hilbert. Se considera el espacio de Hilbert $H^1(\Omega)$, definido como:

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) : \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2} \in L^2(\Omega) \right\}. \quad (4.4)$$

Aquí, $L^2(\Omega)$ representa el conjunto de funciones cuadrado-integrables con el producto

interno definido como:

$$(v, w)_{H^1(\Omega)} = \int_{\Omega} v(x)w(x) dx + \int_{\Omega} \nabla v(x) \cdot \nabla w(x) dx. \quad (4.5)$$

De manera equivalente, esto se puede expresar como:

$$(v, w)_{L^2(\Omega)} + \left(\frac{\partial v}{\partial x_1}, \frac{\partial w}{\partial x_1} \right)_{L^2(\Omega)} + \left(\frac{\partial v}{\partial x_2}, \frac{\partial w}{\partial x_2} \right)_{L^2(\Omega)}. \quad (4.6)$$

La norma en el espacio $H^1(\Omega)$ se define como:

$$\|v\|_{H^1(\Omega)}^2 = \int_{\Omega} |v(x)|^2 dx + \int_{\Omega} |\nabla v(x)|^2 dx. \quad (4.7)$$

Expandiendo la expresión del gradiente, se puede escribir como:

$$\|v\|_{H^1(\Omega)}^2 = \|v\|_{L^2(\Omega)}^2 + \left\| \frac{\partial v}{\partial x_1} \right\|_{L^2(\Omega)}^2 + \left\| \frac{\partial v}{\partial x_2} \right\|_{L^2(\Omega)}^2. \quad (4.8)$$

Cuando la función de prueba $v(x)$ es nula en la frontera $\partial\Omega$, se define el espacio de funciones $H_0^1(\Omega)$ como:

$$H_0^1(\Omega) = \{v \in H^1(\Omega) \mid v = 0 \text{ en } \partial\Omega\}. \quad (4.9)$$

Dado que $H_0^1(\Omega)$ es un subespacio de $H^1(\Omega)$, se tiene la inclusión:

$$H_0^1(\Omega) \subset H^1(\Omega). \quad (4.10)$$

Si en la formulación débil se consideran también las derivadas segundas $\frac{\partial v}{\partial x_i}$ y $\frac{\partial^2 v}{\partial x_i \partial x_j}$, se define el espacio:

$$H^2(\Omega) := \left\{ v \in L^2(\Omega) \mid \frac{\partial v}{\partial x_i}, \frac{\partial^2 v}{\partial x_i \partial x_j} \in L^2(\Omega) \right\}. \quad (4.11)$$

La norma en este espacio se define como:

$$\|v\|_{H^2(\Omega)}^2 = \|v\|_{L^2(\Omega)}^2 + \sum_{i=1}^2 \left\| \frac{\partial v}{\partial x_i} \right\|_{L^2(\Omega)}^2 + \sum_{i=1}^2 \sum_{j=1}^2 \left\| \frac{\partial^2 v}{\partial x_i \partial x_j} \right\|_{L^2(\Omega)}^2. \quad (4.12)$$

Dado que el espacio $H^2(\Omega)$ contiene funciones con mayores propiedades de regularidad, se cumplen las inclusiones:

$$H^2(\Omega) \subset H^1(\Omega) \subset L^2(\Omega). \quad (4.13)$$

A partir de la ecuación variacional previamente obtenida, se considera $V = H_0^1(\Omega)$ y $U = H^1(\Omega)$. Se define la forma bilineal $a(u, v) : U \times V \rightarrow \mathbb{R}$ como:

$$a(u, v) = \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx, \quad \text{para todo } u, v \in H^1(\Omega). \quad (4.14)$$

Asimismo, se define el funcional lineal $L(v)$ como:

$$L(v) = \int_{\Omega} f(x)v(x) dx, \quad \text{para todo } v \in H^1(\Omega). \quad (4.15)$$

Por lo tanto, la formulación débil del problema de Poisson se expresa como:

Calcular $u \in U$ tal que,

$$\begin{cases} a(u, v) = L(v), & \text{para todo } v \in V, \\ u = g, & \text{en } \partial\Omega. \end{cases} \quad (4.16)$$

4.1.2. Formulación de Galerkin

Siguiendo el procedimiento de la sección anterior, la formulación débil se puede escribir como:

$$a(u^*, v) = L(v), \quad \text{para todo } v \in V. \quad (4.17)$$

Aquí, el espacio $V = H_0^1(\Omega)$ es de dimensión infinita. Para resolverlo numéricamente,

se introduce una familia de subespacios finitos V^h con parámetro de refinamiento h , tal que:

$$V^h \rightarrow V \quad \text{cuando } h \rightarrow 0. \quad (4.18)$$

Por lo tanto, la formulación de Galerkin consiste en encontrar una solución aproximada en el subespacio V^h .

Para obtener una solución numérica en el espacio de dimensión finita V^h , se busca una solución aproximada $u_h^* \in V^h$ tal que:

$$a(u_h^*, v_h) = L(v_h), \quad \text{para todo } v_h \in V^h. \quad (4.19)$$

Si $\beta = \{\varphi_1, \varphi_2, \dots, \varphi_{N_h}\}$ es una base de V^h , la solución aproximada u_h^* se puede escribir como una combinación lineal de las funciones base:

$$u_h^* = \sum_{j=1}^{N_h} \alpha_j \varphi_j. \quad (4.20)$$

De este modo, la formulación variacional discreta conduce a un sistema de ecuaciones lineales de la forma:

$$Au_h = b, \quad (4.21)$$

donde:

- La matriz de rigidez A se define como:

$$A = [a_{ij}] = a(u, v)(\varphi_i, \varphi_j), \quad i, j = 1, \dots, N_h. \quad (4.22)$$

- El vector de carga b está dado por:

$$b = [b_i]^T = \mathcal{L}(\varphi_i)^T, \quad i = 1, \dots, N_h. \quad (4.23)$$

- El vector incógnita u_h contiene los coeficientes de la combinación lineal:

$$u_h = [\alpha_1, \alpha_2, \dots, \alpha_{N_h}]^T \in \mathbb{R}^{N_h}. \quad (4.24)$$

Dado que V^h es un subespacio de dimensión finita, el sistema algebraico resultante se puede resolver mediante métodos numéricos estándar, como la eliminación gaussiana o métodos iterativos.

4.1.3. Estimación del error a posteriori

Para el problema de Poisson en dos dimensiones, consideremos la solución exacta u y su aproximación mediante el método de elementos finitos u_h . El objetivo de esta sección es establecer una estimación a posteriori que permita evaluar el error entre ambas soluciones y, con ello, analizar la calidad de la aproximación numérica obtenida [21].

Consideremos la ecuación de Poisson en un dominio $\Omega \subset \mathbb{R}^2$, con condiciones de Dirichlet homogéneas:

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

La formulación variacional consiste en encontrar $u \in V := H_0^1(\Omega)$ tal que

$$a(u, v) = L(v) \quad \text{para todo } v \in V,$$

donde la forma bilineal se define como $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx$. Sea $\mathcal{T}_h$ una triangulación regular de Ω y $V_h \subset V$ un subespacio de dimensión finita. El problema discreto consiste en hallar $u_h \in V_h$ tal que

$$a(u_h, v_h) = L(v_h) \quad \text{para todo } v_h \in V_h.$$

Definimos el error $e = u - u_h$ y la norma de energía $\|v\|_E := \|\nabla v\|_{0,\Omega}$.

Con el objetivo de cuantificar la calidad de la aproximación numérica u_h y evaluar el error cometido, se introduce una estimación a posteriori del error en norma de energía, la cual permite medir la discrepancia entre la solución exacta y su aproximación

discreta a partir de cantidades computables.

Teorema 4.1 (Estimación a posteriori del error para Poisson). *Sea u_h la solución del problema de elementos finitos definido anteriormente. Entonces existe una constante $C > 0$, independiente del tamaño de la malla, tal que*

$$\|u - u_h\|_E^2 \leq C \sum_{K \in \mathcal{T}_h} \rho_K^2(u_h),$$

donde el residuo elemental se define por

$$\rho_K(u_h) = h_K \|f + \Delta u_h\|_{0,K} + \frac{1}{2} h_K^{1/2} \|[n \cdot \nabla u_h]\|_{0,\partial K \cap \Omega},$$

y $[n \cdot \nabla u_h]$ representa el salto del gradiente normal de u_h sobre los bordes interiores:

$$[n \cdot \nabla u_h]|_{\partial K_1 \cap \partial K_2} = n_1 \cdot \nabla u_h|_{K_1} + n_2 \cdot \nabla u_h|_{K_2}.$$

Demostración. Sea $e = u - u_h$ el error y πe un interpolante de e en V_h . Por ortogonalidad de Galerkin, se cumple

$$\|e\|_E^2 = \int_{\Omega} \nabla e \cdot \nabla (e - \pi e) \, dx.$$

Dividimos la integral en la suma sobre los elementos K de la triangulación:

$$\|e\|_E^2 = \sum_{K \in \mathcal{T}_h} \int_K \nabla e \cdot \nabla (e - \pi e) \, dx.$$

Aplicando el teorema de Green en cada elemento y recordando que $-\Delta u = f$, se obtiene

$$\|e\|_E^2 = \sum_{K \in \mathcal{T}_h} \left(\int_K (f + \Delta u_h)(e - \pi e) \, dx - \int_{\partial K} (\nabla u_h \cdot n)(e - \pi e) \, ds \right).$$

Cuando se suman las contribuciones de los elementos, las integrales de borde compartido entre dos elementos adyacentes K_1 y K_2 se expresan mediante los saltos del

gradiente normal:

$$\sum_K \int_{\partial K \cap \Omega} (\nabla u_h \cdot n)(e - \pi e) ds = \sum_E \int_E [n \cdot \nabla u_h](e - \pi e) ds.$$

Usando la desigualdad de Cauchy–Schwarz y las estimaciones de interpolación locales

$$\begin{aligned} \left| \int_K (f + \Delta u_h)(e - \pi e) dx \right| &\leq \|f + \Delta u_h\|_{0,K} \|e - \pi e\|_{0,K} \leq C h_K \|f + \Delta u_h\|_{0,K} \|\nabla e\|_{0,\omega_K}, \\ \left| \int_E [n \cdot \nabla u_h](e - \pi e) ds \right| &\leq \|[n \cdot \nabla u_h]\|_{0,E} \|e - \pi e\|_{0,E} \leq C h_K^{1/2} \|[n \cdot \nabla u_h]\|_{0,E} \|\nabla e\|_{0,\omega_E}. \end{aligned}$$

Sumando sobre todos los elementos y bordes interiores, y aplicando nuevamente Cauchy–Schwarz, se llega a

$$\|e\|_E^2 \leq C \sum_{K \in \mathcal{T}_h} (h_K^2 \|f + \Delta u_h\|_{0,K}^2 + h_K \|[n \cdot \nabla u_h]\|_{0,\partial K \cap \Omega}^2)^{1/2} \|e\|_E.$$

Finalmente, dividiendo ambos lados por $\|e\|_E$ se obtiene la cota del enunciado. $\square$

La estimación a posteriori presentada permite cuantificar el error sin conocer la solución exacta, y constituye la base de los métodos adaptativos de refinamiento de malla en FEM. [21].

Capítulo 5

Uso de Gridap

En el capítulo anterior se presentó el desarrollo del método de elementos finitos en dos dimensiones (2D). La extensión de estos procedimientos al caso tridimensional (3D) es análoga, ya que se conservan los principios fundamentales de formulación variacional, discretización y ensamblaje del sistema y se presenta el uso básico en un ejemplo (3D).

En este capítulo se presenta el uso básico de la librería **Gridap** en el lenguaje de programación Julia, así como un ejemplo de implementación en tres dimensiones, con el objetivo de ilustrar la construcción y resolución de problemas mediante el método de elementos finitos en un entorno computacional moderno. Gridap se ha desarrollado con el objetivo de proporcionar una plataforma flexible y extensible para la simulación de métodos de elementos finitos (FEM) en el lenguaje de programación Julia. Su arquitectura modular y su diseño basado en principios de programación genérica y orientada a objetos permiten a los usuarios abordar una amplia gama de problemas de ecuaciones diferenciales parciales (EDPs) con relativa facilidad y eficiencia computacional [3].

5.1. Componentes principales

a) Módulo de Mallas (Grids):

El núcleo de cualquier simulación FEM es la malla, y Gridap ofrece un conjunto de herramientas robustas para la creación y manipulación de mallas complejas. Las mallas pueden ser generadas dentro de Gridap o importadas desde fuentes externas, proporcionando una interfaz unificada para su gestión.

b) Módulo de Funciones de Forma (Shape Functions):

Este módulo es responsable de definir el espacio de funciones dentro del cual se buscará la solución aproximada. Gridap maneja un rango de funciones de forma, desde las lineales hasta las de mayor orden, lo que permite a los usuarios elegir la precisión y complejidad de su aproximación.

c) Módulo de Ensamblaje (Assembly):

La construcción de matrices de sistema y vectores de carga es una tarea fundamental en FEM. Gridap automatiza este proceso, manejando la integración numérica y la aplicación de condiciones de frontera de manera eficiente.

d) Módulo de Solucionadores (Solvers):

Una vez que el sistema de ecuaciones ha sido ensamblado, se necesita un solucionador para obtener la solución aproximada. Gridap se integra con varios solucionadores de Julia y de terceros, ofreciendo soluciones al sistema de ecuaciones.

e) Módulo de Post-Procesamiento (Postprocessing):

La interpretación de los resultados numéricos es tan crucial como la obtención de los mismos. Gridap proporciona capacidades para evaluar la solución en puntos arbitrarios, calcular errores, y realizar visualizaciones de alta calidad.

5.2. Ejemplo práctico con Gridap

En este capítulo, presentaremos un ejemplo práctico que demuestra la implementación de la metodología de elementos finitos utilizando Gridap en Julia. Nuestro enfoque se centrará en la resolución de la ecuación de Poisson en un dominio tridimensional, empleando condiciones de frontera de Dirichlet. Este ejemplo ilustra el flujo de trabajo típico dentro de un entorno de elementos finitos, desde la generación de mallas hasta la visualización de resultados computacionales.

5.2.1. Declaración del problema

Este trabajo se centra en la solución de la ecuación de Poisson dentro de un dominio específico: una bola tridimensional. La geometría y el mallado de la bola se realizaron utilizando la herramienta Gmsh, un generador de mallas tridimensionales, y posteriormente se exportaron para su uso en Gridap, un entorno de elementos finitos en Julia.

El problema matemático se puede formular de la siguiente manera:

$$\begin{cases} -\Delta u = f & \text{en } \Omega, \\ u = g & \text{en } \partial\Omega. \end{cases}$$

donde Ω representa el dominio interior de la bola y $\partial\Omega$ denota su superficie. En nuestro caso, la función fuente f es constante, con $f(x) = 1$, y la condición de frontera de Dirichlet se especifica con $g(x) = 2$ en toda la superficie de la esfera.

El propósito de esta sección es demostrar el proceso de implementación de la solución numérica de la ecuación de Poisson en un dominio tridimensional, utilizando las herramientas y métodos proporcionados por Gridap.

5.2.2. Esquema numérico

Para abordar la ecuación de Poisson dentro de una bola tridimensional generada en Gmsh, implementamos un esquema numérico basado en el método de elementos finitos de Galerkin. Este método es particularmente adecuado para resolver problemas en dominios con geometrías complejas como la bola.

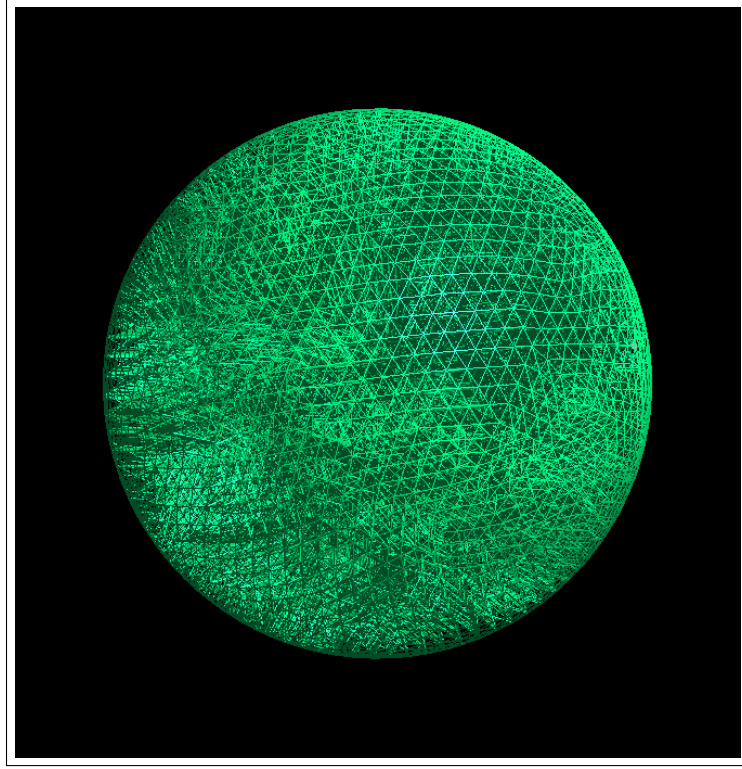


Figura 5.1: Dominio tridimensional con forma de bola Ω utilizado en la resolución de la ecuación de Poisson.

La formulación variacional del problema, o forma débil, es el primer paso para aplicar el método de elementos finitos. La forma débil se establece buscando la solución en un espacio funcional adecuado que satisfaga las ecuaciones en un sentido integral. Para nuestra ecuación de Poisson, la formulación variacional se expresa como:

$$\text{Encontrar } u \in V \text{ tal que } a(u, v) = L(v) \quad \text{para todo } v \in V, \quad (5.1)$$

donde V es un espacio de funciones de prueba apropiado, y las formas bilineal $a(\cdot, \cdot)$ y lineal $L(\cdot)$ están definidas por:

$$a(u, v) = \int_{\Omega} \nabla v \cdot \nabla u \, dx, \quad (5.2)$$

$$L(v) = \int_{\Omega} f v \, dx. \quad (5.3)$$

Aquí, f representa el término fuente constante y u es la función desconocida que deseamos aproximar. Las condiciones de frontera de Dirichlet se imponen adecuadamente en el espacio de funciones de prueba.

El problema de elementos finitos resultante se resuelve numéricamente mediante la discretización del espacio funcional y la aplicación de un solucionador lineal adecuado. La solución numérica se obtiene aproximando el espacio de soluciones y aplicando métodos de integración numérica para evaluar las integrales en las formas bilineal y lineal.

5.2.3. Implementación con Gridap

La implementación del esquema numérico descrito en la sección anterior se lleva a cabo en el lenguaje de programación Julia, utilizando la biblioteca Gridap. Gridap proporciona una serie de herramientas que facilitan la definición de problemas de elementos finitos y la resolución de ecuaciones diferenciales parciales. La simplicidad y potencia de Julia, junto con las abstracciones de alto nivel de Gridap, permiten una implementación eficiente y expresiva del problema de Poisson en la esfera.

5.2.3.1. Modelo discreto y carga de la malla

La geometría de la esfera y la malla asociada fueron creadas en Gmsh y exportadas para su uso en Gridap. El primer paso en la implementación es la carga de la malla en el entorno de Julia. Esto se realiza a través del paquete GridapGmsh, que permite la importación directa de archivos de malla de Gmsh:

```
using GridapGmsh  
  
model = GmshDiscreteModel("esfera.msh")
```

Este modelo discreto importado contiene toda la información necesaria para definir el espacio de elementos finitos y se utiliza como base para la posterior formulación del problema.

5.2.3.2. Espacios de funciones y condiciones de frontera

Una vez que el modelo discreto está en su lugar, el siguiente paso es la definición de los espacios de funciones. En Gridap, se definen espacios de prueba y solución que

reflejan las condiciones de frontera de Dirichlet y las propiedades del material del problema:

```
V0 = TestFESpace(  
    model=model,  
    reffe=:Lagrangian,  
    conformity=:H1,  
    dirichlet_tags=["DirichletBoundary"]  
)  
Ug = TrialFESpace(V0, g)
```

En este fragmento de código, ‘V0’ representa el espacio de funciones de prueba y ‘Ug’ el espacio de funciones de solución, donde ‘g’ es la condición de frontera de Dirichlet aplicada a la superficie de la esfera.

5.2.3.3. Formulación del problema y resolución

La formulación del problema en su forma débil y la configuración del problema de elementos finitos se realizan a continuación. Gridap ofrece una sintaxis intuitiva para expresar la forma bilineal y lineal, y para ensamblar el sistema de ecuaciones:

```
a(u,v) = ((v)(u))d  
l(v) = (v*f)d  
op = AffineFEOperator(a, l, Ug, V0)
```

La solución del sistema se obtiene mediante un solucionador lineal:

```
solver = LUSolver()  
uh = solve(solver, op)
```

La función ‘solve’ proporciona la solución numérica ‘uh’, que puede ser utilizada para la posterior visualización y análisis de resultados.

La Figura 5.2 muestra la distribución de la solución numérica de la ecuación de Poisson en una bola tridimensional. Para visualizar el comportamiento interno de la solución, se realizó un corte transversal del dominio utilizando la herramienta `Clip` de ParaView.

Se observa que los valores mínimos se encuentran sobre la frontera esférica, donde se impone la condición de Dirichlet $u = 2$, mientras que los valores máximos aparecen en la región interior del dominio debido al término fuente constante $f = 1$. Además, la solución presenta un comportamiento radialmente simétrico, consistente con la geometría del dominio y las condiciones del problema.

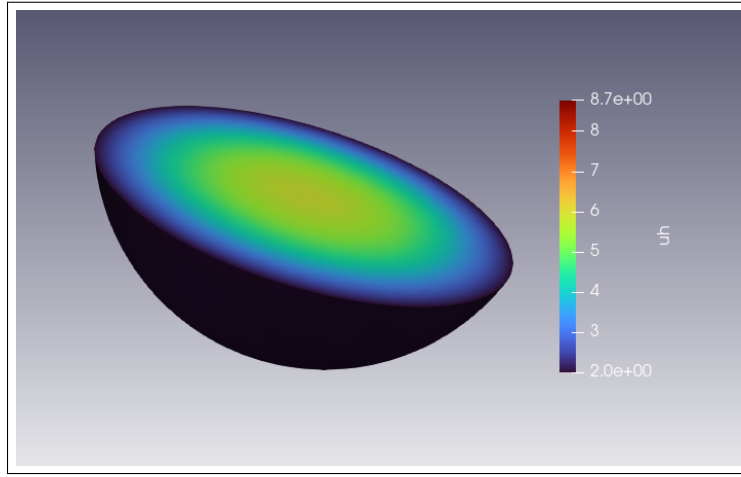


Figura 5.2: Distribución de la solución numérica en una bola tridimensional.

Con estos pasos, se completa la implementación del ejemplo práctico utilizando Gridap, desde la carga de la malla hasta la obtención de la solución numérica.

Además, se desarrolló un cuaderno interactivo en `Pluto.jl` para ilustrar la implementación del método de elementos finitos en dos dimensiones, mostrando paso a paso la generación de la malla, el cálculo de funciones base y la resolución del sistema lineal usando el paquete `Gridap.jl`. Este cuaderno se incluye como material complementario en el Apéndice A.

Capítulo 6

Método generalizado de elementos finitos multiescala (GMsFEM) con malla gruesa

Los métodos de elementos finitos multiescala constan de dos ingredientes principales que son funciones de base multiescala y la formulación numérica global que combina estas funciones base multiescala. Las funciones base están diseñadas para capturar las características multiescala de la solución [11].

6.1. Problema Fino

Consideremos Ecuación de Darcy o ecuación de la presión de un fluido en un medio poroso

$$\begin{cases} -\nabla \cdot (\kappa(x) \nabla u(x)) = f(x) & \text{para } x \in D, \\ u(x) = 0, & x \in \partial D. \end{cases} \quad (6.1)$$

$\kappa(x)$ es un coeficiente multiescala y de alto contraste. se tiene que esta modela distintos tipos de difusión dependiendo de $\kappa(x)$, tal como se muestra a continuación.

- Si $\kappa(x) = \text{constante}$ entonces se tiene difusión homogénea.

La difusión homogénea ocurre cuando el coeficiente $\kappa(x)$ es constante en el dominio.

- Si $\kappa(x) \neq \text{constante}$ entonces se tiene difusión heterogénea.

La difusión heterogénea ocurre cuando dicho coeficiente varía espacialmente.

- Si $\kappa(x) = \begin{pmatrix} k_{11}(x) & 0 \\ 0 & k_{11}(x) \end{pmatrix}$ es una matriz simétrica 2×2 entonces se tiene difusión isotrópica.

La difusión isotrópica describe un medio donde las propiedades son iguales en todas las direcciones, por lo que la difusión ocurre de manera uniforme.

- Si $\kappa(x) = \begin{pmatrix} k_{11}(x) & 0 \\ 0 & k_{22}(x) \end{pmatrix}$ es una matriz simétrica 2×2 , con $k_{11}(x) \neq k_{22}(x)$ entonces se tiene difusión ortotrópica.

La difusión ortotrópica es un caso particular de anisotropía donde las direcciones principales son independientes, pero con distintas intensidades de difusión.

- Si $\kappa(x) = \begin{pmatrix} k_{11}(x) & k_{12}(x) \\ k_{21}(x) & k_{22}(x) \end{pmatrix}$ es una matriz simétrica 2×2 definida positiva entonces se tiene difusión anisotrópica.

La difusión anisotrópica indica que el proceso de difusión depende de la dirección, es decir, el material presenta propiedades distintas en diferentes direcciones.

En esta tesis sólo se consideró $\kappa(x)$ isotrópico.

6.2. Formulación variacional y estabilidad

Para poder visualizar la teoría que sustenta y se relaciona con el GMSFEM, se considera la misma ecuación diferencial parcial elíptica.

El objetivo de este método es hallar una solución aproximada $u(x)$ tal que $u \in H_0^1(D)$ y que cumpla la siguiente igualdad para cualquier función de prueba $v(x)$ que se encuentre en el espacio $H_0^1(D)$

$$\int_D \kappa(x) \nabla u(x) \nabla v(x) \, dx = \int_D f(x) v(x) \, dx. \quad (6.2)$$

La formulación de Galerkin pide encontrar en específico a $u(x)$ en el espacio $V_0^h(D)$ y que siga cumpliendo la ecuación 6.2 con las funciones de test o prueba $v(x)$ en el mismo espacio. Es importante aclarar que, el espacio $V^h(D)$ es aquel conformado por la discretización del dominio D por elementos finitos de funciones continuas lineales por partes, relativas a la triangularización formada en la malla fina $\mathcal{T}^h$ y por tanto, tómesese $V_0^h(D)$ como todas aquellas funciones que pertenecen a $V^h(D)$ y que

son 0 en la frontera. Entonces, las integrales de la ecuación 6.2 definen una forma bilineal y un funcional lineal asociados al problema variacional, las cuales en esta tesis se llamarán a y L , es decir;

$$a(u, v) = \int_D \kappa(x) \nabla u(x) \nabla v(x) \, dx, \quad (6.3)$$

y

$$L(v) = \int_D f(x) v(x) \, dx. \quad (6.4)$$

La formulación variacional del problema consiste en encontrar $u \in H_0^1(D)$ tal que

$$a(u, v) = L(v), \text{ para todo } v \in V_0^h(D), \quad (6.5)$$

o de manera matricial, de la siguiente forma

$$Au = b.$$

Y teniendo en cuenta que toda forma bilineal se puede escribir por medio de matrices, se sigue que para todo $u, v \in V^h(D)$ se tiene que

$$u^T Av = a(u, v) = \int_D \kappa(x) \nabla u(x) \nabla v(x) \, dx, \quad (6.6)$$

y

$$v^T b = L(v) = \int_D f(x) v(x) \, dx. \quad (6.7)$$

A continuación se verifica que la formulación variacional asociada a la ecuación de Darcy satisface las hipótesis establecidas en la definición 2.6

Considérese la ecuación 6.1,

además, supóngase que el coeficiente de permeabilidad satisface

$$0 < \kappa_{\min} \leq \kappa(x) \leq \kappa_{\max} < \infty,$$

para casi todo $x \in D$.

A continuación se verifican las propiedades fundamentales de la formulación variacional.

Proposición 6.1. *La aplicación $a : H_0^1(D) \times H_0^1(D) \rightarrow \mathbb{R}$ define una forma bilineal.*

Demostración. Sean $u, w, v \in H_0^1(D)$ y $\alpha, \beta \in \mathbb{R}$.

Se tiene que

$$a(\alpha u + \beta w, v) = \int_D \kappa(x) \nabla(\alpha u + \beta w) \cdot \nabla v \, dx.$$

Por linealidad del operador gradiente,

$$\nabla(\alpha u + \beta w) = \alpha \nabla u + \beta \nabla w.$$

Sustituyendo esta expresión en la integral anterior, se obtiene

$$a(\alpha u + \beta w, v) = \int_D \kappa(x) (\alpha \nabla u + \beta \nabla w) \cdot \nabla v \, dx.$$

Aplicando la distributividad del producto interno,

$$a(\alpha u + \beta w, v) = \int_D \alpha \kappa(x) \nabla u \cdot \nabla v \, dx + \int_D \beta \kappa(x) \nabla w \cdot \nabla v \, dx.$$

Por linealidad de la integral,

$$a(\alpha u + \beta w, v) = \alpha \int_D \kappa(x) \nabla u \cdot \nabla v \, dx + \beta \int_D \kappa(x) \nabla w \cdot \nabla v \, dx.$$

Por lo tanto,

$$a(\alpha u + \beta w, v) = \alpha a(u, v) + \beta a(w, v).$$

De manera análoga se prueba la linealidad respecto de la segunda variable. Por lo tanto, $a(\cdot, \cdot)$ es una forma bilineal. □

Proposición 6.2. *La forma bilineal $a(\cdot, \cdot)$ es continua sobre $H_0^1(D) \times H_0^1(D)$.*

Demostración. Sean $u, v \in H_0^1(D)$.

Entonces,

$$|a(u, v)| = \left| \int_D \kappa(x) \nabla u \cdot \nabla v \, dx \right|.$$

Aplicando la desigualdad triangular,

$$|a(u, v)| \leq \int_D |\kappa(x) \nabla u \cdot \nabla v| \, dx.$$

Usando la desigualdad

$$|a \cdot b| \leq |a||b|,$$

se obtiene

$$|a(u, v)| \leq \int_D |\kappa(x)| |\nabla u| |\nabla v| \, dx.$$

Como

$$\kappa(x) \leq \kappa_{\max},$$

para casi todo $x \in D$, se sigue que

$$|a(u, v)| \leq \kappa_{\max} \int_D |\nabla u| |\nabla v| \, dx.$$

Aplicando la desigualdad de Cauchy–Schwarz se tiene

$$\int_D |\nabla u| |\nabla v| \, dx \leq \left(\int_D |\nabla u|^2 \, dx \right)^{1/2} \left(\int_D |\nabla v|^2 \, dx \right)^{1/2}.$$

Por tanto,

$$|a(u, v)| \leq \kappa_{\max} \left(\int_D |\nabla u|^2 dx \right)^{1/2} \left(\int_D |\nabla v|^2 dx \right)^{1/2}.$$

Utilizando la definición de la norma $L^2(D)$,

$$|a(u, v)| \leq \kappa_{\max} \|\nabla u\|_{L^2(D)} \|\nabla v\|_{L^2(D)}.$$

Finalmente, dado que

$$\|\nabla u\|_{L^2(D)} \leq \|u\|_{H^1(D)},$$

y

$$\|\nabla v\|_{L^2(D)} \leq \|v\|_{H^1(D)},$$

se concluye que

$$|a(u, v)| \leq \kappa_{\max} \|u\|_{H^1(D)} \|v\|_{H^1(D)}.$$

Por lo tanto, la forma bilineal $a(\cdot, \cdot)$ es continua sobre $H_0^1(D) \times H_0^1(D)$. $\square$

Proposición 6.3. *La forma bilineal $a(\cdot, \cdot)$ es V -elíptica sobre $H_0^1(D)$.*

Demostración. Sea $v \in H_0^1(D)$.

Por definición de la forma bilineal,

$$a(v, v) = \int_D \kappa(x) \nabla v \cdot \nabla v \, dx.$$

Usando la identidad

$$\nabla v \cdot \nabla v = |\nabla v|^2,$$

se obtiene

$$a(v, v) = \int_D \kappa(x) |\nabla v|^2 dx.$$

Como

$$\kappa(x) \geq \kappa_{\min} > 0,$$

para casi todo $x \in D$, se sigue que

$$a(v, v) \geq \kappa_{\min} \int_D |\nabla v|^2 dx.$$

Por lo tanto,

$$a(v, v) \geq \kappa_{\min} \|\nabla v\|_{L^2(D)}^2.$$

Además, dado que $v \in H_0^1(D)$, por la desigualdad de Poincaré existe una constante $C_P > 0$ tal que

$$\|v\|_{L^2(D)} \leq C_P \|\nabla v\|_{L^2(D)}.$$

En consecuencia, la seminorma

$$|v|_{H^1(D)} = \|\nabla v\|_{L^2(D)}$$

es equivalente a la norma $\|v\|_{H^1(D)}$ sobre $H_0^1(D)$.

Por equivalencia de normas, existe una constante $\alpha > 0$ tal que

$$a(v, v) \geq \alpha \|v\|_{H^1(D)}^2, \quad \forall v \in H_0^1(D).$$

Por lo tanto, $a(\cdot, \cdot)$ es V -elíptica sobre $H_0^1(D)$, es decir coerciva. □

Proposición 6.4. *El funcional lineal $L : H_0^1(D) \rightarrow \mathbb{R}$ es continuo.*

Demostración. Sea $v \in H_0^1(D)$.

Entonces,

$$|L(v)| = \left| \int_D f(x)v(x) dx \right|.$$

Aplicando la desigualdad de Cauchy–Schwarz,

$$|L(v)| \leq \left(\int_D |f(x)|^2 dx \right)^{1/2} \left(\int_D |v(x)|^2 dx \right)^{1/2}.$$

Por definición de la norma $L^2(D)$,

$$|L(v)| \leq \|f\|_{L^2(D)} \|v\|_{L^2(D)}.$$

Usando la desigualdad de Poincaré en la expresión anterior

$$\|v\|_{L^2(D)} \leq C_P \|\nabla v\|_{L^2(D)}.$$

Además,

$$\|\nabla v\|_{L^2(D)} \leq \|v\|_{H^1(D)}.$$

Por tanto,

$$|L(v)| \leq C_P \|f\|_{L^2(D)} \|v\|_{H^1(D)}.$$

En consecuencia, $L(\cdot)$ es un funcional lineal continuo sobre $H_0^1(D)$. □

Teorema 6.1. *El problema variacional asociado a la ecuación de Darcy posee una única solución débil $u \in H_0^1(D)$.*

Demostración. De las proposiciones anteriores se tiene que la forma bilineal $a(\cdot, \cdot)$ es continua y coerciva sobre $H_0^1(D) \times H_0^1(D)$, mientras que el funcional lineal $L(\cdot)$ es continuo sobre $H_0^1(D)$.

Por el teorema de Lax–Milgram, existe una única función

$$u \in H_0^1(D)$$

tal que

$$a(u, v) = L(v), \quad \forall v \in H_0^1(D).$$

Por lo tanto, el problema variacional asociado a la ecuación de Darcy está bien planteado. Además, la solución depende continuamente del término fuente f , lo cual garantiza la estabilidad del problema variacional. $\square$

6.3. Construcción de funciones base multiescala

La construcción de las funciones base multiescala presentada en esta sección sigue el enfoque desarrollado en [15]. Al igual que en los métodos FEM clásico y Método de Elementos Finitos Multiescala (MsFEM), para aproximar la función $u(x)$ se considera una partición gruesa $\mathcal{T}^H$ del dominio D y así, para dar una solución $u(x)$ más adaptada a la solución real, se introduce una malla que refina la partición y que se denota como $\mathcal{T}^h$. Sean $\{x_i\}$ los nodos de la malla $\mathcal{T}^h$. Se formarán las funciones base en cada uno de estos nodos. Sin embargo, para hacer estas funciones más robustas en su soporte, se toma la información más importante relativa a la vecindad que lo rodea, para ello se define la vecindad de dicho punto de la siguiente forma.

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H : x_i \in \overline{K_j}\}. \quad (6.8)$$

Las vecindades gruesas ω_i corresponden a las regiones asociadas a cada nodo de la malla gruesa y son utilizadas para construir problemas espectrales locales [10]. En la Figura 6.1 se muestra la vecindad del nodo x_i en el dominio D .

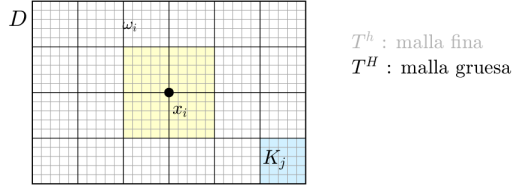


Figura 6.1: Vecindades del dominio D . Elaboración propia en Matlab.

Así, para hallar las funciones base que ayudarán a aproximar la solución real, se considera el siguiente problema de autovalores de alto contraste, tomando como dominio a cada una de las vecindades ω_i .

$$-\operatorname{div}(\kappa(x)\nabla\psi) = \lambda\kappa(x)\psi. \quad (6.9)$$

Este problema espectral local permite identificar los modos dominantes asociados a las heterogeneidades del medio poroso. La selección de los autovectores correspondientes a autovalores pequeños permite construir espacios reducidos capaces de capturar las características multiescala de la solución [10].

Observe que nuevamente en la anterior ecuación se tiene un problema parecido al inicial, pero en un dominio más pequeño, entonces se sigue que la formulación débil del problema pide encontrar autovectores ψ y un número real λ tal que para cualquier función de test $z \in H_0^1(\omega_i)$ se tenga que

$$\int_{\omega_i} \kappa(x)\nabla\psi(x)\nabla z(x) \, dx = \int_{\omega_i} \lambda\kappa(x)z(x) \, dx. \quad (6.10)$$

Esto se puede representar análogamente utilizando la forma bilineal a definida previamente en 6.3 y la forma bilineal m , que se define;

$$m(z_1, z_2) = \int_D \kappa(x)z_1(x)z_2(x) \, dx. \quad (6.11)$$

Entonces, la ecuación 6.10 se puede expresar en términos de formas bilineales como

un problema de autovalores y autovectores generalizados. Tal como se muestra a continuación

$$a_{\omega_i}(\psi, z) = \lambda m_{\omega_i}(\psi, z) \quad \text{con} \quad z \in H_0^1(\omega_i). \quad (6.12)$$

Sin embargo, se está trabajando con la formulación de Galerkin, entonces realmente se está buscando una función de prueba z en $V_0^{\omega_i}(D)$, es decir, en un subespacio de $V_0^h(D)$ que solo tiene las funciones base de FEM definidas sobre los nodos de la malla fina que se encuentren en la vecindad ω_i .

Para mayor claridad, los autovalores y autovectores resultantes de la solución de dicho problema se notarán como $\{\lambda_j^{\omega_i}\}$ y $\{\psi_i^{\omega_i}\}$ respectivamente. Como se mencionó, solo se escogerán los dominantes. Así, al finalizar este procedimiento con todas las vecindades, se conforma el espacio generado por todos los autovectores escogidos y se procede a ordenar estos autovectores según su autovalor asociado (de menor a mayor). A este espacio formado se le notará como V_{gms} el cual será el espacio de las funciones base multiescala generalizadas.

Para la construcción del espacio grueso, se seleccionan los espacios correspondientes a autovalores pequeños. Esta base se define como:

$$\Phi_{i,\ell} = \chi_i \psi_\ell^{\omega_i}, \quad 1 \leq i \leq N_v, \quad 1 \leq \ell \leq L_i,$$

donde L_i es el número de autovalores seleccionados para el nodo i , y χ_i es una partición de la unidad.

Finalmente, el espacio multiescala espectral local se define como

$$V_{gms} = \text{span}\{\Phi_{i,\ell}\} \text{ con } 1 \leq i \leq N_v \text{ y } 1 \leq \ell \leq L_i.$$

Entonces, la solución en la malla gruesa

$$u_0(x) = \sum_i c_i \Phi_i$$

del problema se encontrará tomando las funciones de prueba $v(x)$ en el espacio V_{gms} y solucionando la ecuación $a(u, v) = L(v)$.

La matriz del sistema en malla gruesa se obtiene mediante la transformación

$$A_0 = R_0 A R_0^T,$$

donde

$$R_0^T = [\Phi_1, \dots, \Phi_{N_c}].$$

En este contexto, las funciones Φ_i corresponden a bases discretas definidas sobre la malla fina. Para simplificar la notación, se emplea el mismo símbolo para las bases multiescala continuas.

La solución multiescala de elementos finitos se obtiene proyectando la solución fina sobre el espacio V_{gms} , lo cual se expresa mediante la ecuación

$$A_0 u_0 = b_0,$$

donde

$$b_0 = R_0 b.$$

Finalmente, la solución en la malla gruesa se mapea de vuelta a la malla fina mediante $u = R u_0$.

Con la construcción del espacio multiescala V_{gms} y la formulación del sistema reducido, se obtiene una aproximación eficiente de la solución del problema fino mediante un número significativamente menor de grados de libertad. Sin embargo, dado que esta aproximación se basa en la selección de un subconjunto de funciones base locales, es necesario evaluar la calidad de dicha reducción y cuantificar el error introducido.

En particular, resulta fundamental analizar cómo la omisión de ciertos modos espectrales en cada vecindad afecta la precisión global de la solución. Con este propósito, en la siguiente sección se desarrolla el análisis teórico del error del método

GMSFEM, incluyendo estimaciones a priori y a posteriori que permiten caracterizar la convergencia del método y evaluar la eficiencia de las funciones base seleccionadas. La elección del número de funciones base adicionales m se realiza a partir del problema espectral local asociado al método GMSFEM. En general, se seleccionan las funciones base correspondientes a los autovalores más pequeños, ya que estas contienen la información dominante del comportamiento multiescala de la solución. Un criterio común consiste en fijar un umbral espectral τ y conservar únicamente las funciones base cuyos autovalores satisfacen

$$\lambda_i < \tau.$$

De esta manera, el número de funciones base depende del comportamiento espectral del problema y del nivel de precisión deseado en la aproximación [10].

6.4. Teoría del error del método generalizado de elementos finitos multiescala (GMSFEM)

El Método Generalizado de Elementos Finitos Multiescala (GMSFEM) se basa en resolver la ecuación de Darcy elíptica de segundo orden. El método busca aproximar la solución fina $u(x)$ por una solución multiescala $u_{ms}(x)$ sobre un espacio reducido V_{gms} construido a partir de funciones base locales que capturan las heterogeneidades del coeficiente $\kappa(x)$ [8].

Esta aproximación se lleva a cabo sobre un espacio pequeño generado por las funciones base que codifican los modos principales de cada vecindario grueso.

6.4.1. Espacio de Galerkin multiescala

Recordando la formulación variacional,

$$a(u, v) = L(v), \quad \text{para todo } v \in H_0^1(D),$$

donde

$$a(u, v) = \int_D \kappa(x) \nabla u \cdot \nabla v \, dx, \quad L(v) = \int_D f(x) v(x) \, dx,$$

la solución aproximada se busca en un subespacio multiescala $V_{gms} \subset H_0^1(D)$ generado por las funciones base

$$\Phi_{i,\ell} = \chi_i \psi_\ell^{\omega_i}, \quad 1 \leq i \leq N_v, \quad 1 \leq \ell \leq L_i,$$

donde $\psi_\ell^{\omega_i}$ son los autovectores asociados a los menores autovalores locales $\lambda_\ell^{\omega_i}$ del problema espectral definido en cada vecindario ω_i . De esta forma, el espacio multiescala local se define como

$$V_{gms} = \text{span}\{\Phi_{i,\ell}\}, \quad u_{ms}(x) = \sum_{i,\ell} c_{i,\ell} \Phi_{i,\ell}(x).$$

El sistema algebraico global equivalente se obtiene como

$$A_0 u_0 = b_0, \quad A_0 = R_0 A R_0^T, \quad b_0 = R_0 b,$$

donde $R_0^T = [\Phi_1, \Phi_2, \dots, \Phi_{N_c}]$ es la matriz que proyecta la solución fina sobre la base multiescala.

6.4.2. Norma de energía

Para medir el error entre la solución fina y la aproximada se define la norma de energía asociada a la forma bilineal $a(\cdot, \cdot)$, definida por

$$|v|_a = \sqrt{a(v, v)} = \left(\int_D \kappa(x) \nabla v \cdot \nabla v \, dx \right)^{1/2}, \quad \text{para todo } v \in H_0^1(D).$$

Esta norma es equivalente a la norma del espacio $H^1(D)$ y sirve para medir la energía del campo de presión. El error en norma de energía se escribe como

$$\|u - u_{ms}\|_a = |u - u_{ms}|_a.$$

6.4.3. Estimación a priori del error

Continuando con los resultados de [8] y [1], el error de la solución multiescala se puede acotar en términos de los autovalores locales no capturados en el espacio V_{gms} . Si λ_{i,L_i+1} es el primer autovalor no tomado en cuenta en la vecindad ω_i , entonces se tiene la siguiente estimación a priori:

$$\|u - u_{ms}\|_a \leq C \max_i \lambda_{i,L_i+1}^{-1/2} \|f\|_{L^2(D)},$$

donde C es una constante independiente del contraste de $\kappa(x)$ y del tamaño de la malla gruesa. Esto demuestra que el error se reduce a medida que se incluyen más funciones base locales, ya que los autovalores λ_{i,L_i+1} aumentan para cada modo adicional.

6.4.4. Estimación a posteriori del error

Además, el error global puede evaluarse mediante residuos locales asociados a cada vecindario. Sea $R_i(v)$ el residuo definido por

$$R_i(v) = \int_{\omega_i} f v \, dx - a(u_{ms}, v), \quad \text{para todo } v \in V^{snap}(\omega_i),$$

y sea $\|R_i\|_{V_i^*}$ su norma dual correspondiente. Entonces, el error total satisface la estimación a posteriori:

$$\|u - u_{ms}\|_a^2 \leq C \sum_i \lambda_{i,L_i+1}^{-1} \|R_i\|_{V_i^*}^2.$$

El valor λ_{i,L_i+1}^{-1} es un indicador de la calidad de las bases locales: valores pequeños indican que la información local no está totalmente capturada, y valores grandes muestran una buena representación multiescala.

6.4.5. Discusión e interpretación

Cuando se analizan estas estimaciones, se observa que la exactitud del método depende en gran medida del comportamiento local de la solución en cada vecindad [1]. La cota *a priori* nos da una idea teórica de cómo se propaga el error. [1], mientras que la cota *a posteriori* sirve para localizar las áreas donde más falla y, por lo tanto, donde más necesita enriquecerse el espacio. V_{gms} [8]. Físicamente, esto implica que el método es capaz de detectar cambios en la multiescala del campo de presión y localiza el esfuerzo computacional en las zonas donde el contraste de permeabilidad $\kappa(x)$ es mayor, [1, 8]. Así, el estudio del error no sólo verifica la estabilidad y convergencia del método, sino que también proporciona una guía para ajustar y seleccionar adaptativamente las funciones base [8].

Capítulo 7

Experimentos numéricos

En este capítulo se presentan los experimentos numéricos realizados con el método GMsFEM con malla gruesa, con el propósito de evaluar su desempeño en la aproximación de la solución del problema de flujo en medios porosos heterogéneos. En particular, se analizan tanto la capacidad del método para capturar altos contrastes en el coeficiente de permeabilidad como su comportamiento en términos de precisión y estabilidad al variar el número de funciones base multiescala.

Con este fin, se consideraron dos tipos de escenarios. En primer lugar, se estudia un medio poroso sintético, cuya construcción permite un control directo sobre la heterogeneidad del coeficiente y facilita la validación de la implementación. Posteriormente, el método se aplica a un caso de referencia más complejo, correspondiente al benchmark SPE10 capa 85, el cual presenta una estructura altamente heterogénea y constituye un estándar en la literatura.

7.1. Caso de validación: medio poroso sintético

Como primer experimento, antes de aplicar el GMsFEM al modelo SPE10, se realizó un caso de prueba con un campo de permeabilidad sintético. Este caso busca validar la construcción adecuada de las matrices en la malla fina y gruesa y la estabilidad de la formulación en presencia de heterogeneidad inducida artificialmente.

El dominio tomado fue $\Omega = [0, 1] \times [0, 1]$ con condición de frontera $p(x, y) = y^2 + \frac{1-x}{128}$, con $p = g$ sobre $\partial\Omega$, discretizado en una malla gruesa de $N_x \times N_y = 10 \times 10$ subregiones. Cada una de ellas se discretizó en una malla de $n_x \times n_y = 4 \times 4$ elementos, generando 40×40 celdas en la malla fina.

El campo de permeabilidad $k(x, y)$ se generó aleatoriamente, con valores entre 1 y

10001, según la siguiente fórmula:

$$k(x, y) = 1 + 10^4 \xi(x, y),$$

donde $\xi(x, y)$ corresponde a una variable aleatoria con distribución uniforme en $(0, 1)$.

El siguiente fragmento de código muestra como se crea la malla y como se define la permeabilidad heterogénea.

```
begin
    ax = 0; bx = 1; ay = 0; by = 1; intx = [ax, bx]; inty = [ay, by]
    Nx = 10; Ny = 10; nx = 4; ny = 4; additional_basis = 6;
    Elements, Boundary, vertex_list, mesh_parameter, h, boundary_nodes, free =
        square_mesh(nx, ny, Nx, Ny, intx, inty)

    K = 10000*rand(Nx*nx, Ny*ny)
    println("size(K) = ", size(K))
    coefficient_values_aux = 1 .+ K[1:Nx*nx, 1:Ny*ny]
end
```

El objetivo es verificar que el método se mantiene estable y converge a la solución fina.

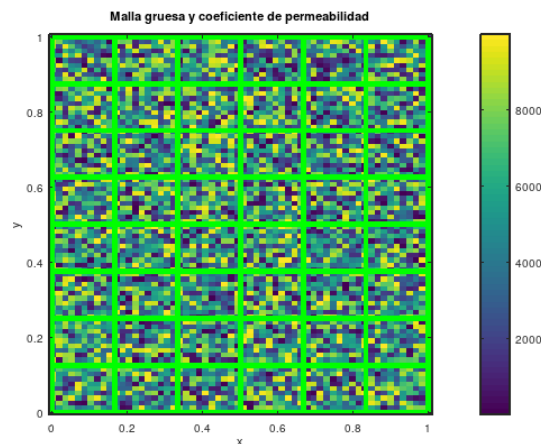


Figura 7.1: Distribución del campo de permeabilidad heterogéneo. Elaboración propia.

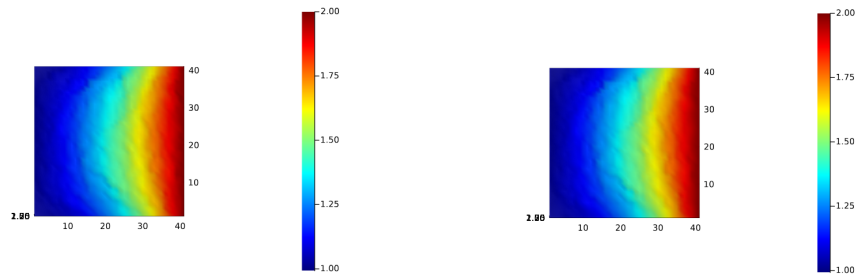
Tabla 7.1: Errores relativos para diferentes números de bases adicionales en el caso sintético ($t = 0$, $N_x = 10$, $N_y = 10$, $n_x = n_y = 4$).

m	L_∞^{rel}	L_2^{rel}	$\ e\ _A^{rel}$	Tiempo (s)
6	0.05362	0.00416	0.01126	2.1
8	0.04081	0.00389	0.00966	2.5
10	0.04341	0.00346	0.00897	3.2
12	0.02907	0.00291	0.00794	2.6
24	0.00250	0.00061	0.00070	4.3
32	0.00250	0.00063	0.00077	6.5

A medida que aumentamos el número de bases adicionales m , los errores relativos en norma L_2 y en energía disminuyen considerablemente, lo que evidencia la convergencia del método multiescala. La mayor disminución se produce entre $m = 12$ y $m = 24$, donde el error en energía se reduce en un factor de 10.

A partir de $m = 24$, las mejoras son muy pequeñas, lo que sugiere una *saturación del espacio multiescala* con la partición gruesa utilizada. El coste computacional (tiempo de ejecución) crece linealmente con m por el crecimiento del tamaño del sistema global. Entonces, un equilibrio entre exactitud y coste se obtiene para $m \in [12, 24]$, donde la mejora en la aproximación todavía justifica el aumento de tiempo. El caso $m = 32$ da el menor error, pero con mucho mayor costo computacional.

A continuación se presentan las gráficas obtenidas al ejecutar el experimento.



(a) Solución de referencia (malla fina) (b) Solución GMsFEM reconstruida

Figura 7.2: Comparación de la solución a escala fina y la multiescala con $m = 12$. Ambas se comportan de manera similar, lo que demuestra que el método multiescala reproduce fielmente la solución fina.

En la Figura 7.2 se comparan la solución multiescala generalizada con $m = 12$ bases adicionales y la solución de referencia en escala fina. Se nota que la solución multiescala reproduce fielmente las características globales del flujo, resolviendo

correctamente las oscilaciones espaciales forzadas por el campo de permeabilidad sintético.

En general, la solución multiescala se comporta de manera consistente con la física del problema, demostrando que con $m = 12$ funciones locales adicionales por vecindad se puede aproximar eficientemente la solución de la malla fina a un costo computacional mucho menor. Este resultado verifica la construcción adecuada de las bases multiescala y la habilidad del método para capturar la información del medio sin tener que refinar la malla global.

La implementación completa de este experimento en Julia se encuentra disponible en el Apéndice A.

7.2. Caso de estudio: SPE10 — capa 85

En esta sección se presenta el caso de estudio SPE10, capa 85, ampliamente utilizado como benchmark en simulación de flujo en medios porosos debido a su alta heterogeneidad y complejidad multiescala.

7.2.1. Metodología de preparación del dato SPE10

Se usó como referencia el benchmark SPE10 (10th SPE Comparative Solution Project), Modelo 2, que ofrece campos de permeabilidad y porosidad altamente heterogéneos; la malla 3D original es $60 \times 220 \times 85$, siendo la capa 85 la heterogénea Upper Ness, un corte 2D muy contrastado, ampliamente utilizado como caso test de flujo en medios porosos. La caracterización del benchmark y la descripción de uso comparativo se encuentran en Christie Blunt (2001) y en el material público de SPE/SINTEF. [7, 9]

Para esta tesis se extrajo la capa 85 a partir de archivos CSV: `SPE10_perm_layer85.csv` y `SPE10_phi_layer85.csv`. Al leerlos se verificaron dimensiones (60, 220) y, si venían

transpuestos $(220, 60)$, se reordenaron. Se aplicaron aserciones para asegurar positividad de la permeabilidad y ausencia de NaN/Inf. Los valores de permeabilidad se convirtieron de miliDarcy a m^2 usando $1 \text{ mD} = 9.869233 \times 10^{-16} \text{ m}^2$, y se regularizaron con un mínimo de $1 \times 10^{-6} \text{ mD}$ antes de la conversión. El dominio se reescaló afinmente a $\Omega = (0, 1)^2$ y el campo $k(x, y)$ se acomodó a la malla numérica usada en los experimentos.

7.2.2. Formulación del problema y parámetros numéricos

Se resolvió la ecuación de Darcy en $\Omega = (0, 1)^2$ con $f = 0$, condiciones de Dirichlet $p = p_L$ en $x = 0$ y $p = p_R$ en $x = 1$, y flujo nulo (Neumann) en $y = 0, 1$. La solución de referencia p_{ref} se obtuvo en malla fina (o con una configuración multiescala muy enriquecida). El número de funciones base adicionales por parche se denota por m . En esta sección se fijó la malla gruesa en $N_x = 6$, $N_y = 22$ y la malla fina por celda en $n_x = n_y = 10$, sin oversampling ($t = 0$).

7.2.3. Resultados numéricos y análisis

Se analizó la sensibilidad del error con respecto a m en $\{6, 8, 10, 12, 24, 32\}$. Se informa del error relativo en L_∞ , error relativo en L_2 , error relativo en norma de energía y tiempo total de ejecución. Se resuelve la ecuación de Darcy en 2D

$$-\nabla \cdot (k(x, y) \nabla p(x, y)) = 0 \quad \text{en } \Omega = (0, 1) \times (0, 1), \quad (7.1)$$

con condiciones de frontera

$$p(0, y) = 1, \quad p(1, y) = 0, \quad \partial_n p(x, 0) = \partial_n p(x, 1) = 0, \quad (7.2)$$

y forzante nulo $f = 0$.

La permeabilidad k se toma de *SPE10*, capa 85 (malla 60×220), se convierte de mD a m^2 y se aplanan los datos en orden *x-first* para coincidir con la numeración de

elementos. La malla fina es la misma que SPE10: $N_x n_x = 60$ y $N_y n_y = 220$.

Se emplea GMsFEM. Para cada vecindario grueso se resuelve un problema eigen local $A_i \phi = \lambda M_i \phi$ y se seleccionan *modos* por *salto relativo* de autovalores,

$$\lambda_{k+1}/\lambda_k > \gamma, \quad \gamma = 10,$$

con tope de 10 modos por vecindario. Las bases se normalizan *dentro del parche* (enmascarando fuera de I_g y dividiendo por el máximo local, umbral 10^{-10}). Las condiciones de Dirichlet se imponen en todas las bases asociadas a nodos de borde $x = 0, 1$. En el ensamblaje grueso se aplica una única vez la corrección del término de frontera: $b_0 = R^\top b - A_0 x_D$.

Los errores se miden respecto a la solución fina p_h :

$$\|e\|_\infty = \|p_{\text{ms}} - p_h\|_\infty, \quad \|e\|_2 = \left(\frac{1}{|\Omega|} \int_\Omega (p_{\text{ms}} - p_h)^2 dx \right)^{1/2}, \quad (7.3)$$

$$\frac{\|e\|_A}{\|p_h\|_A} = \frac{\left(\int_\Omega k |\nabla(p_{\text{ms}} - p_h)|^2 dx \right)^{1/2}}{\left(\int_\Omega k |\nabla p_h|^2 dx \right)^{1/2}}.$$

Para analizar cómo se comporta el error en función del número de bases añadidas, se llevaron a cabo simulaciones fijando la malla gruesa ($N_x = 6$, $N_y = 22$) y la malla fina ($n_x = n_y = 10$). En todos los casos se operó sin oversampling ($t = 0$). El número de funciones base adicionales m se aumentó gradualmente para $m = 6, 8, 10, 12, 24, 32$, y para cada caso se calcularon los errores relativos en norma L_2 , norma L_∞ y norma de energía.

Los resultados se resumen en la Tabla 7.2, donde también se muestra el tiempo total de ejecución de cada caso.

m	L_∞	L_2^{rel}	$\ e\ _A^{rel}$	Tiempo (s)
6	0.11165	0.07733	0.02587	215
8	0.11129	0.07629	0.02355	222
10	0.11103	0.07567	0.02264	196
12	0.11086	0.07465	0.02206	233
24	0.11027	0.07220	0.01970	257
32	0.10866	0.07050	0.01769	304

Tabla 7.2: Errores relativos para diferentes números de bases adicionales ($t = 0$, $N_x = 6$, $N_y = 22$, $n_x = n_y = 10$).

Al aumentar m , los errores relativos L_2 y de energía descienden de forma sostenida; el mayor salto se observa entre $m = 12$ y $m = 24$. El tiempo total crece con m , por el coste del pre-cómputo local y el tamaño efectivo del sistema. Un compromiso razonable se sitúa en el entorno $m \approx 12$ – 24 , donde la mejora de error aún compensa el incremento de tiempo; $m = 32$ produce el menor error pero con el mayor coste.

A continuación se presentan las gráficas obtenidas al ejecutar el experimento.

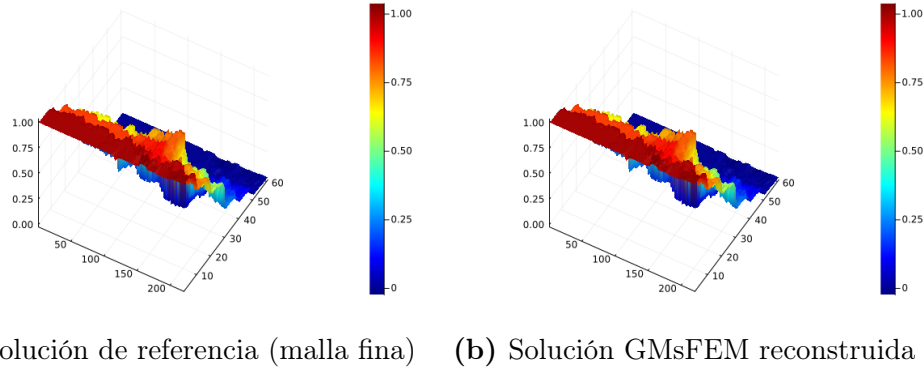


Figura 7.3: Comparación visual entre la solución fina y la solución reconstruida mediante GMsFEM para la capa 85 del modelo SPE10. Ambas imágenes utilizan la misma escala de color para resaltar las diferencias locales.

La solución multiescala reproduce la caída global de presión de izquierda a derecha y las variaciones locales debidas a la heterogeneidad.

Para corroborar los experimentos numéricos sobre nuestro método implementado aplicado a SPE10 capa 85 se construye una curva logarítmica de convergencia.

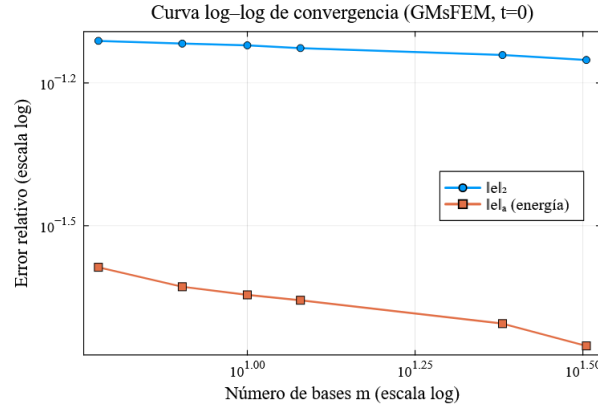


Figura 7.4: Comportamiento del error relativo en función del número de bases adicionales m en cada vecindad. Elaboración propia.

En la Figura 7.4 se ilustra la curva log-log de los errores relativos en norma L_2 y en norma de energía en función del número de bases adicionales m . Se aprecia un comportamiento decreciente en ambas normas, más acusado en la norma de energía, lo que demuestra la convergencia del método GMsFEM. Por otro lado, el error en norma L_2 se estabiliza, como es característico de configuraciones sin oversampling ($t = 0$). La tendencia aproximadamente lineal en escala logarítmica indica que el error se comporta como una potencia de m .

La implementación completa de este experimento en Julia se encuentra disponible en el Apéndice A.

Capítulo 8

Estudios adicionales: GMsFEM sin malla gruesa

En esta sección se presenta una variante del método (GMsFEM) que evita la construcción explícita de una malla gruesa estructurada. Este enfoque, denominado Meshfree Generalized Multiscale Finite Element Method (MFGMsFEM), permite construir espacios multiescala utilizando únicamente una nube de puntos distribuidos sobre el dominio y vecindarios locales asociados. La principal ventaja del método consiste en su capacidad para tratar medios altamente heterogéneos y geometrías complejas sin depender de una partición gruesa conformante del dominio.

La idea fundamental del método es reemplazar la malla gruesa tradicional por una colección de vecindarios superpuestos contruidos alrededor de puntos gruesos seleccionados en el dominio. En cada vecindario se resuelven problemas espectrales locales asociados al coeficiente de permeabilidad $\kappa(x)$, obteniendo funciones base capaces de capturar las principales características multiescala del medio. Posteriormente, dichas funciones locales son combinadas mediante funciones de partición de unidad para construir un espacio multiescala reducido.

Finalmente, el problema original se proyecta sobre este espacio reducido, obteniendo un sistema de menor dimensión computacional. Una vez calculada la solución multiescala, ésta puede reconstruirse en la escala fina mediante una matriz de proyección. Este enfoque ha mostrado buenos resultados en problemas de flujo en medios porosos con altos contrastes de permeabilidad y configuraciones geométricas complejas (véase [22]).

8.1. Modelo matemático

El modelo estudiado está dado por 6.1 acompañado de condiciones iniciales y de frontera apropiadas. Aquí, $\kappa(x)$ representa un coeficiente de permeabilidad altamente heterogéneo y multiescala, el cual será utilizado para la construcción del método GMsFEM sin malla gruesa. El objetivo es reducir la dimensión del problema mediante la construcción

de funciones base multiescala sin requerir una malla gruesa estructurada.

8.2. Construcción de la escala gruesa sin malla

En lugar de utilizar una malla gruesa convencional, el dominio D se descompone en una colección de vecindarios gruesos superpuestos S_i , construidos alrededor de una nube de puntos gruesos

$$\{x_i\}_{i=1}^{N_H}.$$

Estos puntos pueden obtenerse mediante técnicas de teselaciones centroidales de Voronoi o mediante distribuciones adaptadas al comportamiento espacial del coeficiente de permeabilidad $\kappa(x)$, ver Figura 8.1b y Figura 8.1c.

Cada vecindario grueso S_i se define como

$$S_i = \bigcup \left\{ K^h : \max_{y \in K^h} \|y - x_i\| \leq r_i, \quad K^h \in \mathcal{T}^h \right\}, \quad (8.1)$$

donde r_i representa el radio de influencia asociado al punto grueso x_i , y $\mathcal{T}^h$ corresponde a una malla fina de referencia utilizada para resolver los problemas locales, ver Figura 8.1a.

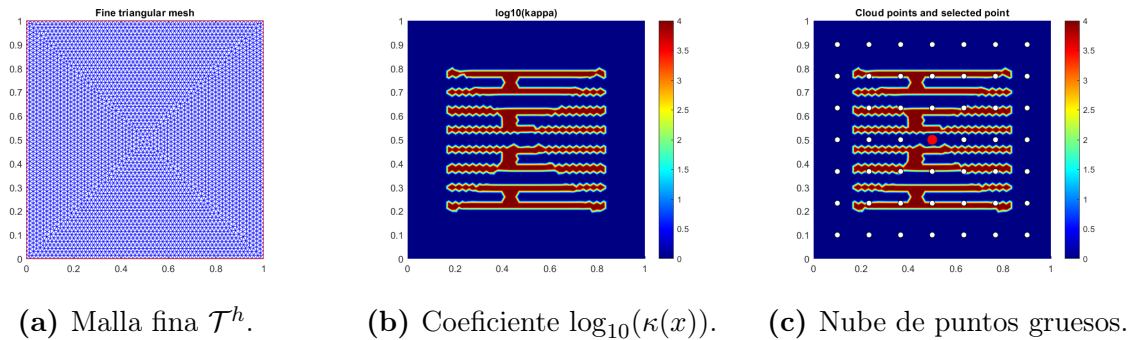


Figura 8.1: Construcción de la discretización meshfree utilizada en el método MFGMs-FEM. Se muestra la malla fina de referencia, la distribución heterogénea del coeficiente de permeabilidad y la nube de puntos empleada para definir los vecindarios locales.

La construcción de esta discretización meshfree permite definir vecindarios locales asociados a la heterogeneidad del medio. Sobre dichos vecindarios se construirán las funciones base multiescala mediante problemas espectrales locales.

8.3. Cálculo de las funciones base multiescala

Para capturar las heterogeneidades locales del coeficiente $\kappa(x)$, se resuelven problemas espectrales locales en cada vecindario grueso S_i . En particular, se buscan autofunciones $\psi_k^{S_i}$ y autovalores $\lambda_k^{S_i}$ tales que

$$-\nabla \cdot (\kappa(x) \nabla \psi_k^{S_i}) = \lambda_k^{S_i} \kappa(x) \psi_k^{S_i}, \quad x \in S_i, \quad (8.2)$$

con condiciones de contorno de Neumann homogéneas sobre ∂S_i .

Los autovalores se ordenan de menor a mayor y se seleccionan los primeros N_b modos dominantes para construir el espacio multiescala local. Estas funciones contienen la información principal asociada a las variaciones multiescala de la permeabilidad.

Para garantizar continuidad global entre los vecindarios, se introducen funciones de partición de unidad $W_i(x)$, definidas mediante funciones spline cúbicas de soporte compacto:

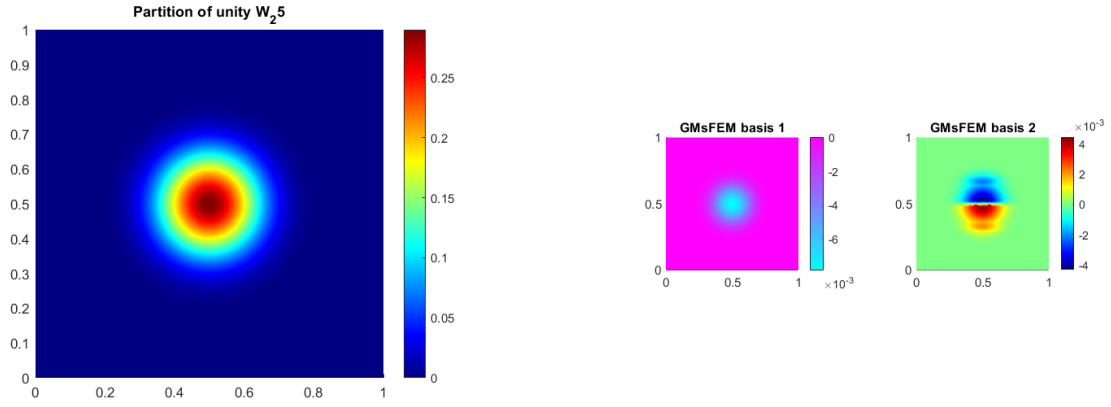
$$W_i(x) = \frac{\phi_i(x)}{\sum_j \phi_j(x)}. \quad (8.3)$$

Donde $\phi_i(x)$ es una función spline cúbica de soporte compacto. Un ejemplo de función de partición de unidad asociada a un vecindario local se muestra en la Figura 8.2a.

Las funciones base multiescala se construyen entonces como

$$\psi_k^{ms} = W_i \psi_k^{S_i}, \quad 1 \leq i \leq N_H, \quad 1 \leq k \leq N_b. \quad (8.4)$$

Ejemplos de funciones base multiescala obtenidas mediante el método MFGMsFEM se presentan en la Figura 8.2b.



(a) Función de partición de unidad $W_i(x)$.

(b) Funciones base multiescala obtenidas mediante MFGMsFEM.

Figura 8.2: Construcción de funciones base multiescala en el método MFGMsFEM. Se muestra una función de partición de unidad asociada a un vecindario local y ejemplos de funciones base multiescala obtenidas mediante problemas espectrales locales.

Finalmente, el espacio multiescala reducido queda definido por

$$V_0 = \text{span} \{ \psi_k^{ms} \}. \quad (8.5)$$

8.4. Formulación variacional en la escala gruesa

El problema de Darcy se proyecta sobre el espacio multiescala V_0 , obteniendo la formulación reducida: encontrar $p_{ms} \in V_0$ tal que

$$a(p_{ms}, v) = F(v_{ms}), \quad \forall v \in V_0, \quad (8.6)$$

donde

$$a(p_{ms}, v) = \int_D \kappa(x) \nabla p_{ms} \cdot \nabla v \, dx, \quad F(v) = \int_D f v \, dx. \quad (8.7)$$

Introduciendo la matriz de proyección multiescala R_0 , las matrices reducidas se obtienen mediante

$$A_0 = R_0 A R_0^T, \quad b_0 = R_0 b. \quad (8.8)$$

Por tanto, el sistema reducido en la escala gruesa queda dado por

$$A_0 p_{ms} = b_0. \quad (8.9)$$

El sistema final en la escala gruesa es:

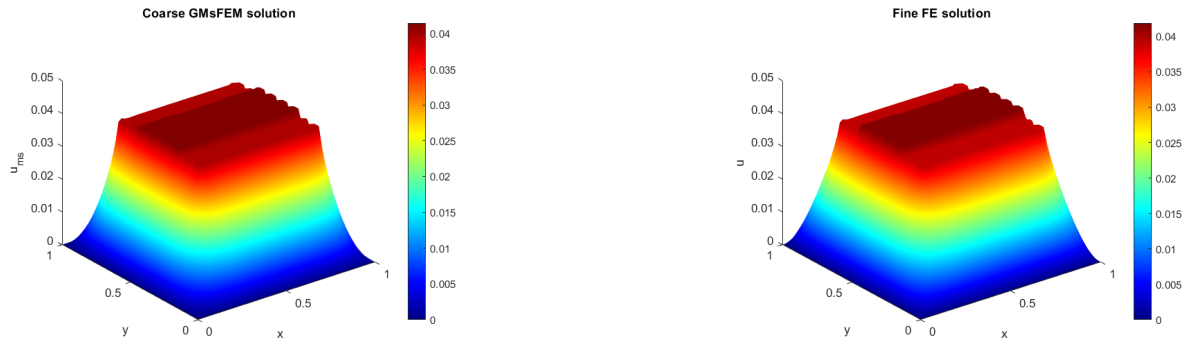
$$A_0 p_{ms} = b_0. \quad (8.10)$$

8.5. Recuperación de la solución a la escala fina

Una vez resuelto el sistema multiescala, la solución en la escala fina se obtiene mediante la interpolación con R_0^T :

$$p \approx R_0^T p_{ms}. \quad (8.11)$$

De esta manera, el método permite aproximar eficientemente soluciones de problemas con coeficientes altamente heterogéneos utilizando un número significativamente menor de grados de libertad respecto al problema original.



(a) Solución multiescala obtenida mediante GMSFEM.

(b) Solución de referencia mediante elementos finitos en malla fina.

Figura 8.3: Comparación entre la solución multiescala obtenida mediante el método GMSFEM y la solución de referencia calculada sobre una malla fina.

Las aproximaciones mostradas fueron obtenidas considerando un problema bidimensional de Darcy con coeficientes altamente heterogéneos. La solución multiescala fue calculada mediante el método GMSFEM sin malla gruesa utilizando funciones base construidas a partir de problemas espectrales locales, mientras que la solución de referencia fue obtenida

resolviendo el problema completo sobre la malla fina.

Capítulo 9

Conclusiones y trabajos futuros

En esta tesis se estudió el Método Generalizado de Elementos Finitos Multiescala (GMsFEM) para la aproximación numérica de la ecuación de Darcy en medios porosos heterogéneos. El trabajo integró aspectos teóricos, variacionales y computacionales relacionados con el método de elementos finitos clásico y su extensión multiescala, enfatizando la construcción de funciones base enriquecidas capaces de capturar información local asociada a coeficientes altamente heterogéneos.

Desde el punto de vista teórico, se presentaron fundamentos de análisis funcional, formulación variacional y métodos de Galerkin necesarios para justificar el marco matemático del método de elementos finitos. En particular, la formulación débil del problema elíptico permitió establecer el problema en espacios de Sobolev apropiados y garantizar propiedades de existencia, unicidad y estabilidad mediante el teorema de Lax–Milgram.

La formulación multiescala desarrollada mediante GMsFEM mostró que la incorporación de funciones base obtenidas a partir de problemas espectrales locales permite construir espacios de aproximación enriquecidos capaces de representar las oscilaciones inducidas por coeficientes de permeabilidad con altos contrastes. La selección de autofunciones asociadas a los autovalores dominantes permitió reducir significativamente la dimensión del problema global sin perder precisión en la aproximación de la solución fina.

Los experimentos numéricos realizados sobre medios heterogéneos sintéticos y sobre el benchmark internacional SPE10, específicamente en la capa 85, evidenciaron que el método reproduce adecuadamente el comportamiento de la solución de referencia calculada sobre malla fina. En particular, se observó una disminución monótona de los errores relativos en norma L^2 y norma de energía al incrementar el número de funciones base adicionales por vecindad, lo cual confirma numéricamente la convergencia del espacio multiescala enriquecido hacia la solución fina.

Los resultados obtenidos muestran además que el parámetro correspondiente al número

de funciones base adicionales influye directamente en la precisión y el costo computacional del método. Se identificó experimentalmente que valores de $m \in [12, 24]$ proporcionan un balance adecuado entre reducción del error y eficiencia computacional para los casos considerados. Aunque el incremento de funciones base mejora la aproximación, también incrementa el tamaño del sistema multiescala y, por consiguiente, el tiempo de ensamblaje y resolución del sistema lineal global.

La implementación computacional realizada en Julia permitió desarrollar de manera eficiente tanto el método de elementos finitos clásico como el método GMsFEM. El uso de Julia demostró ser adecuado para la manipulación de estructuras matriciales dispersas, el ensamblaje de sistemas lineales y la implementación de problemas espectrales locales asociados a métodos multiescala.

Los resultados obtenidos muestran que GMsFEM establece un esquema multiescala robusto y eficiente para la aproximación de problemas con coeficientes altamente heterogéneos ya que combina reducción de costo computacional y precisión numérica.

Adicionalmente, se presentó un estudio complementario relacionado con una formulación del método GMsFEM sin malla gruesa. Este enfoque mostró que es posible construir funciones base multiescala utilizando vecindades locales definidas mediante nubes de puntos, evitando la dependencia explícita de una malla gruesa estructurada. Esta formulación constituye una alternativa prometedora para problemas definidos sobre geometrías complejas o dominios donde la generación de mallas resulta costosa.

Finalmente, los resultados obtenidos permiten concluir que el método GMsFEM constituye una herramienta numérica eficiente para la simulación de flujo en medios porosos altamente heterogéneos. El método logra reducir significativamente el número de grados de libertad respecto a discretizaciones finas globales, conservando una aproximación precisa de la solución y capturando adecuadamente los efectos multiescala inducidos por el coeficiente de permeabilidad.

El material presentado en este trabajo permitió además contribuir en la publicación de tres trabajos en donde se aplican de manera exitosa estas técnicas. En un primer trabajo [20], se realizó un estudio comparativo entre el método de elementos finitos clásico (FEM), el método multiescala de elementos finitos (MsFEM) y el método generalizado de elemen-

tos finitos multiescala (GMsFEM) para problemas de difusión en medios heterogéneos. Asimismo, se desarrollaron implementaciones computacionales en FreeFem++ y MATLAB, analizando el comportamiento de los métodos en medios homogéneos, heterogéneos y de alto contraste.

En un segundo trabajo, [22], se propuso un enfoque combinado entre el método generalizado multiescala de elementos finitos sin malla (MFGMsFEM) y la integración temporal exponencial para resolver problemas parabólicos en medios porosos heterogéneos multiescala. El trabajo desarrolla una aproximación espacial robusta sin requerir una malla gruesa estructurada, además de un análisis riguroso de convergencia y estabilidad. Los resultados numéricos muestran reducciones significativas en el costo computacional y mejoras en estabilidad para problemas multiescala de alto contraste.

En un tercer trabajo [23] se extendió el enfoque MFGMsFEM con integración exponencial hacia problemas tridimensionales de advección-difusión en medios heterogéneos y de alto contraste. El trabajo incorporó nuevas funciones base multiescala con efectos de advección y mostró, mediante experimentos numéricos, mejoras en estabilidad y eficiencia para simulaciones multiescala tridimensionales.

La comunidad de investigadores que desarrolla y aplica el método GMsFEM continúa en crecimiento. Este método ha sido aplicado a distintos tipos de ecuaciones diferenciales en medios heterogéneos o con parámetros heterogéneos, debido a su capacidad para acelerar el cálculo de soluciones en este tipo de problemas. Además, GMsFEM es un método flexible que puede adaptarse a diferentes discretizaciones y formulaciones numéricas. Por ejemplo, podrían estudiarse versiones donde la discretización fina se construya mediante esquemas de diferencias finitas o mediante discretizaciones no conformes. Varias de estas ideas surgieron en conversaciones con el profesor Juan Galvis sobre posibles direcciones interesantes de trabajo futuro.

Otra dirección es el uso de precisión mixta en los cálculos de las funciones base o en el cálculo de la solución multiescala. Este enfoque todavía no ha sido explorado y representa una dirección de investigación interesante, ya que requiere analizar el efecto de la precisión numérica sobre la estabilidad y la precisión del método.

Bibliografía

- [1] E. Abreu, C. Díaz, and J. Galvis. A convergence analysis of generalized multiscale finite element methods. *Computer Physics Communications*, 250:107074, 2020. Sección 8.2: estimación a priori basada en autovalores locales.
- [2] S. Axler. *Linear Algebra Done Right*. Springer, Cham, fourth edition, 2024. Open access book.
- [3] S. Badia and F. Verdugo. Gridap: An extensible finite element toolbox in julia. *Journal of Open Source Software*, 5(52):2520, 2020.
- [4] D. Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2001.
- [5] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, 3rd edition, 2008.
- [6] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Universitext. Springer, New York, NY, 1st edition, 2010.
- [7] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. In *SPE Reservoir Simulation Symposium*, 2001. Modelo 2 con formaciones Tarbert y Upper Ness; malla $60 \times 220 \times 85$.
- [8] E. T. Chung, Y. Efendiev, and G. Li. An adaptive generalized multiscale finite element method for high-contrast flow problems. *Journal of Computational Physics*, 250:477–489, 2013. Sección 4: estimación a posteriori e indicadores adaptativos.
- [9] S. Digital. Access to the spe10 benchmark case. <https://www.sintef.no/contentassets/2551f5f85547478590ceca14bc13ad51/spe10.html>, 2020. Descripción y descarga del modelo SPE10; uso común de cortes por capa.
- [10] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized multiscale finite element methods (gmsfem). *Journal of Computational Physics*, 251:116–135, 2013.
- [11] Y. Efendiev and T. Y. Hou. *Multiscale Finite Element Methods: Theory and Applications*, volume 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*. Springer Science + Business Media, New York, NY, USA, 2009.

- [12] Y. R. Efendiev, T. Y. Hou, and X.-H. Wu. Convergence of a nonconforming multiscale finite element method. *SIAM Journal on Numerical Analysis*, 37(3):888–910, 2000.
- [13] S. Fu, R. Altmann, E. T. Chung, R. Maier, D. Peterseim, and S.-M. Pun. Mesh-free generalized multiscale finite element method. *Computers & Mathematics with Applications*, 74(5):1207–1221, 2016.
- [14] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling & Simulation*, 8(4):1461–1483, 2010.
- [15] L. F. C. Hernandez. On generalized multiscale methods for flow in complex porous media and their applications. Master’s thesis, Universidad Nacional de Colombia, Bogotá, Colombia, 2022.
- [16] J. D. Hernández Ramírez. Método de elementos finitos para la ecuación de reacción-difusión en medios heterogéneos. *Boletín de Matemáticas*, 25(2):123–138, 2018. Disponible en Dialnet.
- [17] T. Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169–189, 1997.
- [18] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press Lecture Notes. Cambridge University Press, Cambridge, UK, 3rd edition, 1987.
- [19] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, New York, 1978.
- [20] A. D. F. Laiton, Y. K. S. Perdomo, and Y. V. T. Ladino. Estudio e implementación del método multiescala generalizado (gmsfem). *Portal de la Ciencia*, 2026. Aceptado para publicación.
- [21] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*, volume 10 of *Texts in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2013.

- [22] D. Nikiforov, L. A. Poveda, D. Ammosov, Y. Sarmiento, and J. Galvis. Meshfree generalized multiscale exponential integration method for parabolic problems. *Journal of Computational and Applied Mathematics*, 459:116367, May 2025.
- [23] D. Nikiforov, L. A. Poveda, D. Ammosov, Y. Sarmiento, J. Galvis, and M. Al Kobaisi. Meshfree gmsfem-based exponential integration for multiscale 3d advection–diffusion problems. *arXiv preprint arXiv:2604.08347*, 2026.
- [24] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, UK, 1st edition, 2003.

Apéndice A

Implementaciones

En este anexo se recogen las implementaciones realizadas a lo largo del trabajo para distintos niveles del método de elementos finitos y extensiones multiescala. Cada una de ellas se desarrolló para verificar los principios teóricos, la formulación numérica y el análisis de errores para los casos de estudio.

- **Autor:** Yesy Karina Sarmiento
- **Implementación para:** Método de elementos-finitos bases lagrangianas de grado 1 y grado 2.
- **URL:** <https://github.com/Karinasarmiento1/Metodo-de-elementos-finitos-Julia.git>
- **Implementación para:** Uso de Gridap para la ecuación de Poisson.
- **URL:** <https://github.com/Karinasarmiento1/FEM-multiescala-en-Julia/tree/59a4d67efee290d9a56101292619195a4580449c/Eq.%20Poisson%20en%20Gridap>
- **Implementación para:** GMsFEM para ecuación de Darcy-ermeabilidad sintética.
- **URL:** <https://github.com/Karinasarmiento1/FEM-multiescala-en-Julia/tree/5fa5094c6081b62eef65621b402b9da6f6163ac6/GMsFEM%20caso%20sintetico>
- **Implementación para:** GMsFEM para ecuación de Darcy-SPE10 capa 85.
- **URL:** <https://github.com/Karinasarmiento1/FEM-multiescala-en-Julia/tree/59a4d67efee290d9a56101292619195a4580449c/GMsFEM-SPE10>
- **Implementación para:** GMsFEM sin malla.
- **URL:** <https://github.com/Karinasarmiento1/FEM-multiescala-en-Julia/tree/152ea6052be14ab1f798eab483f15073905c0867/GMsFEM%20sin%20malla>

Todos los códigos se probaron y están disponibles en los repositorios de GitHub que se enumeran en este apéndice para facilitar su revisión, ejecución y extensión en trabajos futuros.