



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIAS DE LA TIERRA
INSTITUTO DE GEOFÍSICA

Método del Espacio Vectorial Derivado (DVS) y Modelación
Computacional en Paralelo del Transporte de Solutos y
problemas Geofísicos Indefinidos

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRA EN CIENCIAS DE LA TIERRA
(MODELACIÓN DE SISTEMAS TERRESTRES)

P R E S E N T A:

FÍS. L A U R A M I N E R V A S T E L L A R A M Í R E Z

DIRECTOR DE TESIS:
Dr. ISMAEL HERRERA REVILLA
INSTITUTO DE GEOFÍSICA

México, D.F. Enero 2014



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

- A mi Tutor Ismael Herrera Revilla por su apoyo, tiempo y entusiasmo para el desarrollo de esta Tesis.
- A mis sinodales:
Guillermo Hernández García
Ernesto Rubio Acosta
Fabián García Nocetti
David Parra Guevara
por sus comentarios y contribuciones para esta Tesis.
- Al Dr. Antonio Carrillo Ledesma y al C.Dr. Alberto Rosas Medina por sus invaluable sugerencias y soporte para este trabajo.
- A mi Familia con mucha gratitud por ser incondicional.
- Con afecto a Allan, Ángel, Jesús, Roberto y José Luis por compartir estos años en la maestría.
- En especial al programa de becas CONACyT por el impulso a mis estudios de Maestría.

Índice general

| | |
|---|-----------|
| Resumen | 1 |
| Introducción | 3 |
| Objetivo de la Tesis | 5 |
| 1. Formulación Axiomática de los Modelos Matemáticos de Sistemas Físicos Macroscópicos | 7 |
| 1.1. Introducción | 8 |
| 1.2. Propiedades extensivas e intensivas | 9 |
| 1.3. Ecuaciones de balance | 11 |
| 1.4. Modelo general de Sistemas de una Fase | 12 |
| 1.5. Modelo general de los Sistemas Multifásicos | 13 |
| 1.6. Modelos matemáticos completos | 14 |
| 2. Procesos de Transporte y sus Modelos | 17 |
| 2.1. La ecuación general del transporte de solutos, por un fluido libre | 18 |
| 2.2. Transporte de un soluto por un fluido en un medio poroso | 22 |
| 3. Métodos de Descomposición de Dominio | 27 |
| 3.1. Introducción | 28 |
| 3.2. Representaciones Matriciales y Vectoriales | 31 |
| 3.3. Métodos sin Traslape | 33 |
| 4. Métodos DVS | 49 |
| 4.1. Introducción | 50 |
| 4.2. Marco Teórico | 51 |
| 4.3. La Teoría DVS | 57 |
| 4.4. El problema DVS con restricciones | 59 |
| 4.5. Procedimientos Numéricos | 64 |
| 5. La Computación de Alto Rendimiento en la Modelación Matemática y Computacional de las Ciencias de la Tierra | 71 |
| 5.1. Introducción | 72 |
| 5.2. Consideraciones Computacionales para la Implementación | 74 |

| | |
|---|------------|
| 5.3. El procesamiento en Paralelo y los Clusters | 76 |
| 5.4. Métodos y equipos utilizados | 82 |
| 6. Aplicaciones Numéricas | 83 |
| 6.1. Introducción | 84 |
| 6.2. Implementación Computacional | 84 |
| 6.3. Consideraciones del problema | 85 |
| 6.4. Definición de los problemas y resultados | 85 |
| Conclusiones y Perspectivas | 96 |
| Apéndice | 97 |
| A. Matrices Simétricas y no Simétricas, Definidas Positivas e Indefnidas | 99 |
| A.1. La Matriz Transpuesta | 100 |
| A.2. Matriz simétrica | 100 |
| A.3. Matriz Diagonal por bloques | 101 |
| A.4. Número de condición de una matriz | 102 |
| B. Solución de Sistemas Lineales | 105 |
| B.1. Eigenvalores y Número de Condición | 106 |
| B.2. CGM: Método de Gradiente Conjugado | 108 |
| B.3. GMRES: El método de Residuo Mínimo Generalizado | 111 |
| C. La Ecuación de Advección-Difusión y su estabilización | 115 |
| C.1. La ecuación de Advección difusión | 116 |
| C.2. Esquema de diferencias contracorriente | 117 |
| C.3. El caso 2D | 119 |
| C.4. El caso 3D | 121 |
| Referencias | 122 |

Resumen

Los fenómenos de transporte pueden entenderse como los procesos en los que existe una transferencia de materia, energía o momento lineal en un sistema físico. En particular estos fenómenos son de suma importancia en el ámbito de las Ciencias de la Tierra, por tal motivo este trabajo aborda el estudio de las ecuaciones diferenciales parciales que describen el transporte de solutos y energía por fluidos.

El conjunto de estas ecuaciones describen el transporte de sustancias disueltas (solutos) por un fluido, lo cual es esencial para el desarrollo de modelos de dispersión de contaminantes en cauces de ríos, atmósfera, yacimientos petroleros entre otros. Para poder obtener un estudio completo de estos sistemas físicos, es imprescindible poseer el modelo matemático constituido por sistemas de ecuaciones diferenciales parciales, el cual se complementa con información técnica y científica del fenómeno en cuestión.

Es importante acentuar que los sistemas de ecuaciones diferenciales parciales que comúnmente generan los problemas relacionados con las Ciencias de la Tierra, suelen tener un gran número de grados de libertad. Lo anterior aunado a los valores iniciales y de frontera que debe de poseer el modelo, hace muy complicado (salvo para casos muy sencillos) obtener la solución analítica de los sistemas.

Debido a los escasos problemas de esta índole que pueden resolverse por métodos analíticos, ha sido de suma importancia para la materia el desarrollo de los métodos numéricos y los procesos computacionales. También se han introducido los métodos de descomposición de dominio -Domain Decomposition Methods(DDM)-, los cuales hacen más eficiente la solución de sistemas de ecuaciones diferenciales parciales. De forma general, estos métodos hacen una partición del dominio del problema en varios subdominios, en los cuales puede existir o no traslape; al final el problema es reformulado en términos de cada subdominio (véase [2] y [3]).

Recientemente con el objetivo de abordar los DDM de manera más eficiente, se ha introducido el Espacio Vectorial Derivado (Herrera, 2011), por sus siglas en inglés (DVS). Este marco algebraico presume (véase [9]) hacer más eficiente la forma en que se opera dicha Descomposición de Dominio del problema a tratar.

En este trabajo se busca aplicar este nuevo marco algebraico a ejemplos existentes en la literatura (véase [37] y [38]) y comparar sus algoritmos con los algoritmos DVS (véase [9]). En particular, en esta tesis es de interés ejemplificar el uso de los algoritmos DVS para problemas de transporte.

Introducción

Por fenómenos de transporte pueden entenderse como los procesos en los que existe una transferencia de materia, energía o momento lineal en un sistema físico. Existen ejemplos de fenómenos de transporte como la conducción de calor, en donde se transfiere energía, el flujo de fluidos que involucra la transferencia de momento, la difusión molecular debida a la transferencia de masa, la radiación y la carga eléctrica que se transfiere en semiconductores. En particular por su relación con las Ciencias de la Tierra, es de interés en este trabajo el estudio de las ecuaciones diferenciales parciales que describen el transporte de solutos y energía por fluidos.

El conjunto de ecuaciones que describen el transporte de sustancias disueltas (solutos) por un fluido son esenciales para el desarrollo de modelos de dispersión de contaminantes en cauces de ríos, atmósfera, yacimientos petroleros entre otros . El estudio de estos modelos es de suma importancia para el desarrollo, gestión y control de los recursos de agua, así como para el tratamiento de aguas residuales, disminución de la contaminación y el uso racional del vital líquido, así mismo dichos modelos son de gran relevancia en la exploración de yacimientos petroleros y en la optimización para su explotación.

Para poder obtener un estudio completo de estos sistemas físicos, es imprescindible poseer el modelo matemático constituido por sistemas de ecuaciones diferenciales parciales, el cual se complementa con información técnica y científica del fenómeno en cuestión.

Cuando se adquiere el modelo matemático de cierto fenómeno físico, a éste se le aplican métodos numéricos pertinentes que hagan posible transformar dicho modelo en ciertos algoritmos que resuelven las ecuaciones que lo describen. Los cálculos que se realicen dependen del sistema a tratar, generalmente en fenómenos vinculados con las Ciencias de la Tierra dichos sistemas suelen tener un gran número de variables en cuestión. Por tanto, poder traducir dichos pasos en código computacional y explotar su rendimiento hace posible su resolución en menor tiempo y con una mayor eficiencia.

El conocimiento teórico y experimental es utilizado para generar modelos matemáticos, que a su vez generarán los modelos numéricos indispensables para la solución del problema. Los modelos numéricos conllevan a modelos computacionales de los cuales se escriben programas para simular el comportamiento de sistemas del mundo real. Esta aproximación es de suma utilidad si la experimentación es difícil, de alto

costo o simplemente imposible.

Hay varios métodos para convertir las ecuaciones diferenciales parciales a un sistema de ecuaciones algebraicas. Los más comunes son el método de diferencias finitas, método de elementos finitos y el método de volumen finito (véase [3]). En la práctica las derivadas temporales son discretizadas casi exclusivamente usando el métodos de diferencias finitas. Las derivadas espaciales son discretizadas usando diferencias finitas, elementos finitos o volumen finito.

La discretización se puede dividir en dos categorías: una forma, que da lugar a los métodos de diferencias finitas, es representar la función por su valor en un conjunto discreto de puntos de malla. La otra forma es el método de los residuos pesados (WRM por su sigla en inglés, véase [13]).

Este método es conceptualmente diferente del método de diferencias finitas pues asume que la solución puede ser representada por un conjunto de funciones de prueba. Cuando el conjunto de funciones de prueba forma un conjunto ortogonal esta discretización da lugar al método espectral (véase [14]). Cuando las funciones de prueba son diferentes de cero solamente en una pequeña parte del dominio este método da lugar al método de elementos finitos. Técnicas híbridas usando ambos tipos de discretización también existen, por ejemplo el método pseudo-espectral. En este trabajo se utiliza el método de diferencias finitas.

El uso de esquemas como Diferencias Finitas, Elemento Finito y Volumen Finito para la solución numérica de ecuaciones diferenciales parciales reducen la generación de un sistema algebraico de ecuaciones cada vez más grande y el problema que conlleva su solución (véase [3]). Desarrollar códigos eficientes para la solución de problemas de gran número de variables, descansa en buena parte en qué tan desarrollados sean los métodos numéricos para sistema algebraicos grandes. Actualmente existen bibliotecas optimizadas para la solución de sistemas lineales que son usadas al implementar una solución computacional. Estas bibliotecas pueden utilizarse en entornos secuenciales y/o paralelos.

Los métodos de descomposición de dominio (DDM) se fundamentan en la idea básica de resolver múltiples problemas de tamaño reducido sobre un sólo subdominio un cierto número de veces, en lugar de resolver un enorme problema sobre el dominio completo. Además parten de la siguiente suposición: Sea un dominio $\Omega \in \mathbb{R}^n$, se puede particionar en E subdominios Ω_i , donde $i = 1, 2, \dots, E$; tales que: $\Omega = \left(\bigcup_{i=1}^E \Omega_i \right)$, tal que puede o no existir traslape entre los subdominios (véase [2] y [3]).

El problema será reformulado mediante el uso de un método de discretización, en términos de cada subdominio. Con esto es posible obtener una familia de problemas

de tamaño pequeño e independientes entre sí. Estos problemas estarán acoplados por medio de la solución en la interfase de los subdominios, la cual es desconocida. De forma general es posible clasificar a los métodos (DDM) en métodos con y sin traslape entre los subdominios; el interés general se ha fijado en métodos sin traslape (véase [4]), debido a su efectividad en una variedad de problemas en Ciencia e Ingeniería.

Realizando consideraciones fenomenológicas o computacionales, podremos designar cierta descomposición de dominio a algún sistema físico de nuestro interés, ésta descomposición puede dividir en dos o más subdominios contiguos nuestro sistema. La descomposición que se realice del dominio afectará la forma en que se implementará el código, así como el lenguaje que será utilizado, debido a su conveniencia, implementación y mantenimiento (véase [12]). Mediante técnicas de programación orientada a objetos pueden implementarse fácilmente los métodos de descomposición de dominio sin traslape para computadoras paralelas.

De esta forma, los métodos de descomposición de dominio introducen desde la formulación matemática del problema una separación natural de las tareas a efectuarse, además simplifican considerablemente la transmisión de información entre los subdominios (véase [2]). Si a esto le asociamos la programación orientada a objetos y el cómputo en paralelo, se forma una aleación poderosa con la cual es posible ayudar a la solución de problemas que poseen una gran cantidad de grados de libertad, los cuales son problemas recurrentes en las Ciencias de la Tierra.

Objetivo de la Tesis

Los Objetivos de esta Tesis serán agrupados en objetivos generales y objetivos particulares, lo cuales se describen a continuación.

Objetivos Generales

- Realizar un desarrollo de los distintos algoritmos de los métodos de descomposición de dominio.
- Mostrar el reciente esquema de descomposición de dominio en el espacio vectorial derivado (véase [6],[7],[8] y[9]), en el cual es posible aplicar las técnicas de descomposición de dominio que se describen en esta Tesis directamente al sistema de matrices que se genera una vez que el sistema de ecuaciones diferenciales parciales es discretizado. Una de las principales características del esquema DVS, es que no es necesario conocer el sistema de ecuaciones o ecuación diferencial que origina las matrices para la aplicación de los métodos descritos en DVS.
- Mostrar la aplicabilidad del esquema DVS a problemas de Transporte, en particular matrices no simétricas.

- Implementar una estabilización para problemas bidimensionales que mejore los resultados obtenidos por el esquema DVS para problemas no simétricos y con números de Péclet grandes.

Objetivos Particulares

- Mostrar los cuatro algoritmos preconditionados de descomposición de dominio sin superposición del esquema DVS.
- Mostrar el comportamiento que tienen estos métodos en la solución de problemas de advección-difusión.
- Compararlos con ejemplos que han sido reportados en la literatura para distintos valores del número de Péclet. véase ([37] y [38])

Capítulo 1

Formulación Axiomática de los Modelos Matemáticos de Sistemas Físicos Macroscópicos

1.1. Introducción

1.1.1. Modelación matemática y computacional: Generalidades

El proceso general de predecir el comportamiento de un sistema es conocido como *modelar*. En este proceso los modelos se construyen para predecir algún comportamiento de un sistema; por lo que se puede decir que un modelo es un sustituto que imita el comportamiento de dicho sistema. Los modelos pueden ser considerados de dos formas, los que se construyen físicamente y los matemáticos. Los modelos matemáticos son los que actualmente se usan con mayor frecuencia, ya que son los más versátiles y económicos. En aplicaciones específicas estos modelos consisten en códigos computacionales que se adaptan fácilmente para poder realizar cambios en las propiedades de los sistemas. Además los fundamentos y los métodos de la aplicación de los modelos matemáticos poseen una generalidad y unidad notable; estas características tienen importantes implicaciones prácticas ya que conducen a un ahorro en el esfuerzo y en el costo de recursos.

Los modelos matemáticos integran conocimiento técnico y científico con el propósito de predecir el comportamiento de los sistemas, dicho conocimiento es incluido dentro de los códigos computacionales que los equipos ejecutan en la aplicación del modelo. Actualmente los modelos matemáticos basados en simulación numérica permiten la predicción del comportamiento de sistemas complejos y fenómenos naturales que de otra forma sería sumamente costoso, peligroso o simplemente imposible por experimentación directa. Podemos decir, que la modelación matemática y computacional es el método más eficiente y efectivo para predecir el comportamiento de sistemas tanto naturales como artificiales de interés humano. Mundialmente, los modelos matemáticos son utilizados ampliamente en ciencia e ingeniería, en la industria petrolera, los sistemas hidrológicos, la predicción climática, manufactura automotriz y aeronáutica, investigación en medicina, economía, química y astronomía.

Existen dos enfoques para el estudio de la materia y su movimiento: el *enfoque microscópico*, que estudia las moléculas, átomos y partículas elementales; y el *enfoque macroscópico*, el cual estudia y modela sistemas que contienen un número colosal de partículas. La predicción del comportamiento de partículas tales como moléculas, átomos y sus componentes es un enfoque microscópico el cual es materia de estudio de la *mecánica cuántica*.

Predecir el comportamiento de sistemas de grandes dimensiones, como yacimientos de petróleo o la atmósfera, los cuales consisten de una enorme cantidad de moléculas y átomos, es una meta inaccesible si se pretende el enfoque microscópico para su estudio. Por lo tanto, para problemas de importancia práctica en ciencia e ingeniería el enfoque macroscópico es el apropiado. La *mecánica del medio continuo* suministra los fundamentos teóricos de este último enfoque. Muchos sistemas de interés en la ingeniería,

las ciencias de la Tierra, ciencias aplicadas y las ciencias en general requieren modelos basados en la mecánica del medio continuo. La aplicación de los conceptos extraídos de la mecánica del medio continuo y de la modelación computacional y matemática ha conducido a logros muy impresionantes y tratado gran diversidad de problemas de la ciencia y la ingeniería. La teoría de *sistemas macroscópicos* puede aplicarse no sólo a sistemas físicos, sino también a sistemas químicos y algunos sistemas biológicos.

Un enfoque axiomático, es el procedimiento más eficiente para lograr la generalidad y la unidad conceptual de los modelos de sistemas físicos a describir. En particular, el uso de axiomas nos permitirá construir un enfoque eficaz para familiarizarse con los modelos más importantes de la ciencia y la ingeniería. El estudio de esta variedad de problemas de una manera unificada por medio de lo que hemos descrito anteriormente como el *enfoque axiomático* produce un gran ahorro de esfuerzo en el proceso de enseñanza-aprendizaje. Además, esta forma de enseñanza y aprendizaje es muy valiosa en la investigación, ya que la formulación unificada asociada contiene pistas para la solución de muchos problemas hasta ahora imprevistos.

Cuando la materia se estudia mediante el punto de vista macroscópico, los cuerpos llenan completamente el espacio que ocupan, de manera que no hay huecos que existan en ellos, a pesar del hecho de que cuando se examinan con la ayuda de un microscopio, es decir, desde el punto de vista microscópico, uno se encuentra con muchos huecos. Así, desde el punto de vista macroscópico, por ejemplo, el agua llena el recipiente que la contiene y nuestro escritorio de trabajo es un trozo continuo delimitado de materia.

Este punto de vista macroscópico está presente en la física clásica, en particular en la mecánica clásica. La ciencia ha avanzado hasta el punto en el que reconocemos que la mayor parte del espacio ocupado por la materia está vacío, los cuales nuestros sentidos no perciben; también que la energía se mueve en paquetes subatómicos, es decir, que se *cuantifica*. A primera vista parece contradictorio el uso de los dos enfoques, microscópico y macroscópico, para el análisis de los sistemas físicos. Sin embargo, no sólo son compatibles, sino que en realidad son complementarios, y se puede establecer una relación entre ellos por medio de la *mecánica estadística*.

1.2. Propiedades extensivas e intensivas

En la mecánica del medio continuo hay una clase de funciones, tales como la densidad o la temperatura, que han de ser estudiadas y que se definen para cada *partícula* de un *cuerpo* y para cada tiempo. En general, cualquier función de este tipo se dice que es una *propiedad intensiva*. Utilizaremos \underline{X} para identificar a las partículas y t el tiempo. Cuando consideramos una propiedad intensiva, escribiremos $\phi(\underline{X}, t)$, para el valor de la función en la partícula \underline{X} en el tiempo t . Esta función, $\phi(\underline{X}, t)$, se dice que es la *representación Lagrangiana* de la propiedad intensiva.

Por otro lado, podemos considerar esta misma propiedad intensiva desde otro pun-

to de vista; sea $\psi(\underline{x}, t)$ el valor de la propiedad intensiva en el punto \underline{x} del espacio físico al tiempo t . La función $\psi(\underline{x}, t)$ se dice que es la *representación Euleriana* de la propiedad intensiva. Existe una condición para que las representaciones Eulerianas y Lagrangianas correspondan a la misma propiedad intensiva; dicha condición es:

$$\phi(\underline{X}, t) \equiv \psi(\underline{p}(\underline{X}, t), t) = \psi(\underline{x}, t) \quad (1.1)$$

Donde $\underline{p}(\underline{X}, t)$ es la posición de la partícula \underline{X} en el tiempo t . Además la ecuación (1.1) es equivalente a:

$$\psi(\underline{x}, t) \equiv \phi(\underline{p}^{-1}(\underline{x}, t), t) \quad (1.2)$$

La representación Lagrangiana se utiliza con más frecuencia en el estudio de los sólidos, mientras que la representación Euleriana se utiliza más a menudo en el estudio de los fluidos. Esto es probablemente debido al hecho de que los desplazamientos de fluidos son generalmente muy grandes, mientras que los desplazamientos de los sólidos son generalmente pequeños. Las propiedades intensivas pueden ser funciones escalares o funciones vectoriales.

Se dice que $E(\mathbf{B}, t)$ (véase[11]) es una propiedad extensiva cuando para cada cuerpo \mathbf{B} se puede expresar como una integral de una propiedad intensiva sobre el cuerpo, es decir,

$$E(\mathbf{B}, t) = \int_{B(t)} \psi(\underline{x}, t) d\underline{x} \quad (1.3)$$

Con respecto a esta definición, podemos notar lo siguiente. Primero, se asume que las representaciones Eulerianas, $\psi(\underline{x}, t)$, de propiedades intensivas son integrables en el sentido de *Lebesgue*. Entonces, la ecuación (1.3) establece una correspondencia uno a uno entre las propiedades extensivas e intensivas porque la función en el integrando de la ecuación (1.3) siempre se puede tomar como la definición de la representación Euleriana de una propiedad intensiva. A la inversa, dado una propiedad intensiva, su integral sobre cada cuerpo define una propiedad extensiva. En segundo lugar, la propiedad intensiva asociada de esta manera con cada propiedad extensiva es la propiedad extensiva por *unidad de volumen*.

El uso de propiedades intensivas por unidad de volumen permite lograr una mayor sencillez teórica y desarrollos más elegantes. En particular, la correspondencia entre las propiedades intensivas y extensivas es más directa. Por ejemplo, cuando una propiedad extensiva está dada por la ecuación (1.3), su propiedad intensiva correspondiente es el integrando en esta ecuación. Además, al tratar con sistemas complejos, tales como multifásicos y sistemas con multicomponentes, diversas masas pueden ocurrir en los sistemas y la aplicación de fórmulas que utilizan las propiedades definidas por unidad de masa pueden ser confusas. En cambio, el volumen del espacio físico está siempre definido de forma única.

1.3. Ecuaciones de balance

Para establecer la ecuación de balance de una propiedad extensiva, E , notemos que el cambio en la propiedad extensiva, ΔE , está dado por:

$$\Delta E = P + I \quad (1.4)$$

Aquí, hemos escrito P para representar la producción de la propiedad extensiva en el interior de un dominio del espacio físico e I para la importación de la misma a través de la frontera de tal dominio. Recordando que las *partículas materiales* contenidas en un dominio del espacio físico constituyen un cuerpo, podemos utilizar las ideas anteriores para establecer las ecuaciones de balance que una propiedad extensiva de un cuerpo debe cumplir. Si ahora introducimos el concepto de tiempo, es decir, el cambio en la propiedad E por unidad de tiempo, se puede usar la ecuación (1.4) para obtener:

$$\frac{dE}{dt}(t) = \int_{B(t)} g(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} q(\underline{x}, t) d\underline{x} \quad (1.5)$$

La primera integral se toma sobre el volumen del dominio $B(t)$ y la segunda es tomada sobre la superficie $\partial B(t)$ del dominio $B(t)$. Aquí $g(\underline{x}, t)$ se denomina *el suministro externo de la propiedad extensiva, por unidad de volumen y por unidad de tiempo*; esto representa el importe de la propiedad extensiva que entra en el cuerpo, en el punto \underline{x} al tiempo t , por unidad de volumen. En cuanto a $q(\underline{x}, t)$, representa *la cantidad de la propiedad extensiva que entra en el cuerpo a través de su frontera, en el punto \underline{x} al tiempo t , por unidad de área*.

Esta ecuación establece que la tasa de aumento de la propiedad E en el cuerpo $B(t)$ es igual a la tasa de introducción de la propiedad E que se interna en el cuerpo $B(t)$ a través de un suministro externo más la cantidad por unidad de tiempo que entra a través de la frontera $\partial B(t)$. Esta relación se denomina generalmente como la *ecuación general de balance global*, que es un concepto fundamental en la formulación de modelos matemáticos de sistemas continuos.

Las ecuaciones de balance local, son ecuaciones expresadas en términos de las propiedades intensivas de los sistemas continuos. La notación $\Sigma(t)$ es usada para denotar un *choque*: es decir, la superficie a través de la cual las propiedades extensivas son discontinuas, aunque en este trabajo no abundaremos en el tema de los saltos por discontinuidades.

Se introduce a continuación una forma más general de la *ecuación de balance global*:

$$\frac{dE}{dt}(t) = \int_{B(t)} \{g(\underline{x}, t) + \nabla \cdot \underline{\tau}(\underline{x}, t)\} d\underline{x} \quad (1.6)$$

Sea $B(t)$ un cuerpo. Entonces la ecuación de balance global,(1.6), tiene que cumplirse para todos los cuerpos sub-contenidos en $B(t)$ si y sólo si, la siguiente condición es satisfecha:

1 La ecuación diferencial

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\underline{v}\psi) = \nabla \cdot \underline{\tau} + g, \quad (1.7)$$

se cumple para toda $\underline{x} \in B(t)$

La ecuación (1.7) se conoce como la ecuación diferencial de "balance local". Hacemos notar que la misma se expresa en términos de propiedades intensivas exclusivamente. Esta sustitución de propiedades extensivas por sus correspondientes propiedades intensivas es factible debido a que la ecuación de balance global (1.6) es equivalente a las ecuaciones de balance local (1.7), la última de éstas no involucra propiedades extensivas, pero sí su propiedad intensiva asociada. En general, los modelos de sistemas macroscópicos pueden contener discontinuidades las cuales no serán discutidas en este Tesis. La ecuación (1.7) es aplicable en regiones en que dichas discontinuidades no están presentes.(véase[11])

1.4. Modelo general de Sistemas de una Fase

La teoría que hemos desarrollado anteriormente se aplica solamente a los sistemas de una fase; en particular, trataremos sistemas clásicos de sólidos y fluidos de una sola fase. Una característica notable de la teoría de los modelos de sistemas continuos de una fase es que en cualquier momento cada punto del espacio físico está ocupado por una, y sólo una partícula. Por lo mismo, la velocidad de las partículas es única en cada punto y cada tiempo. Sin embargo, existen modelos más generales, los cuales son aplicables a una clase más amplia de sistemas comúnmente hallados en la ciencia y la ingeniería, en los cuales pueden estar definidas más de una velocidad en cada punto y tiempo (dónde una gran cantidad de velocidades de partículas pueden estar definidas para cada punto y tiempo).

El procedimiento general para la construcción de modelos matemáticos de una fase puede ser brevemente descrito como un procedimiento de dos pasos:

- 1 Se identifica una familia de propiedades extensivas,
- 2 A cada propiedad extensiva de la familia se le aplican las condiciones de balance.

Los balances de las propiedades extensivas se expresan por medio de ecuaciones diferenciales de balance. El sistema resultante de ecuaciones diferenciales parciales obtenido de esta forma se conoce como el *modelo matemático básico del sistema*. Así que si N es el número de propiedades extensivas del sistema, entonces de acuerdo a la ecuación (1.3), cada una de ellas se puede expresar como:

$$E^\alpha \equiv \int_{B(t)} \psi^\alpha(\underline{x}, t) d\underline{x}, \quad \alpha = 1, \dots, N \quad (1.8)$$

donde ψ^α es la propiedad intensiva correspondiente. Además, cada propiedad extensiva satisface la ecuación de balance global, por tanto:

$$\frac{dE^\alpha}{dt}(t) = \int_{B(t)} g^\alpha(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} \underline{\tau}^\alpha(\underline{x}, t) d\underline{x} \cdot \underline{n}(\underline{x}, t) d\underline{x} \quad (1.9)$$

Por consiguiente, el modelo matemático se obtiene por medio de la aplicación de la ecuación (1.7) para cada propiedad extensiva. Esto produce la ecuación diferencial:

$$\frac{\partial \psi^\alpha}{\partial t} + \nabla \cdot (\underline{v} \psi^\alpha) = \nabla \cdot \underline{\tau} + g^\alpha, \quad \forall \alpha = 1, \dots, N \quad (1.10)$$

Esta ecuación diferencial constituye el modelo matemático básico de un sistema continuo de una fase.

A fin de que un modelo matemático adquiera la capacidad de predecir el comportamiento, es necesario incorporar en él conocimiento científico y tecnológico sobre los sistemas específicos bajo consideración; en general esta información se incorpora en los modelos matemáticos por medio de las llamadas *ecuaciones constitutivas*. Una vez que las ecuaciones constitutivas se han combinado con la ecuación diferencial de balance, desarrollar el modelo del sistema continuo y derivar predicciones de él se convierte en un problema que pertenece a los campos de *ecuaciones diferenciales parciales* y a los *métodos numéricos*. En general, finalmente el modelo numérico así obtenido se implementa por medios computacionales.

1.5. Modelo general de los Sistemas Multifásicos

Muchos modelos importantes en la ciencia y la ingeniería corresponden a sistemas de una sola fase. Sin embargo, hay muchos otros sistemas también relevantes que están definidos por múltiples fases, como yacimientos de petróleo y campos geotérmicos. Así, la extensión de la teoría a sistemas de varias fases, aunque bastante sencilla, es muy importante. Una característica de los sistemas multifásicos es que cada fase se mueve con su propia velocidad. En general, en cada punto del dominio ocupado por un sistema multifásico hay definidas tantas velocidades de las partículas como fases tenga el sistema. Con esta conceptualización los sistemas de una fase son un caso particular de los multifásicos, en el cual el número de fase es igual a uno. El procedimiento general para la construcción de modelos multifásicos se obtiene modificando ligeramente el procedimiento explicado en la Subsección anterior. Esta modificación conduce a un procedimiento de tres pasos:

- 1 Se identifica una familia de propiedades extensivas;
- 2 Cada propiedad extensiva de la familia está asociada con una de las fases del sistema;
- 3 Las condiciones de balance se aplican a cada propiedad extensiva de la familia, utilizando *la velocidad de las partículas de la fase correspondiente*.

La aplicación de esta estrategia al modelo matemático básico del sistema multifásico conduce al siguiente sistema de ecuaciones diferenciales:

$$\frac{\partial \psi^\alpha}{\partial t} + \nabla \cdot (\underline{v}^\alpha \psi^\alpha) = \nabla \cdot \underline{\tau}^\alpha + g^\alpha, \quad \forall \alpha = 1, \dots, N. \quad (1.11)$$

Debe observarse que las velocidades de las partículas que se utilizan en la ecuación (1.11) corresponde a la fase a la que está asociada la propiedad extensiva correspondiente. La información complementaria sobre los sistemas, como anteriormente mencionamos, juega un papel muy importante ya que es el medio por el cual el conocimiento científico y tecnológico se incorpora en el modelo de un sistema específico.

1.6. Modelos matemáticos completos

En general, los ingenieros y los científicos están interesados en el desarrollo de *modelos completos*, es decir, los modelos capaces de predecir el comportamiento del sistema cuando se conocen ciertas condiciones iniciales y de contorno. A fin de obtener tales modelos, es necesario tener información adecuada sobre el sistema que se está modelando, lo que incluye como hemos visto, conocimientos científicos y tecnológicos; así como las mediciones de laboratorio y de campo de las propiedades del sistema. En particular, las fórmulas conocidas como ecuaciones constitutivas son construidas por medio de teorías científicas y tecnológicas, incluso con el apoyo de experimentos. Estas ecuaciones permiten expresar a $g^\alpha(\underline{x}, t)$ y $\underline{\tau}^\alpha(\underline{x}, t)$ en términos de las propiedades intensivas, que se producen en el modelo y son necesarias; a través de ellas el conocimiento científico y tecnológico sobre el sistema se incorpora en el modelo. En algunos casos, como en ciertos modelos de transporte de solutos, la información sobre las velocidades de las partículas $\underline{v}^\alpha(\underline{x}, t)$ también es necesaria para transformar el modelo matemático básico en un modelo completo.

Como ya se menciona es necesario conocer la velocidad del fluido para considerar un *modelo completo*. Por ejemplo, si el problema involucra a la contaminación atmosférica, la información de la velocidad podría ser suministrada por una red de estaciones de medición del viento; de manera similar, si se está modelando el transporte de los contaminantes en una corriente de agua, se obtienen las mediciones de velocidad del agua. La información acerca de la fuente externa g puede estar dada por ecuaciones constitutivas derivadas de las leyes físicas o químicas, como las leyes del decaimiento nuclear, balances químicos o reacciones químicas. En fluidos en reposo o que se mueven muy lentamente (gases o líquidos), frecuentemente la ecuación constitutiva más utilizada, relativa al flujo difusivo de la distribución de la concentración, es la ley de Fick. Para fluidos que exhiben un flujo no uniforme la relación de flujo puede ser más compleja.

Además, la solución del modelo matemático también requiere condiciones de frontera. En general, hay tres tipos de condiciones de frontera:

- a. La propiedad intensiva ψ se especifica a lo largo de una frontera, llamada la *condición de Dirichlet*.
- b. Se especifican los gradientes de la propiedad intensiva $\frac{\partial\psi}{\partial t}$ a través de una frontera, llamada la *condición de Neumann*.
- c. Los dos valores de la propiedad intensiva ψ a lo largo de una frontera y los gradientes de la propiedad intensiva $\frac{\partial\psi}{\partial t}$ a través de esa frontera se especifican, rindiendo una combinación de (b.) llamada *condición de Cauchy*.

Los métodos para la obtención de la solución de un modelo matemático se puede dividir en dos clases, analíticos y numéricos, a pesar de que no es poco común un híbrido de estos dos métodos. En general, las soluciones analíticas sólo pueden obtenerse en virtud de la simplificación de muchos supuestos.

Capítulo 2

Procesos de Transporte y sus Modelos

2.1. La ecuación general del transporte de solutos, por un fluido libre

Para formular un modelo completo de transporte, es necesario conocer de antemano las velocidades de las partículas del fluido como funciones de la posición y del tiempo. Un procedimiento que se usa frecuentemente para cumplir con este requisito es calcular las velocidades de las partículas por medio de modelos de flujo y una vez que se obtienen, se utilizan en los modelos de transporte. Los modelos de flujo para fluidos libres se basan en la teoría de la dinámica de fluidos y los modelos de flujo de fluidos en un medio poroso. Sin embargo, en muchos casos no es posible calcular las velocidades de fluido por medio de un modelo de flujo, en cuyo caso se necesita realizar mediciones directas de las velocidades de fluido, para lo cual frecuentemente se establece una red de monitoreo. Por ejemplo, cuando se modela el transporte de contaminantes por un río o por la atmósfera, los datos sobre las velocidades del fluido se obtienen a través de mediciones que se llevan a cabo en estaciones de vigilancia que se establecen para este propósito.

En algunos problemas, la concentración de los solutos es tan pequeña que la densidad del fluido es esencialmente independiente de la concentración. Sin embargo, en otros casos es tan grande que la densidad es altamente dependiente de la concentración, y en este último caso, cuando las fuerzas en el cuerpo tales como la gravedad están actuando en el sistema continuo, estos cambios de densidad modifican las condiciones de flujo de manera que las ecuaciones de transporte y flujo son necesariamente acopladas y necesitan ser tratadas simultáneamente. Además, el sistema resultante no es lineal, es decir, la concentración depende de la velocidad y la velocidad depende de la concentración de tal manera que las ecuaciones de transporte y flujo son acopladas. Algo similar sucede cuando se trata de transporte de energía, cuando los cambios en la temperatura son grandes, la densidad del fluido es fuertemente dependiente de la temperatura. Entonces, las ecuaciones de transporte de energía y flujo están también acopladas y deben resolverse simultáneamente. En particular, sólo expondremos el caso en que las ecuaciones de transporte y de flujo están desacopladas, es decir, en donde se considere que la concentración tiene un impacto despreciable en la velocidad.

En el modelo de transporte de solutos por un fluido libre, la masa del soluto es una propiedad extensiva dada a cualquier tiempo t como:

$$M_s \equiv \int_{B(t)} c(\underline{x}, t) d\underline{x} \quad (2.1)$$

En ausencia de choques, la ecuación de balance global es:

$$\frac{dM_s}{dt} = \int_{B(t)} g_s(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} \underline{\tau}_s(\underline{x}, t) \cdot \underline{n} d\underline{x} \quad (2.2)$$

La fuente externa del soluto a consideración, $g_s(\underline{x}, t)$, puede ser diferente a cero por muchas razones, por ejemplo puede ser atribuido a un decaimiento radiactivo. La expresión $\underline{\tau}_s(\underline{x}, t) \cdot \underline{n}$, en la ecuación (2.2), representa la masa por soluto por unidad de área, por unidad de tiempo, que penetra en el cuerpo del fluido. En el caso de un fluido libre el flujo de soluto-masa, $\underline{\tau}_s(\underline{x}, t)$, ocurre principalmente debido a la difusión molecular.

La ecuación diferencial del balance local es:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\underline{v}c) = \nabla \cdot \underline{\tau}_s + g_s \quad (2.3)$$

donde $c(\underline{x}, t)$ es la concentración del soluto y es la propiedad intensiva asociada con la masa del soluto; esto es la masa del soluto por unidad de volumen de solución. Reiteramos aquí, que para poder obtener un modelo completo del transporte de un soluto, la ecuación (2.3) necesita ser complementada con información acerca de la velocidad de la partícula. También es necesario suministrar información acerca de $\underline{\tau}_s$ y g_s . Generalmente, esta última información se suministra por medio de ecuaciones constitutivas. Por ejemplo, en el caso de $\underline{\tau}_s$, como veremos más adelante, en el transporte de solutos por un fluido libre se utiliza la ecuación de Fick.

2.1.1. Proceso de Transporte

La ecuación (2.3) es bastante significativa ya que es la base de muchos tipos de transporte de sustancias diluidas en un fluido libre, como contaminantes que pueden existir en un cuerpo de agua superficial o en la atmósfera. Cabe señalar, que la ecuación (2.3) todavía no es el modelo completo, de hecho es solamente lo que hemos llamado el modelo matemático básico; esto se refiere a el sistema básico en el cual es necesario incorporar información científica y tecnológica disponible acerca de los procesos que ocurren en el sistema de transporte. Tres procesos se distinguirán: advección, difusión y generación de masa.

2.1.2. Advección

Siempre que el fluido está en movimiento la *advección* ocurre, esto es, siempre cuando la velocidad de partícula es distinta de cero $\underline{v} \neq 0$. Este fenómeno, o proceso, es debido al hecho de que la sustancia disuelta es arrastrada por el fluido mientras se mueve, justo como un pasajero es llevado por el autobús. La fuerza del proceso advectivo es caracterizado por la velocidad del fluido, la cual en los modelos de transporte que hemos considerado es considerado como dato. Así, para aplicar la ecuación (2.3) para modelar el transporte de un soluto, es necesario reunir información acerca de la velocidad de partícula por medios adecuados.

2.1.3. Proceso de Difusión

El hecho de que un fluido esté macroscópicamente en reposo no implica que las partículas microscópicas que lo componen estén también en reposo. Es un hecho que las partículas microscópicas que constituyen un fluido (átomos, iones o moléculas) están en agitación permanente y las partículas de soluto que las acompañan efectúan trayectorias aleatorias conocidas como *movimiento Browniano*. Generalmente, la masa de las partículas de soluto que dejan el cuerpo del fluido debido a este movimiento Browniano, no es igual a la masa que entra en él. Este desbalance produce un *flujo de masa* a través de la frontera del cuerpo y en física macroscópica los fenómenos de este tipo son conocidos como *procesos de difusión*. El proceso de difusión que exclusivamente es debido al movimiento Browniano es conocido como *difusión molecular*.

Generalmente, al paso del tiempo, el proceso de difusión tiende a suavizar cualquier rugosidad en la distribución espacial de la concentración. Un simple experimento que evidencia la ocurrencia del proceso de difusión, consiste en depositar una gota de tinta en un recipiente que contenga agua en reposo. Así, al poco tiempo la mancha de tinta se extiende, haciéndose cada vez más grande y después de un largo lapso de tiempo la concentración de la tinta se distribuye tan uniformemente que es imperceptible al ojo humano. En los fluidos libres, los movimientos Brownianos están siempre presentes, es por esto que el transporte de solutos en estos fluidos es siempre difusivo. Sin embargo, cuando el proceso de difusión es muy pequeño en comparación con otros procesos que se estén llevando a cabo, se pueden tener resultados satisfactorios usando modelos que ignoren dichos procesos de difusión. Tales modelos de transporte son llamados *no difusivos*.

Un resultado bien conocido de álgebra lineal muestra que la transformación lineal más general de un vector siempre se puede expresar como el producto del vector y una matriz. Un modelo muy simple para la difusión molecular y que es ampliamente utilizado, es la *Primera ley de Fick*; la cual asume que el campo vectorial que representa el *flujo de masa de soluto*, $\underline{\tau}_S(\underline{x}, t)$, es una función lineal del gradiente de la concentración. Por lo tanto, primera ley de Fick establece que:

$$\underline{\tau}_S(\underline{x}, t) = \underline{\underline{D}} \cdot \nabla c \quad (2.4)$$

Aquí $\underline{\underline{D}}$ es una matriz, conocida como el *tensor de difusión molecular*, el cual es simétrico y no negativo. La ecuación (2.4) es una forma muy general de la primera ley de Fick; por lo general, para fluidos libres la difusión molecular es isotrópica, en cuyo caso no hay dirección preferida del proceso de difusión en el espacio físico. Cuando los procesos de difusión son isotrópicos, la ecuación (2.4) se reduce a

$$\underline{\underline{D}} \equiv D \underline{\underline{I}} \quad (2.5)$$

Aquí D es un escalar conocido como el *coeficiente de difusión*. Para la difusión

isotrópica, el tensor de difusión molecular es no negativo sí y sólo si el coeficiente de difusión es no negativo. Se observa que cuando \underline{D} es despreciable, se cumple la suposición de que el flujo de masa de soluto es una función lineal del gradiente de concentración, por lo que la primera ley de Fick incluye el transporte no difusivo como caso particular.

2.1.4. Proceso de Generación de masa

La razón con la cual la masa es generada es determinada por la fuente externa $g_S(\underline{x}, t)$. Cuando la fuente externa es cero, no hay masa generada y cada cuerpo del fluido conserva la masa de soluto que posee, exceptuando por la masa que llega a través de su frontera debido a difusión. En este caso, el sistema de transporte se dice que es *conservativo*.

Por otra parte, la masa de soluto no se conserva siempre que la fuente externa de soluto es diferente a cero, $g_S(\underline{x}, t) \neq 0$ y tal sistema de transporte se dice que es *no conservativo*. Informalmente, uno dice que existe una *fuentes de masa* o un *sumidero de masa* cuando $g_S(\underline{x}, t) > 0$ o $g_S(\underline{x}, t) < 0$, respectivamente. Los orígenes de esas fuentes y sumideros son diversos, por ejemplo, dos de los que son especialmente significantes son el decaimiento radiactivo y reacciones químicas entre solutos diferentes contenidos en el fluido. Sin embargo, el suministro externo de soluto puede tener fuentes menos ortodoxas. Por ejemplo, en aplicaciones al estudio de transporte de contaminantes por un río, el suministro externo de contaminantes puede provenir de la actividad humana provocada por asentamientos en los bordes de un río.

Un principio básico en Física es la conservación de masa. Así, se puede observar que el hecho de que $g_S(\underline{x}, t) \neq 0$ no significa que ese principio ha sido violado, ya que el principio de conservación de masa se aplica a sistemas aislados solamente. El balance representado por la ecuación (2.2) es válido sólo para la masa de la substancia que ha sido disuelta en el fluido y tiene precisamente la forma química del soluto. Si nosotros nos ocupáramos del balance de masa total de un sistema aislado que contiene un sistema de soluto-fluido, encontraríamos que el principio de conservación de masa se satisface para el sistema. En otras palabras, la masa que se encuentra en el sistema aislado en otra forma química puede transformarse en un compuesto químico que constituye la substancia disolvente. Un ejemplo típico que ilustra este hecho es el decaimiento radiactivo; cuando el decaimiento radiactivo acontece, el principio de conservación de masa no es violado; todo lo que sucede es que la masa de la substancia radiactiva que desaparece se transforma en masa de las partículas que constituyen la radiación, ya sea la radiación *alfa* o *beta*; ya que la radiación *gamma* es no masiva.

2.2. Transporte de un soluto por un fluido en un medio poroso

2.2.1. Supuestos básicos del Modelo de flujo

El modelo de flujo que se presenta en esta sección se basa en un conjunto de supuestos que son ampliamente utilizados en hidrología subterránea. A continuación se listan.

- La matriz porosa está saturada con el fluido. Esto significa que el fluido llena completamente el espacio de los poros de la matriz sólida;
- La matriz sólida permanece en reposo durante todo el proceso de flujo de fluido;
- La matriz sólida es elástica. Más precisamente, la porosidad de la matriz es una función de la presión del fluido. Por lo tanto, la porosidad de la matriz puede cambiar a medida que pasa el tiempo. Esto, a pesar del hecho de que la velocidad macroscópica de la matriz sólida es cero a lo largo de todo el proceso;
- El fluido es compresible. Específicamente, la densidad del fluido satisface una ecuación de estado en la cual la densidad es una función de la presión, exclusivamente;
- La velocidad de las partículas del fluido cumple con la ley de Darcy. Esta es una ecuación constitutiva empírica, que relaciona la velocidad de las partículas a la distribución de la presión del fluido;
- El líquido no se somete a procesos de difusión, de modo que τ es igual a cero.

La presión del fluido también se conoce como la presión de poro, especialmente en mecánica de suelos. La presión de poro empuja sobre las superficies de los granos, ejerciendo una fuerza que tiende a aumentar el volumen de los poros, al igual que un globo se infla bajo la presión del aire contenido en su interior. En general, los granos se mueven en respuesta a los cambios en la presión de fluido de tal manera que la porosidad aumenta a medida que aumenta la presión del fluido.

Al mismo tiempo, una disminución en la presión de poro resulta en una disminución en el volumen de éste, dando lugar a la consolidación. La consolidación puede tener importancia práctica, ya que puede conducir a la deformación del suelo en la superficie de la Tierra con la inestabilidad estructural asociada.

En cuanto a la densidad del fluido, por lo general es una función no sólo de su presión, sino también de la temperatura, lo que implica que la temperatura, en sí, puede ser una función de la presión. Dicha función está determinada por el tipo de procesos termodinámicos que el fluido realiza durante su movimiento. Dos procesos alternativos se suelen suponer: que tal proceso es isotérmico (la temperatura no cambia) o es adiabático (no hay intercambio de calor). Sin embargo, la suposición aquí adoptada

es que la densidad sólo es una función de su presión. Además, la densidad del fluido puede ser una función de la concentración de las distintas especies disueltas. Por lo tanto, en el supuesto de que la densidad es una función de la presión solamente, estamos aceptando implícitamente la dependencia insignificante de la densidad del fluido en la concentración de especies.

2.2.2. El modelo básico para el flujo de un fluido en un medio poroso

En el caso del modelo de un fluido en un medio poroso usualmente se adoptan ciertas suposiciones tales como que el medio poroso está saturado por el fluido y que la masa del fluido se conserva. Generalmente el propósito final de este tipo de modelo es predecir el flujo del fluido (esto es, la velocidad del fluido). El modelo se basa solamente en un propiedad extensiva, la masa del fluido dada por:

$$M_f \equiv \int_{B(t)} \varepsilon(\underline{x}, t) \rho(\underline{x}, t) d\underline{x} \quad (2.6)$$

Donde, $\rho(\underline{x}, t)$ es la densidad del fluido. Claramente podemos notar que la propiedad intensiva asociada con la única propiedad extensiva del modelo es el producto $\varepsilon(\underline{x}, t) \rho(\underline{x}, t)$. Por lo tanto, el modelo matemático básico del flujo de un fluido en un medio poroso es:

$$\frac{\partial \varepsilon \rho}{\partial t} + \nabla \cdot (\varepsilon \rho \underline{v}) = g \quad (2.7)$$

Aquí, el producto $\varepsilon \rho$ es la masa del fluido por unidad de volumen del espacio físico ocupado por un fluido en un sistema de medio poroso.

2.2.3. El modelo de transporte en un medio poroso

Un supuesto básico implícito en los modelos de transporte en esta sección, es que el movimiento del fluido es conocido de antemano; en particular, la velocidad del fluido es un dato. Esto exige la medición de tales velocidades o que se deriven a través de la aplicación de un modelo de flujo. Se asumirá que el fluido está localizado en los poros del medio poroso y tal medio es *saturado*; esto es, que el fluido llena completamente los poros. Así, aunque el medio este saturado el fluido no llena el espacio físico ya que parte de el está ocupado por la *matriz sólida*.

Veamos que la masa del soluto $M_S(t)$, contenido en un cuerpo en un sistema poroso, el cual esta hecho de la matriz sólida y el fluido, está dada por la integral:

$$M_S(t) = \int_{B(t)} \varepsilon(\underline{x}, t) c(\underline{x}, t) d\underline{x}. \quad (2.8)$$

Aquí, $c(\underline{x}, t)$ es la concentración del soluto, esto es la masa del soluto por unidad de volumen del fluido, y $\varepsilon(\underline{x}, t)$ es la porosidad. Por otra parte, la ecuación de balance

global es:

$$\frac{dM_S}{dt}(t) = \int_{B(t)} g_S(\underline{x}, t) dx + \int_{\partial B(t)} \underline{\tau}_S(\underline{x}, t) \cdot \underline{n}(\underline{x}, t) dx. \quad (2.9)$$

Donde $g_S(\underline{x}, t)$ representa las fuentes de masa de soluto y $\underline{\tau}_S(\underline{x}, t)$ es el flujo de masa de soluto, el cual como hemos visto es debido principalmente al proceso de difusión-dispersión. Aplicando ahora la formulación axiomática de los modelos continuos, la ecuación diferencial gobernante es:

$$\frac{\partial \varepsilon c}{\partial t} + \nabla \cdot (\varepsilon c \underline{v}) = g_S + \nabla \cdot \underline{\tau}_S. \quad (2.10)$$

Esta ecuación debe de satisfacerse para cada punto ocupado por el sistema sólido-matriz-fluido.

Transporte

En el caso del transporte de solutos en un medio poroso, los procesos de difusión también son modelados por medio de la ley de Fick, sin embargo, para este tipo de aplicación usualmente se escribe de la forma:

$$\underline{\tau}_S(\underline{x}, t) = \varepsilon \underline{\underline{D}} \nabla c. \quad (2.11)$$

Además, existe una diferencia fundamental entre los procesos de difusión que se producen en un fluido libre y los que tienen lugar en un medio poroso y es que en este último dos tipos de fenómenos coexisten:

- 1 *La difusión molecular*, que se debe al movimiento browniano producido a escala microscópica, y
- 2 *La dispersión mecánica*, que es debida al sinuoso flujo del fluido, atribuible a la aleatoriedad estructural de la matriz sólida.

Cada uno de estos fenómenos contribuye globalmente al proceso de dispersión-difusión donde dichos procesos tienen un efecto aditivo. En efecto, el *tensor de dispersión* $\underline{\underline{D}}$ se expresa como la suma del tensor de *difusión molecular* $\underline{\underline{D}}^m$ más el tensor de *dispersión mecánica* $\underline{\underline{D}}^M$, es decir

$$\underline{\underline{D}} = \underline{\underline{D}}^M + \underline{\underline{D}}^m \quad (2.12)$$

La *dispersión mecánica* no se produce cuando el fluido está en reposo, pero cuando el fluido está en movimiento el efecto de *dispersión mecánica* es generalmente más significativo que el de *difusión molecular*.

La ecuación diferencial básica que gobierna el transporte de solutos en un medio poroso se puede obtener, de una forma general, mediante la sustitución del flujo difusivo del soluto, dada por la ecuación (2.11), en la ecuación (2.10):

$$\frac{\partial \varepsilon c}{\partial t} + \nabla \cdot (\varepsilon c \underline{v}) = \nabla \cdot (\varepsilon \underline{\underline{D}} \nabla c) + g_S \quad (2.13)$$

Esta ecuación puede transformarse en:

$$\varepsilon c \left(\frac{D \text{Ln } c}{Dt} + \frac{D \text{Ln } \varepsilon}{Dt} + \nabla \cdot \underline{v} \right) = \nabla \cdot (\varepsilon \underline{\underline{D}} \nabla c) + g_S \quad (2.14)$$

Ésta a su vez, puede transformarse en:

$$\frac{Dc}{Dt} + c \left(\frac{D \text{Ln } \varepsilon}{Dt} + \nabla \cdot \underline{v} \right) = \nabla \cdot (\underline{\underline{D}} \nabla c) + (\nabla \text{Ln } \varepsilon) \cdot \underline{\underline{D}} \nabla c + \varepsilon^{-1} g_S \quad (2.15)$$

Esta es una forma muy general de la ecuación gobernante, y las aplicaciones de la misma a ciertos casos particulares se derivan a continuación.

Cuando el fluido es incompresible se tiene que:

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon \underline{v}) = 0 \quad (2.16)$$

De forma equivalente:

$$\frac{D \text{Ln } \varepsilon}{Dt} + \nabla \cdot \underline{v} = 0 \quad (2.17)$$

Por lo tanto,

$$\frac{Dc}{Dt} = \nabla \cdot (\underline{\underline{D}} \nabla c) + (\nabla \text{Ln } \varepsilon) \cdot \underline{\underline{D}} \nabla c + \varepsilon^{-1} g_S \quad (2.18)$$

Si la matriz sólida es homogénea (en realidad, si $\nabla \text{Ln } \varepsilon$ puede despreciarse),

$$\frac{Dc}{Dt} = \nabla \cdot (\underline{\underline{D}} \nabla c) + \varepsilon^{-1} g_S \quad (2.19)$$

Más explícitamente se tiene:

$$\frac{\partial c}{\partial t} + \underline{v} \cdot \nabla c = \nabla \cdot (\underline{\underline{D}} \nabla c) + \varepsilon^{-1} g_S \quad (2.20)$$

Cuando el fluido está en reposo, esta ecuación se reduce a:

$$\frac{\partial c}{\partial t} = \nabla \cdot (\theta \underline{\underline{D}} \nabla c) + \varepsilon^{-1} g_S \quad (2.21)$$

En el caso de transporte no difusivo la ecuación (2.20) se reduce a:

$$\frac{\partial c}{\partial t} + \underline{v} \cdot \nabla c = \varepsilon^{-1} g_S \quad (2.22)$$

Todas estas ecuaciones, de (2.20) a (2.22), se utilizan ampliamente en la modelación de contaminantes en aguas subterráneas. Éstas muestran explícitamente que al igual que en el transporte por un fluido libre, en el transporte de solutos por un fluido en

un medio poroso, la velocidad de advección es también la velocidad de las partículas del fluido.

En la ecuación (2.20) la *matriz de difusión-dispersión* es generalmente anisotrópica. En cuanto a la ecuación (2.22), que gobierna el *transporte no difusivo*, es una ecuación de primer orden.

Las ecuaciones que gobiernan los estados estacionarios y que se derivan de las ecuaciones (2.20) y (2.21) son:

$$\underline{v} \cdot \nabla c = \nabla \cdot \left(\underline{\underline{D}} \nabla c \right) + \varepsilon^{-1} g_S \quad (2.23)$$

$$\nabla \cdot (\theta D \nabla c) = -\varepsilon^{-1} g_S \quad (2.24)$$

y

$$\underline{v} \cdot \nabla c = \varepsilon^{-1} g_S \quad (2.25)$$

Con esto se presenta de manera general las ecuaciones diferenciales parciales que modelan el transporte de solutos en un medio poroso y en fluidos libres, en base a ésta última se planteará el modelo a analizar en esta tesis.

Capítulo 3

Métodos de Descomposición de Dominio

3.1. Introducción

Las técnicas de descomposición de dominio permiten que los procesos debidos a problemas de gran escala puedan ser divididos en partes susceptibles de ser resueltas de forma independiente. (véase [1]). La idea original de estos métodos representa una extensión, a arquitecturas paralelas, del clásico método alternativo de Schwarz (véase [2]). Consideremos un dominio rectangular Ω :

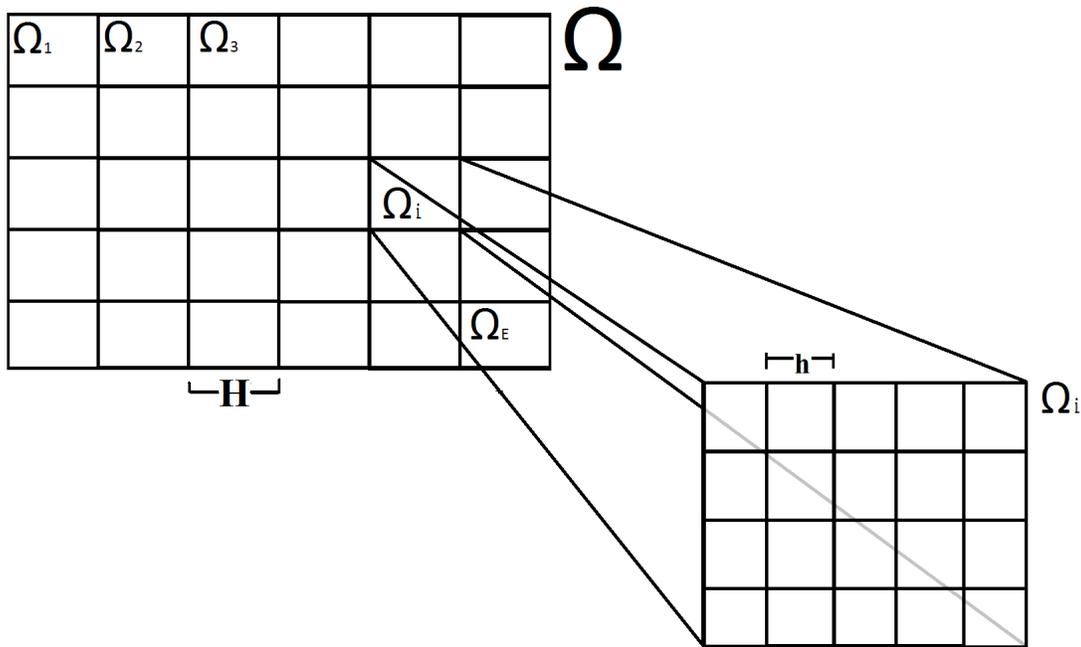


Figura 3.1: Partición en E subdominios

Cualquier método de descomposición de dominio se basa en la suposición de que dado un dominio computacional Ω , se puede particionar en subdominios $\Omega_i, i = 1, 2, \dots, E$ entre los cuales puede o no existir superposición (ver Figura: 3.1).

Además consideraremos una *partición gruesa* o *malla gruesa* a la partición del dominio Ω en correspondientes subdominios $\Omega_1, \dots, \Omega_E$ y una *partición fina* o *malla local* a la referente a cada subdominio Ω_i . Entonces, el problema es reformulado en términos de cada subdominio utilizando algún método de discretización y se obtiene una familia de subproblemas de tamaño reducido, estos subproblemas en principio son independientes entre sí y están acoplados a través de la solución en la interfaz de los subdominios que es desconocida. Los métodos de descomposición de dominio permiten tratar problemas de tamaño considerable mediante el uso de algoritmos paralelos en computadoras, ya sean secuenciales o paralelas. Últimamente la atención se ha desplazado de los métodos con superposición en el dominio a los métodos sin superposición

en el dominio, principalmente ya que estos últimos son más efectivos para una gran variedad de problemas.

3.1.1. Métodos de Discretización

El proceso por medio del cual se obtiene la solución aproximada de un problema gobernado por una ecuación diferencial en derivadas parciales, está constituido por dos etapas. La primera etapa, llamada *discretización*, consiste en transformar el dominio continuo en una malla de nodos, para luego convertir a la ecuación diferencial parcial continua y a las condiciones auxiliares, ya sean de frontera o iniciales, en un sistema de ecuaciones algebraicas. La segunda etapa del proceso de aproximación requiere un método adecuado para obtener la solución del sistema de ecuaciones algebraicas planteado. Existe una gran variedad de métodos numéricos para resolver las ecuaciones diferenciales parciales. No obstante, los más usados en la actualidad son: el Método de diferencias finitas y el Método de elementos finitos

La aproximación por medio de diferencias finitas es el método más antiguo aplicado para obtener la solución numérica de ecuaciones diferenciales. Es de una gran sencillez conceptual y constituye un procedimiento muy sencillo para la resolución de una ecuación diferencial parcial. El primer paso para la aplicación del método consiste en discretizar el dominio en el que se quiere resolver la ecuación con una malla, por conveniencia puede ser cuadrada como en (3.1). Los puntos de la malla están separados una distancia h en ambas direcciones x e y ; se reemplazan las derivadas continuas de la ecuación diferencial por las expresiones equivalentes en diferencias finitas:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad \text{por ejemplo, diferencias hacia delante} \quad (3.1)$$

La resolución del sistema de ecuaciones queda planteado como consecuencia de la anterior sustitución. El FDM es tal vez el método más simple para aplicar a mallas con una geometría uniforme particularmente. Su mayor desventaja consiste en su incapacidad para tratar efectivamente la solución de problemas sobre formas geométricas irregulares.

Al aplicar la ecuación discretizada en cada punto de la malla se obtiene un sistema de ecuaciones denominado *sistema de ecuaciones de diferencias finitas*. El proceso de aproximación requiere de la selección de un método adecuado para obtener la solución del sistema de ecuaciones algebraicas planteado. Una vez resuelto el sistema de ecuaciones de diferencias finitas se obtiene el valor de la función en los nodos de la malla, es decir, que al emplear el método de diferencias finitas se obtiene una solución aproximada discreta.

Los métodos de *elemento finito* (FEM, Finite Element Method) son una estrategia numérica alternativa muy popular para la simulación de sólidos. La teoría de elemento

finito es más compleja que los métodos de diferencias finitas (FDM), el FEM tiene ventajas significativas sobre el FDM en algunas aplicaciones prácticas.

El FEM se basa en considerar al cuerpo o estructura dividido en elementos discretos, con determinadas condiciones de vínculo entre sí, generándose un sistema de ecuaciones que se resuelve numéricamente y proporciona el estado de tensiones y deformaciones.

3.1.2. Ideas Básicas

Las ideas básicas de la descomposición de Dominio son naturales y simples. Consideremos la ecuación de Poisson en una región Ω , en dos o tres dimensiones, con frontera tipo Dirichlet cero en $\partial\Omega$, la cual es la frontera de Ω . Supongamos también que Ω está particionada en dos subdominios Ω_i sin traslape:

$$\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}, \quad \Omega_1 \cap \Omega_2 = \emptyset, \quad \Gamma = \partial\Omega_1 \cap \Omega_2; \quad (3.2)$$

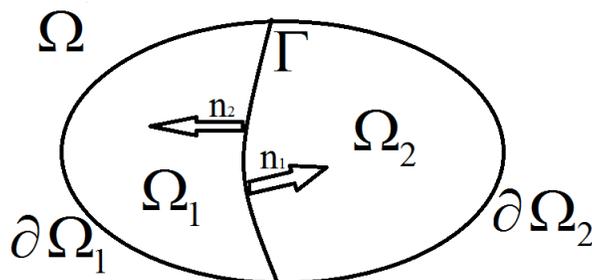


Figura 3.2: Partición en dos subdominios sin traslape

Observando la figura (3.2) podemos asumir que:

$$\text{medida}(\partial\Omega_1 \cap \Omega) \geq 0, \quad \text{medida}(\partial\Omega_2 \cap \Omega) \geq 0, \quad (3.3)$$

y así las fronteras de los subdominios son Lipschitz-continuas, lo que quiere decir que existe una constante $K > 0$, para funciones definidas sobre espacios euclídeos $f : N \rightarrow M$, tal que:

$$\|f(x) - f(y)\| \leq K \|x - y\|, \quad \forall x, y \in \mathbb{R}^n, \quad f : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (3.4)$$

Considerando el siguiente problema:

$$\begin{aligned} -\Delta u &= f \text{ en } \Omega \\ u &= 0 \text{ sobre } \partial\Omega. \end{aligned} \quad (3.5)$$

Entonces, bajo condiciones regularmente adecuadas en f y en las fronteras de los subdominios, regularmente f es cuadrado sumable y las fronteras tienen continuidad de Lipschitz, el problema (3.5) es equivalente al siguiente problema acoplado:

$$\begin{aligned} -\Delta u_1 &= f && \text{en } \Omega_1, \\ u_1 &= 0 && \text{sobre } \partial\Omega_1 \setminus \Gamma, \\ u_1 &= u_2 && \text{sobre } \Gamma, \\ \frac{\partial u_1}{\partial n_1} &= -\frac{\partial u_2}{\partial n_2} && \text{sobre } \Gamma, \\ -\Delta u_2 &= f && \text{en } \Omega_2, \\ u_2 &= 0 && \text{sobre } \partial\Omega_2 \setminus \Gamma. \end{aligned} \quad (3.6)$$

Aquí, u_i es la restricción de u para Ω_i y \mathbf{n}_i la normal exterior a Ω_i . Esta equivalencia puede ser probada considerando los correspondientes problemas variacionales (véase [2]).

Las condiciones sobre la interfaz Γ son llamadas condiciones de transmisión y también son equivalentes a la igualdad de las dos combinaciones lineales independientes de las trazas de las funciones y sus derivadas normales. En lo siguiente, nos referiremos a la derivada normal como el *flujo*.

3.2. Representaciones Matriciales y Vectoriales

Empezando por cualquier algoritmo de descomposición de dominio escrito en términos de funciones u operadores, debemos ser capaces de reescribirlo en forma matricial como un método iterativo preconditionado para un cierto sistema lineal.

Consideraremos una descomposición del dominio Ω y una aproximación por medio de alguna discretización del problema (3.5). Siempre asumiremos que los subdominios consisten en uniones de elementos, o equivalentemente, que las fronteras de los subdominios no corten a través de cualquier elemento. Por lo que la aproximación da lugar a un sistema lineal:

$$Au = f \quad (3.7)$$

la cual para una malla de tamaño h , tiene normalmente un número de condición (véase A.4) del orden de $1/h^2$. La matriz A aquí expuesta suele subagruparse de acuerdo a las características de los nodos en la malla, al igual que los vectores f y u . Los nodos

serán clasificados como *nodos interiores* y *nodos de frontera interna*: un nodo es interior si pertenece sólo a un subdominio cerrado y es de frontera interna si pertenece a más de uno; a su vez los nodos de frontera interna pueden ser clasificados como *nodos primales* y *duales*. Se define:

A_{II} el subgrupo de la matriz A formado solamente por nodos Interiores.

$A_{I\Gamma}$ el subgrupo de la matriz A formado por nodos Interiores y nodos de Frontera interna.

$A_{\Gamma I}$ el subgrupo de la matriz A formado por nodos de Frontera interna y nodos Interiores.

$A_{\Gamma\Gamma}$ el subgrupo de la matriz A formado solamente por nodos de Frontera interna.

Subagrupando el sistema (3.7) obtenemos para el ejemplo con dos subdominios (Ω_1, Ω_2) :

$$A = \begin{pmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{pmatrix}, \quad u = \begin{pmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_\Gamma \end{pmatrix}, \quad f = \begin{pmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_\Gamma \end{pmatrix}, \quad (3.8)$$

En donde hemos dividido los grados de libertad en aquellos que, son internos a Ω_1 y a Ω_2 y en los interiores de Γ . La matriz A y el vector f pueden obtenerse subagrupando las componentes correspondientes a los dos subdominios. En general,

$$f^{(i)} = \begin{pmatrix} f_I^{(i)} \\ f_\Gamma^{(i)} \end{pmatrix}, \quad A^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma} \end{pmatrix}, \quad i = 1, \dots, E. \quad (3.9)$$

son la parte derecha y la matriz del sistema para los problemas de Poisson (ecuación (3.5)), respectivamente; con condiciones a la frontera tipo Dirichlet sobre $\partial\Omega_i \setminus \Gamma$ y condiciones tipo Neumann sobre Γ , tenemos:

$$A_{\Gamma\Gamma} = A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)}, \quad f_\Gamma = f_\Gamma^{(1)} + f_\Gamma^{(2)}. \quad (3.10)$$

En vista de las condiciones de transmisión vistas en (3.6), buscaremos una aproximación de las derivadas normales sobre Γ . Teniendo la solución exacta local u_i , su derivada normal puede definirse como un funcional lineal usando la fórmula de Green. Así, si ϕ_j es una función de base nodal para un nodo sobre Γ , usando (3.6) tenemos que:

$$\int_\Gamma \frac{\partial u_i}{\partial n_i} \phi_j ds = \int_{\Omega_i} (\Delta u_i \phi_j + \nabla u_i \cdot \nabla \phi_j) dx = \int_{\Omega_i} (-f \phi_j + \nabla u_i \cdot \phi_j) dx. \quad (3.11)$$

Una aproximación, $\lambda^{(i)}$, del funcional que representa a la derivada normal, puede ser encontrado reemplazando la solución exacta u_i en el lado derecho con su aproximación por elemento finito. Dejando a j correr a través de los nodos sobre Γ y usando la definición de la matriz local A , introducimos la expresión:

$$\lambda^{(i)} = A_{\Gamma I}^{(i)} u_I^{(i)} + A_{\Gamma \Gamma}^{(i)} u_{\Gamma}^{(i)} - f_{\Gamma}^{(i)}. \quad (3.12)$$

Esta aproximación coincide con el residual correspondiente a los nodos sobre Γ del problema de Poisson (3.5) con condiciones Neumann sobre Γ ; véase (3.9). También notamos que $\lambda^{(i)}$ no es obtenida de la derivada normal de la solución por elemento finito, pero es una aproximación del funcional lineal de la solución exacta. Usando estas definiciones, podemos encontrar una aproximación al problema (3.6):

$$\begin{aligned} A_{II}^{(1)} u_I^{(1)} + A_{\Gamma I}^{(1)} u_{\Gamma}^{(1)} &= f_I^{(1)}, \\ u_{\Gamma}^{(1)} &= u_{\Gamma}^{(2)} = u_{\Gamma}, \\ (A_{\Gamma I}^{(1)} + A_{\Gamma \Gamma}^{(1)} u_{\Gamma}^{(1)} - f_{\Gamma}^{(1)}) &= - (A_{\Gamma I}^{(2)} u_I^{(2)} + A_{\Gamma \Gamma}^{(2)} u_{\Gamma}^{(2)} - f_{\Gamma}^{(2)}) = \lambda_{\Gamma}, \\ A_{II}^{(2)} u_I^{(2)} + A_{\Gamma I}^{(2)} u_{\Gamma}^{(2)} &= f_I^{(2)}. \end{aligned} \quad (3.13)$$

donde notamos que la primera y última ecuación de (3.13) son discretizaciones del problema de Poisson para las funciones interiores $u_I^{(i)}$ con valores Dirichlet que se cancelan sobre $\partial\Omega_i \setminus \Gamma$ y es igual a un valor común u_{Γ} sobre Γ . Alternativamente, la primer y tercera ecuaciones proveen una discretización del problema de Poisson en Ω_1 para la función local u_1 con valores Neumann iguales a λ_{Γ} y valores Dirichlet que se cancelan sobre $\partial\Omega_1 \setminus \Gamma$. Un problema Neumann análogo en Ω_2 es provisto por las ecuaciones del tercer y cuarto lugar en (3.13). La solución de estos problemas locales con los valores Dirichlet y Neumann adecuados proveen los bloques para la construcción de los métodos sin traslape.

3.3. Métodos sin Traslape

Primero consideraremos algunos métodos *iterativos subestructurados* simples que se basan en la partición de subdominios sin traslape. Estos métodos, son de hecho métodos iterativos preconditionados por el valor de la frontera u_{Γ} o por el de la derivada normal λ_{Γ} . Empezaremos con estos métodos desde que son derivados directamente por medio de los problemas acoplados (3.6) y (3.13). Los métodos de descomposición de dominio basados en el traslape de las particiones fueron los primeros en ser divididos (nombrados, alternativamente método de Schwarz en subdominios con traslape, véase [2]).

3.3.1. Una ecuación para u_{Γ} : El Sistema del complemento de Schur

Consideremos el sistema lineal (3.7) con A , u y f definidas en (3.8). En un primer paso de muchos métodos iterativos de descomposición de dominio, las incógnitas del

interior de los subdominios ($u_I^{(i)}$) son eliminadas. Esto corresponde a una factorización de bloque de la matriz de (3.8):

$$A = LR = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ A_{\Gamma I}^{(1)} A_{II}^{(1)-1} & A_{\Gamma I}^{(2)} A_{II}^{(2)-1} & I \end{pmatrix} \begin{pmatrix} A_{II}^{(1)} & 0 & A_{\Gamma I}^{(1)} \\ 0 & A_{II}^{(2)} & A_{\Gamma I}^{(2)} \\ 0 & 0 & S \end{pmatrix} \quad (3.14)$$

y el sistema lineal resultante:

$$\begin{pmatrix} A_{II}^{(1)} & 0 & A_{\Gamma I}^{(1)} \\ 0 & A_{II}^{(2)} & A_{\Gamma I}^{(2)} \\ 0 & 0 & S \end{pmatrix} u = \begin{pmatrix} f_I^{(1)} \\ f_I^{(2)} \\ g_\Gamma \end{pmatrix}. \quad (3.15)$$

Aquí I es la matriz identidad y $S = A_{\Gamma\Gamma} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} A_{\Gamma I}^{(1)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} A_{\Gamma I}^{(2)}$ es el complemento de Schur relativo a las incógnitas sobre Γ . Por un cálculo directo, vemos que S y g_Γ pueden ser encontrados subarreglando contribuciones locales. En particular, retomando la forma de las matrices locales en (3.9) y definiendo el complemento local de Schur por medio de:

$$S^{(i)} := A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} A_{II}^{(i)-1} A_{\Gamma I}^{(i)} \quad i = 1, \dots, E. \quad (3.16)$$

Encontramos que el sistema del complemento de Schur para u_Γ será:

$$S u_\Gamma = g_\Gamma, \quad (3.17)$$

con:

$$S = S^{(1)} + S^{(2)}, \\ g_\Gamma = \left(f_\Gamma^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} f_I^{(1)} \right) + \left(f_\Gamma^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} f_I^{(2)} \right) =: g_\Gamma^{(1)} + g_\Gamma^{(2)}.$$

Notamos que una vez que u_Γ es encontrado, resolviendo (3.17), las componentes interiores pueden encontrarse usando (3.15):

$$u_I^{(i)} = A_{II}^{(i)-1} \left(f_I^{(i)} - A_{\Gamma I}^{(i)} u_\Gamma \right); \quad (3.18)$$

estas son precisamente soluciones a dos problemas Dirichlet no homogéneos. Los sistemas del complemento de Schur, los cuales proveen una ecuación para la aproximación de la traza de la solución exacta sobre Γ , son puramente derivados del álgebra por eliminación Gaussiana en bloque. Es interesante notar que también se pueden obtener usando las condiciones de transmisión del sistema acoplado. También podemos encontrar una ecuación para la traza de la solución exacta sobre Γ trabajando directamente con el problema continuo (3.6). El operador correspondiente es llamado el operador *Steklov-Poincaré*. El sistema del complemento de Schur (3.17) es una aproximación de la ecuación de Steklov-Poincaré, *determinada* directamente por la aproximación en elemento finito, particularmente, por la aproximación de las derivadas normales (3.12).

3.3.2. El Algoritmo Dirichlet-Neumann

Los métodos de este tipo fueron considerados en principio en (véase [26], [27], [28], [29], [30] y [31]). El algoritmo básico Dirichlet-Neumann consiste en dos pasos fraccionales correspondientes a las dos subregiones Ω_i , $i = 1, 2$. Dando un valor inicial u_Γ^0 , se resuelve primero un problema Dirichlet en Ω_1 con el valor de frontera Dirichlet u_Γ^0 sobre Γ y después un problema mixto Neumann-Dirichlet en Ω_2 con condiciones Neumann sobre Γ , determinado por la solución en Ω_1 se obtiene en el paso previo y con condiciones Dirichlet en el resto de $\partial\Omega_2$. El nuevo valor a iterar u_Γ^1 es elegido como la traza de la solución en Ω_2 o, más general, como una combinación lineal de su traza y u_Γ^0 , usando un parámetro de relajación *theta* adecuado. En términos de los operadores diferenciales (véase (3.6)), podemos escribir, para $n \geq 0$:

$$(D) \begin{cases} -\Delta u_1^{n+1/2} = f & \text{en } \Omega_1, \\ u_1^{n+1/2} = 0 & \text{sobre } \partial\Omega_1 \setminus \Gamma, \\ u_1^{n+1/2} = u_\Gamma^n & \text{sobre } \Gamma, \end{cases} \quad (N) \begin{cases} -\Delta u_2^{n+1} = f & \text{en } \Omega_2, \\ u_2^{n+1} = 0 & \text{sobre } \partial\Omega_2 \setminus \Gamma, \\ \frac{\partial u_2^{n+1}}{\partial n_2} = -\frac{\partial u_1^{n+1/2}}{\partial n_1} & \text{sobre } \Gamma, \end{cases} \quad (3.19)$$

$$u_\Gamma^{n+1} = \theta u_2^{n+1} + (1 - \theta) u_\Gamma^n \text{ sobre } \Gamma,$$

con $\theta \in (0, \theta_{max})$. Usando nuestra aproximación para las derivadas normales, es decir (3.12), podemos derivar la iteración correspondiente para el problema discreto. Si nosotros definimos los vectores de grados de libertad internos como $v_1 = u_\Gamma^{(1)}$ y $w_2 = u_\Gamma^{(2)}$, encontraremos:

$$(D) \quad A_{II}^{(1)} v_1^{n+1/2} + A_{I\Gamma}^{(1)} u_\Gamma^n = f_I^{(1)},$$

$$(N) \quad \begin{pmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{pmatrix} \begin{pmatrix} w_2^{n+1} \\ \widetilde{u}_\Gamma^{n+1} \end{pmatrix} = \begin{pmatrix} f_I^{(2)} \\ f_\Gamma^{(2)} - \lambda_\Gamma^{n+1/2} \end{pmatrix}, \quad (3.20)$$

$$u_\Gamma^{n+1} = \theta \widetilde{u}_\Gamma^{n+1} + (1 - \theta) u_\Gamma^n,$$

con

$$\lambda_\Gamma^{n+1/2} = A_{\Gamma I}^{(1)} v_1^{n+1/2} + A_{\Gamma\Gamma}^{(1)} u_\Gamma^n - f_\Gamma^{(1)}.$$

Es claro que (3.20) surge de un corte al sistema original (3.13) y esto nos provee de un método iterativo consistente para su solución, es decir, el límite de cualquier secuencia va a satisfacer el conjunto correcto de ecuaciones. Eliminando $v_1^{n+1/2}$ de (3.20), encontrando $\lambda_\Gamma^{n+1/2}$ y usando una factorización de bloque para la matriz local $A^{(2)}$ y eliminando w_2^{n+1} de (3.20) tenemos la siguiente ecuación:

$$S^{(2)}(u_\Gamma^{n+1} - u_\Gamma^n) = \theta(g_\Gamma - S u_\Gamma^n),$$

la cual nos muestra que el algoritmo Dirichlet-Neumann es una iteración preconditionada de Richardson (véase [2]) para el sistema del complemento de Schur (3.17), con el preconditionador $S^{(2)^{-1}}$. El operador preconditionador es:

$$S^{(2)^{-1}}S = I + S^{(2)^{-1}}S^{(1)},$$

La aplicación de esto a un vector involucra la solución para un problema Dirichlet (una multiplicación por $S^{(1)}$) y una multiplicación por $S^{(2)^{-1}}$ que corresponde a resolver un problema con condiciones Neumann sobre Γ y condiciones Dirichlet para el resto de $\partial\Omega_2$.

En el caso especial donde Ω_1 y Ω_2 tienen la misma forma y tamaño, son simétricas con respecto a la interfaz Γ y tienen las mismas triangulaciones, entonces $\kappa(S^{(2)^{-1}}S) = 1$. Generalmente, el método Dirichlet-Neumann para dos subdominios es óptimo si se pueden encontrar extensiones estables: el número de condición satisface (véase B.1):

$$\kappa(S^{(2)^{-1}}S) \leq C,$$

con C una constante independiente de la dimensión del problema discretizado.

3.3.3. El Algoritmo Neumann-Neumann

Los métodos de este tipo son considerados en (véase [32],[33] y [34]).

El algoritmo básico Neumann-Neumann puede ser descrito de la siguiente forma: Empezamos con un vector inicial u_Γ^0 . Un paso del algoritmo Neumann-Neumann consiste primero en resolver problemas Dirichlet para cada Ω_i con el valor u_Γ^0 sobre Γ , y después un problema para cada subdominio, con valores Neumann, sobre Γ , elegidos como la diferencia de las derivadas normales de las soluciones de los dos problemas Dirichlet. Los valores sobre Γ de las soluciones de estos problemas Neumann son empleados para corregir el valor inicial u_Γ^0 y encontrar el nuevo término en la iteración u_Γ^1 . En términos de operadores diferenciales (ver (3.6)), podemos escribir para $n \geq 0$:

$$(D_i) \left\{ \begin{array}{ll} -\Delta u_i^{n+1/2} = f & \text{en } \Omega_i, \\ u_i^{n+1/2} = 0 & \text{sobre } \partial\Omega_i \setminus \Gamma, \\ u_i^{n+1/2} = u_\Gamma^n & \text{sobre } \Gamma, \end{array} \right\}, i = 1, 2,$$

$$(N_i) \left\{ \begin{array}{ll} -\Delta \psi_i^{n+1} = 0 & \text{en } \Omega_i, \\ \psi_i^{n+1} = 0 & \text{sobre } \partial\Omega_i \setminus \Gamma, \\ \frac{\partial \psi_i^{n+1}}{\partial n_i} = \frac{\partial u_1^{n+1/2}}{\partial n_1} + \frac{\partial u_2^{n+1/2}}{\partial n_2} & \text{sobre } \Gamma, \end{array} \right\}, i = 1, 2,$$
(3.21)

$$u_\Gamma^{n+1} = u_\Gamma^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) \text{ sobre } \Gamma,$$

con un valor adecuado $\theta \in (0, \theta_{max})$. Usando nuestra aproximación para las derivadas normales, podemos derivar una iteración para el problema discreto. Si definimos los vectores de los grados de libertad interiores como $v_i = u_\Gamma^{(i)}$ y $w_i = \psi_\Gamma^{(i)}$, encontramos :

$$\begin{aligned}
 (D_i) \quad & A_{II}^{(i)} v_i^{n+1/2} + A_{\Gamma\Gamma}^{(i)} u_\Gamma^n = f_I^{(i)}, \quad i = 1, 2, \\
 (N) \quad & \begin{pmatrix} A_{II}^{(i)} & A_{\Gamma\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} w_i^{n+1} \\ \eta_i^{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ r_\Gamma \end{pmatrix}, \quad i = 1, 2, \\
 & u_\Gamma^{n+1} = u_\Gamma^n - \theta(\eta_1^{n+1} + \eta_2^{n+1}),
 \end{aligned} \tag{3.22}$$

donde r_Γ esta definido como:

$$\begin{aligned}
 r_\Gamma = & (A_{\Gamma I}^{(1)} v_1^{n+1/2} + A_{\Gamma\Gamma}^{(1)} u_\Gamma^n - f_\Gamma^{(1)}) + \\
 & (A_{\Gamma I}^{(2)} v_2^{n+1/2} + A_{\Gamma\Gamma}^{(2)} u_\Gamma^n - f_\Gamma^{(2)});
 \end{aligned}$$

A continuación eliminamos $v_i^{n+1/2}$ y w_i^{n+1} de (3.22). De los problemas (D_i) obtenemos:

$$r_\Gamma = -(g_\Gamma - S u_\Gamma^n), \tag{3.23}$$

que muestra que la diferencia r_Γ de los flujos locales es igual a menos el residual del sistema del complemento de Schur. Usando una factorización en bloque de las matrices locales $A^{(i)}$, entonces de los problemas (N_i) tenemos:

$$\eta_i^{n+1} = S^{(i-1)} r_\Gamma = -S^{(i-1)} (g_\Gamma - S u_\Gamma^n),$$

Por tanto, encontramos

$$u_\Gamma^{n+1} - u_\Gamma^n = \theta (S^{(1)-1} + S^{(2)-1}) (g_\Gamma - S u_\Gamma^n),$$

lo cual muestra que el algoritmo Neumann-Neumann es también una iteración preconditionada de Richardson del sistema del complemento de Schur (véase [2]), con el preconditionador $S^{(1)-1} + S^{(2)-1}$. La matriz preconditionadora es:

$$FS = (S^{(1)-1} + S^{(2)-1}) S = (S^{(1)-1} + S^{(2)-1}) (S^{(1)} + S^{(2)}), \tag{3.24}$$

La aplicación de esto a un vector involucra la solución de los dos problemas Dirichlet y dos problemas con valores Neumann sobre Γ . La optimización de este método se sigue fácilmente del resultado en el algoritmo Dirichlet-Neumann por medio del uso del teorema del mapeo espectral. La matriz preconditionadora en (3.24) puede ser escrita como $S^{(2)-1} S^{(1)} + 2I + (S^{(2)-1} S^{(1)})^{-1}$, donde los eigenvalores de $S^{(2)-1} S^{(1)}$ son uniformemente acotados superior e inferiormente.

Notemos que, dados los pesos positivos δ_1^\dagger y δ_2^\dagger con:

$$\delta_1^\dagger + \delta_2^\dagger = 1,$$

el residual r_Γ del lado derecho de la igualdad de los problemas Neumann en (3.22) puede ser reemplazado por $\delta_1^\dagger r_\Gamma$ y $\delta_2^\dagger r_\Gamma$ para Ω_1 y Ω_2 , respectivamente. Similarmente, cuando se encuentra el nuevo iterante u_Γ^{n+1} la suma de las dos correcciones η_1^{n+1} y η_2^{n+1} puede ser reemplazada por el promedio con sus respectivos pesos:

$$u_\Gamma^{n+1} = u_\Gamma^n - \theta(\delta_1^\dagger \eta_1^{n+1} + \delta_2^\dagger \eta_2^{n+1}).$$

Sea $D^{(i)}$ la matriz diagonal con los elementos $\delta_i^\dagger(x)$ correspondientes al punto $x \in \partial\Omega_{i,h} \cap \Gamma_h$.

$$D := \begin{pmatrix} D^{(1)} & 0 & \cdots & 0 \\ 0 & D^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D^{(E)} \end{pmatrix}. \quad (3.25)$$

Esto nos da paso al preconditionador:

$$D^{(1)}S^{(1)-1}D^{(1)} + D^{(2)}S^{(2)-1}D^{(2)},$$

donde $D^{(i)}$ es una matriz diagonal, cuyas entradas en las diagonales son iguales a δ_i^\dagger . En la práctica comúnmente se emplea el escalamiento de matrices $D^{(i)}$ para mejorar la convergencia de los algoritmos Neumann-Neumann para subdominios con cruce de puntos y en particular para problemas con grandes cambios en los coeficientes.

Los algoritmos Dirichlet-Neumann y Neumann-Neumann es posible generalizarlos, usando condiciones más generales, como las del tipo Robin. Nótese que el uso de condiciones de interfaz más generales son requeridas usualmente para problemas indefnidos o no simétricos.

3.3.4. El Algoritmo Dirichlet-Dirichlet o el Método FETI

En esta parte consideraremos el método dual al algoritmo Neumann-Neumann. Empezamos con un vector inicial λ_Γ^0 del flujo sobre Γ . Primero se resuelven dos problemas Neumann sobre Ω_i con los valores λ_Γ^0 sobre Γ , y después el problema en cada subdominio, con valores Dirichlet, sobre Γ elegidos como la diferencia de la traza de las soluciones de los dos problemas Neumann sobre Γ . Los valores de las derivadas normales sobre Γ de las soluciones de estos problemas Dirichlet son empleadas para corregir el valor inicial λ_Γ^0 y encontrar la nueva iteración λ_Γ^1 . Recordemos que λ_Γ es una aproximación de la derivada normal en la dirección \mathbf{n}_1 . Para simplificar la notación se subcribe Γ para las derivadas normales y establecemos:

$$\lambda^n = \lambda_\Gamma^n, \quad \lambda_1^n = -\lambda_2^n = \lambda^n.$$

En términos de operadores diferenciales (ver(3.6)), podemos escribir, para $n \geq 0$:

$$(N_i) \left\{ \begin{array}{ll} -\Delta u_i^{n+1/2} = f & \text{en } \Omega_i, \\ u_i^{n+1/2} = 0 & \text{sobre } \partial\Omega_i \setminus \Gamma, \\ \frac{\partial u_i^{n+1/2}}{\partial n_i} = \lambda_i^n & \text{sobre } \Gamma, \end{array} \right\}, i = 1, 2,$$

$$(D_i) \left\{ \begin{array}{ll} -\Delta \psi_i^{n+1} = 0 & \text{en } \Omega_i, \\ \psi_i^{n+1} = 0 & \text{sobre } \partial\Omega_i \setminus \Gamma, \\ \psi_i^{n+1} = u_1^{n+1/2} - u_2^{n+1/2} & \text{sobre } \Gamma, \end{array} \right\}, i = 1, 2,$$

$$\lambda^{n+1} = \lambda^n - \theta \left(\frac{\partial \psi_1^{n+1}}{\partial n_1} + \frac{\partial \psi_2^{n+1}}{\partial n_2} \right) \text{ sobre } \Gamma \quad (3.26)$$

con un adecuado valor de $\theta \in (0, \theta_{max})$. Usando nuestra aproximación de las derivadas normales, podemos obtener una iteración del problema discreto. Si definimos los grados de libertad de los vectores interiores por $v_i = u_i^{(i)}$ y $w_i = \psi_i^{(i)}$, encontramos:

$$(N_i) \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} v_i^{n+1/2} \\ \gamma_i^{n+1/2} \end{pmatrix} = \begin{pmatrix} f_I^{(i)} \\ f_\Gamma^{(i)} + \lambda_i^n \end{pmatrix}, i = 1, 2, \quad (3.27)$$

$$(D_i) A_{II}^{(i)} w_i^{n+1} + A_{I\Gamma}^{(i)} r_\Gamma = 0, i = 1, 2,$$

$$\lambda^{n+1} = \lambda^n - \theta(\eta_1^{n+1} + \eta_2^{n+1}),$$

donde el residuo r_Γ está definido por

$$r_\Gamma = \gamma_1^{n+1/2} - \gamma_2^{n+1/2}$$

y los flujos η_i^{n+1} por:

$$\eta_i^{n+1} = A_{\Gamma I}^{(i)} w_i^{n+1/2} + A_{\Gamma\Gamma}^{(i)} r_\Gamma$$

Procedemos eliminando $v_i^{n+1/2}$, $\gamma_i^{n+1/2}$ y w_i^{n+1} de (3.28). Usando la factorización de bloque de las matrices locales $A^{(i)}$, de los problemas (N_i) tenemos:

$$r_\Gamma = -(d_\Gamma - F \lambda^n),$$

lo que nos dice que la diferencia r_Γ de la solución local es igual al negativo del residual del sistema $F \lambda_\Gamma = d_\Gamma$. Los problemas (D_i) nos proveen de las siguientes ecuaciones:

$$\eta_i^{n+1} = S^{(i)} r_\Gamma = -S^{(i)} (d_\Gamma - F \lambda^n)$$

Por lo tanto, se obtiene:

$$\lambda^{n+1} - \lambda^n = \theta (S^{(1)} + S^{(2)}) (d_\Gamma - F \lambda^n),$$

lo que muestra que este algoritmo Dirichlet-Dirichlet es también una iteración preconditionada de Richardson para el sistema $F\lambda_\Gamma = d_\Gamma$ (véase [2]), con el preconditionador $S^{(1)} + S^{(2)}$. La matriz preconditionadora es:

$$SF = S \left(S^{(1)^{-1}} + S^{(2)^{-1}} \right) = \left(S^{(1)} + S^{(2)} \right) \left(S^{(1)^{-1}} + S^{(2)^{-1}} \right) \quad (3.28)$$

la aplicación de esto a un vector involucra la solución de dos problemas Neumann y dos problemas con valores Dirichlet en Γ . Cabe mencionar que en la literatura de descomposición de dominio, el término Dirichlet-Dirichlet es también usado para el método de Richardson o gradiente conjugado aplicado al sistema no preconditionado del complemento de Schur (3.17), entonces la aplicación de S involucra la solución de un problema Dirichlet en cada subdominio.

3.3.5. El Método FETI-DP

En todos los algoritmos FETI-DP se hacen cumplir un número relativamente pequeño de restricciones de continuidad a través de la interfaz en cada paso de iteración. En los primeros algoritmos de este tipo sólo los valores en los vértices de subdominios se clasifican como primales, es decir, adquieren valores únicos. Además los algoritmos presentados en (véase [35] y [36]) utilizan restricciones en los promedios sobre las aristas y las caras. Las restricciones en las aristas y las caras fueron originalmente introducidas para mejorar la tasa de convergencia de la iteración y pueden ser opcionales. Podemos ver que es posible reemplazar la totalidad o la mayor parte de las restricciones primales en los vértices por restricciones escritas en términos de los promedios sobre las aristas (y las caras). Tales restricciones primales en aristas suelen ofrecer mejores tasas de convergencia de algoritmos FETI-DP que cuando sólo se usan las restricciones primales de vértices. Para aplicar los métodos duales-primales, los nodos de frontera interna serán subclasificados como nodos primales y duales. La matriz A será subagrupada por bloques, tal como se mostró en la sección (3.2), con la diferencia que se introducirá esta nueva clasificación. Se define:

- A_{II} el subgrupo de la matriz A formado solamente por nodos Interiores.
- A_{III} el subgrupo de la matriz A formado por nodos Interiores y nodos primales.
- A_{III}^T el subgrupo de la matriz A formado por nodos de Frontera interna y nodos primales, transpuesto.
- $A_{I\Delta}$ el subgrupo de la matriz A formado por nodos Interiores y nodos duales.
- $A_{I\Delta}^T$ el subgrupo de la matriz A formado por nodos de Frontera interna y nodos duales, transpuesto.
- A_{III} el subgrupo de la matriz A formado solamente por nodos primales.

- $A_{\Delta\Delta}$ el subgrupo de la matriz A formado solamente por nodos duales.
- $A_{\Pi\Delta}$ el subgrupo de la matriz A formado por nodos primales y nodos duales.
- $A_{\Pi\Delta}^T$ el subgrupo de la matriz A formado por nodos primales y nodos duales, transpuesto.

Denotaremos un espacio de funciones lineales continuas a trozos en Ω_i como $W^h(\Omega_i)$. Asumiremos que las funciones de $W^h(\Omega_i)$ se anulan sobre $\partial\Omega_i$. Además definiremos un espacio $W_i := W^h(\partial\Omega_i \cap \Gamma)$ con $i = 1, \dots, E$, y a $W := \prod_{i=1}^E W_i$ como el espacio producto asociado. Dada $u \in W$, denotaremos a su i -ésima componente como u_i .

Los valores del lado derecho de (3.7) cambiarán cuando las variables interiores sean modificadas. En tal caso, denotaremos a los vectores resultantes como f_i , los cuales representan la modificación en Ω_i ; a su vez se modificarán los valores en los vectores locales de interfase nodal u_i . Así podremos reformular el problema, reducido a la interfase Γ , como un problema de minimización con restricciones, dado por el requisito de continuidad a lo largo de Γ : Encontrar $u \in W$ tal que:

$$J(u) := \left. \begin{array}{l} \frac{1}{2} \langle Su, u \rangle - \langle f, u \rangle \rightarrow \min \\ Bu = 0 \end{array} \right\}, \text{ donde} \quad (3.29)$$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_E \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_E \end{pmatrix}, \quad \text{y} \quad S = \begin{pmatrix} S^{(1)} & 0 & \cdots & 0 \\ 0 & S^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & S^{(E)} \end{pmatrix}.$$

La matriz

$$B = (B^{(1)}, B^{(2)}, \dots, B^{(E)}) \quad (3.30)$$

Está construida de $\{0, 1, -1\}$ de tal manera que los valores de la solución u asociada a más de un subdominio coincidan cuando $Bu = 0$. Podemos notar que la elección de B está lejos de ser única. Mientras que hay pocas opciones de cómo escribir las restricciones para un punto nodal que pertenece a una cara, hay muchas opciones para un punto en una esquina o en un vértice.

Trabajaremos siempre en subespacios $\tilde{W} \subset W$ para que una cantidad suficiente de restricciones se apliquen de manera tal que la matriz diagonal por bloques (véase (A.3)) resultante del problema de punto de silla, aunque ya no diagonal por bloques, sea estrictamente positiva definitiva. Análogamente denotaremos por $\tilde{W}^h(\Omega)$ el subespacio de $\prod_i W^h(\Omega_i)$ que es igual a \tilde{W} cuando se restringe a la interfaz Γ . También se introducen dos subespacios $\tilde{W}_\Pi \subset \tilde{W}$ y \tilde{W}_Δ correspondientes a la parte primal y dual del espacio \tilde{W} . Éstos juegan un rol importante en la descripción y el análisis de nuestros métodos iterativos y su suma directa es igual a \tilde{W} . Notemos que el subespacio dual \tilde{W}_Δ está directamente asociado con saltos a través de la interfaz y con los multiplicadores de

Lagrange que se introducen para eliminar estos saltos (véase [10]). Cada uno de los algoritmos FETI-DP está expresado en términos del complemento de Schur \tilde{S} relacionado con el espacio dual \tilde{W}_Δ . Así, \tilde{W} consiste de funciones en W que toman el mismo valor en los vértices de los subdominios y pueden ser escritas como:

$$\tilde{W} = \tilde{W}_\Pi \oplus \tilde{W}_\Delta. \quad (3.31)$$

donde $\tilde{W}_\Pi \subset \tilde{W}$ es el espacio de funciones continuas de interfaz que se anulan en todos los puntos nodales sobre Γ_h excepto en los vértices de los subdominios y \tilde{W}_Δ es la suma directa de los subespacios locales $\tilde{W}_{\Delta,i}$:

$$\tilde{W}_\Delta := \prod_{i=1}^N \tilde{W}_{\Delta,i}. \quad (3.32)$$

Aquí $\tilde{W}_{\Delta,i} \subset W_i$ consiste en funciones locales sobre $\partial\Omega_i$ que se anulan en los vértices de Ω_i . Los grados de libertad (continuos) asociados con la subestructura de vértices y con el subespacio \tilde{W}_Π son llamados *primales* (Π), mientras los asociados a los subespacios $\tilde{W}_{\Delta,i}$ (potencialmente discontinuos a través de Γ) y al interior de la subestructura de bordes son llamados *duales* (Δ).

Sea \tilde{A} la matriz de rigidez, que se obtiene mediante la restricción

$$A = \text{diag}\{A^{(1)}, A^{(2)}, \dots, A^{(N)}\}$$

de $\prod_i W^h(\Omega_i)$ a $\tilde{W}^h(\Omega)$. Notemos que \tilde{A} no es diagonal por bloques desde que existe el acoplamiento entre las subestructuras que tienen vértices en común. Podemos particionar \tilde{A} como:

$$\tilde{A} = \begin{pmatrix} A_{II} & A_{I\Pi} & A_{I\Delta} \\ A_{I\Pi}^T & A_{\Pi\Pi} & A_{\Pi\Delta} \\ A_{I\Delta}^T & A_{\Pi\Delta}^T & A_{\Delta\Delta} \end{pmatrix} \quad (3.33)$$

dónde el subíndice I se refiere a los grados de libertad interiores de los subdominios, Π a los asociados a los vértices de los subdominios y Δ a los que se encuentran en las fronteras interiores de los subdominios. A_{II} y $A_{\Delta\Delta}$ son diagonales por bloques y cada bloque corresponde a un sólo subdominio y cada entrada distinta de cero de $A_{\Delta I}$ (y $A_{I\Delta}$) representa un acoplamiento entre los grados de libertad asociados a la misma subestructura. \tilde{A} puede obtenerse ensamblado parcialmente contribuciones locales asociadas con las subestructuras.

Las variables de los conjuntos I y Π son eliminadas y el complemento de Schur, asociado con los grados de libertad del conjunto Δ , de las fronteras interiores se define:

$$\tilde{S} := A_{\Delta\Delta} - (A_{I\Delta}^T A_{\Pi\Delta}^T) \begin{pmatrix} A_{II} & A_{I\Pi} \\ A_{I\Pi}^T & A_{\Pi\Pi} \end{pmatrix}^{-1} \begin{pmatrix} A_{I\Delta} \\ A_{\Pi\Delta} \end{pmatrix} \quad (3.34)$$

También obtendremos el lado derecho reducido \tilde{f}_Δ de la construcción de vectores asociados con los subdominios individuales. Denotaremos por $u_\Delta \in \tilde{W}_\Delta$ al vector de grados de libertad asociado con la frontera. Podemos reformular el problema reduciéndonos al subespacio \tilde{W}_Δ , como un problema de minimización con restricciones dado por el requerimiento de continuidad a través de todo Γ : encontrando $u_\Delta \in \tilde{W}_\Delta$ de manera que:

$$\begin{cases} J(u_\Delta) := \frac{1}{2} \langle \tilde{S} u_\Delta, u_\Delta \rangle - \langle \tilde{f}_\Delta, u_\Delta \rangle \rightarrow \min \\ B_\Delta u_\Delta = 0 \end{cases} \quad (3.35)$$

La matriz B_Δ es construida de $\{0, 1, -1\}$ (ecuación (3.30)) y de tal manera que los valores de la solución u , en los nodos comunes a más de un subdominio, coinciden cuando $B_\Delta u = 0$. Mediante la introducción de un conjunto de multiplicadores de Lagrange $\lambda \in V := \text{rango}(B_\Delta)$ para hacer cumplir las restricciones de continuidad. Como \tilde{A} es simétrica, positiva definida, entonces lo es \tilde{S} . Entonces podemos eliminar el subvector u_Δ , y obtener el siguiente sistema para multiplicadores de Lagrange:

$$F\lambda = d, \quad (3.36)$$

con

$$F := B_\Delta \tilde{S}^{-1} B_\Delta^T, \quad d := B_\Delta \tilde{S}^{-1} \tilde{f}_\Delta.$$

una vez que se encuentra λ , podemos regresar y resolver

$$u_\Delta = \tilde{S}^{-1}(\tilde{f}_\Delta - B_\Delta^T \lambda) \in \tilde{W}_\Delta.$$

Se introduce un preconditionador para F, para esto se definen matrices diagonales escalonadas $D_\Delta^{(i)}$. Cada uno de los elementos de la diagonal corresponde a un multiplicador de Lagrange el cual asegura la continuidad entre los valores en los nodos de algún $w_i \in W_i$ y $w_j \in W_j$ en algún punto $x \in \Gamma_h$, esto está dado por $\delta_j^\dagger(x)$. También definiremos un operador de salto, dado por:

$$B_{D,\Delta} := (D_\Delta^{(1)} B_\Delta^{(1)}, \dots, D_\Delta^{(N)} B_\Delta^{(N)}),$$

el bloque $B_\Delta^{(i)}$ se obtiene extrayendo las columnas de B_Δ asociadas al espacio local W_i . Se resuelve el sistema dual (3.36) usando el algoritmo de gradiente conjugado preconditionado con el preconditionador:

$$M^{-1} := B_{D,\Delta} S_\Delta B_{D,\Delta}^T = \sum_{i=1}^N D_\Delta^{(i)} B_\Delta^{(i)} S_\Delta^{(i)T} B_\Delta^{(i)T} D_\Delta^{(i)}, \quad (3.37)$$

donde $S_\Delta^{(i)}$ es la restricción del complemento de Schur local $S^{(i)}$ por $\tilde{W}_{\Delta,i} \subset W_i$. El método FETI-DP es el algoritmo del gradiente conjugado preconditionado estándar que resuelve el sistema preconditionado:

$$M^{-1}F\lambda = M^{-1}d$$

Este último método de descomposición de dominio hace cumplir la solución de la interfaz de los subdominios vía los multiplicadores de Lagrange, excepto en las esquinas de éstos, las cuales siguen siendo variables primales. El método ha mejorado haciendo que el promedio de la solución se satisfaga a través de los bordes o caras en la interfaz de los subdominios, lo cual es importante para la escalabilidad en paralelo en problemas tridimensionales. El algoritmo FETI-DP es una simplificación y una versión mejorada de FETI. Los eigenvalores usados en FETI-DP (véase B.1) son los mismos que los de BDDC, excepto para el eigenvalor igual a uno, fuera de eso el rendimiento de FETI-DP y BDDC es esencialmente el mismo. Por lo anterior, estos métodos son muy adecuados para el cómputo de alto rendimiento.

3.3.6. El Método BDDC

El método de descomposición de dominio balanceado con restricciones, mejor conocido como BDDC (Balancing Domain Decomposition with Constraints) comparte algunas similitudes con el método FETI y FETI-DP presentados en la sección anterior. Los métodos FETI-DP y BDDC no requieren la solución de sistemas lineales con matrices singulares, como en el caso de los métodos FETI de primer nivel (sección (3.3.4)). En el método BDDC las variables primarias de la solución iterativa son desplazamientos mientras que en el método FETI-DP son multiplicadores de Lagrange. Las restricciones en BDDC se introducen en los espacios locales (partición dentro de cada subdominio) y los correspondientes a la malla gruesa (subdominios), descomponiéndolos por medio de una suma directa.

Para cada subdominio Ω_i introducimos restricciones en las variables sobre Γ_i por medio de una matriz de restricciones C_i de tamaño $n_{c_i} \times n_i$, donde n_{c_i} es el número de grados de libertad locales y n_i es el número de grados de libertad sobre Γ_i . Consideraremos el método BDDC basado en vértices. En cada subestructura Ω_i nos referimos a una esquina como el vértice de Γ_i que comparte más de dos subdominios, denotaremos por \mathcal{N}_i la colección de dichos nodos sobre Γ_i . En el método BDDC basado en vértices, n_{c_i} es la dimensión de \mathcal{N}_i y para cada vector $u_{\Gamma}^{(i)}$ sobre Γ_i , el vector $C_i u_{\Gamma}^{(i)}$ serán los valores de $u_{\Gamma}^{(i)}$ sobre los nodos en \mathcal{N}_i .

En los métodos BDDC, requerimos dos condiciones de consistencia:

1. Un espacio local consistente con las matrices de peso en el sentido que: para cualquier $u_i \in \Gamma_i$ tal que $C_i u_i = 0$ entonces

$$C_j \bar{R}_j \bar{R}_i^T D_i^{-1} u_i = 0 \quad \forall j = 1, \dots, E. \quad (3.38)$$

2. Un espacio global consistente entre subestructuras con el fin de definir una interpolación en el sentido que: para cada v sobre Γ , existe u_c sobre \mathbb{R}^{n_c} tal que

$$C_i \bar{R}_i v = R_{c_i} u_c \quad \forall i = 1, \dots, E. \quad (3.39)$$

Las restricciones en vértices imponen condiciones Dirichlet en los vértices de las esquinas en los problemas locales y funciones base globales (referente a la malla gruesa).

El problema Global

En cada subestructura Ω_i definimos las funciones base locales como $\Phi_i^j := w_\Gamma$ ($j = 1, \dots, n_{c_i}$, donde n_{c_i} es el número de grados de libertad locales), como las soluciones de:

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} & 0 \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} & C_i^T \\ 0 & C_i & 0 \end{pmatrix} \begin{pmatrix} w_I \\ w_\Gamma \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e_j \end{pmatrix} \quad (3.40)$$

donde e_j es la columna j -ésima de la matriz identidad de tamaño n_{c_i} . Denotaremos $\Phi_j = [\Phi_i^1 \cdots \Phi_i^{n_{c_i}}]$ la matriz cuyas columnas son las funciones base locales. Por lo tanto, en el caso del método BDDC basado en vértices, este es un problema con una condición Dirichlet impuesta sobre uno de los nodos j -ésimos y cero en los demás nodos de \mathcal{N}_i , además de tener valores Neumann cero en los nodos de interfaz restantes.

Los espacios locales $V_0^{(i)}$ se definen como:

$$V_0^{(i)} \equiv \{u_0 \in \Gamma_i; u_0 = \Phi_i u_{c_i} \text{ y } u_{c_i} \in \mathbb{R}^{n_{c_i}}\} \quad (3.41)$$

En el caso del método BDDC basado en vértices, la segunda condición de consistencia sobre las restricciones impone continuidad en los vértices de esquina entre espacios vecinos $V_0^{(i)}$. De hecho, esta condición de consistencia impone continuidad en las variables primales. En el método BDDC preconditionado, la construcción de la matriz S_0^c se realiza como:

$$S_0^c = \sum_{i=1}^E R_{c_i}^T \Phi_i^T S_i \Phi_i R_{c_i}, \quad (3.42)$$

aquí es interesante notar que esta construcción puede realizarse en los subdominios con muy pocos pasos de comunicación entre los subdominios, ya que estos cálculos involucran solamente a las subestructuras que comparten grados de libertad. El problema en la malla es definido como:

$$Q_0 = R_0^T S_0^c R_0 \quad (3.43)$$

donde R_0 es la matriz de restricción de grados de libertad sobre Γ a los grados de libertad de la malla gruesa (global), definida como:

$$R_0 \equiv \sum_{i=1}^E R_{c_i}^T \Phi_i^T D_i^{-1} \bar{R}_i \quad (3.44)$$

con R_{c_i} la matriz de ceros y unos la cual mapea la malla global a los grados de libertad de las mallas locales. También notamos que S_0^c es distinta a la matriz global $S_0 = R_0 S R_0^T$ cuyo procesamiento es más costoso, ya que su cálculo incluye los vecinos de los subdominios vecinos de los subdominios involucrados en el cálculo de S_0^c

Los problemas locales

Los problemas locales son definidos en las variables duales del espacio de la malla. Los espacios locales son definidos como:

$$V_\Gamma^{(i)} \equiv \{u_i \in \Gamma_i ; C_i u_i = 0\}. \quad (3.45)$$

La primer condición de consistencia del método BDDC impone una consistencia sobre los pesos D_i^{-1} , en el sentido que si u_i está en $V_\Gamma^{(i)}$ entonces para cualquiera $j = 1, \dots, E$ el vector $\bar{R}_j \bar{R}_i^T D_i^{-1} u_i$ está en $V_\Gamma^{(j)}$.

Definiremos los problemas locales como:

$$Q_i = \bar{R}_i^T D_i^{-1} S_i^{c-1} D_i^{-1} \bar{R}_i \quad (3.46)$$

donde $u_\Gamma^{(i)} = S_i^{c-1} \tilde{b}_i$ es definida como la solución de

$$\begin{pmatrix} S_i & C_i^T \\ C_i & 0 \end{pmatrix} \begin{pmatrix} u_\Gamma^{(i)} \\ \lambda_i \end{pmatrix} = \begin{pmatrix} \tilde{b}_i \\ 0 \end{pmatrix} \quad (3.47)$$

lo cual es equivalente a resolver:

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} & 0 \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} & C_i^T \\ 0 & C_i & 0 \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_\Gamma^{(i)} \\ \lambda_i \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{b}_i \\ 0 \end{pmatrix} \quad (3.48)$$

Las restricciones y su aplicación

Un vértice de esquina es un vértice tal que comparte más de dos subdominios. La arista entre dos subestructuras Ω_i y Ω_j es la colección de vértices en un segmento conectado de $\bar{\Omega}_i \cap \bar{\Omega}_j$. La colección de esquinas de una subestructura Ω_i será denotada como \mathcal{N}_i y la colección de aristas como \mathcal{E}_i . Consideraremos las siguientes restricciones:

1. Fijar los valores de las esquinas, o
2. Fijar los promedios en las aristas

Como ya se mencionó anteriormente, las restricciones en los valores en las esquinas imponen una condición Dirichlet en el problema local y en la definición de las funciones base globales por medio de la matriz de restricción C_i , donde $C_i u_i$ es el valor de u_i en los nodos de esquina de la subestructura Ω_i . Estas restricciones hacen los problemas locales (3.48) solubles sin necesidad de un problema extra sobre la malla, como es el caso en el método Neumann-Neumann balanceado.

La introducción de las restricciones en el vértice se hace imponiendo la condición de Dirichlet en el sistema lineal. Definiremos $\Gamma_\Pi = \mathcal{N}_i$ y $\Gamma_\Delta = \Gamma \setminus \Gamma_\Pi$, donde Π se establece para nodos primales y Δ para nodos duales. Por tanto, el problema local (3.48) se convierte en:

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma_\Delta}^{(i)} & 0 \\ A_{\Gamma_\Delta I}^{(i)} & A_{\Gamma_\Delta \Gamma_\Delta}^{(i)} & C_i^T \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_{\Gamma_\Delta}^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{b}_i \end{pmatrix} \quad (3.49)$$

Y las funciones de la malla $\Phi_i^j = \begin{pmatrix} u_{\Gamma_\Delta}^{(i)} \\ e_j \end{pmatrix}$ se calculan como:

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma_\Delta}^{(i)} & 0 \\ A_{\Gamma_\Delta I}^{(i)} & A_{\Gamma_\Delta \Gamma_\Delta}^{(i)} & C_i^T \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_{\Gamma_\Delta}^{(i)} \end{pmatrix} = \begin{pmatrix} -A_{I\Gamma_\Pi}^{(i)} e_j \\ -A_{\Gamma_\Delta \Gamma_\Pi}^{(i)} e_j \end{pmatrix}, \quad (3.50)$$

donde e_j es la columna j de la matriz identidad de dimensión $n_{c_i} \times n_{c_i}$.

En la versión que se basa en las aristas, impondremos un promedio a cada arista. A cada renglón j de C_i le asociaremos una arista E_j . La matriz C_i es definida de tal manera que el elemento de la línea j del vector $C_i u_i$ es el promedio del vector u_i sobre la arista. Ahora, en una malla estructurada, los renglones de C_i son las entradas cero, con unos en las columnas asociadas a los nodos de la arista asociada.

Forma matricial del Precondicionador

Los espacios de la malla local $V_0^{(i)}$ y los espacios locales V_i proveen una descomposición en suma directa del espacio V_Γ de la siguiente manera:

$$V_\Gamma = \tilde{R}_0^T V_0 \oplus \sum_{i=1}^E R_i^T V_i \quad (3.51)$$

donde $V_0 = \prod_{i=1}^E V_0^{(i)}$ es discontinua sobre los grados de libertad duales y continua en los grados de libertad primales. Para cada $v = (v_i)_{i=1}^E \in V_0$ definimos el operador extensión $\tilde{R}_0^T v = \sum_{i=1}^E \tilde{R}_i^T D_i^{-1} v_i$.

El preconditionador tiene una forma de preconditionador de Schwarz aditivo de dos niveles, esto es:

$$M^{-1} = \sum_{i=1}^E Q_i + Q_0 \quad (3.52)$$

donde Q_i es definido en (3.46) y Q_0 en (3.43). Por tanto, el operador preconditionado es:

$$M^{-1}S = \sum_{i=1}^E Q_i S + Q_0 S \quad (3.53)$$

En resumen, se trataron en este capítulo dos de los algoritmos más usados en la actualidad de los métodos de descomposición de dominio sin traslape: FETI (Finite Element Tearing and Interconnect) y BDDC (Balancing Domain Decomposition by Constraints). Ambos algoritmos inician con la discretización de la ecuación diferencial parcial y en ellos los grados de libertad están asociados con las funciones base usadas en la discretización. El algoritmo FETI es un método indirecto que usa multiplicadores de Lagrange, por otro lado, el algoritmo BDDC es un método directo que no usa multiplicadores de Lagrange. En el capítulo siguiente se retomarán dichos métodos situándolos dentro del marco DVS (Derived Space Vector).

Capítulo 4

Métodos DVS

4.1. Introducción

Como ya se menciono anteriormente, existen varios métodos de descomposición de Dominio o DDM's. Los que son de superposición y los que no. En este caso, el método de Descomposición utilizado en el espacio Vectorial Derivado es un método sin superposición de subdominios y por tanto el manejo del código para computo en paralelo es más preciso y efectivo en el momento de operar.

El espacio Vectorial Derivado (DVS) es un esquema primal similar a la formulación BDD; con la diferencia que en la formulación DVS el problema es transformado en otro definido en el espacio vectorial derivado, el cual es un espacio producto conteniendo funciones discontinuas en el cual todo el trabajo del método es realizado en este espacio. En la formulación BDD por otro lado, el espacio original de funciones continuas nunca es abandonado completamente y constantemente se regresa a los grados de libertad asociados con el espacio de funciones continuas pertenecientes a las subestructuras, el cual en su formulación juega el rol del espacio producto.

Los métodos más eficientes hasta ahora son: BDDC y FETI-DP, los cuales fueron puestos en el marco primal del espacio vectorial derivado, el cual permite una efectiva y sintética presentación de ambos métodos: Formulaciones primal y dual.

Cada producto interior $\langle \cdot, \cdot \rangle$ en un espacio vectorial H , que puede ser real o complejo, da lugar a una norma $\|\cdot\|$, que se define como:

$$\|x\| = \sqrt{\langle x, x \rangle} \quad (4.1)$$

Se dice que H es un espacio de Hilbert si es completo con respecto esta norma. Completo en este contexto significa que cualquier sucesión de Cauchy ($\{x_n\}_{n \in \mathbb{N}}$ es de Cauchy, si $\forall \varepsilon \in \mathbb{R} > 0 \exists N \in \mathbb{N} \ni \forall m, n > N; |x_m - x_n| < \varepsilon$) de elementos del espacio converge a un elemento en el espacio, en el sentido que la norma de las diferencias tiende a cero.

El espacio vectorial derivado constituye un espacio de Hilbert con respecto al adecuado producto interior -el producto interior Euclidiano- y mediante el uso de la formulación DVS, tomamos ventaja de la estructura del espacio de Hilbert obteniendo de esta manera una gran simplicidad para el algoritmo definido en el espacio de funciones definidas por tramos y es usado para establecer una clara correspondencia entre los problemas a nivel continuo y aquellos obtenidos después de la discretización.

La formulación algorítmica se ve simplificada al desarrollar las operaciones en el espacio vectorial sujeto a restricciones, además que su aplicación a problemas prácticos incluyen en su discretización tanto a matrices simétricas como no simétricas e indefinidas. Esta formulación se resume en un conjunto de ocho fórmulas matriciales, que se aplican a matrices simétricas, no simétricas e indefinidas generadas mediante la discretización de la ecuación o sistemas de ecuaciones diferenciales parciales.

4.2. Marco Teórico

La formulación DVS inicia con la discretización del sistema de ecuaciones lineales que se obtiene después de la ecuación diferencial parcial, o un sistema de dichas ecuaciones. Podemos llamar a este sistema de ecuaciones lineales el “*problema original*” y nos referiremos a los nodos usados en este proceso como ‘*nodos originales*’ (Figura 4.1). Independientemente del método de discretización a usar, se asume que un conjunto de nodos y una partición del dominio son definidos y tanto los nodos como los subdominios son numerados. Generalmente, algunos nodos pertenecen a más de un subdominio (ver Figura 4.2); para la formulación de los métodos de descomposición sin superposición esto es un inconveniente.

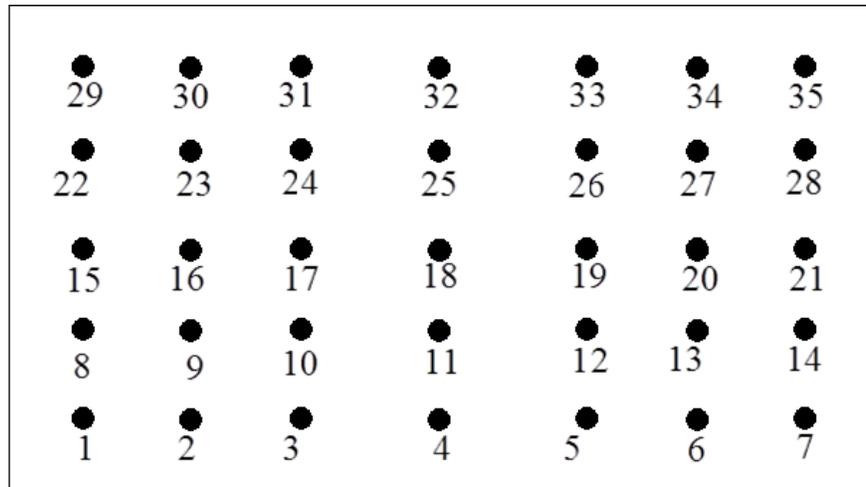


Figura 4.1: Nodos originales

Que cada nodo pertenezca a uno y sólo una de las particiones del subdominio (malla gruesa) puede ser ventajoso para alcanzar el *paradigma DDM*: “Obtener la solución del problema *global* resolviendo problemas *locales*”. Con el fin de lograr dicho paradigma, el esquema DVS introduce un nuevo conjunto de nodos, que son llamados ‘*vectores derivados*’ (ver Figura 4.3); un *nodo derivado* es una pareja de números naturales definido como: en su primer entrada un índice de nodo y en la segunda un índice de subdominio. A cada subdominio podemos asociarle un subconjunto local de nodos derivados, donde la segunda entrada del vector derivado corresponde al subdominio en cuestión.

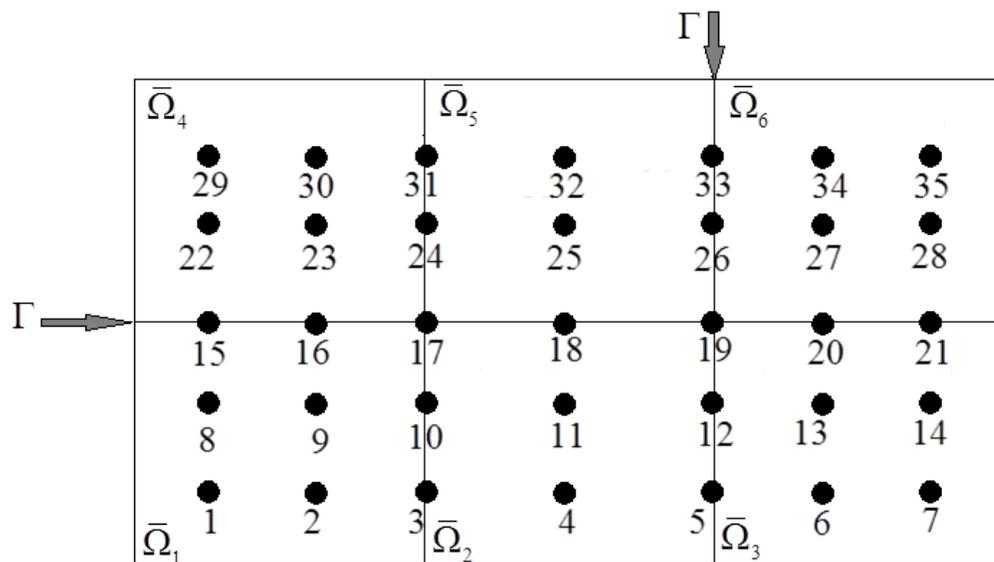


Figura 4.2: Nodos originales dentro de la malla gruesa

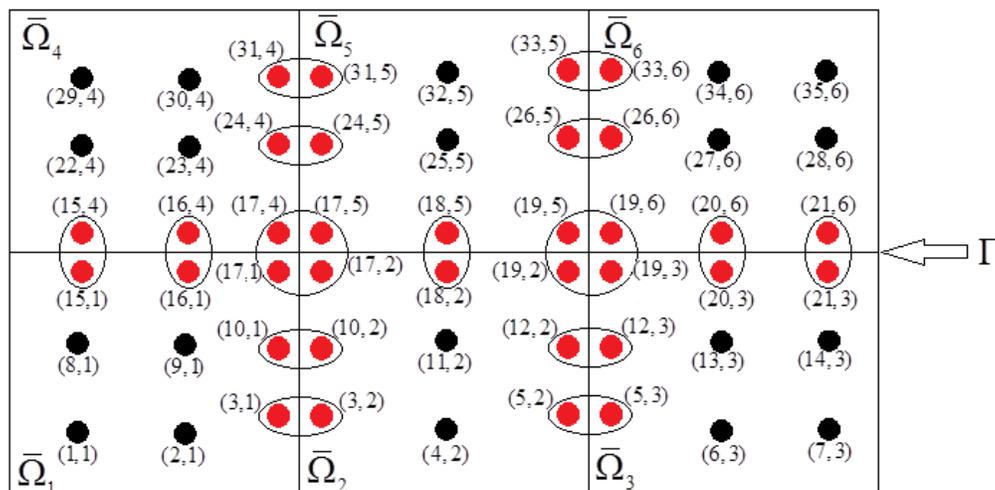


Figura 4.3: Nodos derivados

La familia de subconjuntos locales de nodos derivados obtenidos, una por cada subdominio, constituye una partición realmente disjunta (es decir, no superpuesta) de todo el conjunto de nodos derivados. Un vector derivado se define por ser una función real-valuada definida dentro del conjunto de nodos derivados; el conjunto de todos los nodos derivados constituye un espacio lineal: El espacio de vectorial derivado, (DVS, por sus siglas en inglés, Derived-Vector Space).

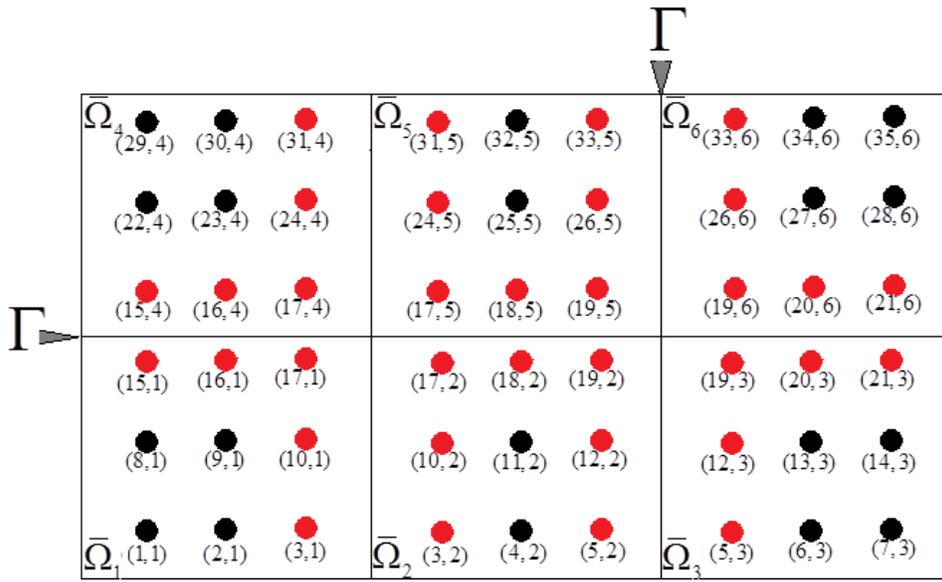


Figura 4.4: Los *Nodos derivados* distribuidos en la malla gruesa. Los nodos de color rojo indican aquellos que sufrieron de mitosis.

Un nuevo problema que es equivalente al problema original, es definido en el espacio de vectorial derivado. Desde luego, la matriz de este problema es distinta a la Matriz original, la cual no está definida en el espacio de vectorial derivado, y la teoría suministra la fórmula para obtenerla; el procedimiento para construirla es similar a sub-estructurarla. De ahora en adelante, en el marco DVS, todo el trabajo está hecho en el espacio vectorial derivado y del cual no se regresará al espacio vectorial original.

El espacio vectorial derivado es un tipo de espacio producto; nombrado así por el producto de las familias de subconjuntos locales de nodos derivados mencionados anteriormente.

Más aún, constituye un espacio de Hilbert (de dimensión finita) con respecto a un adecuado producto interior, llamado el producto interior Euclidiano y se trabaja con él, en consecuencia, a lo largo del desarrollo de la teoría. A pesar que el esquema DVS se desarrolló originalmente teniendo en mente aplicaciones a matrices simétricas y definidas, este esquema puede extenderse a matrices no simétricas e indefinidas.

4.2.1. El Problema Original

El esquema DVS se aplica a la Matriz del sistema de ecuaciones obtenida en la discretización. Este procedimiento es independiente del método de discretización usado, que puede ser Volumen Finito, Diferencias Finitas, FEM o algún otro. Sin embargo, se requiere de algunas asunciones (o axiomas) que satisfacer. Estos axiomas son establecidos en términos de la matriz del sistema de ecuaciones y en dos conceptos adi-

cionales: los nodos originales y una familia de subconjuntos de dichos nodos, que está asociada con la partición del dominio (o descomposición de dominio). Para ilustrar como estos conceptos son introducidos, consideremos la formulación variacional de la versión discretizada de un problema general de valores a la frontera. Esto consiste en encontrar $\widehat{u} \in V$, tal que:

$$\tilde{a}(\widehat{u}, v) = (g, v), \quad \forall v \in V \quad (4.2)$$

Donde V es un espacio lineal de dimensión finita de funciones real-valuadas definidas en un cierto dominio espacial Ω , en donde $g \in V$ es una función dada.

Sea $\widehat{N} \equiv \{1, \dots, n\}$ el conjunto de índices, que numeran los nodos usados en la discretización y sea $\{\varphi_1, \dots, \varphi_n\} \subset V$ una base de V , tal que para cada $i \in \widehat{N}$, $\varphi_i = 1$ en el nodo i y cero en cualquier otro nodo. Entonces, ya que $\widehat{u} \in V$, tenemos lo siguiente:

$$\widehat{u} = \sum_{i=1}^n \widehat{u}_i \varphi_i \quad (4.3)$$

donde \widehat{u}_i es el valor de \widehat{u} en el nodo i . Sean $\underline{\widehat{u}}$ y $\underline{\widehat{f}}$ los vectores $\underline{\widehat{u}} \equiv (\widehat{u}_1 \dots \widehat{u}_n)$ y $\underline{\widehat{f}} \equiv (\widehat{f}_1 \dots \widehat{f}_n)$ con:

$$\widehat{f}_i \equiv (g, \varphi_i), \quad i \in \widehat{N} \quad (4.4)$$

La formulación variacional de (4.2) es equivalente a:

$$\underline{\widehat{A}} \underline{\widehat{u}} = \underline{\widehat{f}} \quad (4.5)$$

La matriz $\underline{\widehat{A}}$, a la que nos referiremos como la 'Matriz Original' está dada por:

$$\underline{\widehat{A}} = (\widehat{A}_{ij}) \quad \text{con} \quad \widehat{A}_{ij} \equiv \tilde{a}(\varphi_j, \varphi_i), \quad i, j = 1, \dots, n \quad (4.6)$$

Después de que el problema ha sido discretizado, se introduce una partición de Ω en un conjunto de subdominios no superpuestos, $\{\Omega_1, \dots, \Omega_E\}$; siendo precisos, para cada $\alpha = 1, \dots, E$; Ω_α , es abierto y:

$$\Omega_\alpha \cap \Omega_\beta = \emptyset \quad \text{y} \quad \Omega \subset \bigcup_{\alpha=1}^E \overline{\Omega}_\alpha \quad (4.7)$$

donde $\overline{\Omega}_\alpha$ conlleva a la cerradura de Ω_α . El conjunto de 'índices de subdominio' será:

$$\widehat{E} = \{1, \dots, E\} \quad (4.8)$$

\widehat{N}^α , donde $\alpha = 1, \dots, E$, será usado para el subconjunto de nodos originales que corresponden a nodos pertenecientes a Ω_α . Como es usual, los nodos serán clasificados

como 'nodos interiores' y 'nodos de frontera interna': un nodo es interior si pertenece sólo a un subdominio cerrado y es de frontera interna si pertenece a más de uno. Para aplicar los métodos duales-primales, los nodos de frontera interna serán clasificados como nodos 'primales' y 'duales'. Se define:

- $\widehat{N}_I \subset \widehat{N}$ el conjunto de nodos Interiores.
- $\widehat{N}_\Gamma \subset \widehat{N}$ el conjunto de nodos de Frontera interna.
- $\widehat{N}_\pi \subset \widehat{N}$ el conjunto de nodos Primales.
- $\widehat{N}_\Delta \subset \widehat{N}$ el conjunto de nodos Duales.

El conjunto $\widehat{N}_\pi \subset \widehat{N}_\Gamma$ se escoge arbitrariamente y \widehat{N}_Δ es definido como $\widehat{N}_\Delta \equiv \widehat{N}_\Gamma - \widehat{N}_\pi$. Cada uno de los dos siguientes subconjuntos de familias de nodos son disjuntos: $\{\widehat{N}_I, \widehat{N}_\Gamma\}$ y $\{\widehat{N}_I \widehat{N}_\pi, \widehat{N}_\Delta\}$. Además estos subconjuntos de nodos cumplen con las siguientes relaciones:

$$\widehat{N} = \widehat{N}_I \cup \widehat{N}_\Gamma = \widehat{N}_I \cup \widehat{N}_\pi \cup \widehat{N}_\Delta \quad \text{y} \quad \widehat{N}_\Gamma = \widehat{N}_\pi \cup \widehat{N}_\Delta \quad (4.9)$$

Las funciones real-valuadas definidas en $\widehat{N} = \{1, \dots, n\}$ constituyen un espacio vectorial lineal que será denotado por \widehat{W} y nos referiremos a él como 'el espacio vectorial original'. Los vectores $\widehat{u} \in \widehat{W}$ serán escritos como $\widehat{u} = (\widehat{u}_1, \dots, \widehat{u}_n)$, donde \widehat{u}_i para $i = 1, \dots, n$, son las componentes del vector \widehat{u} . Ahora, para el 'problema original' la solución consiste en : Dada una $\widehat{f} \in \widehat{W}$, encontrar una $\widehat{u} \in \widehat{W}$ tal que la ecuación (4.5) se satisfaga. A lo largo de nuestro desarrollo la matriz original \widehat{A} se asume no singular. (es decir, define una biyección de \widehat{W} en sí mismo.)

Sean los índices $i \in \widehat{N}^\alpha$ y $j \in \widehat{N}^\beta$ índices de nodos originales interiores, entonces:

$$\widehat{A}_{ij} = 0, \quad \text{cuando} \quad \alpha \neq \beta \quad (4.10)$$

Nodos Derivados

Cuando una partición sin superposición es introducida, algunos de los nodos usados en la discretización pertenecen a más de un subdominio de la partición. Para superar este inconveniente, en el esquema DVS además de los nodos originales, otro conjunto de nodos es introducido, el conjunto de nodos derivados. En general, se tiene una mejor interpretación del desarrollo al introducir un ejemplo.

Consideremos el conjunto de 35 nodos en una descomposición de dominio sin superposición; que consiste en seis subdominios como lo muestra la figura 4.2. Tendremos un conjunto de nodos y uno de subdominios, los cuales son numerados usando el conjunto de índices: $\widehat{N} \equiv \{1, \dots, 35\}$ y $\widehat{E} \equiv \{1, 2, 3, 4, 5, 6\}$, respectivamente. Ahora, el conjunto de nodos correspondiente a esta descomposición de dominio sin superposición, está actualmente superpuesta ya que los cuatro subconjuntos son:

$$\begin{aligned}
\widehat{N}^1 &\equiv \{1, 2, 3, 8, 9, 10, 15, 16, 17\} \\
\widehat{N}^2 &\equiv \{3, 4, 5, 10, 11, 12, 17, 18, 19\} \\
\widehat{N}^3 &\equiv \{5, 6, 7, 12, 13, 14, 19, 20, 21\} \\
\widehat{N}^4 &\equiv \{15, 16, 17, 22, 23, 24, 29, 30, 31\} \\
\widehat{N}^5 &\equiv \{17, 18, 19, 24, 25, 26, 31, 32, 33\} \\
\widehat{N}^6 &\equiv \{19, 20, 21, 26, 27, 28, 33, 34, 35\}
\end{aligned} \tag{4.11}$$

y como se puede observar en la Figura 4.2, estos son subconjuntos traslapados. Por ejemplo, en este caso:

$$\widehat{N}^1 \cap \widehat{N}^2 = \{3, 10, 17\} \tag{4.12}$$

Con el objetivo de obtener una verdadera descomposición sin superposición, reemplazaremos el conjunto de nodos originales por otro, el de nodos derivados (ver figura 4.3). Un nodo derivado se define por una pareja de números: (p, α) , donde p corresponde a un nodo que pertenece a \widehat{N}_α . En símbolos podemos decir que un nodo derivado es una pareja ordenada (p, α) , tal que $p \in \widehat{N}^\alpha$.

Denotaremos como X al conjunto de nodos derivados; observamos que el total de nodos derivados es 54 (ver Figura 4.4), mientras que el de nodos originales es 35. Ahora, definimos X^α como el conjunto de nodos derivados que pueden ser escritos como (p, α) , donde α se mantiene fija. Tomando α sucesivamente como 1,2,3,4,5 y 6, obtenemos la familia de seis subconjuntos, $\{X^1, X^2, X^3, X^4, X^5, X^6\}$, que es una verdadera descomposición disjunta de X , en el sentido que:

$$X = \bigcup_{\alpha=1}^6 X^\alpha \text{ y } X^\alpha \cap X^\beta = \emptyset, \text{ donde } \alpha \neq \beta \tag{4.13}$$

Desde luego, la cardinalidad de cada uno de estos subconjuntos es $54/6$ igual a 9. El conjunto del total de nodos derivados satisface:

$$X = \{(p, \alpha) \mid \alpha \in \widehat{E} \ \& \ p \in \widehat{N}^\alpha\} \tag{4.14}$$

Definiremos algunos subconjuntos de X útiles en el desarrollo de la teoría:

$$I \equiv \{(p, \alpha) \mid p \in \widehat{N}_I\} \text{ y } \Gamma \equiv \{(p, \alpha) \mid p \in \widehat{N}_\Gamma\} \tag{4.15}$$

$$\pi \equiv \{(p, \alpha) \mid p \in \widehat{N}_\pi\} \text{ y } \Delta \equiv \{(p, \alpha) \mid p \in \widehat{N}_\Delta\} \tag{4.16}$$

Para cada $\alpha = 1, \dots, E$ asociaremos un único subconjunto local de nodos derivados:

$$X^\alpha \equiv \{(p, \alpha)\} \tag{4.17}$$

La familia de subconjuntos $\{X^1, \dots, X^E\}$ es una verdadera descomposición de dominio de todo el conjunto de nodos derivados, en el sentido que:

$$X = \bigcup_{\alpha=1}^E X^\alpha \text{ y } X^\alpha \cap X^\beta = \emptyset, \text{ cuando } \alpha \neq \beta \quad (4.18)$$

4.3. La Teoría DVS

Retomaremos la definición del espacio vectorial \widehat{W} ; veamos que para cada $\alpha = 1, \dots, E$, definimos un subespacio vectorial tal que: $\widehat{W}^\alpha \subset \widehat{W}$, el cual se construye de vectores que tienen la propiedad de que para cada $i \notin \widehat{N}^\alpha$, su i -ésima componente se anula. Con esta notación se puede definir el espacio Producto W como:

$$W \equiv \prod_{\alpha=1}^E \widehat{W}^\alpha = \widehat{W}^1 \times \dots \times \widehat{W}^E \quad (4.19)$$

El problema original de la ecuación (4.5) es un problema formulado en el espacio vectorial \widehat{W} y en el desarrollo siguiente se transformará este problema en uno que es formulado en el espacio producto W , el cual es un espacio de funciones discontinuas.

Por 'vector derivado' entendemos una función de valores reales definida en el conjunto de nodos derivados X . El conjunto de nodos derivados constituye un espacio lineal, al cual nos referimos como 'El Espacio Vectorial Derivado'. Correspondiente a cada subconjunto local de nodos derivados, X^α , existe un subespacio local de vectores derivados, W^α , el cual está definido con la condición que los vectores en W^α se anulan en cada nodo derivado que no pertenezca a X^α . Una manera formal de escribir esta definición es:

$$\underline{u} \in W^\alpha \subset W, \text{ si y solo si } \underline{u}(p, \beta) = 0 \text{ siempre y cuando } \alpha \neq \beta \quad (4.20)$$

Una diferencia notable entre los subespacios W y W^α que debe ser observada es que $W^\alpha \subset W$, mientras que $\widehat{W}^\alpha \not\subset W$. En particular:

$$W \equiv \prod_{\alpha=1}^E \widehat{W}^\alpha = W^1 \oplus \dots \oplus W^E \quad (4.21)$$

En palabras, podremos definir la ecuación (4.21) como: El espacio W es el producto de una familia de subespacios $\{\widehat{W}^1, \dots, \widehat{W}^E\}$, pero al mismo tiempo es la suma directa de la familia $\{W^1, \dots, W^E\}$. Es natural establecer una biyección (de hecho, un isomorfismo) entre el espacio vectorial derivado y el espacio producto.

Para cada par de vectores $\underline{u} \in W$ y $\underline{w} \in W$, el producto interior Euclidiano es definido como:

$$\underline{u} \cdot \underline{w} = \sum_{(p, \alpha) \in X} \underline{u}(p, \alpha) \underline{w}(p, \alpha) \quad (4.22)$$

En aplicaciones a la teoría de sistema de ecuaciones, cuando $\underline{u}(p, \alpha)$ por sí mismo es un vector, la ecuación (4.22) es remplazada por:

$$\underline{u} \cdot \underline{w} = \sum_{(p, \alpha) \in X} \underline{u}(p, \alpha) \odot \underline{w}(p, \alpha) \quad (4.23)$$

Aquí, $\underline{u} \odot \underline{w}$ se refiere al producto interior de los vectores involucrados. Una propiedad importante es que el espacio vectorial derivado, W , constituye un Espacio de Hilbert finito con respecto al producto interior Euclidiano. Podemos definir este producto interior Euclidiano independientemente de la naturaleza de la matriz original $\underline{\hat{A}}$; en particular esta puede ser simétrica, no simétrica o indefnida.

La inyección natural, $R : \widehat{W} \rightarrow W$, de \widehat{W} en W , es definida por la condición de que para toda $\widehat{u} \in \widehat{W}$, se tiene que:

$$(R \widehat{u})(p, \alpha) = \widehat{u}(p), \quad \forall (p, \alpha) \in X \quad (4.24)$$

la multiplicidad, $m(p)$, para cualquier nodo original $p \in \widehat{N}$ se caracteriza por la propiedad (véase [6] y [7]):

$$\sum_{\alpha=1}^E (R \widehat{u})(p, \alpha) = m(p) \widehat{u}(p) \quad (4.25)$$

El espacio W será descompuesto en dos subespacios ortogonales complementarios: $W_{11} \subset W$ y $W_{12} \subset W$, entonces tenemos que:

$$W = W_{11} + W_{12} \quad \text{y} \quad \{0\} = W_{11} \cap W_{12} \quad (4.26)$$

Aquí, el subespacio $W_{12} \subset W$ es la inyección natural de \widehat{W} en W ; es decir:

$$W_{12} \equiv R \widehat{W} \subset W \quad (4.27)$$

y $W_{11} \subset W$ es el complemento ortogonal con respecto al producto interior Euclidiano. Para usos posteriores, señalamos que la inversa de $R : \widehat{W} \rightarrow W$, cuando es restringida a $W_{12} \subset W$, existe y se define por: $R^{-1} : W_{12} \rightarrow \widehat{W}$. Renombramos como es costumbre, usando la notación de suma directa:

$$W = W_{11} \oplus W_{12} \quad (4.28)$$

Esto se satisface cuando el par de igualdades de la ecuación (4.26) se cumplen. El *subespacio de vectores continuos* es definido como $W_{12} \subset W$, mientras que el *subespacio de vectores de promedio cero* se define como $W_{11} \subset W$.

Se introducen dos matrices; $\underline{\underline{a}} : W \rightarrow W$ y $\underline{\underline{j}} : W \rightarrow W$; estas matrices son operadores de proyección, con respecto al producto interior Euclidiano, en W_{11} y W_{12} , respectivamente.

El primero al que hacemos referencia es el *operador promedio* y el segundo será el *operador de salto*, respectivamente. Observamos que en la ecuación (4.26), cada vector, $\underline{u} \in W$, puede ser escrito de forma única como la suma de un vector de promedio cero más un vector continuo (se podría decir: un vector de salto cero) de hecho:

$$\underline{u} = \underline{u}_{11} + \underline{u}_{12} \quad \text{con} \quad \begin{cases} \underline{u}_{11} \equiv \underline{ju} \in W_{11} \\ \underline{u}_{12} \equiv \underline{au} \in W_{12} \end{cases} \quad (4.29)$$

Los vectores \underline{ju} y \underline{au} son llamados el 'salto' y el 'promedio' de \underline{u} , respectivamente.

A continuación definimos varios subespacios de W que usaremos dentro de la teoría. Estos son $W_I, W_\Gamma, W_\pi, W_\Delta$ y W_Π ; los vectores que pertenecen a cada uno de estos subespacios, se anulan en cada nodo derivado que no pertenezca a I, Γ, π, Δ y Π , respectivamente. Además, para especificar las restricciones que serán usadas en los algoritmos, introducimos el subespacio $W_r \subset W$ y se asume que tiene la siguiente propiedad:

$$W_r \equiv W_I \oplus \underline{a}'W_\pi \oplus W_\Delta \quad (4.30)$$

aquí, \underline{a}' representa el operador de proyección en W_r . Los subespacios lineales que se introdujeron anteriormente satisfacen lo siguiente:

$$W = W_I \oplus W_\Gamma = W_I \oplus W_\pi \oplus W_\Delta \quad \text{y} \quad W_r = W_\Pi \oplus W_\Delta \quad (4.31)$$

Además $W_\Pi \equiv W_I + \underline{a}W_\pi$

4.4. El problema DVS con restricciones

Un vector $\widehat{\underline{u}} \in \widehat{W}$ es solución del problema original, sí y sólo si, $\underline{u}' = R\widehat{\underline{u}} \in W$ satisface las siguientes igualdades:

$$\underline{a}A\underline{u}' = \underline{\bar{f}} \quad \text{y} \quad \underline{ju}' = 0 \quad (4.32)$$

El vector $\underline{\bar{f}} \equiv (R\widehat{\underline{f}}) \in W_{12} \subset W_r$ puede ser escrito como $\underline{\bar{f}} = \underline{\bar{f}}_{-\Pi} + \underline{\bar{f}}_{-\Delta}$, con $\underline{\bar{f}}_{-\Pi} \in W_\Pi$ y $\underline{\bar{f}}_{-\Delta} \in W_\Delta$.

Destacamos que este problema está formulado en W : el espacio vectorial derivado. En lo consecuente, la matriz \underline{A} , cuando es restringida a W_r (i.e., $\underline{A} : W_r \rightarrow W_r$), se asume que es invertible. En muchos casos esto es posible cuando se toma un número suficientemente grande de *nodos primales* adecuadamente localizados. Sea $\underline{u}' \in W_r$ la solución, entonces $\underline{u}' \in W_{12} \subset W$ necesariamente, ya que $\underline{ju}' = 0$ y uno puede aplicar la inversa de la *inyección natural* para obtener:

$$\widehat{\underline{u}} = R^{-1}\underline{u}' \quad (4.33)$$

De este modo observamos que el mapeo R^{-1} sólo está definido cuando \underline{u}' es un *vector continuo* (i.e., $\underline{u}' \in W_{12}$). La condición $\underline{u}' \in W_{12}$ solo puede lograrse cuando no se presentan errores de redondeo. Debido a este hecho, en aplicaciones numéricas la ecuación (4.33) debe ser remplazada por:

$$\underline{\hat{u}} = R^{-1} \underline{au}' \quad (4.34)$$

En efecto, cuando \underline{u}' no es continua, \underline{au}' es el vector continuo más cercano a \underline{u}' (aquí, 'cercano' es con respecto a la *distancia Euclidiana*). La ecuación (4.32) es un elemento clave del *marco DVS*, ya que suministra una formulación del problema original en el espacio vectorial derivado, en el cual los nodos tienen que ser particionados en paquetes disjuntos de nodos, que a su vez permite descomponer la matriz en sub-matrices acopladas por la frontera, que pueden ser tratadas en diferentes procesadores con un pequeño costo de coordinación y comunicación entre ellos.

4.4.1. La construcción de la matriz $\underline{\underline{A}}^t$

El enfoque del marco DVS inicia cuando la matriz es obtenida mediante alguna discretización y la aplicación de ésta no requiere ninguna información acerca de la ecuación diferencial parcial que la generó. Esta matriz no está definida dentro del espacio vectorial derivado. Definiremos $\underline{\underline{A}}^t : W \rightarrow W$, que es utilizada para formular el problema en el espacio Vectorial Derivado.

Sea la matriz $\widehat{\underline{\underline{A}}} : \widehat{W} \rightarrow \widehat{W}$ dada por (4.5), podemos expresarla como: $\widehat{\underline{\underline{A}}} \equiv (\widehat{A}_{pq})$. Para cada pareja (p, q) tal que $p \in \widehat{N}$ y $q \in \widehat{N}$, tal que:

$$\delta_{pq}^\alpha \equiv \begin{cases} 1, & \text{si } p, q \in \widehat{N}^\alpha \\ 0, & \text{si } p \notin \widehat{N}^\alpha \text{ o } q \notin \widehat{N}^\alpha \end{cases} \quad \alpha = 1, \dots, E. \quad (4.35)$$

además definimos:

$$m(p, q) \equiv \sum_{\alpha=1}^E \delta_{pq}^\alpha \quad (4.36)$$

y

$$s(p, q) \equiv \begin{cases} 1, & \text{cuando } m(p, q) = 0 \\ 0, & \text{cuando } m(p, q) \neq 0 \end{cases} \quad (4.37)$$

la función $m(p, q)$ es llamada la *multiplicidad* de la pareja ordenada (p, q) . Ahora definiremos:

$$\underline{\underline{\widehat{A}}}^\alpha \equiv (\widehat{A}_{pq}^\alpha) \quad \text{con} \quad \widehat{A}_{pq}^\alpha = \frac{\widehat{A}_{pq} \delta_{pq}^\alpha}{s(p, q)} \quad (4.38)$$

Notemos que surge la identidad: $\underline{\underline{\widehat{A}}} = \sum_{\gamma=1}^E \underline{\underline{\widehat{A}}}^{\gamma}$

Debido a que se utiliza una descomposición de dominio sin traslape. Además para cada $\gamma = 1, \dots, E$, la matriz $\underline{\underline{\widehat{A}}}^{\gamma} : W \rightarrow W$ se define como:

$$\underline{\underline{\widehat{A}}}^{\gamma} \equiv \left(A_{(p,\alpha)(q,\beta)}^{\gamma} \right) \quad (4.39)$$

con $A_{(p,\alpha)(q,\beta)}^{\gamma} \equiv \delta_{\alpha\gamma}^{\underline{\underline{\widehat{A}}}^{\gamma}} \delta_{\beta\gamma}$

Por tanto la matriz total $\underline{\underline{\widehat{A}}}^{\dagger} : W \rightarrow W$ es definida por:

$$\underline{\underline{\widehat{A}}}^{\dagger} \equiv \sum_{\gamma=1}^E \underline{\underline{\widehat{A}}}^{\gamma} \quad (4.40)$$

Una propiedad relevante de $\underline{\underline{\widehat{A}}}^{\dagger}$ es la siguiente:

$$\left(\underline{\underline{\widehat{A}}}^{\dagger} \right)^{-1} = \sum_{\gamma=1}^E \left(\underline{\underline{\widehat{A}}}^{\gamma} \right)^{-1} \quad (4.41)$$

La matriz $\underline{\underline{\widehat{A}}}^{\dagger}$ puede ser expresada en más de una manera. La ecuación anterior implica que si el complemento de Schur local se define como:

$$\underline{\underline{\widehat{S}}}^{\alpha} = \underline{\underline{\widehat{A}}}^{\alpha}_{\Gamma\Gamma} - \underline{\underline{\widehat{A}}}^{\alpha}_{\Gamma I} \left(\underline{\underline{\widehat{A}}}^{\alpha}_{II} \right)^{-1} \underline{\underline{\widehat{A}}}^{\alpha}_{I\Gamma} \quad (4.42)$$

de cada $\underline{\underline{\widehat{A}}}^{\alpha}$ existe y es invertible, entonces *la matriz total del complemento de Schur*, $\underline{\underline{\widehat{S}}}^{\dagger} : W_r \rightarrow W_r$ satisface que:

$$\underline{\underline{\widehat{S}}}^{\dagger} = \sum_{\alpha=1}^E \underline{\underline{\widehat{S}}}^{\alpha} \quad (4.43)$$

$$\left(\underline{\underline{\widehat{S}}}^{\dagger} \right)^{-1} = \sum_{\alpha=1}^E \left(\underline{\underline{\widehat{S}}}^{\alpha} \right)^{-1} \quad (4.44)$$

debido a que se debe a una verdadera descomposición de dominio y ya que satisface las ecuaciones (4.40) y (4.41)

4.4.2. Los algoritmos DVS con restricciones

La matriz $\underline{\underline{A}}$ de la ecuación (4.32), puede ser escrita como:

$$\underline{\underline{A}} = \begin{pmatrix} \underline{\underline{A}}_{\Pi\Pi} & \underline{\underline{A}}_{\Pi\Delta} \\ \underline{\underline{A}}_{\Delta\Pi} & \underline{\underline{A}}_{\Delta\Delta} \end{pmatrix} \quad (4.45)$$

Usando esta notación definimos 'la matriz del complemento de Schur con restricciones' como:

$$\underline{\underline{S}} = \underline{\underline{A}}_{\Delta\Delta} - \underline{\underline{A}}_{\Delta\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{A}}_{\Pi\Delta} \quad (4.46)$$

Además se tiene lo siguiente:

$$\underline{\underline{f}}_{-\Delta} = \left(R \hat{\underline{\underline{f}}} \right)_{\Delta} - \underline{\underline{A}}_{\Delta\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \left(R \hat{\underline{\underline{f}}} \right)_{\Pi} \quad (4.47)$$

Después la ecuación (4.32) es transformada en:

$$\underline{\underline{a}} \underline{\underline{S}} \underline{\underline{u}}_{\Delta} = \underline{\underline{f}}_{-\Delta} \quad \text{y} \quad \underline{\underline{j}} \underline{\underline{u}}_{\Delta} = 0 \quad (4.48)$$

Junto con:

$$\hat{\underline{\underline{u}}} = R^{-1} \underline{\underline{a}} \left\{ \left(\underline{\underline{I}} - \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{A}}_{\Pi\Delta} \right) \underline{\underline{u}}_{\Delta} + \underline{\underline{A}}_{\Pi\Pi}^{-1} \left(R \hat{\underline{\underline{f}}} \right)_{\Pi} \right\} \quad (4.49)$$

Generalmente, los algoritmos DDM se clasifican en *primales* y *duales*. La primera de estas clases incluye los que son directos, es decir, sin recurrir a multiplicadores de Lagrange y las segunda hace uso de ellos. Sin embargo, el marco DVS suministra una configuración *primal* muy general que permite incluir las dos clases en él, pero para hacer esto se cambian las directrices de la formulación.

En realidad, cada uno de los algoritmos está determinado completamente por la *información buscada*, es decir, la información que los algoritmos buscan de manera inmediata. La *información buscada* puede ser elegida de muchas formas alternativas; todo lo que se requiere es que la solución del *problema DVS* pueda ser derivada de esta información (véase [9]), de una forma económica (computacionalmente hablando).

4.4.3. La versión DVS del algoritmo BDDC

Cuando se realiza la incorporación del método BDDC en el esquema DVS se encuentran diferencias importantes. Por ejemplo, cuando las inversas del complemento de Schur local existen en el esquema DVS la inversa $\underline{\underline{S}}^t$ está dada por:

$$\left(\underline{\underline{S}}^t \right)^{-1} = \sum_{\alpha=1}^E \left(\underline{\underline{S}}^{\alpha} \right)^{-1} \quad (4.50)$$

una relación similar no es satisfecha por el algoritmo BDDC. Si tomamos el operador restricción $\bar{R}_i : \Gamma \rightarrow \Gamma_i$, podemos obtener que: $S = \sum_{i=1}^E \bar{R}_i^T S_i \bar{R}_i$. De esta forma podremos escribir:

$$\underline{\underline{S}} = \sum_{\alpha=1}^E \underline{\underline{\bar{R}}}_{\alpha}^T \underline{\underline{S}}_{\alpha} \underline{\underline{\bar{R}}}_{\alpha} \quad (4.51)$$

Sin embargo es importante notar que:

$$\left(\underline{\underline{S}}\right)^{-1} \neq \sum_{\alpha=1}^E \left(\underline{\underline{\bar{R}}}_{\alpha}^T \underline{\underline{S}}_{\alpha} \underline{\underline{\bar{R}}}_{\alpha}\right)^{-1} \quad (4.52)$$

aquí, la igualdad no se satisface debido a que los rangos de $\underline{\underline{\bar{R}}}_{\alpha}^T \underline{\underline{S}}_{\alpha} \underline{\underline{\bar{R}}}_{\alpha}$ y $\underline{\underline{\bar{R}}}_{\beta}^T \underline{\underline{S}}_{\beta} \underline{\underline{\bar{R}}}_{\beta}$ no son ajenos cuando $\alpha \neq \beta$. Esta última limitación es debido al hecho que en BDDC, la formulación no se establece directamente en el espacio producto y que en su implementación se retorna a los grados de libertad asociados con los nodos originales, mientras **en el esquema DVS uno se olvida completamente de los nodos originales y se trabaja exclusivamente con los nodos derivados**. Por lo tanto, las fórmulas desarrolladas para el esquema DVS y sus códigos computacionales se simplifican de manera notable. La unificación y simplificación lograda de esta manera permite producir Software genérico, robusto y eficiente.

La *información buscada* se define como $\underline{u}_{\Delta} \in W_{\Delta}$. Con esta elección se obtiene una versión del *algoritmo DVS-BDDC* (véase [9]): Encuentre $\underline{u}_{\Delta} \in W_{\Delta}$ tal que:

$$\underline{a} \underline{S}^{-1} \underline{a} \underline{S} \underline{u}_{\Delta} = \underline{a} \underline{S}^{-1} \underline{f}_{\Delta} \quad \text{y} \quad \underline{j} \underline{u}_{\Delta} = 0 \quad (4.53)$$

Una vez que se obtiene $\underline{u}_{\Delta} \in W_{\Delta}$, la solución $\hat{u} \in \hat{W}$ de la ecuación (4.5) está dada por:

$$\hat{u} = R^{-1} \underline{a} \left\{ \left(I - \frac{A^{-1} A}{\underline{\underline{\Pi}} \underline{\underline{\Pi}} \Delta} \right) \underline{u}_{\Delta} + \frac{A^{-1}}{\underline{\underline{\Pi}} \underline{\underline{\Pi}}} \left(R \hat{f} \right)_{\Pi} \right\} \quad (4.54)$$

4.4.4. La versión DVS del algoritmo FETI-DP

La *información buscada* se elige como $\underline{\lambda} \equiv -\underline{j} \underline{S} \underline{u}$. Con esta elección se obtiene una versión del *algoritmo DVS-FETI-DP* (véase [9]). Dado $\underline{f}_{\Delta} \in \underline{a} W_{\Delta}$, encuentre $\underline{\lambda} \in W_{\Delta}$ tal que:

$$\underline{j} \underline{S} \underline{j} \underline{S}^{-1} \underline{\lambda} = \underline{j} \underline{S} \underline{j} \underline{S}^{-1} \hat{f}_{\Delta} \quad \text{y} \quad \underline{a} \underline{\lambda} = 0 \quad (4.55)$$

Una vez que se obtiene $\underline{\lambda} \in \underline{j} W_{\Delta}$, $\underline{u}_{\Delta} \in \underline{a} W_{\Delta}$ está dado por:

$$\underline{u}_{\Delta} = \underline{a} \underline{S}^{-1} \left(\underline{f}_{\Delta} - \underline{\lambda} \right) \quad (4.56)$$

Entonces la ecuación (4.54) puede aplicarse para obtener la solución $\hat{u} \in \widehat{W}$ de la ecuación (4.5).

4.4.5. El algoritmo primal DVS

La *información buscada* se elige como $\underline{v}_\Delta \equiv \underline{S}^{-1} \underline{jS} u$. Este algoritmo consiste en buscar un vector derivado $\underline{v}_\Delta \in W_\Delta$ (véase [9]), que cumple con:

$$\underline{S}^{-1} \underline{jS} \underline{jv}_\Delta = -\underline{S}^{-1} \underline{jS} \underline{jS}^{-1} \hat{f}_\Delta \quad y \quad \underline{aS} \underline{v}_\Delta = 0 \quad (4.57)$$

Una vez que se obtiene $\underline{v}_\Delta \in \underline{S}^{-1} \underline{jS} W_\Delta$, entonces:

$$\underline{u}_\Delta = \underline{a} \left(\underline{S}^{-1} \hat{f}_\Delta + \underline{v}_\Delta \right) \quad (4.58)$$

La ecuación (4.54) puede aplicarse para obtener la solución $\hat{u} \in \widehat{W}$ de la ecuación (4.5).

4.4.6. El algoritmo dual DVS

La *información buscada* se elige como $\underline{\mu} = \underline{S} u$. Este algoritmo consiste en buscar una función $\underline{\mu} \in W_\Delta$ (véase [9]), que satisface lo siguiente:

$$\underline{aS}^{-1} \underline{a\mu} = \underline{aS}^{-1} \underline{f}_\Delta \quad y \quad \underline{jS}^{-1} \underline{\mu} = 0 \quad (4.59)$$

Una vez que se obtiene $\underline{\mu} \in \underline{aS}^{-1} W_\Delta$, $\underline{u}_\Delta \in \underline{a} W_\Delta$ está dado por:

$$\underline{u}_\Delta = \underline{aS}^{-1} \underline{\mu}_\Delta \quad (4.60)$$

Y así la ecuación (4.54) puede aplicarse para obtener la solución $\hat{u} \in \widehat{W}$ de la ecuación (4.5).

4.5. Procedimientos Numéricos

Los experimentos numéricos que se muestran en esta tesis, se llevan a cabo con cada uno de los *algoritmos-DVS preconditionados* con restricciones (véase [6],[7],[8] y [9]), los cuales se han mencionado en la sección anterior y en resumen son:

$$\underline{aS}^{-1} \underline{aS} \underline{u}_\Delta = \underline{aS}^{-1} \underline{f}_\Delta \quad y \quad \underline{j} \underline{u}_\Delta = 0 \quad \text{DVS - BDDC} \quad (4.61)$$

$$\underline{jS} \underline{jS}^{-1} \underline{\lambda} = \underline{jS} \underline{jS}^{-1} \hat{f}_\Delta \quad y \quad \underline{a} \underline{\lambda} = 0 \quad \text{DVS - FETI - DP} \quad (4.62)$$

$$\underline{\underline{S}}^{-1} \underline{\underline{jS}} \underline{\underline{jv}}_{\Delta} = -\underline{\underline{S}}^{-1} \underline{\underline{jS}} \underline{\underline{jS}}^{-1} \underline{\underline{f}}_{\Delta} \quad y \quad \underline{\underline{aS}} \underline{\underline{v}}_{\Delta} = 0 \quad \text{DVS - PRIMAL} \quad (4.63)$$

y

$$\underline{\underline{SaS}}^{-1} \underline{\underline{a\mu}} = \underline{\underline{SaS}}^{-1} \underline{\underline{f}}_{\Delta} \quad y \quad \underline{\underline{jS}}^{-1} \underline{\underline{\mu}} = 0 \quad \text{DVS - DUAL} \quad (4.64)$$

La uniformidad excepcional de las fórmulas dadas por las ecuaciones (4.61) a (4.64) ofrece ventajas en el desarrollo del código, especialmente cuando los códigos son construidos usando programación orientada a objetos. Dichas ventajas incluyen:

- I La construcción de códigos robustos. Esta es una ventaja de los *algoritmos DVS*, ya que la definición de dichos algoritmos depende exclusivamente del sistema de ecuaciones discretizado (al cual nos referimos como el *problema original*) que se obtiene mediante la discretización de las ecuaciones diferenciales parciales consideradas, pero esto es independiente del problema que lo motiva. De esta forma, por ejemplo, esencialmente el mismo código es usado para tratar problemas bidimensionales y tridimensionales; en realidad, sólo la parte que define la geometría debe modificarse, y esto es una pequeña parte del código.
- II Los códigos pueden usar distintos resolvidores locales, que pueden ser resolvidores directos o iterativos.
- III Se requieren de modificaciones mínimas para transformar códigos secuenciales en paralelos.
- IV Dichas fórmulas permiten a su vez desarrollar códigos en donde “la solución del *problema global* se obtiene mediante la resolución de *problemas locales* exclusivamente”.

Esta última propiedad debe destacarse, porque hace de los algoritmos DVS una herramienta muy adecuada para usarla en la construcción de software altamente paralelizable, que es necesario para la programación eficiente de la mayoría de computadoras en paralelo de alto rendimiento disponibles hoy en día. Por lo tanto, los procedimientos para construir códigos mencionados en la propiedad IV, son esbozados y analizados a continuación.

Todos los algoritmos DVS de las ecuaciones (4.61) a (4.64) son iterativos y pueden implementarse recurriendo al Método de Gradiente Conjugado (CGM, por sus siglas en inglés. véase sección B.2), en el caso en que la matriz es simétrica y definida, o puede utilizarse algún otro procedimiento iterativo como el GMRES (sección (B.3)) en el caso en que la matriz no cumpla las definiciones anteriores. En cada paso de la iteración se tiene que calcular la acción en un *vector derivado* de una de las siguientes matrices: $\underline{\underline{aS}}^{-1} \underline{\underline{aS}}$, $\underline{\underline{jS}} \underline{\underline{jS}}^{-1}$, $\underline{\underline{S}}^{-1} \underline{\underline{jS}} \underline{\underline{j}}$ o $\underline{\underline{SaS}}^{-1} \underline{\underline{a}}$, dependiendo del algoritmo DVS que se aplique.

Estas matrices en cuestión son permutaciones diferentes de las matrices $\underline{\underline{S}}$, $\underline{\underline{S}}^{-1}$, $\underline{\underline{a}}$ y $\underline{\underline{j}}$. Por lo tanto, para implementar cualquiera de los *algoritmos DVS* preconditionados,

sólo se necesita desarrollar códigos capaces de calcular la acción de una de las matrices $\underline{\underline{S}}, \underline{\underline{S}}^{-1}, \underline{\underline{a}}$ o $\underline{\underline{j}}$ en un vector arbitrario de W , *el espacio vectorial derivado*. Por consiguiente, a continuación se explicará cómo calcular la aplicación de cada una de las matrices $\underline{\underline{S}}$ y $\underline{\underline{S}}^{-1}$ respectivamente. En cuanto a $\underline{\underline{a}}$ y $\underline{\underline{j}}$, su aplicación requiere del intercambio de información entre los nodos derivados que son descendientes del mismo *nodo original* y esto consta de una sencilla operación la cual requiere de un intercambio de información mínimo.

4.5.1. La aplicación de $\underline{\underline{S}}$

Retomamos la definición de la matriz $\underline{\underline{S}}$ de la ecuación (4.46), en dónde $\underline{\underline{S}} \equiv \underline{\underline{A}}_{\Delta\Delta} - \underline{\underline{A}}_{\Delta\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{A}}_{\Pi\Delta}$. Para evaluar la acción de $\underline{\underline{S}}$ en cualquier *vector derivado*, necesitamos evaluar correc-

tamente la acción de las siguientes matrices: $\underline{\underline{A}}_{\Pi\Delta}, \underline{\underline{A}}_{\Pi\Pi}^{-1}, \underline{\underline{A}}_{\Delta\Pi}$ y $\underline{\underline{A}}_{\Delta\Delta}$. No se requiere nada especial, excepto para $\underline{\underline{A}}_{\Pi\Pi}^{-1}$, lo cual se explica a continuación.

Se tiene que:

$$\underline{\underline{A}}_{\Pi\Pi} \equiv \begin{pmatrix} \underline{\underline{A}}_{\Pi\Pi} & \underline{\underline{A}}_{\Pi\pi} \\ \underline{\underline{A}}_{\pi\Pi} & \underline{\underline{A}}_{\pi\pi} \end{pmatrix} = \begin{pmatrix} \underline{\underline{A}}_{\Pi\Pi}^t & \underline{\underline{A}}_{\Pi\pi}^t \underline{\underline{a}}^r \\ \underline{\underline{a}}^r \underline{\underline{A}}_{\Pi\Pi}^t & \underline{\underline{a}}^r \underline{\underline{A}}_{\pi\pi}^t \underline{\underline{a}}^r \end{pmatrix} \quad (4.65)$$

Sea $\underline{\underline{w}} \in W$ un *vector derivado* arbitrario, escribimos:

$$\underline{\underline{v}} = \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{w}} \quad (4.66)$$

Entonces, $\underline{\underline{v}}_{\pi} \in W(\pi)$ está caracterizado por:

$$\sigma_{\pi\pi} \left(\underline{\underline{A}}_{\Pi\Pi} \right) \underline{\underline{v}}_{\pi} = \underline{\underline{w}}_{\pi} - \underline{\underline{A}}_{\pi\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{w}}_{\Pi}, \quad \text{sujeto a } \underline{\underline{j}}^{\pi} \underline{\underline{v}}_{\pi} = 0 \quad (4.67)$$

y puede obtenerse iterativamente. Aquí,

$$\sigma_{\pi\pi} \left(\underline{\underline{A}}_{\Pi\Pi} \right) \equiv \left\{ \underline{\underline{A}}_{\pi\pi} - \underline{\underline{A}}_{\pi\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \underline{\underline{w}}_{\Pi} \right\} \quad (4.68)$$

y, con $\underline{\underline{a}}^{\pi}$ como la matriz de proyección en $W_r(\pi)$,

$$\underline{\underline{j}}^{\pi} \equiv \underline{\underline{I}} - \underline{\underline{a}}^{\pi} \quad (4.69)$$

Podemos observar que el cálculo en paralelo de la acción de $\underline{\underline{A}}_{\Pi\Pi}^{-1}$ no requiere mayor costo, ya que:

$$\underline{\underline{A}}_{\Pi\Pi}^{-1} = \sum_{\alpha=1}^E \left(\underline{\underline{A}}_{\Pi\Pi}^{\alpha} \right)^{-1} \quad (4.70)$$

Una vez que se obtiene $\underline{v}_\pi \in W_r(\pi)$, se puede deducir \underline{v}_I aplicando lo siguiente:

$$\underline{v}_I = \underline{\underline{A}}^{-1} \left(\underline{w}_I - \underline{\underline{A}}_{I\pi} \underline{v}_\pi \right) \quad (4.71)$$

Esto completa la evaluación de $\underline{\underline{S}}$.

4.5.2. La aplicación de $\underline{\underline{S}}^{-1}$

Definimos

$$\Sigma \equiv I \cup \Delta \quad (4.72)$$

Una propiedad relevante para el siguiente análisis es:

$$W_r(\Sigma) = W(\Sigma) \quad (4.73)$$

Por consiguiente, la matriz $\underline{\underline{A}}^{-1}$ puede ser escrita como:

$$\underline{\underline{A}}^{-1} = \begin{pmatrix} \left(\underline{\underline{A}}^{-1} \right)_{\Pi\Pi} & \left(\underline{\underline{A}}^{-1} \right)_{\Pi\Delta} \\ \left(\underline{\underline{A}}^{-1} \right)_{\Delta\Pi} & \left(\underline{\underline{A}}^{-1} \right)_{\Delta\Delta} \end{pmatrix} = \begin{pmatrix} \left(\underline{\underline{A}}^{-1} \right)_{\Sigma\Sigma} & \left(\underline{\underline{A}}^{-1} \right)_{\Sigma\pi} \\ \left(\underline{\underline{A}}^{-1} \right)_{\pi\Sigma} & \left(\underline{\underline{A}}^{-1} \right)_{\pi\pi} \end{pmatrix} \quad (4.74)$$

Entonces, $\underline{\underline{S}} : W_\Delta \rightarrow W_\Delta$ satisface que

$$\underline{\underline{S}}^{-1} = \left(\underline{\underline{A}}^{-1} \right)_{\Delta\Delta} \quad (4.75)$$

Para cualquier $\underline{w} \in W$, podemos escribir:

$$\underline{v} \equiv \underline{\underline{A}}^{-1} \underline{w} \quad (4.76)$$

Entonces \underline{v}_π cumple lo siguiente:

$$\sigma_{\pi\pi} \left(\underline{\underline{A}} \right) \underline{v}_\pi = \underline{w}_\pi - \underline{\underline{A}}_{\pi\Sigma} \left(\underline{\underline{A}}^t_{\Sigma\Sigma} \right)^{-1} \underline{w}_\Sigma, \quad \text{sujeito a } \underline{j}^r \underline{v}_\pi = 0 \quad (4.77)$$

Aquí $\underline{j}^r \equiv \underline{I} - \underline{a}^r$, en donde la matriz \underline{a}^r es el operador de proyección en W_r , mientras que

$$\sigma_{\pi\pi} \left(\underline{\underline{A}} \right) \equiv \underline{\underline{A}}_{\pi\pi} - \underline{\underline{A}}_{\pi\Sigma} \left(\underline{\underline{A}}^t_{\Sigma\Sigma} \right)^{-1} \underline{\underline{A}}_{\Sigma\pi} \quad (4.78)$$

Además observamos que:

$$\left(\underline{\underline{A}}^t_{\Sigma\Sigma} \right)^{-1} = \sum_{\alpha=1}^E \left(\underline{\underline{A}}^\alpha_{\Sigma\Sigma} \right)^{-1} \quad (4.79)$$

La ecuación (4.77) se resuelve iterativamente. Una vez que \underline{v}_π se obtiene, aplicamos:

$$\underline{v}_\Sigma = \left(\underline{\underline{A}}^t_{\Sigma\Sigma} \right)^{-1} \left(\underline{w}_\Sigma - \underline{\underline{A}}_{\Sigma\pi} \underline{v}_\pi \right) \quad (4.80)$$

Este procedimiento permite obtener $\underline{\underline{A}}^{-1}\underline{w}$ en general; no obstante, solamente necesitamos $\left(\underline{\underline{A}}^{-1}\right)_{\Delta\Delta}\underline{w}$. Observemos lo siguiente:

$$\left(\underline{\underline{A}}^{-1}\right)_{\Delta\Delta}\underline{w} = \left(\underline{\underline{A}}^{-1}\underline{w}_{\Delta}\right)_{\Delta} \quad (4.81)$$

El vector $\underline{\underline{A}}^{-1}\underline{w}_{\Delta}$ puede obtenerse por el proceso general presentado anteriormente. En consecuencia, tomando $\underline{w} \equiv \underline{w}_{\Delta} \in W_{\Delta} \subset W$ y :

$$\underline{v} \equiv \underline{\underline{A}}^{-1}\underline{w}_{\Delta} \quad (4.82)$$

Por lo tanto, se tiene que:

$$\underline{v}_I + \underline{v}_{\Delta} = \underline{v}_{\Sigma} = -\left(\underline{\underline{A}}^t_{\Sigma\Sigma}\right)^{-1} \underline{\underline{A}}_{\Sigma\pi} \underline{v}_{\pi} = -\left(\underline{\underline{A}}^t_{\Sigma\Sigma}\right)^{-1} \underline{\underline{A}}^t_{\Sigma\pi} \underline{a}^r \underline{v}_{\pi} \quad (4.83)$$

Usando la ecuación (4.79) estas operaciones pueden ser completamente paralelizadas.

4.5.3. La aplicación de \underline{a} y \underline{j}

Usaremos la notación

$$\underline{a} = \left(a_{(i,\alpha)(j,\beta)}\right) \quad (4.84)$$

Además sabemos que $W^{\alpha} \subset W$, $\alpha = 1, \dots, E$ es el conjunto de los vectores derivados, los cuales se anulan en cada nodo derivado que no está asociado a Ω_{α} :

$$a_{(i,\alpha)(j,\beta)} = \frac{1}{m(i)} \delta_{ij}, \quad \forall \alpha \in Z(i) \text{ y } \beta \in Z(j) \quad (4.85)$$

Mientras que el operador de salto cumple con:

$$\underline{j} = \underline{I} - \underline{a} \quad (4.86)$$

Por lo tanto:

$$\underline{j}\underline{w} = \underline{w} - \underline{a}\underline{w}, \quad \text{para cada } \underline{w} \in W \quad (4.87)$$

En cuanto el lado derecho de las ecuaciones (4.53), (4.55), (4.57) y (4.59) todas se pueden obtener aplicando a \underline{f}_{Δ} alguno de los operadores que ya hemos discutido. Retomando la ecuación (4.47), tenemos que:

$$\underline{f}_{\Delta} = \left(R\hat{f}\right)_{\Delta} - \underline{\underline{A}}_{\Delta\Pi} \underline{\underline{A}}_{\Pi\Pi}^{-1} \left(R\hat{f}\right)_{\Pi} \quad (4.88)$$

El cálculo de $R\hat{f}$ no presenta costo alguno y la evaluación de las acciones $\underline{\underline{A}}_{\Pi\Pi}^{-1}$ y $\underline{\underline{A}}_{\Delta\Pi}$ fueron ya analizadas.

Al definir un sistema lineal virtual del problema, éste podrá ser implementado mediante el método de Gradiente Conjugado o alguna versión de GMRES, esto depende del tipo de matriz que sea $\underline{\underline{S}}$. En el caso en que $\underline{\underline{S}}$ es simétrica y definida positiva, entonces $\underline{\underline{S}}^{-1}$ también será simétrica y definida positiva y por tanto será pertinente usar el método de Gradiente Conjugado; para el caso contrario se utilizará el método GMRES o algún otro. Es importante señalar que la elección correcta de \underline{u}^0 puede llegar a ser costosa (computacionalmente hablando) debido a que la elección no adecuada de \underline{u}^0 generalmente provoca más iteraciones para converger en el método que si eligiéramos $\underline{u}^0 = 0$.

Capítulo 5

La Computación de Alto Rendimiento en la Modelación Matemática y Computacional de las Ciencias de la Tierra

5.1. Introducción

El cómputo es una herramienta esencial en la rama de las ciencias y la ingeniería, particularmente en la relacionada al estudio de sistemas Terrestres, debido a que la naturaleza de dichos problemas involucra un número enorme de grados de libertad. Debido a lo anterior, es necesario trabajar con modernos equipos de cómputo capaces de hacer un número de cálculos considerables y además optimizar los procesos computacionales involucrados en el cálculo.

Al plantear un problema en particular y definir su modelo matemático es posible obtener un conjunto de ecuaciones, éstas se obtienen debido a la discretización la cual generará los sistemas de ecuaciones lineales que podrán ser manipulados computacionalmente, es sencillo notar que si estos sistemas son consecuencia de problemas con un número muy grande de grados de libertad, los sistemas de ecuaciones lineales asociados también lo serán. A grandes rasgos, es importante percatarse que existen dos aspectos importantes cuando se requiere tratar con este tipo de problemas, si se requiere un espacio grande de memoria para poder manipular los datos, no importando el tiempo que lleve su solución; o si se requiere un cálculo en tiempo real, dónde el tiempo de ejecución sea vital. Las necesidades computacionales dependerán del caso al que nos enfrentemos.

El procesamiento en paralelo se basa en la idea de dividir un problema en un conjunto de partes resolubles de forma concurrente mientras que un equipo en paralelo es el conjunto de procesadores capaces de trabajar cooperativamente en la resolución de problemas computacionales. La definición incluye un amplio espectro de equipos, tales como supercomputadoras, procesadores masivamente paralelos (MPP), clusters, etc.

La computación de alto desempeño ha dejado de ser una herramienta inaccesible gracias a los avances en diferentes tecnologías, como el poder de procesamiento debido a microprocesadores, las redes de comunicación de datos y el desarrollo de bibliotecas e interfaces para programación. La tecnología ha avanzado al grado de disponer de máquinas paralelas “caseras” emulando clusters con computadoras de bajo costo; además con el desarrollo de la tecnología grid y cloud (véase [25]) se pueden compartir recursos informáticos (locales o remotos) como si fueran parte de un único equipo, además brindan la capacidad de gestionar y distribuir la potencia de cálculo.

5.1.1. La programación en Paralelo y su Arquitectura

A pesar de los avances en términos de velocidad de procesamiento que se han obtenido en los procesadores, es evidente que hay un límite físico dado por la velocidad de la luz: aunque dentro de un componente electrónico se puedan realizar operaciones muy rápidamente, si se necesita comunicar con otro componente, el tiempo de comunicación de las señales está limitado y se pierde la ganancia de velocidad.

Por lo tanto, la única forma de tratar algunos problemas es la utilización de procesamiento paralelo (véase [15] [16], [17] y [18]). Si varias operaciones pueden ser ejecutadas simultáneamente, el tiempo total de procesamiento se verá reducido, aún cuando cada una de las operaciones no se lleve a cabo más rápidamente. Tradicionalmente, el término supercomputadora estuvo relacionado de una manera u otra con el procesamiento en paralelo. Al mismo tiempo la especificación de algoritmos ejecutables sobre una arquitectura paralela, ha estado asociada por la programación concurrente (véase [19], [20] y [21]).

Se han propuesto múltiples formas de organizar las arquitecturas de procesamiento paralelo, e incluso se considera un problema definir qué significa de manera precisa una “arquitectura paralela” (véase [22] y [23]).

El problema de definir qué es, y luego dar una taxonomía de las arquitecturas de procesamiento paralelo, se encuentra en la gran cantidad de características que se tienen en una computadora paralela y que no todas son de fácil descripción, comparación y clasificación.

La taxonomía clásica de las arquitecturas de procesamiento establecida inicialmente en (véase [24]) se centra en la forma en que se ejecutan las instrucciones sobre los datos:

- **SISD: Single Instruction. Single Data.**
Los sistemas de este tipo se caracterizan por tener un único flujo de instrucciones sobre un único flujo de datos, es decir, se ejecuta una instrucción detrás de otra. Este es el concepto de arquitectura serie de Von Neumann donde, en cualquier momento, sólo se ejecuta una única instrucción, un ejemplo de estos sistemas son las máquinas secuenciales convencionales.
- **MISD: Multiple Instructions. Single Data.**
Sistemas con múltiples instrucciones que operan sobre un único flujo de datos. Este tipo de sistemas no ha tenido implementación hasta hace poco tiempo. Los sistemas MISD se contemplan de dos maneras distintas:
 1. Varias instrucciones operando simultáneamente sobre un único dato.
 2. Varias instrucciones operando sobre un dato que se va convirtiendo en un resultado que será la entrada para la siguiente etapa. Se trabaja de forma segmentada, todas las unidades de proceso pueden trabajar de forma concurrente.
- **SIMD: Single Instruction. Multiple Data.**
Estos sistemas tienen un único flujo de instrucciones que operan sobre múltiples flujos de datos. Existen ejemplos de estos sistemas en las máquinas vectoriales con hardware escalar y vectorial. El procesamiento es síncrono, la ejecución de

las instrucciones sigue siendo secuencial como en el caso anterior, todos los elementos realizan una misma instrucción pero sobre una gran cantidad de datos. Por este motivo existirá concurrencia de operación, es decir, esta clasificación es el origen de la máquina paralela.

El funcionamiento de este tipo de sistemas consiste en que la unidad de control manda una misma instrucción a todas las unidades de proceso (ALUs). Las unidades de proceso operan sobre datos diferentes, pero con la misma instrucción recibida. Existen dos alternativas distintas que aparecen después de realizarse esta clasificación:

1. Arquitectura Vectorial con segmentación, una CPU única particionada en unidades funcionales independientes trabajando sobre flujos de datos concretos.
 2. Arquitectura Matricial (matriz de procesadores), varias ALUs idénticas a las que el procesador da instrucciones, asigna una única instrucción pero trabajando sobre diferentes partes del programa.
- MIMD: Multiple Instruction. Multiple Data.
Sistemas con un flujo de múltiples instrucciones que operan sobre múltiples datos. Estos sistemas empezaron a utilizarse antes de la década de los ochentas. Son sistemas con memoria compartida que permiten ejecutar varios procesos simultáneamente (sistema multiprocesador). Cuando las unidades de proceso reciben datos de una memoria no compartida estos sistemas reciben el nombre de MULTIPLE SISD (MSISD). En arquitecturas con varias unidades de control (MISD Y MIMD), existe otro nivel superior con una unidad de control que se encarga de controlar todas las unidades de control del sistema (ejemplo de estos sistemas son las máquinas paralelas actuales).

5.2. Consideraciones Computacionales para la Implementación

Uno de los grandes retos del área de cómputo científico es poder analizar a priori una serie de consideraciones dictadas por factores externos al problema de interés que repercuten directamente en la forma de solucionar el problema, estas consideraciones influirán de manera decisiva en la implementación computacional de la solución numérica. Algunas de estas consideraciones son:

- Número de Procesadores Disponibles
- Tamaño y Tipo de Partición del Dominio

- Tiempo de Ejecución Predeterminado (El cual es el tiempo esperado para encontrar la solución, con el fin de tener una cota para hallarla)

Siendo común que ellas interactúan entre sí. Para tratar de tener una idea clara de cómo afectan a la aplicación computacional estos factores, a continuación detallamos algunas consideraciones acerca de cada uno de ellos.

5.2.1. Número de Procesadores Disponibles

El número de procesadores disponibles es una barrera en constante evolución pero las principales limitantes son económicas y tecnológicas, es común ahora contar con miles de procesadores interactuando de manera conjunta en grids, pero todo ese poder de cómputo creciente es aún insuficiente para las necesidades del cómputo científico. Por otro lado, la gran mayoría de los proyectos científicos y tecnológicos no cuentan con recursos computacionales grandes, ya que este es un recurso costoso y por ende muy limitado y en la mayoría de los casos no es el adecuado a las necesidades propias del problema. Así, para poder atacar el problema de forma eficiente es necesario adaptarse al equipo de cómputo disponible y tratar de conocer de antemano los factores que mermaran el rendimiento de la implementación computacional para buscar alternativas que mejoren la eficiencia de la implementación.

5.2.2. Tamaño y Tipo de Partición del Dominio

Normalmente cuando se plantea un problema de sistemas continuos la determinación del dominio y tipo de malla a usar en la solución del mismo está sujeta a restricciones fenomenológicas, por ello la malla deberá de adaptarse de la mejor forma posible para tratar de capturar los rasgos esenciales del fenómeno estudiado. Pero en el momento de la implementación computacional, es común hacer adecuaciones a esta, para que pueda ser soportada por el equipo de cómputo y su ejecución sea en un tiempo razonable. Esto puede ser un problema, sobre todo al implementar la solución con el uso del cómputo en paralelo, ya que comúnmente una elección de una malla no homogénea ocasionará problemas de mal balanceo de carga de trabajo entre los procesadores utilizados en la implementación computacional.

5.2.3. Tiempo de Ejecución Predeterminado

Otro factor determinante es el tiempo de solución esperado de la implementación computacional del problema, esto es crítico en problemas de control en tiempo real, comunes en la ciencia e ingeniería actual. Por ello en estos casos es permisible el aumento en el número y capacidades del equipo de cómputo necesario en la implementación con el fin de lograr un tiempo de ejecución por debajo del máximo posible dictado por las especificaciones propias del problema.

Todos estos factores influirán en la versión final implementada en la solución computacional de un problema particular, debiendo ser todas ellas sopesadas antes de tomar una decisión en cuanto a las especificaciones que el programa de cómputo deberá satisfacer.

5.3. El procesamiento en Paralelo y los Clusters

La programación en paralelo y el diseño de programas paralelos eficientes han sido bastante abordados en la computación de alto rendimiento. La simulación de los problemas científicos es un área importante en las ciencias naturales y la ingeniería, donde ha adquirido una creciente importancia. Es necesario mayor poder de cómputo y espacio de memoria para llevar a cabo simulaciones más precisas o simulaciones de problemas de gran tamaño. En las últimas décadas, la investigación en el cómputo de alto rendimiento incluye novedades en hardware paralelo y tecnologías de software, y notablemente un progreso constante en el cómputo en paralelo de alto rendimiento. Los ejemplos más populares son simulaciones de pronóstico del tiempo sobre la base de modelos matemáticos complejos que involucran ecuaciones diferenciales parciales o simulaciones de choques de la industria automotriz basado en métodos de elementos finitos.

En la programación en paralelo primero debe considerarse el diseño de un algoritmo paralelo o programa para un problema dado. El diseño comienza con la descomposición de los cálculos de una aplicación o programa en varias partes, llamadas tareas, que pueden calcularse en paralelo en los núcleos o procesadores del hardware paralelo.

Una clasificación a grandes rasgos de la organización de la memoria distingue entre las máquinas de memoria compartida y las de memoria distribuida. A menudo el término *hilo* está relacionado con la memoria compartida y el término *proceso* está relacionado con memoria distribuida. Para las máquinas de memoria compartida, una memoria compartida global almacena los datos de una aplicación y se puede acceder a todos los procesadores o núcleos de los sistemas de hardware. El intercambio de información entre los hilos se hace por medio de variables compartidas en las cuales un hilo escribe y otro hilo lee. Por medio de la sincronización entre hilos se obtiene un correcto comportamiento del programa en su totalidad, es por esto que el acceso a los datos compartidos es coordinado, es decir, un hilo puede leer un dato hasta que la operación de escritura realizada por otro hilo finalice. Dependiendo del lenguaje de programación o el ambiente de software utilizado, la sincronización se realiza por medio del sistema de ejecución o por el programador.

El procesamiento en paralelo refiere al concepto de aumentar la velocidad de ejecución de un programa a través de dividir el programa en múltiples fragmentos que pueden ser ejecutados simultáneamente, cada uno sobre un procesador. Con n proce-

sadores es posible resolver el problema n veces más rápido que haciendo uso de uno sólo (salvo por el retraso que supone el reparto de trabajo inicial y la recolección de datos final).

Las ventajas que tiene el cómputo en paralelo relacionado con el uso de cluster es que cada máquina en un cluster puede ser un sistema completo, utilizable por un amplio rango de otras aplicaciones de otras computadoras. Esta ventaja, sugiere que el cluster puede utilizar todos los ciclos desaprovechados de las estaciones de trabajo inactivas.

La actual explosión de los sistemas conectados en red significa que la mayor parte del hardware para construir un cluster esta siendo vendida en gran volumen, y relativamente a bajo costo. Más allá de ahorrar dinero viene del hecho que se necesita una sola tarjeta de vídeo, un monitor, un teclado, etc. para cada cluster.

Las desventajas del uso de clusters es que generalmente el hardware de red no está diseñado para procesamiento paralelo. El rendimiento del hardware de red es lo suficientemente pobre como en un cluster aislado. Si la red no está aislada de otro tráfico, como es usualmente el caso cuando se usan máquinas que de casualidad están en red; más que un sistema diseñado como un cluster, el rendimiento puede ser sustancialmente peor.

Entre los cluster existentes podemos destacar los utilizados para el procesamiento de datos relacionados con las Ciencias de la Tierra:

Cluster Beowulf o de alto rendimiento: Es un cluster de componentes commodity de cómputo dedicados a un problema paralelo. El primer Cluster de este tipo fue desarrollado por Thomas Sterling, de la división de Ciencias de la Tierra de la NASA en JPL California, esta solución popular ha sido ampliamente aceptada en varios ambientes de producción, principalmente laboratorios de investigación y sitios académicos.

5.3.1. Patrones de la Programación en Paralelo

Los programas en paralelo consisten en una colección de tareas que son ejecutadas por procesos o subprocesos en múltiples procesadores. Para estructurar un programa en paralelo, diversas formas de organizar dichas tareas pueden utilizarse y ser capturadas por un patrón específico de programación. Estos patrones proporcionan estructuras específicas de coordinación para procesos o hilos, las cuales han resultado ser eficaces para una amplia gama de aplicaciones.

Creación de Procesos o Hilos

La creación de los procesos o hilos puede llevarse a cabo estáticamente o dinámicamente. En el caso estático, se genera un número fijo de procesos o hilos al ejecutarse el programa. Estos procesos o hilos existen durante la ejecución del programa en paralelo y se finalizan cuando la ejecución ha terminado. Un método alternativo es permitir

dinámicamente la creación y destrucción de procesos o hilos en puntos arbitrarios durante la ejecución del programa. Una vez que el programa inicia, un sólo proceso o hilo es activado y ejecuta el programa principal.

El proceso Maestro-Esclavo

En el modelo maestro-esclavo, hay un sólo maestro que controla la ejecución del programa. El hilo maestro regularmente ejecuta la función principal del programa en paralelo y crea hilos esclavos en los puntos apropiados del programa para realizar los cálculos pertinentes. Dependiendo del sistema en específico, los hilos esclavos pueden ser creados estáticamente o dinámicamente. La asignación del trabajo a los hilos esclavos es usualmente hecho por el hilo maestro, sin embargo los hilos esclavos pueden a su vez generar nuevo trabajo o tareas para calcular. En este caso, el hilo maestro será responsable únicamente de la coordinación, por ejemplo, realizar inicializaciones, tiempos y operaciones de salida.

5.3.2. Clusters

El desarrollo de sistemas operativos y compiladores del dominio público (Linux y software GNU), estándares para el pase de mensajes (MPI), conexión universal a periféricos (PCI), etc., han hecho posible tomar ventaja de los económicos recursos computacionales de producción masiva (CPU, discos, redes). La principal desventaja que presenta a los proveedores de multicomputadoras es que deben satisfacer una amplia gama de usuarios, es decir, deben ser generales. Esto aumenta los costos de diseños y producción de equipos, así como los costos de desarrollo de software que va con ellos: sistema operativo, compiladores y aplicaciones. Los usuarios con una necesidad de procesamiento y un mecanismo de pase de mensajes óptimo, son los que actualmente están impulsando el uso de clusters, principalmente de computadoras personales (PC), cuya arquitectura se muestra a continuación:

- *Cluster homogéneo*: si todos los procesadores y/o nodos participantes en el equipo paralelo son iguales en capacidad de cómputo (en la cual es permitido variar la cantidad de memoria o disco duro en cada procesador).
- *Cluster heterogéneo* : es aquel en que al menos uno de los procesadores y/o nodos participantes en el equipo paralelo son de distinta capacidad de cómputo.

5.3.3. Tipos de Cluster

Básicamente existen tres tipos de clusters, cada uno de ellos ofrece ventajas y desventajas, el tipo más adecuado para el cómputo científico es del de alto rendimiento, pero existen aplicaciones científicas que pueden usar más de un tipo al mismo tiempo.

- Alta-disponibilidad (Fail-over o High-Availability): este tipo de cluster está diseñado para mantener uno o varios servicios disponibles incluso a costa de rendimiento, ya que su función principal es que el servicio jamás tenga interrupciones como por ejemplo un servicio de bases de datos.
- Alto-rendimiento (HPC o High Performance Computing): este tipo de cluster está diseñado para obtener el máximo rendimiento de la aplicación utilizada incluso a costa de la disponibilidad del sistema, es decir el cluster puede sufrir caídas, este tipo de configuración está orientada a procesos que requieran mucha capacidad de cálculo.
- Balanceo de Carga (Load-balancing): este tipo de cluster está diseñado para balancear la carga de trabajo entre varios servidores, lo que permite tener, por ejemplo, un servicio de cálculo intensivo multiusuarios que detecte tiempos muertos del proceso de un usuario para ejecutar en dichos tiempos procesos de otros usuarios.

5.3.4. Consideraciones

Actualmente, en muchos centros de cómputo es una práctica común usar directivas de compilación en equipos paralelos sobre programas escritos de forma secuencial, con la perspectiva que sean puestos por el compilador como programas paralelos. Esto en la gran mayoría de los casos genera códigos poco eficientes, pese a que corren en equipos paralelos y pueden usar toda la memoria compartida de dichos equipos, el algoritmo ejecutado continua siendo secuencial en la gran mayoría del código. Si la arquitectura paralela donde se implemente el programa es UMA de acceso simétrico, los datos serán introducidos a una velocidad de memoria constante.

En caso contrario, al acceder a un conjunto de datos, es común que una parte de estos sean locales a un procesador (con un acceso del orden de nano segundos), pero el resto de los datos deberán de ser introducidos mediante red (con acceso del orden de mili segundos), implicando un alto costo en tiempo de procesamiento. Por ello, si usamos métodos de descomposición de dominio es posible hacer que el sistema algebraico asociado pueda distribuirse en la memoria local de múltiples computadoras y que para encontrar la solución al problema se requiera poca comunicación entre los procesadores.

Por lo anterior, si se cuenta con computadoras con memoria compartida o que tengan interconexión por bus, salvo en casos particulares no será posible explotar éstas características eficientemente. Pero en la medida en que se adecuen los programas para usar bibliotecas y compiladores acordes a las características del equipo disponible (algunos de ellos sólo existen de manera comercial) la eficiencia aumentará de manera

importante.

5.3.5. Lenguaje C++ y MPI

El lenguaje de programación C, desde su creación tenía por lo menos una característica auto-definida: modularidad. El lenguaje C fue diseñado para ser modular y esta modularidad se llevo a cabo a través del uso de funciones. Casi todo en C es una función. El lenguaje C se compone básicamente de dos componentes, un núcleo de especificación de lenguaje el cual tiene tipos de datos básicos y construcciones y una colección de bibliotecas, cada una de las cuales tiene muchas funciones pre-definidas. C++ se construyo con esta filosofía e introdujo la “clase” como un segundo bloque fundamental de construcción del lenguaje. En C++, las funciones y las clases se entrelazan para crear el programa deseado.

Comencemos por definir que es lo que se entiende por una función y una clase. Las funciones y las clases se pueden distinguir por sus premisas fundamentales. La premisa principal de una función gira alrededor de lo que la función hace; mientras que la premisa fundamental de una clase gira en torno a los datos que contiene la clase. Las funciones están diseñadas para ser abstracciones de algoritmos; las clases son una abstracción de datos y en las operaciones de esos datos.

La premisa básica detrás de MPI es que los múltiples procesos paralelos funcionen simultáneamente hacia un objetivo común con “mensajes” como medio de comunicación con los demás. Múltiples procesos MPI pueden ejecutarse en diferentes procesadores, y estos procesos comunicarse a través de la infraestructura proporcionada por MPI. Como usuarios, no necesitamos conocer la aplicación de esta infraestructura, sólo tenemos que saber cómo sacar provecho de él. Con este fin, casi todo en MPI puede resumirse en la idea única de “Mensaje enviado - Mensaje recibido”. La interfaz de programación en MPI es una colección de funciones. MPI es una biblioteca de funciones diseñadas para manejar todos los detalles esenciales de paso de mensajes sobre la arquitectura en el que se desea ejecutar.

Los ejemplos que se resolvieron y que se muestran en esta Tesis, fueron resueltos por medio de un código paralelizable escrito en C++ (véase [9] y [40]). Se escogió dicho lenguaje de programación debido a múltiples razones. En primer lugar el lenguaje C++ provee una infraestructura orientada a objetos la cual adapta una descomposición natural del problema en una colección de estructuras de datos y operaciones en dichas estructuras. Además, el uso del lenguaje C++ trasciende a muchas disciplinas más allá de la Ingeniería en donde tradicionalmente el lenguaje prevaeciente ha sido FORTRAN. Otra razón es que C++ es un lenguaje naturalmente compatible con los conceptos algorítmicos básicos como:

- particionar,
- llamada recursiva de una función,

- asignación dinámica de memoria, y
- encapsulación

También se utilizó la librería MPI (Message Passing Interface) la cual es una librería de pase de mensajes. Se utilizó MPI debido a que se adapta a una división natural y fácil del problema, además de que proporciona portabilidad y eficiencia. Esta librería ha recibido gran aceptación en la academia y la industria.

5.3.6. Aceleración y eficiencia

lo más común es medir la eficiencia de un sistema paralelo atendiendo a una de estas dos medidas:

- Tiempo de ejecución (o respuesta) de una aplicación.
- Productividad (*throughput*) o número de aplicaciones que es capaz de procesar por unidad de tiempo.

Además se introducen otras medidas como: *speed-up*, eficiencia de un sistema, utilización, redundancia, etc. véase ([22])

Rapidez y eficiencia del sistema Sea $O(n)$ el número total de operaciones elementales realizadas por un sistema con n elementos de proceso y $T(n)$ el tiempo de ejecución en pasos unitarios de tiempo. En general $T(n) < O(n)$ si los n procesadores realizan más de una operación por unidad de tiempo, donde $n \geq 2$.

Supongamos que $T(1) = O(1)$ en un sistema mono-procesador, lo cual supone que IPC=1 (IPC: Instrucciones Por Ciclo). El *factor de mejora del rendimiento* (*speed-up*, aceleración o rapidez) se define como:

$$S(n) = T(1)/T(n) \tag{5.1}$$

La eficiencia del sistema para un sistema con n procesadores se define como:

$$E(n) = \frac{S(n)}{n} = \frac{T(1)}{n * T(n)} \tag{5.2}$$

La eficiencia es una comparación de grado de speed-up conseguido frente al valor máximo. Dado que $1 \leq S(n) \leq n$, tenemos $1/n \leq E(n) \leq 1$.

La eficiencia más baja $E(n) \rightarrow 0$ corresponde al caso en que todo el programa se ejecuta en un único procesador de forma serial. La eficiencia máxima $E(n) = 1$, se obtiene cuando todos los procesadores están siendo completamente utilizados durante todo el periodo de ejecución.

Escalabilidad Un sistema se dice que es *escalable* para un determinado rango de procesadores $[1, \dots, n]$ si la eficiencia $E(n)$ del sistema se mantiene constante y en todo momento por encima de un factor 0,5.

Normalmente todos los sistemas tienen un determinado número de procesadores a partir del cual la eficiencia empieza a disminuir de forma más o menos abrupta. Un sistema es más escalable que otro si este número de procesadores, a partir del cual la eficiencia disminuye, es menor que el otro.

5.4. Métodos y equipos utilizados

El modelo computacional generado fue implementado en el lenguaje C++ en su forma secuencial y en su forma paralela, además de la interfaz de paso de mensajes (MPI) bajo el esquema Maestro-Esclavo. Cabe mencionar que el paradigma de programación orientada a objetos sacrifica algo de eficiencia computacional por requerir mayor manejo de recursos computacionales al momento de la ejecución. Sin embargo, permite mayor flexibilidad cuando se trata de adaptar los códigos a nuevas especificaciones. Otra ventaja es que disminuye notoriamente el tiempo invertido en el mantenimiento y búsqueda de errores dentro del código.

En el caso de las pruebas realizadas en modo secuencial se utilizó el equipo:

- Notebook Lenovo IdeaPad , Intel Atom N550 1.5GHz, 2GB RAM. en Linux Ubuntu. Compilador C++ GNU.

Para el caso de las pruebas realizadas en modo paralelo se utilizó un Cluster al cual se tuvo acceso y pertenece al grupo de Geofísica Computacional del Instituto de Geofísica de la Universidad Nacional Autónoma de México:

- Cluster homogéneo Olintlali de 108 Cores emulando 216 hilos, Intel Xeon a 2.67 GHz, 9 nodos con 6 Cores y 8 hilos de ejecución con 48 GB RAM interconectados con GIGE 10/100/1000 Gb/s, a cargo del Dr. Ismael Herrera Revilla del Departamento de Recursos Naturales del Instituto de Geofísica de la UNAM.

Capítulo 6

Aplicaciones Numéricas

6.1. Introducción

Los algoritmos de los métodos de descomposición de dominio en el espacio vectorial derivado ((4.61)-(4.64)) fueron puestos a prueba en diversos ejemplos. Estos métodos sólo requieren el cálculo de los valores sobre los nodos duales en la frontera interna, que se realiza mediante el uso de métodos iterativos. Partiendo de la ecuación elíptica:

$$-\nu \nabla^2 \underline{u} + \underline{\beta} \cdot \nabla \underline{u} + c \underline{u} = f \quad (6.1)$$

En los ejemplos tratados en este capítulo, ν , c son constantes, mientras que $\underline{\beta}$ es un vector también constante de dimensión n . El dominio $\Omega \subset \mathbb{R}^n$, fue tomado con $n = 2$ donde Ω es el cuadrado unitario. Se eligieron los valores $\nu \sim [10^{-6}, 10^2]$, $c = 0$, $\underline{\beta} = (1, 3)$ y $\underline{\beta} = (1, 0)$ de la ecuación (6.1) con valores en la frontera que se definen para cada ejemplo.

Las ecuaciones de Advección-Difusión generan matrices no simétricas al ser discretizadas, por tanto es preciso usar el método GMRES (Generalized Minimal Residual Method) para los ejemplos no simétricos.

6.2. Implementación Computacional

La implementación computacional de los métodos de descomposición de dominio en el Espacio Vectorial Derivado (DVS) queda fácilmente estructurado mediante el esquema Maestro-Esclavo, tanto para la implementación del código secuencial como paralelo. En este caso cada subdominio de la descomposición genera sus matrices locales y el método que resuelve el sistema global virtual (ya sea CGM o GMRES) es tal que necesita sólo una porción de la información que generan los subdominios.

Una forma óptima de dividir la carga computacional requerida para resolver un problema de descomposición de dominio sin traslapes se obtiene mediante el esquema Maestro-Esclavo, en el cual cada subdominio es asignado a un nodo esclavo, tanto en su implementación secuencial (en dónde cada nodo esclavo es un objeto), como en su implementación paralela (donde cada nodo esclavo está asignado a un procesador); mientras el nodo maestro controla las tareas que requiere el esquema DVS de forma síncrona, éstas tareas son ejecutadas por los nodos esclavos, la comunicación es sólo entre el nodo maestro y cada uno de los nodos esclavos, todo esto sin comunicación entre los nodos esclavos.

Durante el diseño de la implementación computacional de los métodos de descomposición de dominio en el Espacio Vectorial Derivado, se utilizó una jerarquía de clases que especializó a una clase abstracta llamada `DPMETHOD`, la cual permite implementar uno o más de los métodos de descomposición de dominio desarrollados, además se utiliza el método iterativo de Gradiente Conjugado o el método Residual

Mínimo Generalizado dependiendo de la ecuación diferencial parcial a resolver, es decir, dependiendo si la matriz global virtual fuera simétrica o no simétrica.

6.3. Consideraciones del problema

Es importante resaltar el hecho de que cuando la velocidad del fluido no es grande, los problemas de transporte se resuelven numéricamente en forma satisfactoria casi por cualquier método. Sin embargo, cuando se considera un problema de transporte dominado por advección se presentan considerables dificultades; es por esto que el *Número de Péclet* es un parámetro útil al implementar la solución numérica de las ecuaciones de transporte, además de medir la importancia relativa de la advección. El número de Péclet se aborda de manera global y local. En el caso global nos estamos refiriendo a un análisis del dominio (longitud total), donde está definida la ecuación, para el caso local nos referimos a la forma de cómo se comporta la solución a partir del tamaño de cada subintervalo de la partición del dominio (véase Figura(3.1)).

El número de Péclet global y local se define como:

$$Pe_{global} = \frac{|\beta|}{2\nu}H \quad Pe_{local} = \frac{|\beta|}{2\nu}h \quad (6.2)$$

donde H es el tamaño del dominio, h es el tamaño del subintervalo de la partición del dominio, ν es el término advectivo y β es el campo difusivo.

Para los problemas de Advección-Difusión nos interesa encontrar una malla -lo más gruesa posible- en la cual el problema sea soluble sin obtener un error considerable al usar valores de viscosidad pequeños que comúnmente generan inestabilidad numérica en los métodos de discretización, las cuales siempre se eliminan al refinar de forma adecuada la malla. Además con el fin de probar la eficiencia del esquema DVS, se muestran a continuación los ejemplos a los cuales se les aplicaron los cuatro algoritmos vistos en el capítulo anterior y su comparativa con otros esquemas de resolución reportados anteriormente en la literatura, en estos ejemplos se toman diversos valores del número de Péclet.

6.4. Definición de los problemas y resultados

1. Para el primer ejemplo, se tiene la ecuación:

$$\begin{aligned} \nu \nabla^2 u + \underline{\beta} \cdot \nabla u &= 0 \\ \text{en } (x, y) \in [0, 1] \times [0, 1], \text{ donde} & \\ u(x, y) &= \begin{cases} 0, & (x, y) \in \xi_1 \\ 1, & (x, y) \in \xi_2 \end{cases} \end{aligned} \quad (6.3)$$

y $\underline{\beta} = (1, 3)$ como se muestra a continuación:

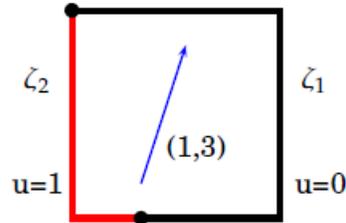


Figura 6.1: Dominio del problema

En este ejemplo, la discretización se realizó con el método de diferencias finitas y se utilizó el método de difusión artificial (Ap.(C)), como se ve en ([37]: Conceição, 2006) los resultados de la ecuación de Advección-Difusión para el método BDDC son utilizados y comparados con los algoritmos DVS ((4.61)-(4.64)).

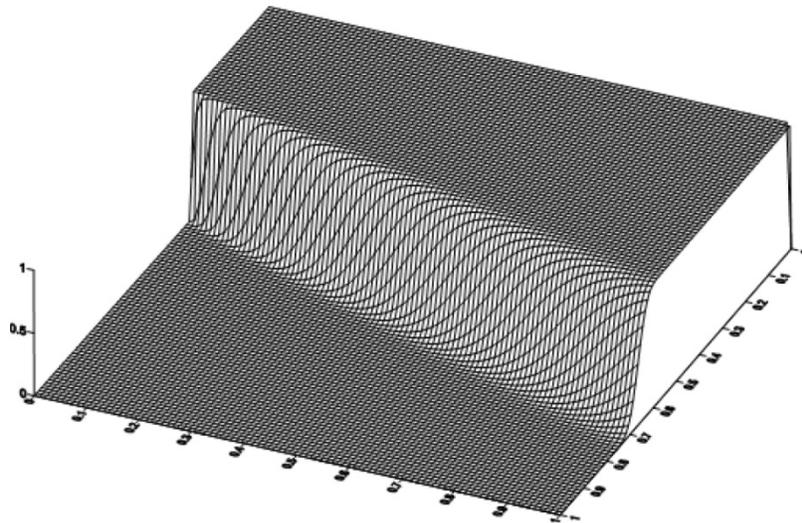


Figura 6.2: Solución del problema para $Pe = 3,16 \times 10^5$

Para su resolución, se utilizó el método GMRES con una tolerancia de 10^{-6} tomando la norma infinita en una malla de 512×512 para distintos valores del número de Péclet como se muestra en la siguiente tabla.

| Partición | Sub-dominios | Grados de Libertad | Primales | DVS-BDDC | DVS-FETI-DP | DVS-PRIMAL | DVS-DUAL | BDDC | Pèclet Global |
|--------------------|--------------|--------------------|----------|----------|-------------|------------|----------|------|------------------------|
| (8 × 8)(64 × 64) | 64 | 262144 | 49 | 12 | 11 | 11 | 11 | 12 | 3,16 × 10 ² |
| (8 × 8)(64 × 64) | 64 | 262144 | 49 | 8 | 8 | 8 | 7 | 9 | 3,16 × 10 ³ |
| (8 × 8)(64 × 64) | 64 | 262144 | 49 | 7 | 7 | 7 | 7 | 9 | 3,16 × 10 ⁴ |
| (8 × 8)(64 × 64) | 64 | 262144 | 49 | 7 | 7 | 7 | 7 | 9 | 3,16 × 10 ⁵ |
| (16 × 16)(32 × 32) | 256 | 262144 | 255 | 19 | 17 | 17 | 18 | 20 | 3,16 × 10 ² |
| (16 × 16)(32 × 32) | 256 | 262144 | 255 | 14 | 14 | 13 | 13 | 17 | 3,16 × 10 ³ |
| (16 × 16)(32 × 32) | 256 | 262144 | 255 | 13 | 13 | 13 | 13 | 15 | 3,16 × 10 ⁴ |
| (16 × 16)(32 × 32) | 256 | 262144 | 255 | 13 | 13 | 13 | 13 | 16 | 3,16 × 10 ⁵ |
| (32 × 32)(16 × 16) | 1024 | 262144 | 961 | 33 | 29 | 29 | 31 | 33 | 3,16 × 10 ² |
| (32 × 32)(16 × 16) | 1024 | 262144 | 961 | 26 | 25 | 25 | 25 | 30 | 3,16 × 10 ³ |
| (32 × 32)(16 × 16) | 1024 | 262144 | 961 | 25 | 25 | 25 | 25 | 28 | 3,16 × 10 ⁴ |
| (32 × 32)(16 × 16) | 1024 | 262144 | 961 | 25 | 25 | 25 | 26 | 29 | 3,16 × 10 ⁵ |
| (64 × 64)(8 × 8) | 4096 | 262144 | 3969 | 53 | 52 | 53 | 59 | 52 | 3,16 × 10 ² |
| (64 × 64)(8 × 8) | 4096 | 262144 | 3969 | 46 | 46 | 46 | 47 | 53 | 3,16 × 10 ³ |
| (64 × 64)(8 × 8) | 4096 | 262144 | 3969 | 45 | 47 | 45 | 47 | 53 | 3,16 × 10 ⁴ |
| (64 × 64)(8 × 8) | 4096 | 262144 | 3969 | 45 | 47 | 45 | 48 | 54 | 3,16 × 10 ⁵ |

Cuadro 6.1: Se muestran los resultados de las iteraciones para los distintos métodos DVS y BDDC. (Conceição, 2006, véase [37])

2. Para el segundo ejemplo se tiene la ecuación:

$$v \nabla^2 u + \underline{\beta} \cdot \nabla u = 0$$

en $(x, y) \in [0, 1] \times [0, 1]$, donde

$$u(x, y) = 1 \begin{cases} y = 1, & 0 \leq x \leq 1 \\ x = 0, & d < y < 1 \\ x = 1, & \sqrt{v} + d < y < 1 \end{cases} \tag{6.4}$$

$$u(x, y) = 0 \begin{cases} y \leq \sqrt{vx} + d, & 0 \leq x \leq 1 \end{cases}$$

y $\underline{\beta} = (1, 0)$ como se muestra a continuación:

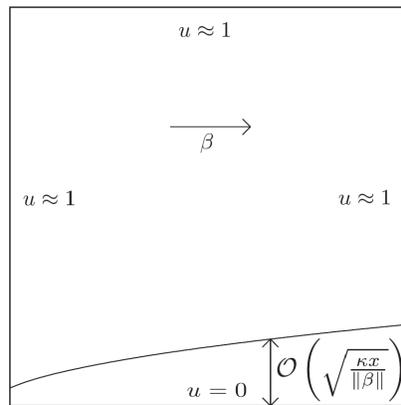


Figura 6.3: Problema de Advección-Difusión en un Dominio cuadrado unitario (Yano,2010)

El problema anterior fue tratado por Yano (2010)(véase [38]) utilizando el método BDDC para un problema con mallas uniformes, utilizando un factor de tolerancia de 10^{-8} en la norma euclidiana. En esta tesis se utilizaron los cuatro algoritmos basados en el Espacio Vectorial Derivado (DVS): DVS-BDDC, DVS-FETI-DP, DVS-PRIMAL y DVS DUAL (véase ((4.61)-(4.64))).

Para la resolución del problema, al igual que en el ejemplo anterior se utilizó el método GMRES, en este caso se consideraron tolerancias de 10^{-6} tomando la norma infinita para distintas mallas y distintos valores del número de Péclet. Basado en lo anterior se comparan el número de iteraciones obtenidas en BDDC (véase [38]) con las obtenidas usando los cuatro algoritmos DVS (Herrera,2011, véase [7]).

| Partición | Sub-dominios | Primales | DVS-BDDC | DVS-FETI-DP | DVS-PRIMAL | DVS-DUAL | BDDC p=1 | Péclet Global |
|------------------------------|--------------|----------|----------|-------------|------------|----------|----------|-----------------------|
| Péclet $\sim 10^{-2}$ | | | | | | | | |
| $(2 \times 2)(8 \times 8)$ | 4 | 1 | 3 | 3 | 3 | 3 | 4 | $1,41 \times 10^{-2}$ |
| $(4 \times 4)(8 \times 8)$ | 16 | 9 | 8 | 8 | 9 | 8 | 9 | $2,82 \times 10^{-2}$ |
| $(8 \times 8)(8 \times 8)$ | 64 | 49 | 11 | 9 | 11 | 10 | 11 | $5,66 \times 10^{-2}$ |
| $(16 \times 16)(8 \times 8)$ | 256 | 225 | 11 | 10 | 12 | 11 | 11 | $1,13 \times 10^{-1}$ |
| $(4 \times 4)(4 \times 4)$ | 16 | 9 | 7 | 6 | 7 | 6 | 7 | $2,83 \times 10^{-2}$ |
| $(4 \times 4)(8 \times 8)$ | 16 | 9 | 8 | 8 | 9 | 8 | 9 | $2,83 \times 10^{-2}$ |
| $(4 \times 4)(16 \times 16)$ | 16 | 9 | 10 | 9 | 10 | 9 | 10 | $2,83 \times 10^{-2}$ |

Cuadro 6.2: Se muestra el número de iteraciones para $Pe \sim 10^{-2}$, usando los cuatro algoritmos DVS (tolerancia de 10^{-6} norma infinito) y comparándolo con el mejor resultado del método BDDC (con una corrección de $p = 1$ y una tolerancia de 10^{-8} norma euclidiana), reportado en (véase [38]).

| Partición | Sub-dominios | Primales | DVS-BDDC | DVS-FETI-DP | DVS-PRIMAL | DVS-DUAL | BDDC p=1 | Péclet Global |
|------------------------------|--------------|----------|----------|-------------|------------|----------|----------|--------------------|
| Péclet $\sim 10^2$ | | | | | | | | |
| $(2 \times 2)(8 \times 8)$ | 4 | 1 | 1 | 2 | 2 | 3 | 4 | $1,41 \times 10^2$ |
| $(4 \times 4)(8 \times 8)$ | 16 | 9 | 5 | 5 | 4 | 7 | 6 | $2,83 \times 10^2$ |
| $(8 \times 8)(8 \times 8)$ | 64 | 49 | 9 | 9 | 8 | 13 | 9 | $5,66 \times 10^2$ |
| $(16 \times 16)(8 \times 8)$ | 256 | 225 | 17 | 17 | 14 | 22 | 15 | $1,13 \times 10^3$ |
| $(4 \times 4)(4 \times 4)$ | 16 | 9 | 6 | 4 | 4 | 8 | 6 | $2,83 \times 10^2$ |
| $(4 \times 4)(8 \times 8)$ | 16 | 9 | 5 | 5 | 4 | 7 | 6 | $2,83 \times 10^2$ |
| $(4 \times 4)(16 \times 16)$ | 16 | 9 | 5 | 5 | 4 | 7 | 6 | $2,83 \times 10^2$ |

Cuadro 6.3: Se muestra el número de iteraciones para $Pe \sim 10^2$, usando los cuatro algoritmos DVS el mejor resultado del método BDDC (con una corrección de $p = 1$), reportado en (véase [38]).

Para el caso en que se tienen números de Péclet pequeños ($Pe \sim 10^{-2}$) el método DVS-FETI-DP mostró un número menor de iteraciones que el método BDDC con una corrección de $p=1$, el resultado con menor número de iteraciones reportado en (véase [38]). El esquema DVS-FETI-DP mostró resolver el sistema con menos iteraciones, lo que prueba la utilidad con que el esquema DVS divide el problema original. Ahora bien, cuando el número de Péclet adquiere valores ($Pe \sim 10^2$), el método con el menor número de iteraciones es el DVS-PRIMAL; reafirmando que el esquema DVS cumple su tarea de dividir el problema de la forma más provechosa para resolver el problema en menos iteraciones y con la menor transmisión de información entre subdominios.

Ahora bien, para mostrar el uso de los algoritmos DVS en problemas que involucran mayor número de grados de libertad, se resolvió el problema 2 de advección-difusión reportado en [38]. En la siguiente serie de resultados se tienen 32,400 grados de libertad, en distintas métricas y se asignó un procesador por cada subdominio generado.

| Partición | Grados de Libertad | Procesadores Esclavos | Procesadores Totales | Tiempo de Ejecución [seg] | Iteraciones GMRES | Péclet Global |
|-----------------------|--------------------|-----------------------|----------------------|---------------------------|-------------------|-----------------------|
| Péclet $\sim 10^{-2}$ | | | | | | |
| (3 × 2)(60 × 90) | 32400 | 6 | 7 | 10 | 9 | $1,80 \times 10^{-2}$ |
| (3 × 3)(60 × 60) | 32400 | 9 | 10 | 7 | 12 | $2,12 \times 10^{-2}$ |
| (3 × 4)(50 × 54) | 32400 | 12 | 13 | 6 | 11 | $2,50 \times 10^{-2}$ |
| (3 × 5)(40 × 54) | 32400 | 15 | 16 | 5 | 15 | $2,92 \times 10^{-2}$ |
| (3 × 6)(36 × 50) | 32400 | 18 | 19 | 4 | 14 | $3,35 \times 10^{-2}$ |
| (5 × 4)(45 × 36) | 32400 | 20 | 21 | 4 | 17 | $3,20 \times 10^{-2}$ |
| (3 × 8)(25 × 54) | 32400 | 24 | 25 | 3 | 12 | $4,27 \times 10^{-2}$ |

Cuadro 6.4: Se muestra el tiempo de ejecución, procesadores e Iteraciones para $Pe \sim 10^{-2}$, se utilizó el algoritmo DVS-PRIMAL en este caso.

| Partición | Grados de Libertad | Procesadores Esclavos | Procesadores Totales | Tiempo de Ejecución [seg] | Iteraciones GMRES | Péclet Global |
|--------------------|--------------------|-----------------------|----------------------|---------------------------|-------------------|--------------------|
| Péclet $\sim 10^2$ | | | | | | |
| (3 × 2)(60 × 90) | 32400 | 6 | 7 | 8 | 6 | $1,80 \times 10^2$ |
| (3 × 3)(60 × 60) | 32400 | 9 | 10 | 5 | 8 | $2,12 \times 10^2$ |
| (3 × 4)(50 × 54) | 32400 | 12 | 13 | 5 | 7 | $2,50 \times 10^2$ |
| (3 × 5)(40 × 54) | 32400 | 15 | 16 | 3 | 8 | $2,92 \times 10^2$ |
| (3 × 6)(36 × 50) | 32400 | 18 | 19 | 3 | 9 | $3,35 \times 10^2$ |
| (5 × 4)(45 × 36) | 32400 | 20 | 21 | 3 | 11 | $3,20 \times 10^2$ |
| (3 × 8)(25 × 54) | 32400 | 24 | 25 | 3 | 11 | $4,27 \times 10^2$ |

Cuadro 6.5: Se muestra el tiempo de ejecución, procesadores e Iteraciones para $Pe \sim 10^2$, se utilizó el algoritmo DVS-PRIMAL en este caso.

| Partición | Grados de Libertad | Procesadores Esclavos | Procesadores Totales | Tiempo de Ejecución [seg] | Iteraciones GMRES | Pèclet Global |
|--------------------|--------------------|-----------------------|----------------------|---------------------------|-------------------|--------------------|
| Pèclet $\sim 10^6$ | | | | | | |
| (3 × 2)(60 × 90) | 32400 | 6 | 7 | 4 | 2 | $1,80 \times 10^6$ |
| (3 × 3)(60 × 60) | 32400 | 9 | 10 | 6 | 11 | $2,12 \times 10^6$ |
| (3 × 4)(50 × 54) | 32400 | 12 | 13 | 3 | 5 | $2,50 \times 10^6$ |
| (3 × 5)(40 × 54) | 32400 | 15 | 16 | 1 | 2 | $2,92 \times 10^6$ |
| (3 × 6)(36 × 50) | 32400 | 18 | 19 | 2 | 2 | $3,35 \times 10^6$ |
| (5 × 4)(45 × 36) | 32400 | 20 | 21 | 2 | 5 | $3,20 \times 10^6$ |
| (3 × 8)(25 × 54) | 32400 | 24 | 25 | 1 | 2 | $4,27 \times 10^6$ |

Cuadro 6.6: Se muestra el tiempo de ejecución, procesadores e Iteraciones para $Pe \sim 10^6$, se utilizó el algoritmo DVS-PRIMAL en este caso.

Un comportamiento a resaltar es el que tienen las iteraciones y el tiempo respecto a la variación del número de Pèclet utilizando las variaciones en la malla que se mostraron en los Cuadros anteriores; dichos comportamientos pueden observarse en las figuras (6.4 y 6.5).

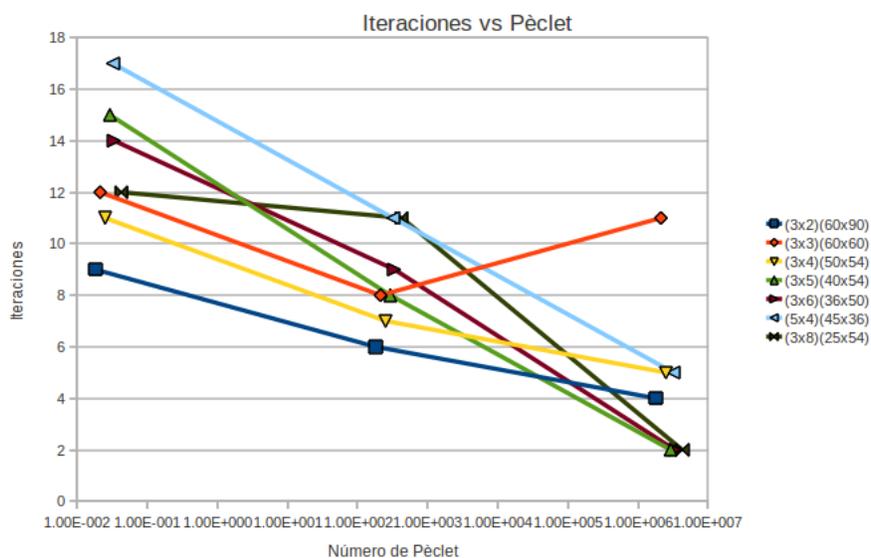


Figura 6.4: Se utilizan los datos mostrados en los Cuadros (6.4, 6.5 y 6.6), se grafican los números de Iteraciones respecto al número de Pèclet para una partición específica

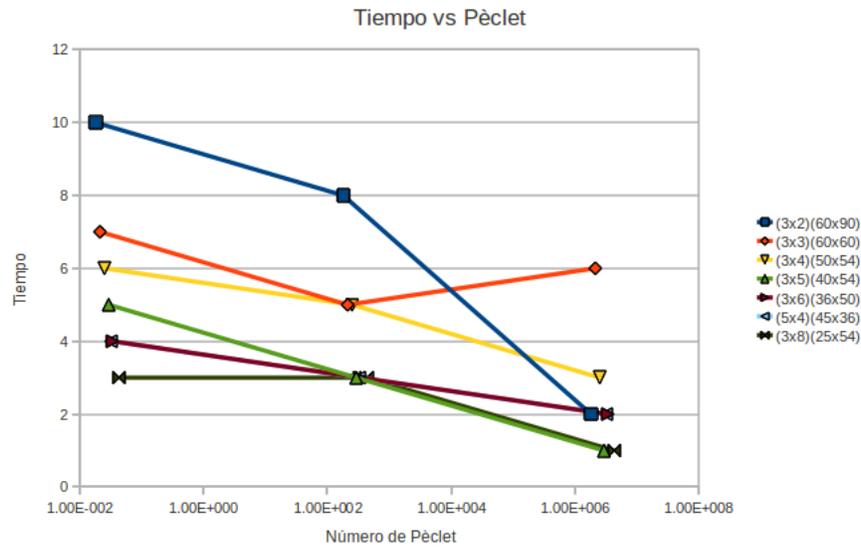


Figura 6.5: Se utilizan los datos mostrados en los Cuadros (6.4,6.5 y 6.6), se grafica el tiempo respecto al número de Péclet para una partición específica

6.4.1. Ejecuciones en Paralelo

Se muestran a continuación las gráficas de tiempo para las ejecuciones en paralelo, en el caso en que se asigna un procesador a cada subdominio para su resolución.

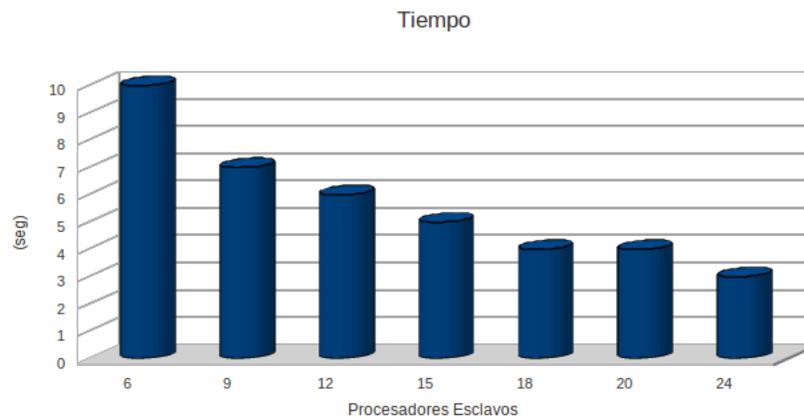


Figura 6.6: Se utilizaron las métricas correspondientes (Cuadro 6.6) para cada número de procesador, el tiempo se estabiliza para 18 y 20 procesadores

Aquí, en todas las métricas usadas se conserva el número de grados de libertad con el fin de comparar resultados de las mismas dimensiones.

| Partición | Grados de Libertad | Procesadores Esclavos | Procesadores Totales | Tiempo de Ejecución [seg] | Iteraciones GMRES |
|--------------------------|--------------------|-----------------------|----------------------|---------------------------|-------------------|
| $Pe=4,27 \times 10^{-2}$ | | | | | |
| (3 × 8)(125 × 270) | 810000 | 1 | 2 | 12887 | 9 |
| (3 × 8)(125 × 270) | 810000 | 2 | 3 | 3638 | 9 |
| (3 × 8)(125 × 270) | 810000 | 3 | 4 | 2723 | 9 |
| (3 × 8)(125 × 270) | 810000 | 4 | 5 | 2112 | 9 |
| (3 × 8)(125 × 270) | 810000 | 5 | 6 | 1925 | 9 |
| (3 × 8)(125 × 270) | 810000 | 6 | 7 | 1649 | 9 |
| (3 × 8)(125 × 270) | 810000 | 7 | 8 | 1595 | 9 |
| (3 × 8)(125 × 270) | 810000 | 8 | 9 | 1365 | 9 |
| (3 × 8)(125 × 270) | 810000 | 9 | 10 | 1384 | 9 |
| (3 × 8)(125 × 270) | 810000 | 10 | 11 | 1398 | 9 |
| (3 × 8)(125 × 270) | 810000 | 11 | 12 | 1385 | 9 |
| (3 × 8)(125 × 270) | 810000 | 12 | 13 | 1460 | 9 |
| (3 × 8)(125 × 270) | 810000 | 13 | 14 | 1674 | 9 |
| (3 × 8)(125 × 270) | 810000 | 14 | 15 | 1423 | 9 |
| (3 × 8)(125 × 270) | 810000 | 15 | 16 | 1358 | 9 |
| (3 × 8)(125 × 270) | 810000 | 16 | 17 | 1654 | 9 |
| (3 × 8)(125 × 270) | 810000 | 17 | 18 | 1788 | 9 |
| (3 × 8)(125 × 270) | 810000 | 18 | 19 | 1666 | 9 |
| (3 × 8)(125 × 270) | 810000 | 19 | 20 | 1529 | 9 |
| (3 × 8)(125 × 270) | 810000 | 20 | 21 | 1540 | 9 |
| (3 × 8)(125 × 270) | 810000 | 21 | 22 | 1477 | 9 |
| (3 × 8)(125 × 270) | 810000 | 22 | 23 | 1479 | 9 |
| (3 × 8)(125 × 270) | 810000 | 23 | 24 | 1377 | 9 |
| (3 × 8)(125 × 270) | 810000 | 24 | 25 | 1342 | 9 |

Cuadro 6.7: Se muestra el tiempo de ejecución, procesadores e Iteraciones para $Pe=4,2710^{-2}$, se utilizó el algoritmo DVS-PRIMAL.

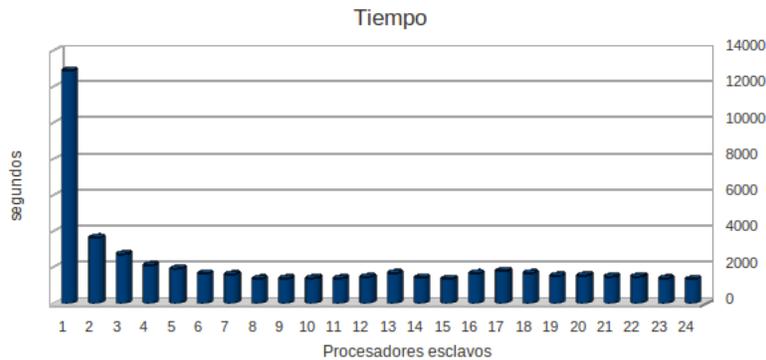


Figura 6.7: Se utilizó la partición (3 × 8)(125 × 270) para cada número de procesador

Se muestra el comportamiento temporal de la tabla (Cuadro 6.7), este sistema cuenta con 810,000 grados de libertad y se utilizaron hasta 25 procesadores en su solución. Se tiene un comportamiento descendente hasta que se utilizan 9 procesadores, ya que al hacer la ejecución con 10 procesadores el tiempo sube 19 unidades. Al usar 11 procesadores se sigue con la tendencia de pequeños aumentos en el tiempo de ejecución.

Cuando se ejecuta el ejemplo con 12 procesadores mejora respecto al resultado anterior, pero al seguir con la ejecución a 13 procesadores se vuelve con la tendencia de aumento. Cuando se utilizan 16 procesadores se mejora el tiempo que se obtuvo con 12 procesadores y en adelante aumenta el tiempo.

Se siguen con tendencias de aumentos y descensos hasta llegar a la ejecución con 24 procesadores, donde por fin se disminuye el tiempo de ejecución obtenido para 16 procesadores. Por lo que los mínimos locales se encuentran al utilizar ejecuciones con 9,12,16,20 22, 25.

Por lo anterior es necesario siempre hacer corridas de prueba buscando la descomposición en la que se obtenga el menor tiempo de ejecución posible para el equipo paralelo con el que se cuente. Estas pruebas dependen fuertemente de la capacidad computacional de cada Core y de la red usada para interconectar los Cores que forman parte del equipo paralelo.

También cabe resaltar que el refinamiento y la dimensión que queramos dar al problema se verá afectada por las características del equipo de cómputo, sin embargo, es notable que con estos algoritmos es posible adaptarse a las restricciones técnicas del equipo y obtener aún buenos resultados.

Estos ejemplos fueron ejecutados por un equipo Lenovo IdeaPad , Intel Atom N550 1.5GHz, 2GB RAM, en el caso secuencial. Con lo que mostramos no solamente el poder de resolución de los algoritmos DVS en equipos de cómputo profesionales, sino que se hace accesible su uso en equipos regulares de mediana capacidad computacional.

En el caso paralelo se utilizó el cluster Olintlali, de 108 Cores emulando 216 hilos, Intel Xeon a 2.67 GHz, 9 nodos con 6 Cores y 8 hilos de ejecución con 48 GB RAM interconectados con GIGE 10/100/1000 Gb/s, perteneciente al grupo de Geofísica Computacional.

Conclusiones y Perspectivas

Recapitulación

Este trabajo de tesis inicia considerando la formulación de los principales modelos matemáticos que son utilizados en los Sistemas Físicos macroscópicos, continuando con una revisión de las ecuaciones de transporte y los modelos más significativos utilizados en el ramo de la Física Terrestre. El trabajo prosigue tratando los métodos de Descomposición de Dominio y formulando sus distintos algoritmos para después dar lugar al marco del método de descomposición de dominio en el Espacio Vectorial Derivado (DVS); aquí se expone como se sitúa dentro de las formulaciones continuas para completar el modelo matemático del esquema DVS. Tomando en cuenta la necesidad actual que tiene el cómputo y sus aplicaciones, se incluyó un capítulo con las generalidades más notables del cómputo de alto rendimiento en las Ciencias de la Tierra.

Posteriormente se tratan las características generales para implementación computacional y mediante ejemplos numéricos se realiza un análisis y discusión de resultados para problemas representativos no simétricos de la ecuación de Advección-Difusión. Por último se establecen los resultados alcanzados en esta tesis y se dan algunas perspectivas del trabajo venidero.

Conclusiones

En esta tesis se utilizan los cuatro algoritmos preconditionados (véase ((4.61)-(4.64))) del método de descomposición de dominio en el Espacio Vectorial Derivado, este esquema unifica a dos de los métodos de descomposición de dominio sin traslape más comúnmente usados: Balancing Domain Decomposition by Constraints (BDDC) y Finite Element Tearing and Interconnect Dual-Primal (FETI-DP).

Una de las principales características del esquema DVS es que permite aplicar técnicas de descomposición de dominio directamente al sistema matricial que se obtiene después de que la ecuación diferencial o el sistema de ecuaciones que han sido discretizadas. Con lo cual se puede aplicar dicho procedimiento sin tener conocimiento alguno sobre la ecuación diferencial que origina las matrices.

Otra de las características del esquema DVS es que es aplicable tanto a matrices simétricas, no simétricas e indefinidas. Aún tratando matrices no simétricas, la eficiencia numérica de los algoritmos preconditionados están en el mismo orden que los algoritmos de descomposición de dominio.

Los algoritmos del esquema DVS se destacan ya que el código implementado es independiente de la geometría, además que con pequeños cambios en éste se pueden aplicar a problemas de dos y tres dimensiones, destacando que pueden soportar matrices simétricas, no simétricas e indefinidas. La ventaja de éstos algoritmos se extiende al ser paralelizables con un bajo costo computacional.

Mostramos como es el comportamiento que tienen estos métodos en la solución de problemas de advección-difusión y al compararlos con ejemplos que han sido reportados en ([38]) para distintos valores del número de Péclet, se obtuvieron los mejores resultados en iteraciones con las formulaciones DVS-FETI-DP y DVS-PRIMAL para el caso difusivo y el caso con advección dominante respectivamente.

Con el objetivo de extender estos resultados a problemas advectivos y como contribución al código utilizado en este trabajo, el cual fue realizado por el grupo de Geofísica Computacional del Instituto de Geofísica, se implementó una estabilización Upwind con la cual se mejoró la solución de problemas de Advección-Difusión con valores del número de Péclet muy grandes ($Pe \gg 1$).

Al modificar la partición del dominio teniendo fijo el número de grados de libertad (figuras 6.4 y 6.5), se muestra que entre más se refine la malla gruesa mayor número de iteraciones se obtiene, sin embargo éstas iteraciones tardan un tiempo mucho menor en ejecutarse. El comportamiento general de una partición revela que al aumentar el número de Péclet las iteraciones disminuyen (figura 6.4) y a su vez que al aumentar el número de Péclet, menor tiempo tardará la ejecución tomando en cuenta que entre menos iteraciones se realicen mayor es el tiempo empleado (figura 6.5).

Expectativas

El objetivo es tener en un futuro próximo mejores resultados en este tipo de problemas, sobre todo en la caracterización de métricas para el cluster Olintlali; debido a que los resultados secuenciales fueron temporalmente superiores a los paralelos. Se pretende resolver esto por medio de la optimización del estabilizador utilizado en esta tesis, además vale la pena poder implementar estos algoritmos en equipos de memoria compartida utilizando las librerías OpenMP y CUDA. Esto propiciará poder tener una eficiencia computacional a un bajo costo como vía alterna a los equipos de alto desempeño.

Apéndice

Apéndice A

Matrices Simétricas y no Simétricas, Definidas Positivas e Indefinidas

A.1. La Matriz Transpuesta

La matriz transpuesta de A se denota por A^T . Sus columnas se toman directamente de los renglones de A : *el i -ésimo renglón de A se convierte en la i -ésima columna de A^T* . Al mismo tiempo, las columnas de A se convierten en los renglones de A^T . Si A es una matriz de m por n , entonces A^T es de n por m . El efecto final es doblar la matriz respecto a su diagonal principal, y el elemento en el renglón i , columna j de A^T proviene del elemento que está en el renglón j , columna i de A :

$$\text{Elementos de } A^T : (A^T)_{ij} = A_{ji} \quad (\text{A.1})$$

i La transpuesta de AB es: $(AB)^T = B^T A^T$

ii La transpuesta de A^{-1} es: $(A^{-1})^T = (A^T)^{-1}$

A.2. Matriz simétrica

Una matriz es simétrica si se cumple que $a_{ij} = a_{ji}$. Es decir, cada elemento de A ubicado en la fila i , columna j , es igual al elemento ubicado en la fila j , columna i . Por tanto, una matriz simétrica es una matriz que es igual a su propia transpuesta $A = A^T$. La matriz es necesariamente cuadrada, cada elemento en un miembro de la diagonal es igual a su "imagen especular" en el otro lado. Una matriz simétrica no es necesariamente invertible, *pero si A^{-1} existe, también es simétrica*.

La transpuesta de A^{-1} siempre es igual a $(A^T)^{-1}$; para una matriz simétrica lo anterior es justo A^{-1} .

Este tipo de matrices nos puede ahorrar espacio de memoria en su almacenamiento, ya que con almacenar los elementos localizados por debajo de la diagonal principal de la misma (triángulo inferior), se conocen los valores de los elementos ubicados por encima de la diagonal superior (triángulo superior) y viceversa. Del mismo modo podemos ahorrar operaciones al aplicar algoritmos que exploten esta estructura.

A.2.1. Matriz definida positiva

Una matriz $A \in \mathbb{R}^{n \times n}$ es definida positiva si es simétrica y $x^T A x > 0$ para todo $x \in \mathbb{R}^n$ diferente de cero. Los sistemas que se asocian con matrices definidas positivas constituyen una de las clases más importantes de problemas $Ax = b$. Las matrices definidas positivas tienen una diagonal de peso, es decir con valores relativamente grandes.

Además, las matrices definidas positivas cumplen las siguientes propiedades:

- Una matriz A definida positiva es no singular.

- Si una matriz A es definida positiva entonces todas sus submatrices principales son definidas positivas. En particular, todas las entradas de la diagonal son positivas.
- Si una matriz A es definida positiva entonces existe una factorización $A = LDM^T$ y la matriz diagonal D tiene valores positivos.
- Si una matriz A es simétrica y definida positiva entonces existe una matriz triangular inferior única G con entradas positivas en la diagonal, tal que $A = GG^T$ (factorización de Cholesky).
- Una matriz simétrica A es definida positiva sí y sólo si sus primeras submatrices principales tienen determinante positivo.

A.2.2. Matriz indefinida

Una matriz A es considerada indefinida cuando su forma cuadrática $x^T Ax$ toma valores tanto positivos como negativos. Aunque la matriz A puede tener una factorización LDL^T , las entradas en los factores pueden tener una magnitud arbitraria.

A.3. Matriz Diagonal por bloques

Sean $B_1 \in \mathcal{M}_{k_1}(\mathbb{F})$, $B_2 \in \mathcal{M}_{k_2}(\mathbb{F})$, ..., $B_m \in \mathcal{M}_{k_m}(\mathbb{F})$. Entonces la matriz

$$\begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_m \end{bmatrix} \quad (\text{A.2})$$

se denota por

$$\text{diag}(B_1, B_2, \dots, B_m)$$

Ejemplo: Sean B_1, B_2 y B_3 las siguientes matrices:

$$B_1 = \begin{bmatrix} 3 & -5 \\ 7 & 1 \end{bmatrix} \quad B_2 = 4 \quad B_3 = \begin{bmatrix} -1 & 2 & 3 \\ 4 & -3 & 7 \\ 1 & -3 & 4 \end{bmatrix}$$

Entonces

$$\text{diag}(B_1, B_2, B_3) = \begin{bmatrix} 3 & -5 & 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 3 \\ 0 & 0 & 0 & 4 & -3 & 7 \\ 0 & 0 & 0 & 1 & -3 & 4 \end{bmatrix}$$

A.3.1. Propiedades básicas

Sean

$$A = \text{diag}(B_1, \dots, B_m), \quad C = \text{diag}(D_1, \dots, D_m)$$

donde para cualquier $j \in \{1, \dots, m\}$ las matrices B_j y D_j son cuadradas y del mismo orden: $C_j, D_j \in \mathcal{M}_{d_j}(\mathbb{F})$. Entonces:

1. $A + C = \text{diag}(B_1 + D_1, \dots, B_m + D_m)$.
2. $\lambda A = \text{diag}(\lambda B_1, \dots, \lambda B_m)$.
3. $AC = \text{diag}(B_1 D_1, \dots, B_m D_m)$.
4. $A^k = \text{diag}(B_1^k, \dots, B_m^k)$.
5. A es invertible sí, y sólo si todos los bloques B_1, \dots, B_m son invertibles.
6. $f(A) = \text{diag}(f(B_1), \dots, f(B_m))$ para todo $f \in \mathcal{P}(\mathbb{F})$.
7. $r(A) = \sum_{j=1}^m r(B_j)$.
8. $\det(A) = \prod_{j=1}^m \det(B_j)$.
9. $\lambda I_n - A = \text{diag}(\lambda I_{d_1} - B_1, \dots, \lambda I_{d_m} - B_m)$.
10. $C_A(\lambda) = \prod_{j=1}^m C_{B_j}(\lambda)$.
11. $\text{sp}(A) = \bigcup_{j=1}^m \text{sp}(B_j)$.

A.4. Número de condición de una matriz

Sea $A \in \mathbb{C}^{m \times n}$, con $\text{rg}(A) = n \leq m$. Se define el número de condición de A asociado a una norma $\|\cdot\|$ como

$$c(A) = \frac{\max_{\|x\|=1} \|Ax\|}{\min_{\|x\|=1} \|Ax\|}. \quad (\text{A.3})$$

El cociente anterior está bien definido debido a:

$$\min_{\|x\|=1} \|Ax\| = 0 \Leftrightarrow \exists x \neq 0, Ax = 0 \Leftrightarrow \text{rg}(A) < n \quad (\text{A.4})$$

En el caso $\text{rg}(A) < n$, se define $c(A) = \infty$. El número de condición proporciona una cota superior para el error en la resolución de un sistema de ecuaciones. Veámoslo en dos casos.

A.4.1. Perturbación del término independiente b

Se considera el sistema $Ax = b$, con $A \in \mathbb{C}^{m \times n}$ y $\text{rg}(A) = n \leq m$. Supongamos que tiene solución única $x_0 \neq 0$. Se desea estudiar la variación del vector solución x_0 ante variaciones del vector b . De este modo, se considera el sistema $Ax = b + \Delta b$, y se asume que también tiene solución única $x_0 + \Delta x_0$. Se pretende estimar $\|\Delta x_0\|$. Observamos

$$Ax_0 = b \quad (\text{A.5})$$

$$A(x_0 + \Delta x_0) = b + \Delta b \quad (\text{A.6})$$

Restando ambas ecuaciones

$$A(\Delta x_0) = \Delta b \quad (\text{A.7})$$

Denotando $M = \max_{\|x\|=1} \|Ax\|$ y $m = \min_{\|x\|=1} \|Ax\|$, se tiene:

$$m = \min_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} \leq \frac{\|A\Delta x_0\|}{\|\Delta x_0\|} = \frac{\|\Delta b\|}{\|\Delta x_0\|} \Rightarrow \|\Delta x_0\| \leq \frac{\|\Delta b\|}{m}. \quad (\text{A.8})$$

$$M = \max_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} \geq \frac{\|Ax_0\|}{\|x_0\|} = \frac{\|b\|}{\|x_0\|} \Rightarrow \frac{1}{\|x_0\|} \leq \frac{M}{\|b\|}. \quad (\text{A.9})$$

Multiplicando (A.8) y (A.9):

$$\frac{\|\Delta x_0\|}{\|x_0\|} \leq \frac{M}{m} \frac{\|\Delta b\|}{\|b\|} = c(A) \frac{\|\Delta b\|}{\|b\|}. \quad (\text{A.10})$$

En conclusión, el número de condición proporciona una cota superior para la amplificación del error relativo.

A.4.2. Perturbación de la matriz A

De nuevo se considera el sistema $Ax = b$, que se (asume) tiene solución única $x_0 \neq 0$. Ahora se desea estudiar la variación del vector solución x_0 ante variaciones de la matriz A . De este modo, se considera el sistema $(A + \Delta A)x = b$. Supongamos que este nuevo sistema sigue teniendo solución única $X_0 + \Delta x_0$, con $\Delta x_0 \neq 0$. En este caso y de forma similar al anterior se verifica:

$$\frac{\|\Delta x_0\|}{\|x_0 + \Delta x_0\|} \leq c(A) \frac{\|\Delta A\|}{\|A\|}. \quad (\text{A.11})$$

De nuevo, el número de condición proporciona una cota superior para la amplificación del error relativo.

Apéndice B

Solución de Sistemas Lineales

A lo largo de este apéndice consideraremos la solución de sistemas lineales

$$Au = b \quad (\text{B.1})$$

con $u, b \in (\mathbb{R})^n$, y A una matriz $n \times n$ real e invertible. Usaremos la notación $\langle u, v \rangle = u^T v$, para $u, v \in (\mathbb{R})^n$.

B.1. Eigenvalores y Número de Condición

Dada una matriz $A \in \mathbb{R}^{n \times n}$, sus eigenvalores $\lambda \in \mathbb{C}$ y eigenvectores $u \in \mathbb{C}^n \setminus \{0\}$ son soluciones para

$$Au = \lambda u. \quad (\text{B.2})$$

El conjunto de los eigenvalores de A , también llamado *espectro*, es denotado por $\sigma(A)$. El radio espectral $\rho(A)$ es definido como:

$$\rho(A) := \max_{\lambda \in \sigma(A)} \{|\lambda|\}. \quad (\text{B.3})$$

Dada una norma matricial $\|\cdot\|_*$, definimos el número de condición de una matriz invertible A como

$$\kappa_*(A) := \|A\|_* \|A^{-1}\|_*. \quad (\text{B.4})$$

De la misma manera, dada una segunda matriz M , podemos considerar el eigenproblema generalizado:

$$Au = \lambda Mu. \quad (\text{B.5})$$

Una matriz A se dice que es *definida positiva* si todos sus eigenvalores tienen parte real positiva o equivalentemente si $u^T Au$ tiene parte real positiva para $u \in \mathbb{C} \setminus \{0\}$. En este caso podemos observar

$$u^T Au = u^T \frac{A + A^T}{2} u > 0, \quad u \in \mathbb{R}^n \setminus \{0\}. \quad (\text{B.6})$$

Si además A es simétrica, entonces sus eigenvalores son reales y estrictamente positivos. A lo largo de este repaso, se hará uso de esta propiedad.

Lemma B.0

Sean A y M dos matrices simétricas y definidas positivas de orden n . Para una matriz arbitraria $B \in \mathbb{R}^{n \times n}$, sea

$$\|B\|_A := \sup_{u \in \mathbb{R}^n} \frac{\|Bu\|_A}{\|u\|_A}, \quad (\text{B.7})$$

con $\|u\|_A^2 := u^T Au$ e igualmente para $\|B\|_M$. Por lo tanto:

1. El siguiente problema de eigenvalores tiene los mismos n eigenvalores

$$Au = \lambda Mu, \quad (\text{B.8})$$

$$M^{-1}Au = \lambda u, \quad (\text{B.9})$$

$$(M^{-1/2}AM^{-1/2})u = \lambda u, \quad (\text{B.10})$$

$$(A^{1/2}M^{-1}A^{1/2})u = \lambda u. \quad (\text{B.11})$$

Todos son reales y estrictamente positivos.

2. Los eigenvalores máximos y mínimos de los problemas anteriores satisfacen

$$\lambda_{\min} = \inf_{u \in \mathbb{R}^n} \frac{u^T Au}{u^T Mu}, \quad \lambda_{\max} = \sup_{u \in \mathbb{R}^n} \frac{u^T Au}{u^T Mu}. \quad (\text{B.12})$$

3. Tenemos que

$$\|M^{-1}A\|_A = \|M^{-1}A\|_M = \lambda_{\max} = \rho(M^{-1}A),$$

$$\|(M^{-1}A)^{-1}\|_A = \|(M^{-1}A)^{-1}\|_M = 1/\lambda_{\min}$$

y por tanto

$$\kappa_A(M^{-1}A) = \kappa_2(M^{-1/2}AM^{-1/2}) = \lambda_{\max}/\lambda_{\min}. \quad (\text{B.13})$$

4. Notemos que

$$(u^T Au \geq cu^T Mu, \quad u \in \mathbb{R}^n) \implies \lambda_{\min} \geq c.$$

Análogamente

$$(u^T Au \leq Cu^T Mu, \quad u \in \mathbb{R}^n) \implies \lambda_{\max} \leq C,$$

y esto nos lleva a

$$\kappa_A(M^{-1}A) \leq C/c. \quad (\text{B.14})$$

usaremos la notación:

$$\kappa(M^{-1}A) = \kappa_A(M^{-1}A) = \kappa_M(M^{-1}A)$$

Observemos que $M^{-1}A$ es no simétrica, por tanto su norma $\|M^{-1}A\|_2$ en general no es igual a su eigenvalor mayor. Sin embargo, $M^{-1}A$ es simétrica con respecto al producto escalar inducido por A y M .

B.2. CGM: Método de Gradiente Conjugado

Sea A simétrica y positiva definida. En el método de Gradiente Conjugado las direcciones $\{p^k\}$ de búsqueda se escogen conjugadas respecto a A , i.e.,

$$\langle p^i, Ap^j \rangle = \langle p^i, p^j \rangle_A = 0, \quad i \neq j. \quad (\text{B.15})$$

Además, el algoritmo provee una forma económica de generar direcciones conjugadas. Asumiremos ahora que se dará una solución inicial u^0 y un conjunto de direcciones conjugadas $\{p^k\}$. Así podremos generar la secuencia:

$$u^{k+1} = u^k + \alpha^k p^k, \quad k = 0, 1, \dots \quad (\text{B.16})$$

donde α^k es el minimizador unidimensional que resuelve

$$\min_{\alpha \in \mathbb{R}} \phi(u^k + \alpha p^k). \quad (\text{B.17})$$

Así tenemos:

$$\alpha = \alpha^k = \frac{\langle r^k, p^k \rangle}{\langle p^k, Ap^k \rangle} \quad (\text{B.18})$$

Debido a que los vectores $\{p^k\}$ son ortogonales respecto al producto escalar $\langle \cdot, \cdot \rangle_A$, también son linealmente independientes y proveen una base para \mathbb{R}^n . Tenemos la siguiente propiedad

Lemma B.1

Sea $u^0 \in \mathbb{R}^n$ y $\{p^k\}$ un conjunto arbitrario de direcciones conjugadas. Para $k \geq 1$, se tiene:

$$\langle r^k, p^i \rangle = 0, \quad i = 0, 1, \dots, k-1,$$

y u^k minimiza $\phi(x)$ sobre el espacio

$$u^0 + \text{span}\{p_i, \quad i = 0, 1, \dots, k-1\}.$$

Consecuentemente, la secuencia generada por (B.16) converge a la solución de (B.1) a lo mucho en n pasos.

Notemos que el resultado previo se cumple para cualquier conjunto de direcciones conjugadas. El algoritmo de gradiente conjugado provee una elección particular para las direcciones de búsqueda: estas son de la forma:

$$p^k = r^k + \beta^k p^{k-1}, \quad (\text{B.19})$$

donde el escalar β^k es determinado únicamente imponiendo la condición de que la nueva dirección es conjugado de las anteriores. Así podremos escribir el algoritmo:

Sólo una aplicación de la matriz A es necesaria en cada paso. Tenemos la siguiente propiedad:

1. Inicializa: $\mathbf{r}^0 = \mathbf{b} - A\mathbf{u}^0$
2. Itera: $k = 1, 2, \dots$ hasta la convergencia

$$\beta^k = \langle \mathbf{r}^{k-1}, \mathbf{r}^{k-1} \rangle / \langle \mathbf{r}^{k-2}, \mathbf{r}^{k-2} \rangle \quad [\beta^1 = 0]$$

$$\mathbf{p}^k = \mathbf{r}^{k-1} + \beta^k \mathbf{p}^{k-1} \quad [\mathbf{p}^1 = \mathbf{r}^0]$$

$$\alpha^k = \langle \mathbf{r}^{k-1}, \mathbf{r}^{k-1} \rangle / \langle \mathbf{p}^k, A\mathbf{p}^k \rangle$$

$$\mathbf{u}^k = \mathbf{u}^{k-1} + \alpha^k \mathbf{p}^k$$

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha^k A\mathbf{p}^k$$

Figura B.1: Gradiente conjugado sin Precondicionar

Lemma B.2

Supóngase que la k -ésima iteración generada por el método de Gradiente Conjugado no es la solución de (B.1), Entonces

$$\begin{aligned} \langle \mathbf{r}^k, \mathbf{r}^i \rangle &= 0, \quad i = 0, \dots, k-1, \\ \text{span}\{\mathbf{r}^i, \quad i = 0, \dots, k\} &= \text{span}\{A^i \mathbf{r}^0, \quad i = 0, \dots, k\} \\ \text{span}\{\mathbf{p}^i, \quad i = 0, \dots, k\} &= \text{span}\{A^i \mathbf{r}^0, \quad i = 0, \dots, k\} \\ \langle \mathbf{p}^k, A\mathbf{p}^i \rangle &= 0, \quad i = 0, \dots, k-1, \end{aligned}$$

Por lo tanto la secuencia \mathbf{u}^k converge a la solución de (B.1) a lo más en n pasos.

Podemos notar que tanto los residuales como las direcciones conjugadas proveen bases para los *espacios de Krylov* (véase [2]) $\mathcal{K}_k = \mathcal{K}_k(\mathbf{r}_0, A) = \text{span}\{A^i \mathbf{r}^0, \quad i = 0, \dots, k-1\}$, y gracias al Lemma B.1, la iteración \mathbf{u}^k minimiza $\phi(x)$ sobre el espacio $\mathbf{u}^0 + \mathcal{K}_k(\mathbf{r}_0, A)$.

La convergencia del Gradiente Conjugado sin precondicionar depende del número de condición de A , lo que nos lleva a:

Lemma B.3

Sea A simétrica y definida positiva. Entonces, el método del Gradiente Conjugado satisface la cota de error:

$$\|\mathbf{e}^k\|_A \leq 2\eta_A^k \|\mathbf{e}^0\|_A$$

donde el factor de convergencia es:

$$\eta_A = \frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}.$$

Podemos notar que el resultado previo puede proveer solo un estimado burdo de la velocidad de convergencia. Si los eigenvalores de A están agrupados, es bien sabido que la convergencia es rápida. (véase [2]) La iteración del Gradiente Conjugado también nos da un estimado de los eigenvalores de la matriz A (y por lo tanto $\kappa_2(A)$). En efecto y gracias al Lemma B.2 las columnas de

1. Inicializa: $\mathbf{r}^0 = \mathbf{b} - A\mathbf{u}^0$
 2. Itera: $k = 1, 2, \dots$ hasta la convergencia
- Precondicionador: $\mathbf{z}^{k-1} = M^{-1}\mathbf{r}^{k-1}$
- $$\beta^k = \langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1} \rangle / \langle \mathbf{r}^{k-2}, \mathbf{r}^{k-2} \rangle \quad [\beta^1 = \mathbf{0}]$$
- $$\mathbf{p}^k = \mathbf{z}^{k-1} + \beta^k \mathbf{p}^{k-1} \quad [\mathbf{p}^1 = \mathbf{z}^0]$$
- $$\alpha^k = \langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1} \rangle / \langle \mathbf{p}^k, A\mathbf{p}^k \rangle$$
- $$\mathbf{u}^k = \mathbf{u}^{k-1} + \alpha^k \mathbf{p}^k$$
- $$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha^k A\mathbf{p}^k$$

Figura B.2: Gradiente conjugado Precondicionado

$$R_k = [r_0 / \|r_0\|, \dots, r_{k-1} / \|r_{k-1}\|] \quad (\text{B.20})$$

nos genera una base ortonormal del espacio de Krylov $\mathcal{K}_k(r_0, A)$. Uno puede probar que la restricción de A hacia $\mathcal{K}_k(r_0, A)$

$$T_k = R_k^T A R_k \quad (\text{B.21})$$

es una matriz simétrica y tridiagonal (véase [2]). Las entradas de ésta matriz pueden construirse de los coeficientes α^i y β^i de la iteración del Gradiente Conjugado. Calculando los eigenvalores de T_k puede obtenerse fácilmente estimados de los eigenvalores de A grandes y pequeños. El preconditionador es necesario en los casos donde $\kappa_2(A)$ es grande y en el caso de aproximaciones por elemento finito y espectral. Dada una matriz M simétrica y positiva definida, consideremos el sistema lineal modificado:

$$M^{-1/2} A M^{-1/2} \mathbf{v} = M^{-1/2} \mathbf{b}, \quad \mathbf{v} = M^{1/2} \mathbf{u}. \quad (\text{B.22})$$

Podemos notar que $M^{-1/2} A M^{-1/2}$ es simétrico y positivo definido y se reduce a la identidad en el caso $M = A$. Entonces, podemos considerar a M como un *precondicionador* de A y aplicar el método de Gradiente Conjugado sin preconditionar a éste sistema modificado. Después de algunas manipulaciones, podemos definir el algoritmo mostrado en la figura B.2.

Este algoritmo no involucra la aplicación de $M^{-1/2}$, solamente la de M^{-1} . Sin embargo, no es el método de Gradiente Conjugado aplicado un sistema involucrando la matriz $M^{-1}A$, la cual no es simétrica en general. Además, se requiere una aplicación de la matriz origina A y una aplicación de M^{-1} a un vector (véase [2]).

Lemma B.4

Sean A y M simétricas y definidas positivas. Entonces, el método de Gradiente Conjugado preconditionado satisface la misma cota de error que el Lema B.3

$$\eta_A = \frac{\sqrt{\kappa(M^{-1}A)} - 1}{\sqrt{\kappa(M^{-1}A)} + 1}.$$

La estimación de los eigenvalores de $M^{-1}A$ en este caso pueden obtenerse usando los coeficientes α^i y β^i .

B.3. GMRES: El método de Residuo Mínimo Generalizado

Se describirá brevemente el algoritmo GMRES y se establecerá un teorema de convergencia.

Sea A una matriz invertible, no necesariamente simétrica o positiva definida y $u^0 \in \mathbb{R}^n$ un vector inicial. El algoritmo GMRES básicamente recae en dos ideas. La primera es construir bases ortonormales $\{v^1, v^2, \dots, v^k\}$ para subespacios de Krylov.

$$\mathcal{K}_k = \mathcal{K}_k(r^0, A) = \text{span}\{r^0, Ar^0, \dots, A^{k-1}r^0\},$$

donde $r^0 = b - Au^0$ es el residual inicial. Esto lo hace por la llamada *iteración de Arnoldi* lo cual produce la representación de A en \mathcal{K}_k . Una solución aproximada w^j se encuentra resolviendo el problema de mínimos cuadrados.

$$\min_{u \in u^0 + \mathcal{K}_j} \|b - Au\|_2 = \min_{z \in \mathcal{K}_j} \|r^0 - Az\|_2 \quad (\text{B.23})$$

1. Inicializa: $r^0 = b - Au^0$

$$r^0 = b - Au^0$$

$$v^1 = r^0 / \|r^0\|_2$$

2. Itera: $k = 1, 2, \dots, j$

$$q^k = Av^k$$

$$h_{i,k} = \langle q^k, v^i \rangle, \quad i = 1, \dots, k$$

$$\widehat{v}^k = q^k - \sum_{i=1}^k h_{i,k} v^i$$

$$h_{k+1,k} = \|\widehat{v}^k\|_2$$

$$v^{k+1} = \widehat{v}^k / h_{k+1,k}$$

3. Forma la solución aproximada:

$$u^j = u^0 + V_j w^j, \text{ donde } w^j \text{ minimiza } \|\beta e_1 - H_j w\|_2, w \in \mathbb{R}^j$$

Figura B.3: Iteración GMRES

donde $\beta = \|r^0\|$, e_1 es la primer columna de la matriz identidad $(j+1) \times (j+1)$, V_j es la matriz, las columnas de ésta son los vectores v^k , $k = 1, \dots, j$, y H_j es una matriz de $(j+1) \times j$, las entradas distintas de cero son los elementos $h_{i,k}$.

El segundo paso en la iteración es un paso de la iteración de Arnoldi. En efecto, fácilmente encontramos

$$Av^k = \sum_{i=1}^{k+1} h_{i,k} v^i, \quad k = 1, \dots, j, \quad (\text{B.24})$$

por lo tanto

$$AV_j = V_{j+1}H_j \quad (\text{B.25})$$

Ya que las columnas de V_j y V_{j+1} son ortonormales, encontramos que

$$V_j^T AV_j = \tilde{H}_j, \quad (\text{B.26})$$

donde \tilde{H}_j es una matriz Hessemberg (véase [2]) de $j \times j$, las entradas distintas a cero de dicha matriz son los elementos $h_{i,k}$. La representación de A en \mathcal{K}_k es por tanto \tilde{H}_j y puede ser utilizada para extraer información sobre el espectro de A .

A continuación haremos énfasis en el tercer paso de algoritmo. Este claramente resuelve el problema de minimización (B.23). A fin de ver esto, reescribimos la ecuación (B.23) como

$$\min_{w \in \mathbb{R}^j} \|\beta v^1 - AV_j w\|_2 \quad (\text{B.27})$$

Usando la representación (B.25) y el hecho de que las columnas de V_{j+1} son ortonormales, finalmente podemos escribir:

$$\|\beta v^1 - AV_j w\|_2 = \|V_{j+1}(\beta e^1 - H_j w)\|_2 = \|\beta e^1 - H_j w\|_2. \quad (\text{B.28})$$

El problema de mínimos cuadrados en el tercer paso del algoritmo puede resolverse usando una factorización QR para H_j . Dicha factorización puede obtenerse de H_{j-1} en los pasos previos. Además, la norma residual de la solución aproximada u^j puede calcularse de la factorización QR sin costo extra, sin encontrar u^j . Gracias a la propiedad de minimización (B.23), la solución exacta puede ser alcanzada en no más de n iteraciones si se usa aritmética exacta. La velocidad de convergencia del método GMRES puede ser caracterizada en términos del mínimo eigenvalor de la parte simétrica del operador y de la norma del operador. Estos son definidos por:

$$c_p = \min_{u \in \mathbb{R}^n} \frac{\langle u, Au \rangle}{\langle u, u \rangle}, \quad C_p = \max_{u \in \mathbb{R}^n} \frac{\|Au\|_2}{\|u\|_2}; \quad (\text{B.29})$$

Considerando el decremento de la norma del residual en un sólo paso, el teorema siguiente se enuncia:

Lema B.5

Si $c_p > 0$, entonces, después de k pasos, la norma del residual es acotada por

$$\|r^k\|_2 \leq \left(1 - \frac{c_p^2}{C_p^2}\right)^{k/2} \|r^0\|_2.$$

La velocidad de convergencia puede mejorarse usando un preconditionador adecuado. Sea M una matriz invertible. Consideramos el problema preconditionado, derivado de (B.1),

$$M^{-1}Au = M^{-1}b. \quad (\text{B.30})$$

El algoritmo GMRES puede escribirse en este caso como en la figura (B.4). Aquí β , V_j y H_j son definidas como se hizo anteriormente.

1. Inicializa: $r^0 = b - Au^0$

$$r^0 = b - Au^0$$

Precondiciona: $z^0 = M^{-1}r^0$

$$v^1 = z^0 / \|z^0\|_2$$

2. Itera: $k = 1, 2, \dots, j$

$$q^k = Av^k$$

Precondiciona: $z^k = M^{-1}q^k$

$$h_{i,k} = \langle z^k, v^i \rangle, \quad i = 1, \dots, k$$

$$\widehat{v}^k = z^k - \sum_{i=1}^k h_{i,k} v^i$$

$$h_{k+1,k} = \|\widehat{v}^k\|_2$$

$$v^{k+1} = \widehat{v}^k / h_{k+1,k}$$

3. Forma la solución aproximada:

$$u^j = u^0 + V_j w^j, \text{ donde } w^j \text{ minimiza } \|\beta e_1 - H_j w\|_2, w \in \mathbb{R}^j$$

Figura B.4: Iteración GMRES Precondicionada

Apéndice C

La Ecuación de Advección-Difusión y su estabilización

C.1. La ecuación de Advección difusión

Considerando el transporte de una sustancia de concentración u en un medio fluido con campo de velocidades β , el cual en el caso estacionario ($\partial u/\partial t = 0$) y al no existir reacción y producción (ecuación 2.3) se obtiene la *ecuación de advección-difusión*. Consideremos el caso 1D en un intervalo de longitud L con condiciones de frontera Dirichlet.

$$\begin{aligned}\beta \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} &= 0, \text{ en } 0 \leq x \leq L \\ u &= 0, \text{ en } x = 0 \\ u &= 1, \text{ en } x = L\end{aligned}\tag{C.1}$$

Esto representa el transporte de temperatura u por un fluido con difusividad ν y velocidad β . En este caso la solución puede encontrarse por métodos operacionales estándar, proponiendo soluciones de la forma $e^{\lambda x}$, resolviendo el polinomio característico en λ y buscando la combinación lineal que satisface las condiciones de frontera. La solución es:

$$u = \frac{e^{2Pe(x/L)} - 1}{e^{2Pe} - 1}, \quad Pe = \frac{\beta L}{2\nu}\tag{C.2}$$

Para valores de β muy pequeños la solución se aparta poco de la solución de conducción pura

$$u = x/L\tag{C.3}$$

A medida que β aumenta las temperaturas bajan, debido a que el movimiento del fluido tiende a contrarrestar el efecto de la condición de frontera en $x = L$ y refrigera más que cuando el fluido está quieto. La importancia relativa de ambos términos (difusivo y convectivo) se puede cuantificar a través del “número de Péclet” (Pe). A medida que el número de Péclet aumenta, el gradiente de u se concentra más y más cerca de la pared $x = 1$, formando una capa límite; estos altos gradientes son una fuente de problemas para los métodos numéricos.

Si la velocidad se invierte entonces la discontinuidad se produce en $x = 0$ que es la nueva salida ($x = L$ sería la entrada). Una forma diferente de ver este fenómeno es considerar que, a medida que $\nu \rightarrow 0$, el problema se hace cada vez más advectivo ($Pe \rightarrow \infty$). En el caso del problema advectivo puro la ecuación es de primer orden y por lo tanto requiere una sola condición de frontera. La teoría de sistemas hiperbólicos indica que la condición de frontera debe aplicarse donde las líneas características entran. El valor de la variable en una frontera donde las características salen resulta de la integración de la ecuación dentro del dominio. Si se pretende imponer un valor diferente como condición de frontera Dirichlet, la diferencia se absorbe en una capa límite.

Debido a que la ecuación de advección pura propaga los valores a lo largo de las líneas características, también se pueden producir altos gradientes en el interior del dominio.

C.1.1. Discretización de la ecuación de advección-difusión

Consideremos el problema unidimensional de advección difusión con coeficientes constantes (ecuación C.1). Consideremos una malla uniforme de N segmentos de longitud $\Delta x = L/N$. Los nodos son $x_j = (j-1)\Delta x$, para $j = 1, \dots, N+1$. Una discretización centrada de segundo orden es:

$$\beta \frac{u_{j+1} - u_{j-1}}{2\Delta x} - \nu \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} = 0 \quad (\text{C.4})$$

Es relevante notar que la derivada de primer orden introduce un término antisimétrico. Este esquema funciona bien mientras la velocidad se mantenga por debajo de un cierto límite, que resulta ser $v_{crit} = 2\nu/\Delta x$. Para velocidades mayores la solución numérica se vuelve oscilatoria y para velocidades mucho más grandes que la crítica las oscilaciones contaminan todo el dominio. Nótese que la velocidad crítica se produce cuando el número de Péclet de la malla es:

$$Pe_{\Delta x} = \frac{\beta \Delta x}{2\nu} = 1 \quad (\text{C.5})$$

Estas oscilaciones pueden asociarse a una falta de estabilidad del esquema numérico, sin embargo el esquema es estable, estrictamente hablando, ya que si refinamos suficientemente entonces $Pe_{\Delta x}$ pasará a ser menor que uno y se recupera la convergencia $O(\Delta x^2)$.

C.2. Esquema de diferencias contracorriente

Notemos que, en el caso de advección pura ($Pe = \infty$) la solución numérica debería ser

$$u_j = \begin{cases} 0 & ; \text{si } j \leq N \\ 1 & ; \text{si } j = N + 1 \end{cases} \quad (\text{C.6})$$

y esto puede lograrse al reemplazar la derivada centrada por una derivada lateral izquierda (que usualmente también se le llama “contracorriente” o “upwinded”)

$$\beta \frac{u_j - u_{j-1}}{\Delta x} = 0, \quad j = 2, \dots, N \quad (\text{C.7})$$

Esto se puede reescribir como:

$$\beta \frac{u_{j+1} - u_{j-1}}{2\Delta x} - \left(\frac{\beta \Delta x}{2} \right) \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} = 0 \quad (\text{C.8})$$

nombramos $v_{num} = \beta\Delta x/2$,

$$\beta \frac{u_{j+1} - u_{j-1}}{2\Delta x} - v_{num} \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} = 0 \quad (C.9)$$

donde v_{num} es una difusión “numérica artificial” que “estabiliza” el esquema. Notar que si $\beta < 0$, entonces el esquema debe estar descentrado a la derecha

$$\beta \frac{u_{j+1} - u_j}{\Delta x} \quad (C.10)$$

y entonces debe ser $v_{num} = -\beta\Delta x/2$, de manera que, en general

$$v_{num} = \frac{|\beta|\Delta x}{2} \quad (C.11)$$

Ahora bien, para $Pe < \infty$ podemos tomar

$$\beta \frac{u_{j+1} - u_{j-1}}{2\Delta x} - (v + v_{num}) \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} = 0 \quad (C.12)$$

Este esquema no presenta más inestabilidades y para $Pe \gg 1$ se aproxima al upwindado pero para Pe pequeños sigue agregando difusión, incluso para $Pe_{\Delta x} < 1$ cuando sabemos que el esquema es estable, resultando en un esquema demasiado difusivo. Entonces surge la idea de tratar de reducir la difusión numérica para valores $Pe_{\Delta x}$ pequeños. Notemos que (C.11) puede reescribirse como:

$$v_{num} = \frac{|\beta|\Delta x}{2} = v|Pe_{\Delta x}| = vPe_{\Delta x} f(Pe_{\Delta x}) = \frac{\beta\Delta x}{2} f(Pe_{\Delta x}) \quad (C.13)$$

con $f(x) = \text{sign}(x)$

de manera que podríamos reemplazar la “función de upwinding” $\text{sign}()$ por otra que anule la difusión numérica para $|Pe_{\Delta x}| \leq 1$, por ejemplo podemos usar

$$v_{num} = \frac{\beta\Delta x}{2} f_1(Pe_{\Delta x}) \quad (C.14)$$

con

$$f_1(x) = \begin{cases} \text{sign}(x) & ; \text{si } |x| > 1 \\ 0 & ; \text{si } |x| \leq 1 \end{cases} \quad (C.15)$$

o también, para hacerlo más continuo

$$f_2(x) = \begin{cases} \text{sign}(x) & ; \text{si } |x| > 1 \\ x & ; \text{si } |x| \leq 1 \end{cases} \quad (C.16)$$

Puede demostrarse que, si elegimos $f(x)$, como la siguiente “función mágica”

$$f_3(x) = \alpha(x) = \frac{1}{\tanh(x)} - \frac{1}{x} \quad (C.17)$$

entonces, en este caso simple se obtiene la solución exacta. Sin embargo, esto sólo vale mientras el problema sea unidimensional, con coeficientes constantes, paso de la malla constante y sin términos fuente. De todas formas es una función upwinding interesante, ya que en forma muy suave satisface todos los límites necesarios.

Como $(d\alpha/dx)|_{x=0} = 1/3$ otra función comúnmente utilizada es

$$f_4(x) = \begin{cases} x/3 & ; si \ |x| < 3 \\ sign(x) & ; si \ |x| > 3 \end{cases} \quad (C.18)$$

C.3. El caso 2D

En el intento por extender el esquema upwindado a 2D se agrega una viscosidad numérica según cada una de las direcciones principales de la malla. Sea un dominio rectangular $0 \leq x \leq L_x$, $0 \leq y \leq L_y$, dividido en N_x, N_y intervalos de longitud $h_x = L_x/N_x$, $h_y = L_y/N_y$. Los puntos de la malla están ubicados en $x_{ij} = ((i-1)h_x, (j-1)h_y)$ y sea $u(x_{ij}) \approx u_{i,j}$.

Considerando el caso general en que $h_x \neq h_y$, el esquema es:

$$\begin{aligned} & \beta_x \frac{u_{i+1,j} - u_{i-1,j}}{2h_x} + \beta_y \frac{u_{i,j+1} - u_{i,j-1}}{2h_y} - (\nu + \nu_{num,xx}) \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} \\ & - (\nu + \nu_{num,yy}) \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} - 2\nu_{num,xy} \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h_x h_y} = 0 \end{aligned} \quad (C.19)$$

Además:

$$\nu_{ij} = \frac{\beta h_s}{2} s_i s_j f(Pe_s) \quad (C.20)$$

, donde $\widehat{\mathbf{s}} = \overline{\beta}/\beta$ es el vector según la línea de corriente; además se define:

$$Pe_s = \frac{\beta h_s}{2\nu} \quad (C.21)$$

y

$$f(Pe_s) = \begin{cases} sign(Pe_s) & ; si \ |Pe_s| > 1 \\ Pe_s & ; si \ |Pe_s| \leq 1 \end{cases} \quad (C.22)$$

el subíndice s indica valores según la línea de corriente, así por ejemplo.

$$h_s = \sqrt{2}h \quad (C.23)$$

Para facilitar la implementación de éste método en lenguaje computacional, nos será útil transformar la ecuación (C.19) a la forma matricial $Au = 0$, dónde A será una

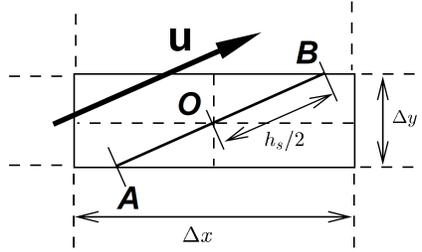


Figura C.1: Definición del tamaño de la celda según la línea de corriente

C.4. El caso 3D

Análogamente a la sección anterior se extiende el esquema upwindado a 2D a un esquema 3D, para esto se agrega una viscosidad numérica según cada una de las direcciones principales de la malla. Sea un dominio cuadrangular $0 \leq x \leq L_x$, $0 \leq y \leq L_y$, $0 \leq z \leq L_z$ dividido en N_x, N_y, N_z intervalos de longitud $h_x = L_x/N_x$, $h_y = L_y/N_y$, $h_z = L_z/N_z$. Los puntos de la malla están ubicados en $x_{ijk} = ((i-1)h_x, (j-1)h_y, (k-1)h_z)$ y sea $u(x_{ijk}) \approx u_{i,j,k}$.

Considerando el caso general en que $h_x \neq h_y \neq h_z$, el esquema es:

$$\begin{aligned}
 & \beta_x \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2h_x} + \beta_y \frac{u_{i,j+1,k} - u_{i,j-1,k}}{2h_y} + \beta_z \frac{u_{i,j,k+1} - u_{i,j,k-1}}{2h_z} \\
 & - (v + v_{num,xx}) \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h_x^2} - (v + v_{num,yy}) \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{h_y^2} \\
 & - (v + v_{num,zz}) \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{h_z^2} - 2v_{num,xy} \frac{u_{i+1,j+1,k} - u_{i+1,j-1,k} - u_{i-1,j+1,k} + u_{i-1,j-1,k}}{4h_x h_y} \quad (C.27) \\
 & - 2v_{num,xz} \frac{u_{i+1,j,k+1} - u_{i+1,j,k-1} - u_{i-1,j,k+1} + u_{i-1,j,k-1}}{4h_x h_z} - 2v_{num,yz} \frac{u_{i,j+1,k+1} - u_{i,j+1,k-1} - u_{i,j-1,k+1} + u_{i,j-1,k-1}}{4h_y h_z} \\
 & - 6v_{num,xyz} \frac{u_{i+1,j+1,k+1} - u_{i+1,j+1,k-1} - u_{i+1,j-1,k+1} + u_{i+1,j-1,k-1} - u_{i-1,j+1,k+1} + u_{i-1,j+1,k-1} + u_{i-1,j-1,k+1} - u_{i-1,j-1,k-1}}{8h_x h_y h_z} = 0
 \end{aligned}$$

Referencias

- [1] DDM Organization, *Proceedings of International Conferences on Domain Decomposition Methods*. <http://www.ddm.org> and <http://www.domaindecomposition.com>, 1988-2012.
- [2] Toselli, A., Widlund O. *Domain decomposition methods- Algorithms and theory*, Springer Series in Computational Mathematics, Springer- Verlag, Berlin, 2005, 450 p.
- [3] Quarteroni A., Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Clarendon Press Oxford, 1999.
- [4] Pavarino L. F., Toselli A. *Recent Developments in Domain Decomposition Methods*. Springer, 2003.
- [5] Herrera I., R. Yates, *Unified Multipliers-Free Theory of Dual Primal Domain Decomposition Methods*. NUMER.METH. PART D. E. 25(3): 552- 581, May 2009, (Published on line May 13, 2008) DOI 10.1002/num.20359.
- [6] Herrera I., Yates R. A., *The Multipliers-Free Domain Decomposition Methods*, NUMER. METH. PART D. E., 26(4): pp. 874-905, July 2010. (Published on line: 23 April 2009, DOI 10.1002/num.20462)
- [7] Herrera I., Yates R. A., *The multipliers-Free Dual Primal Domain Decomposition Methods for Nonsymmetric Matrices*. NUMER. METH. PART D. E. DOI 10.1002/num.20581 (Published on line April 28, 2010).
- [8] Herrera I., Carrillo-Ledesma A., Rosas-Medina A., *A Brief Overview of Non-Overlapping Domain Decomposition Methods*, Geofísica Internacional, Vol, 50, 4, October-December, 2011.
- [9] Herrera I., Rosas-Medina A. , *The Derived-Vector Space Framerwork and Four General purposes massively parallel DDM Algorithms*, Engineering Analysis with Boundary Elements, 2013, in press.
- [10] Glowinski R., Kuznetsov Y.A., *On the solution of the Dirichlet problem for linear elliptic operators by a distributed Lagrange multiplier method*. C. R. Acad. Sci. Paris Ser. I Math., 327(7):693-698, 1998.
- [11] Herrera I., Pinder G. *Mathematical Modeling in Science and Engineering: An Axiomatic Approach*, Wiley,2012,243 p.
- [12] Ortega-Arjona J.L. *Patterns for Parallel Software Design*, Wiley series in Software Design Patterns, 2010.
- [13] Atluri S.N. *The Meshless Method (MLPG) for Domain & Bie Discretizations*, CREST, Tech Science Press,2002,680 p.
- [14] Gottlieb D., Orzag S. *Numerical Analysis of Spectral Methods : Theory and Applications*, SIAM, Philadelphia, PA,1977.
- [15] Leighton F. T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers, 1992.
- [16] Morse F., *Practical Parallel Computing*, AP Professional, 1994.

- [17] H. Lawson, *Parallel processing in industrial real time applications*, Prentice Hall 1992.
- [18] M. Coffin, *Parallel programming- A new approach*, Prentice Hall, Englewood Cliffs, 1992.
- [19] Hoare C. A. R., *Communicating Sequential Processes*, Prentice-Hall, 1985.
- [20] Andrews G., *Concurrent Programming*, The Benjaming/Cummings Publishing Company, 1991.
- [21] Burns A., Davies G., *Concurrent Programming*, Addison Wesley, 1993.
- [22] Hwang K., *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.
- [23] Codenotti B., Leoncini M., *Introduction to Parallel Processing*, Addison Wesley, 1992.
- [24] Flynn M. J., *Some Computer Organizations and Their Effectiveness*, IEEE Transactions on Computers, C-21, No 9, September, 1972.
- [25] Kumar M., Aswatha R; et al. *Proceedings of International Conference on Advances in Computing*, Advances in Intelligent Systems and Computing, Vol. 174, 2012, 1178 p.
- [26] Marini L., Quarteroni A., *A relaxation procedure for domain decomposition methods using finite elements*. Numer. Math, 55(5):575-598, 1989.
- [27] Marini L., Quarteroni A., *Iterative procedure for domain decomposition methods: a finite element approach*. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Pèriaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 129-143, Philadelphia, PA, 1988. SIAM. Paris, France, January 7-9, 1987.
- [28] Funaro D., Quarteroni A., Zanoui P., *An iterative procedure with interface relaxation for domain decomposition methods*. SIAM J. Numer. Anal, 25(6):1213-1236, 1988.
- [29] Bramble J.H., Pasciak J.E., Schatz A.H., *An iterative method for elliptic problems on regions partitioned into substructures*. Math. Comp., 46(173):361-369, 1986.
- [30] Björstad P.E., Hvidsten A., *Iterative methods for substructured elasticity problems in structural analysis*. In Roland Glowinski, Gene H. Golub, Gerard A. Meurant, and Jacques Periaux, editors. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 301-312, Philadelphia, PA, 1988. SIAM. Paris, Prance, January 7-9, 1987.
- [31] Björstad P.E., Widlund O., *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*. SIAM J. Numer. Anal., 23(6):1093-1120, 1986.
- [32] Bourgat J., Glowinski R., Le Tallec P., Vidrascu M., *Variational formulation and algorithm for trace operator in domain decomposition calculations*. In Tony Chan, Roland Glowinski, Jacques Periaux, and Olof Widlund, editors. *Domain Decomposition Methods. Second International Symposium on Domain Decomposition Methods*, pages 3-16, Philadelphia, PA, 1989. SIAM. Los Angeles, California, January 14-16, 1988.
- [33] De Roeck Y., Le Tallec P., *Analysis and test of a local domain decomposition preconditioner*. In Roland Glowinski, Yuri A. Kuznetsov, Gerard A. Meurant, Jacques Periaux, and Olof Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 112-128, Philadelphia, PA, 1991. SIAM. Moscow, Russia, May 21-25, 1990.
- [34] Le Tallec P., De Roeck Y., Vidrascu M., *Domain decomposition methods for large linearly elliptic three-dimensional problems*. J. Comput. Appl. Math., 34(1):93-117, 1991.
- [35] Farhat C., Lesoinne M., Pierson K., *A scalable dual-primal domain decomposition method*. Numer. Linear Algebra Appl., 7(7-8):687-714, 2000.

-
- [36] Pierson K., *A family of domain decomposition methods for the massively parallel solution of computational mechanics problems*. PhD thesis, University of Colorado at Boulder, Aerospace Engineering, 2000.
- [37] Da Conceição, D. T. Jr., *Balancing Domain Decomposition Preconditioners for Non-symmetric Problems*. Instituto Nacional de Matemática pura e Aplicada, Agenda Nacional do Petróleo PRH-32, Rio de Janeiro, May. 9, 2006.
- [38] Yano M., Darmofal D.L., *BDDC preconditioning for high-order Galerkin Least-Squares methods using inexact solvers*. Computational Methods Appl. Mech. Engrg. 199 (2010), 2958-2969.
- [39] Toselli, A., *FETI Domain Decomposition Methods for Scalar Advection-Diffusion Problems*. Computational Methods Appl. Mech. Engrg. 190. (2001), 5759-5776.
- [40] Carrillo, A., *Métodos de Descomposición de Dominio en el Espacio de Vectores Derivados y su Implementación Computacional en Paralelo*. Tesis de Doctorado. Universidad Nacional Autónoma de México, México, 2012.