



**I
N
A
O
E**

Solución Numérica de Ecuaciones en Derivadas Parciales mediante Funciones Radiales

por

José Antonio Muñoz Gómez

Tesis sometida como requisito parcial para obtener el grado de

**DOCTOR EN CIENCIAS EN EL ÁREA DE CIENCIAS
COMPUTACIONALES**

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Septiembre - 2007

Tonantzintla, Puebla

Supervisada por:

Dr. Pedro González-Casanova Henríquez

Universidad Nacional Autónoma de México

Unidad de Investigación en Cómputo Aplicado – DGSCA

Dr. Gustavo Rodríguez Gómez

Instituto Nacional de Astrofísica, Óptica y Electrónica

Ciencias Computacionales

©INAOE 2007

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir y distribuir copias de esta tesis
en su totalidad o en partes.



Resumen

La presente tesis trata sobre la solución numérica de ecuaciones en derivadas parciales mediante funciones de base radial. En la primera parte investigamos el orden de convergencia en h - c para una ecuación dependiente del tiempo de tipo convección-difusión en una dimensión. Con base en la función radial multi-cuádrica, y con un esquema implícito y explícito en la discretización temporal, observamos una tasa de convergencia exponencial en h - c , en donde el coeficiente de la exponencial es reducido conforme incrementamos el número de Péclet. Adicionalmente, mostramos numéricamente que el valor óptimo del parámetro c decrece monotónicamente conforme el coeficiente de difusión es reducido.

En general, cuando utilizamos funciones de base radial para resolver ecuaciones en derivadas parciales, la matriz resultante es por lo general densa y mal condicionada. Por lo cual, el uso de métodos directos es aplicable solo a problemas de moderado tamaño. En la segunda parte de este trabajo abordamos dicho problema empleando descomposición de dominio con nodos distribuidos uniformemente. La estrategia propuesta es aplicada a un problema dependiente del tiempo en 2-dimensiones. Empleando un solo procesador, observamos una disminución lineal en el tiempo de procesamiento conforme incrementamos el número de particiones, por lo tanto, el esquema propuesto puede ser aplicado a problemas de gran escala con cúmulos de computadoras.

El incremento uniforme en la densidad de los nodos induce una disminución en el error de aproximación; sin embargo, aún con cúmulos de computadoras la estrategia de refinamiento global de nodos es un método computacionalmente ineficiente. En problemas en donde existen capas límite, zonas de alto gradiente o una gran variación espacial en la solución, es conveniente aproximar dichos problemas con un esquema de refinamiento local de nodos. La idea de refinamiento local, consiste en densificar el número de nodos en las regiones en donde se requiere de mayor exactitud. Con base en el error de interpolación local y

el esquema de celda \times celda, se obtiene un método eficiente para el esquema de refinamiento local con funciones de base radial. Este esquema es probado en distintas ecuaciones diferenciales parciales en una y dos dimensiones, mostrando la efectividad del método propuesto. El esquema desarrollado no requiere de una malla para el proceso de refinamiento y puede extenderse a tres o más dimensiones con fronteras complejas.

Finalmente, basándose en el esquema de refinamiento local de nodos (RLN) y el método de descomposición de dominio (MDD) se desarrolla un algoritmo que integra ambos métodos numéricos en donde se requiere MDD para disminuir la complejidad computacional y poder manejar miles de nodos, así como el RLN para aumentar el desempeño computacional. Obteniéndose así un método numérico libre de malla con el cual se pueden abordar problemas de gran escala. El esquema propuesto es aplicado a problemas particulares mostrando la efectividad del método.

Agradecimientos

Una tarea prodigiosa es el expresar en este papel inerte la latente gratitud y admiración que tengo hacia mis directores de Tesis. Conciente de tal proeza y aún más de mis limitaciones, solo quisiera expresar mis agradecimientos a ambos: Dr. Pedro González-Casanova Henríquez y Dr. Gustavo Rodríguez Gómez, por el apoyo constante brindado a lo largo de la investigación y en cuyas pláticas me sentí atraído como luciérnaga a la luz del fascinante mundo del Cómputo Científico. En particular a la Teoría de Aproximación y a los Métodos Numéricos para Ecuaciones en Derivadas Parciales.

Adicionalmente, quiero expresar mi agradecimiento a los investigadores del departamento en Ciencias Computacionales del INAOE, los cuales en distintas maneras influenciaron en mi el estudio de las Ciencias de la Computación.

Al Consejo Nacional de Ciencias y Tecnología (CONACyT) por el apoyo otorgado a través de la beca No. 145052, con lo cual fue posible realizar este trabajo de investigación. Adicionalmente, al proyecto CONACyT-2002-C01-4022 que apoyo el desarrollo de esta investigación en las primeras etapas.

Dedicatorias

A la memoria de mi madre Esperanza Gómez Ortega

Al recuerdo del amor de Rosa María Rocha Morales

Índice general

1. Interpolación con Funciones Radiales	1
1.1. Introducción	1
1.2. Planteamiento	5
1.2.1. Funciones Positivas Definidas	6
1.2.2. Funciones Condicionalmente Positivas Definidas	9
1.2.3. Funciones Radiales de Soporte Compacto	13
1.3. Espacios Nativos	15
1.3.1. Convergencia	18
1.3.2. Ejemplo de Espacio Nativo	23
1.4. Parámetro c	25
1.5. Miscelanea	26
2. Esquema de Colocación Asimétrico	29
2.1. Esquemas de Colocación	29
2.1.1. Colocación Asimétrica	30
2.1.2. Colocación Simétrica	32
2.2. Problemas Lineales Estacionarios	34
2.2.1. Problema de Poisson	35
2.3. Problemas Dependientes del Tiempo	37
2.3.1. Esquema Explícito e Implícito	40

Análisis de Estabilidad	42
2.3.2. Algoritmo	43
Análisis de Complejidad	44
2.3.3. Caso Particular: Convección-Difusión	46
2.4. Convergencia para Convección-Difusión	47
2.4.1. Convección-Difusión	47
2.4.2. Planteamiento	48
2.4.3. Método Implícito	49
2.4.4. Relación β vs. c	54
2.4.5. Método Explícito	56
2.4.6. Comparación Implícito y Explícito	59
3. Descomposición de Dominio	61
3.1. Introducción	61
3.2. Formulación Clásica	63
3.3. Lineal Temporal	67
3.3.1. Análisis de Complejidad	73
3.3.2. Caso Particular $P = 1$	75
3.3.3. Partición de Datos	77
3.4. Resultados Numéricos	78
3.4.1. Convección en $2-d$	79
3.4.2. Incremento en el Número de Subdominios	81
3.4.3. Diferencias Finitas	84
4. Refinamiento Local de Nodos	89
4.1. Motivación	90
4.2. Métodos Adaptivos	90

4.3.	Aproximación Local	94
4.3.1.	Indicador de Error Local	95
4.4.	Esquema de Celda×Celda	97
4.4.1.	Refinamiento de Celdas	98
4.4.2.	Restricción 2:1	101
4.4.3.	Estructura de Datos	101
4.5.	Esquema General	103
4.5.1.	Criterios de Paro	104
4.6.	Resultados Numéricos	105
4.6.1.	Ejemplo en 1- d	106
	Adaptivo vs. no-Adaptivo	107
4.6.2.	Poisson en 2- d	110
4.6.3.	Convección-Difusión en 1- d	114
5.	Refinamiento Local de Nodos y Descomposición de Dominio	117
5.1.	Motivación	117
5.2.	Partición de Datos	118
5.3.	Zonas de Traslape	119
5.3.1.	Descripción General	119
5.3.2.	Traslape en una Dimensión	120
5.3.3.	Traslape en dos Dimensiones	121
5.3.4.	Niveles de Expansión	125
5.3.5.	Actualización de la Frontera Artificial	126
5.4.	Esquema General	127
5.4.1.	Número de Particiones Adaptable	129
5.5.	Resultados Numéricos	130
5.5.1.	Poisson en 2- d	131

Partición Estática	132
Placa Delgada vs. Multicuadrado	133
Porcentaje de Refinamiento	135
5.5.2. Convección-Difusión en 2- d	140
Péclet 10^0	140
Péclet 10^1	141
Péclet 10^2	141
Péclet 10^3	143
6. Conclusiones y Aportaciones	149
6.1. Publicaciones	152
6.2. Trabajo Futuro	153
Apéndices	154
A. Número de Condición	155
B. Extensión del Algoritmo 3	159
C. Acrónimos	161
D. Símbolos	163

Prefacio

La solución numérica de ecuaciones en derivadas parciales de fenómenos físicos complejos, demanda una gran cantidad de recursos computacionales en términos de procesamiento y almacenamiento en memoria. En las últimas décadas ha habido un interés creciente en la investigación de métodos eficientes para esta clase de problemas en gran escala. La restricción principal de los métodos tradicionales, como diferencias finitas, volumen finito y elemento finito, es la generación de la malla. Los métodos libres de malla han emergido como una alternativa para abordar este clase de problemas.

Las funciones de base radial originalmente han sido utilizadas como una herramienta efectiva para la interpolación de datos no equiespaciados en \mathbb{R}^d , en la construcción del interpolante no se requiere de una malla. En la década pasada, a partir de las funciones de base radial se formuló un nuevo esquema numérico para aproximar ecuaciones en derivadas parciales, el cual está libre de la principal restricción que es la generación de la malla. Adicionalmente, para cierto tipo de problemas, numéricamente ha mostrado tener una tasa de convergencia superior a la de los métodos tradicionales; es decir, una tasa de convergencia exponencial.

En esta tesis, nos enfocamos en la aplicación de las funciones radiales para resolver ecuaciones en derivadas parciales con base en el método de colocación asimétrico. En particular, abordamos numéricamente la determinación de la tasa de convergencia para un problema de tipo convección-difusión dependiente del tiempo, y proponemos un nuevo método computacionalmente eficiente para problemas con altos gradientes; es decir, se plantea un nuevo esquema que combina el refinamiento local de nodos y el método de descomposición de dominio.

La tesis está organizada de la siguiente manera:

En el capítulo 1 se muestran los fundamentos teóricos de la interpolación con funciones de base radial.

En el capítulo 2 se describe en detalle el método de colocación asimétrico para problemas estacionarios y dependientes del tiempo. Se investiga numéricamente la influencia del número de Péclet en la tasa de convergencia para una ecuación diferencial parcial dependiente del tiempo de tipo convección-difusión.

En el capítulo 3 se formula la aplicación del método clásico de Schwarz a problemas lineales dependientes del tiempo con el esquema de colocación asimétrico. Se analiza, para una ecuación convectiva, la influencia del número de subdominios en el tiempo de procesamiento y en la exactitud numérica.

En el capítulo 4 se propone un esquema nuevo para el refinamiento local de nodos basado en el error de interpolación local. La efectividad del método se ejemplifica con un problema de tipo convección-difusión temporal y dos problemas elípticos.

En el capítulo 5 se describe la integración en un solo algoritmo del esquema de refinamiento local de nodos y el método de descomposición de dominio. El algoritmo propuesto, es el primer esquema en la literatura que aborda la solución numérica de ecuaciones en derivadas parciales estacionarias para datos no equiespaciados en dos dimensiones con descomposición de dominio y adaptividad local; bajo el esquema de colocación asimétrico. El esquema propuesto es aplicado a dos problemas con gradientes abruptos, para mostrar la efectividad del algoritmo propuesto.

Las conclusiones finales son mostradas en el capítulo 6.

Capítulo 1

Interpolación con Funciones Radiales

En este capítulo se muestran las bases teóricas del problema de interpolación multivariada con base en funciones de base radial. Lo anterior nos provee las bases para abordar el método numérico libre de malla denominado esquema de colocación asimétrico.

1.1. Introducción

Muchos procesos físicos de importancia práctica como la dispersión de contaminantes, terremotos, vientos solares, etc., tienen un común denominador: la física del problema puede ser aproximada por una ecuación diferencial parcial. La solución numérica de ecuaciones diferenciales parciales es considerada como uno de los grandes retos actuales en ciencias computacionales [1]. La simulación numérica requiere de la cooperación interdisciplinaria entre diversos campos científicos: computación, matemáticas e ingeniería, lo cual es necesario para ampliar perceptiblemente nuestra capacidad de reproducir y de predecir procesos físicos.

Las simulaciones numéricas proporcionan una herramienta alternativa en

la investigación científica, los experimentos costosos e incluso peligrosos, están siendo substituidos cada vez más por simulaciones en computadora; esto nos permite examinar procesos que no pueden ser probados experimentalmente. Las herramientas numéricas son a menudo más útiles que los métodos experimentales tradicionales, en términos de proporcionar información completa de todo el proceso, la cual que no puede ser medida, observada directamente o es difícil de adquirir vía otros medios. El desarrollo de productos nuevos está siendo acelerado por la eliminación de estos experimentos físicos costosos [2].

La simulación numérica desempeña un papel importante en la comprensión de los modelos teóricos, ofrece evidencia de la posible validez teórica de nuevos resultados, y provee un puente entre el modelo experimental y las predicciones teóricas.

Tradicionalmente, la simulación de fenómenos físicos complejos es realizada vía métodos numéricos como son: diferencias finitas, volumen finito y elemento finito. En estos métodos numéricos se requiere de una malla definida en una región espacial para construir el sistema algebraico de ecuaciones que corresponde a la discretización del problema analizado. La mayor limitación de los esquemas tradicionales radica en el proceso del mallado [3, 4].

El problema con la creación de mallas radica en que para geometrías complejas y especialmente en tres dimensiones hasta la fecha no se ha obtenido un esquema automático para la creación de la malla, que satisfaga los requerimientos necesarios como son: controlar el ángulo mínimo de la malla, anisotrópicos y recursos computacionales restringidos [5]. Aunado a ello, *la mayor dificultad en el mallado es extender los algoritmos a más dimensiones* [6]. Una pregunta que surge de forma natural es: ¿qué tipo de elemento genera el resultado numérico más exacto?, desafortunadamente no existe una respuesta única, en general la selección del elemento está en función del problema a resolver. El lector interesado en una amigable descripción y clasificación de métodos para la generación de mallas con y sin estructura puede consultar [7, 8]. Una extensa base de datos de artículos libres acerca de la generación de mallas se encuentra

en el sitio web de la “International Meshing Roundtable”.

Los métodos libres de malla han emergido en las últimas tres décadas como una alternativa para la solución numérica de ecuaciones en derivadas parciales [9, 10, 11, 12, 13]. El objetivo de esta nueva clase de métodos numéricos es construir la aproximación del problema analizado mediante un conjunto de nodos, en donde no se requiere una relación entre los nodos; es decir, no requieren de una malla. En problemas con largas deformaciones y distorsiones, los métodos libres de mallas son más apropiados [14]. Para capturar correctamente las deformaciones es necesario de la remoción e inserción de nodos o elementos, en los métodos tradicionales es necesario incluir estrategias sofisticadas para evitar la degeneración en los elementos de la malla.

A partir de la década pasada, se desarrollaron los esquemas numéricos libres de malla basados en las funciones de base radial. Inicialmente, las funciones de base radial fueron desarrolladas para problemas de interpolación con datos no equiespaciados en \mathbb{R}^d . Cabe resaltar que no se requiere de una malla para construir el interpolante, y que el problema de interpolación no crece en función de la dimensión espacial en donde están definidos los datos, lo cual ha motivado el desarrollo teórico y práctico para utilizar dicho interpolante en la solución numérica de ecuaciones en derivadas parciales. Dentro de los esquemas numéricos basados en funciones de base radial se tienen: método de colocación asimétrico [15, 16], esquema de colocación simétrico [17], método de soluciones fundamentales [18], esquema de Galerkin [19], método de reciprocidad dual [20, 21], método de cuadratura diferencial local [22], *boundary knot method* [23, 24] y método de interpolación puntual [25].

En esta investigación nos centraremos en el método de colocación asimétrico introducido por Kansa para la solución numérica de ecuaciones en derivadas parciales [15, 16]; esto se debe a su generalidad, su independencia de la dimensión espacial, que no requiere de integrales y que permite el reúso de código. Adicionalmente, para cierto tipo de funciones, ha mostrado tener una tasa de convergencia superior a la de los métodos tradicionales; es decir, una ta-

sa de convergencia exponencial. Por último, se ha aplicado con éxito en una gran variedad de problemas: transferencia de calor [26], convección-difusión [27, 28, 29, 30], valor inicial [31], ecuación no lineal de Burger [32], ecuaciones diferenciales elípticas [33], matemáticas financieras [34, 35], modelación de aguas de poca profundidad [36, 37, 38], campo superficial del viento [39], modelo mixto bifásico y trifásico [40, 41], flujo en cavidades (Navier Stokes) [42]. Por otra parte, con el esquema de colocación asimétrico no se ha demostrado la invertibilidad del sistema algebraico de ecuaciones, y presenta un mal número de condición conforme se incrementa el número de nodos o se incrementa el parámetro libre presente en distintas funciones radiales. Para reducir el mal condicionamiento se han desarrollado distintas estrategias: preconditionamiento, descomposición de dominio, parámetro c variable y optimización en la locación de los nodos. El uso de descomposición de dominio ha mostrado ser una herramienta efectiva en el manejo de grandes volúmenes de nodos, y en una disminución en el número de condición [43, 44].

En esta tesis nos enfocamos en la aplicación de las funciones radiales para resolver ecuaciones en derivadas parciales con base en el esquema de colocación asimétrico. En particular, abordamos numéricamente la determinación de la tasa de convergencia para un problema de tipo convección-difusión dependiente del tiempo, y proponemos un nuevo método computacionalmente eficiente para problemas con altos gradientes. Planteamos un nuevo esquema que combina el refinamiento local de nodos y el método de descomposición de dominio.

El par de temas principales que tratamos en la presente investigación no se han abordado en la literatura. Ambos constituyen un primer esfuerzo en el entendimiento y desarrollo del método de colocación asimétrico con funciones radiales para abordar problemas más complejos, con miles de datos no cartesianos, así como la determinación del orden de convergencia del modelo simplificado (convección-difusión) de las ecuaciones de Navier-Stokes en dinámica de fluidos.

1.2. Planteamiento

Dado un conjunto de puntos $X = \{x_1, x_2, \dots, x_N\}$ con $x_i \in \mathbb{R}^d$ y sus correspondientes valores $f(x_1), f(x_2), \dots, f(x_N)$ con $f(x_i) \in \mathbb{R}$ siendo $f : \mathbb{R}^d \rightarrow \mathbb{R}$. El problema de interpolación multivariada con nodos distribuidos de manera no uniforme está definido como el determinar una función continua o suficientemente diferenciable $s : \mathbb{R}^d \rightarrow \mathbb{R}$ tal que

$$s(x_i) = f(x_i), \quad i = 1, 2, \dots, N. \quad (1.1)$$

La determinación del interpolante s no es sólo útil para construir superficies de datos provenientes de mediciones físicas. Es posible que la función f sea muy costosa de evaluar, en cuyo caso se requiere tener un interpolante que sea menos costoso de evaluar. Otra aplicación es compresión de datos en donde el tamaño de los datos originales exceden la capacidad de almacenamiento en disco duro. En este caso queremos seleccionar un subconjunto de datos de tal forma que con base al interpolante podamos reconstruir el resto de los valores.

Una manera conveniente para resolver el problema anterior es suponer que la función s es una combinación lineal de funciones de base radial

$$s(x) = \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|), \quad x \in \mathbb{R}^d, \quad (1.2)$$

donde $\lambda = [\lambda_1, \dots, \lambda_N]^T$ es el vector de incógnitas, ϕ corresponde a la función radial y $\|\cdot\|$ denota la norma Euclidiana.

El sistema matricial resultante de (1.2) queda expresado como

$$\begin{bmatrix} \phi(\|x_1 - x_1\|) & \phi(\|x_1 - x_2\|) & \dots & \phi(\|x_1 - x_N\|) \\ \phi(\|x_2 - x_1\|) & \phi(\|x_2 - x_2\|) & \dots & \phi(\|x_2 - x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_N - x_1\|) & \phi(\|x_N - x_2\|) & \dots & \phi(\|x_N - x_N\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix},$$

escribiéndolo en una forma compacta

$$A\lambda = f, \quad (1.3)$$

donde $A_{i,j} = \{\phi(\|x_i - x_j\|)\}_{1 \leq i,j \leq N}$ en $\mathbb{R}^{N \times N}$ y $f, \lambda \in \mathbb{R}^N$. Para determinar la solución del sistema lineal de ecuaciones (1.3) requerimos que la matriz A sea no singular.

Los siguientes teoremas y definiciones, mostrados en este capítulo, están basados principalmente en un trabajo de Fasshauer [45], unas notas de A. Iske de la escuela de verano Europea en 2001, y en el único libro a la fecha de funciones de base radial de Buhmann [46].

La teoría subyacente en la interpolación multivariada con funciones de base radial puede ser dividida en dos grandes ramas: el enfoque algebraico y el enfoque Hilbertiano. El primero aborda la determinación de la invertibilidad del sistema lineal de ecuaciones. El segundo, por otra parte, trata la estimación de errores (tasa de convergencia) para los distintos núcleos radiales con base en el espacio nativo asociado. Ambos enfoques son necesarios y complementarios; dada una función radial determinaremos bajo que condiciones el sistema lineal de ecuaciones es invertible, y además cual es la tasa de convergencia asociada a dicha función.

1.2.1. Funciones Positivas Definidas

En esta sección estamos interesados en determinar para que tipo de funciones radiales el sistema lineal de ecuaciones (1.3) es no singular. Primero definiremos las condiciones para que el sistema lineal de ecuaciones sea invertible o no singular.

Definición 1. *Una matriz real simétrica A es llamada positiva semi-definida si su asociada forma cuadrática es no negativa*

$$\sum_{j=1}^N \sum_{k=1}^N \lambda_j \lambda_k A_{j,k} \geq 0, \quad (1.4)$$

para $\lambda = [\lambda_1, \dots, \lambda_N]^T \in \mathbb{R}^N$. Si el único vector λ que satisface la igualdad en (1.4) es el vector cero, entonces A es llamada positiva definida.

Una característica importante de las matrices positivas definidas es que todos los valores propios son positivos. El determinante de una matriz positiva definida es positivo y la matriz es invertible o no singular. Sí A en (1.3) es positiva definida, entonces podemos determinar la solución del sistema lineal de ecuaciones

$$\lambda^* = A^{-1}f,$$

donde A^{-1} representa la inversa de la matriz A . La matriz es generada por la evaluación de la función radial en los distintos puntos, para establecer el vínculo entre la función radial y el sistema lineal de ecuaciones primero definiremos cuando una función es (estrictamente) positiva definida y posteriormente radial.

Definición 2. *Una función continua ϕ que toma valores en los reales es positiva definida en \mathbb{R}^d si y sólo si es una función par y es no negativa*

$$\sum_{j=1}^N \sum_{k=1}^N \lambda_j \lambda_k \phi(x_j - x_k) \geq 0,$$

para cualesquiera N distintos puntos $x_1, \dots, x_N \in \mathbb{R}^d$, y $\lambda = [\lambda_1, \dots, \lambda_N]^T \in \mathbb{R}^N$. La función ϕ es estrictamente positiva definida en \mathbb{R}^d si el único vector λ que satisface la igualdad es el vector cero.

En la definición anterior entenderemos por función par a una función tal que para cualquier x en el dominio de la función f se satisface que $f(-x) = f(x)$. Adicionalmente, una función f es impar si para cualquier x en el dominio de f se satisface $f(-x) = -f(x)$.

Definición 3. *Una función $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ es llamada radial si existe una función univariada $\varphi : [0, \infty) \rightarrow \mathbb{R}$, tal que $\phi(x) = \varphi(r)$, donde $r = \|x\|$ y $\|\cdot\|$ es alguna norma en \mathbb{R}^d - usualmente la norma Euclidiana.*

La definición anterior de función radial indica que el problema de interpolación con dichas funciones es insensible a la dimensión del espacio d en donde se encuentren los datos. En lugar de trabajar con una función multivariada ϕ cuya complejidad aumenta con la dimensión del espacio, podemos trabajar con

la misma función univariada φ para todas las opciones de d . Esto no sucede con las funciones radiales de soporte compacto, ver sección 1.2.3. Para definir cuando ϕ es estrictamente positiva definida, primero definiremos la monotonicidad completa.

Definición 4. Una función $\varphi : [0, \infty) \rightarrow \mathbb{R}$ que es $C[0, \infty) \cap C^\infty(0, \infty)$, y que satisface

$$(-1)^\ell \varphi^\ell(r) \geq 0, \quad r > 0, \quad \ell = 0, 1, 2, \dots$$

es llamada completamente monótona en $[0, \infty)$.

Como es usual, $C[0, \infty)$ denota a las funciones que son continuas en dicho intervalo $[0, \infty)$. Entenderemos por $C^\infty(0, \infty)$ a las funciones que tienen derivadas de cualquier orden en el abierto $(0, \infty)$. El caso C^ℓ indica que tiene derivadas continuas hasta el orden $\ell \in \mathbb{N}$.

El siguiente teorema de Schoenberg demostrado en 1938 nos indica bajo que circunstancias el sistema algebraico de ecuaciones es invertible.

Teorema 5. Si la función $\varphi : [0, \infty) \rightarrow \mathbb{R}$ es completamente monótona pero no constante, entonces $\varphi(\|\cdot\|^2)$ es estrictamente positiva y radial en \mathbb{R}^d para cualquier d .

Observe que se empleó el cuadrado de la norma, lo cual es diferente a la definición 3. El teorema anterior nos provee una manera directa para determinar cuando la matriz A es no singular, a partir de la completa monotonicidad de la función φ . Por lo tanto, las funciones que satisfagan las condiciones del teorema 5 las podemos utilizar para interpolar datos no equiespaciados en \mathbb{R}^d .

Ejemplos de funciones completamente monótonas pero no constantes son: $\varphi(r) = (r + c^2)^{-\beta}$ con $c, \beta > 0$ y $\varphi(r) = e^{-cr}$ con $c > 0$. Con base en dichas funciones φ , el interpolante (1.2) adquiere la forma particular

$$s(x) = \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|^2 + c^2)^{-\beta} \quad \text{y} \quad s(x) = \sum_{j=1}^N \lambda_j e^{-c\|x - x_j\|}, \quad x \in \mathbb{R}^d,$$

los cuales reciben el nombre de interpolante con la función multicuádrica inversa e interpolante con la función Gaussiana respectivamente.

1.2.2. Funciones Condicionalmente Positivas Definidas

En el problema de interpolación con funciones radiales existe una clase de funciones, por ejemplo placa delgada (ver tabla 1.1), que requieren la inclusión de un polinomio

$$s(x) = \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|) + \sum_{k=1}^M \mu_k p_k(x), \quad x \in \mathbb{R}^d, \quad (1.5)$$

donde p_1, \dots, p_M forman una base $M = \binom{d+m-1}{m-1}$ - dimensional en el espacio lineal π_{m-1}^d de polinomios de grado total menor o igual a $m-1$ en d variables: $p \in \pi_{m-1}^d$.

Los coeficientes de la función s en (1.5) son definidos de manera única por el sistema

$$\left. \begin{aligned} s(x_i) &= f(x_i), & i &= 1, \dots, N \\ \sum_{j=1}^N \lambda_j p_k(x_j) &= 0, & k &= 1, \dots, M \end{aligned} \right\}. \quad (1.6)$$

La determinación de los vectores de incógnitas $\lambda \in \mathbb{R}^N$ y $\mu \in \mathbb{R}^M$ del problema de interpolación (1.6) se representa de forma matricial como

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} f \\ \bar{0} \end{bmatrix}, \quad (1.7)$$

donde $A \in \mathbb{R}^{N \times N}$ con $A_{i,j} = \{\phi(\|x_i - x_j\|)\}_{1 \leq i,j \leq N}$, $f \in \mathbb{R}^N$, $P \in \mathbb{R}^{N \times M}$, $\bar{0}$ es un vector de ceros de longitud M , y 0 es una matriz de ceros de dimensión $M \times M$. Por lo tanto, la dimensión de la matriz de interpolación es $N + M \times N + M$. La representación matricial de P es

$$P = \begin{bmatrix} p_1(x_1) & \dots & p_M(x_1) \\ \vdots & & \vdots \\ p_1(x_N) & \dots & p_M(x_N) \end{bmatrix}. \quad (1.8)$$

En analogía a la definición 2, requerimos definir las funciones condicionalmente positiva definida y estrictamente condicional positiva definida de orden m .

Definición 6. Una función continua ϕ par y que toma valores en los reales es llamada condicional positiva definida de orden m en \mathbb{R}^d si es no negativa

$$\sum_{j=1}^N \sum_{k=1}^N \lambda_j \lambda_k \phi(x_j - x_k) \geq 0, \quad (1.9)$$

para cualesquiera N puntos $x_1, \dots, x_n \in \mathbb{R}^d$, y $\lambda = [\lambda_1, \dots, \lambda_N]^T \in \mathbb{R}^N$ satisfaciendo

$$\sum_{j=1}^N \lambda_j q(x_j) = 0, \quad \forall q \in \pi_{m-1}^d. \quad (1.10)$$

La función ϕ es llamada estrictamente condicional positiva definida de orden m en \mathbb{R}^d si los $x_1, \dots, x_n \in \mathbb{R}^d$ son distintos, y $\lambda \neq 0$ implica que (1.9) sea mayor a cero.

Para determinar la no singularidad de (1.7) primero definimos la unisolvencia de un conjunto de puntos en \mathbb{R}^d , la cual se debe satisfacer dependiendo del grado del polinomio.

Definición 7. Sea $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, y sea $p \in \pi_{m-1}^d$ cualquier polinomio a lo más de grado $m-1$. El conjunto X es llamado unisolvente si al interpolar datos con valor de cero el polinomio que los interpola es el polinomio cero

$$p \in \pi_{m-1}^d \quad y \quad p(x_i) = 0, \quad i = 1, \dots, N \quad \implies \quad p \equiv 0.$$

Por ejemplo, decimos que X es 0-unisolvente para π_0^d si todos los puntos son distintos, es 1-unisolvente en π_1^2 si contiene 3 puntos no colineales, es 2-unisolvente en π_2^2 si contiene 6 puntos que no satisfacen una forma cónica.

En 1986 Michelli [47] proveyó un criterio suficiente para las funciones condicionales positivas de orden m , esto generaliza el resultado de Schoenberg para las funciones positivas definidas radiales [48, 49]. Adicionalmente, Michelli conjetura la necesidad de este criterio. Esto fue finalmente mostrado por K. Guo et al. [50]. Los siguiente dos teoremas engloban los principales resultados de los citados trabajos.

Teorema 8. Sea $\varphi \in C[0, \infty) \cap C^\infty(0, \infty)$. La función $\phi = \varphi(\|\cdot\|^2)$ es condicional positiva definida de orden m y radial en \mathbb{R}^d para toda d , si y sólo si $(-1)^m \varphi^m$ es completamente monótona en $(0, \infty)$.

Teorema 9. Si φ es de la forma del teorema 8 y no un polinomio de grado a lo más m , entonces ϕ es estrictamente condicional positiva de orden m y radial en \mathbb{R}^d para toda dimensión del espacio d .

A partir de los teoremas 5 y 9, mostramos en la tabla 1.1 una selección de las funciones de base radial $\phi(r)$ más frecuentemente utilizadas en la literatura. Adicionalmente, indicamos el grado del polinomio requerido en las funciones condicionales positivas definidas, así como el rango de los parámetros adicionales.

Nombre	$\phi(r)$	Parámetros	Polinomio
Multicuádrico	$(-1)^{\lceil \beta/2 \rceil} (r^2 + c^2)^{\beta/2}$	$\beta > 0, \beta \notin 2\mathbb{N}$	$\lceil \beta \rceil$
Placa Delgada	$(-1)^{1+\beta/2} r^\beta \log r$	$\beta > 0, \beta \in 2\mathbb{N}$	$1 + \beta/2$
Potencias	$(-1)^{\lceil \beta/2 \rceil} r^\beta$	$\beta > 0, \beta \notin 2\mathbb{N}$	$\lceil \beta/2 \rceil$
Gausiana	$e^{-r^2 c}$	$c > 0$	0
Inverso Multicuádrico	$(r^2 + c^2)^{-\beta/2}$	$\beta > 0, \beta \notin 2\mathbb{N}$	0
Wendland	$(1 - r)_+^\beta p(r)$	$\beta > 0$	0

Tabla 1.1: Funciones de base radial frecuentemente utilizadas y grado del polinomio.

Existe una gran variedad de métodos con los cuales podemos interpolar datos no equiespaciados en \mathbb{R}^d . R. Franke, en un trabajo relevante realizado en la década de los 80's [51], comparó 29 métodos de interpolación para datos en dos dimensiones, determinando experimentalmente que la interpolación con la función radial multicuádrica $\phi(r) = (r^2 + c^2)^{1/2}$, sugerida por Hardy [52], generó los resultados más exactos. Adicionalmente, con la función placa delgada $r^2 \log r$ se obtienen resultados similares a los obtenidos con la función multicuádrica, en términos de calidad visual y suavidad en la reconstrucción, con la ganancia

de no requerir el ajuste de un parámetro. Este trabajo motivó la investigación teórica del núcleo multicuádrico para determinar la invertibilidad del sistema, así como su tasa de convergencia. La función radial multicuádrica requiere de la adición de una constante para garantizar la invertibilidad del sistema lineal de ecuaciones

$$s(x) = \sum_{j=1}^N \lambda_j \sqrt{\|x - x_j\|^2 + c^2} + \mu_1, \quad (1.11)$$

Michelli [47] demuestra que la interpolación con una función estrictamente condicional positiva definida de orden uno es posible realizarla sin la inclusión del polinomio constante.

Teorema 10. *Suponga que ϕ es estrictamente condicional positiva de orden 1 y que $\phi(0) \leq 0$. Para cualesquiera puntos distintos $x_1, \dots, x_N \in \mathbb{R}^d$ la matriz A con entradas $A_{i,j} = \phi(x_i - x_j)$ tiene $N - 1$ eigenvalores positivos y 1 negativo, y por lo tanto es no singular.*

Este resultado generaliza el resultado de Schoenberg para el caso $m = 1$. Por lo tanto, en la práctica cuando se trabaja con el núcleo multicuádrico (1.11) no agregamos la constante, numéricamente se ha observado que esto no afecta la exactitud en la interpolación.

Es conveniente recordar que el polinomio requerido ($m > 0$) es necesario para garantizar la invertibilidad del sistema lineal de ecuaciones. En la tabla 1.1 se incluyó la función de soporte compacto de Wendland, la cual es tratada en la sección siguiente. Para la función de placa delgada $r^2 \log r$ en dos dimensiones el polinomio requerido es bivariado lineal $p \in \pi_1^2$. Usando la notación $\bar{x} = (x, y) \in \mathbb{R}^2$, la base de este espacio de polinomios π_1^2 es $\{1, x, y\}$. El interpolante (1.5) tiene la forma particular

$$s(\bar{x}) = \sum_{j=1}^N \lambda_j r_j^2 \log(r_j) + \mu_1 + \mu_2 x + \mu_3 y, \quad \bar{x} \in \mathbb{R}^2,$$

con las condiciones

$$\sum_{k=1}^N \lambda_k = \sum_{k=1}^N \lambda_k x_k = \sum_{k=1}^N \lambda_k y_k = 0,$$

obteniendo un sistema lineal de ecuaciones de dimensiones $N + 3 \times N + 3$, el cual es invertible. Para la función radial $\phi(r) = r^4 \log r$ la base del polinomio requerido es $\{1, x, y, xy, x^2, y^2\}$, y por lo tanto en la construcción de $s(x)$ se requiere agregar 6 condiciones adicionales.

1.2.3. Funciones Radiales de Soporte Compacto

En esta sección se aborda el tema de funciones radiales de soporte compacto. El objetivo es mostrar las condiciones para las cuales las funciones son de soporte compacto, estrictamente positivas definidas y radiales en \mathbb{R}^d para una dimensión del espacio d fijo.

En la literatura existe una diversidad de funciones radiales de soporte compacto [53, 54, 55], centraremos nuestra atención en el trabajo desarrollado por Wendland el cual presenta la primera familia de funciones radiales de soporte compacto.

Las funciones radiales de soporte compacto están definidas en el intervalo $[0, 1]$ fuera de dicho intervalo toman un valor constante cero. Pueden ser escaladas para tener un soporte compacto en el intervalo $[0, \delta]$ lo cual se logra substituyendo r por r/δ para $\delta > 0$. El factor de escalamiento δ puede ser constante o puede ser variable en los distintos centros. La determinación del valor óptimo δ , con lo cual se obtenga la mayor exactitud numérica, es una pregunta de investigación actual. El siguiente teorema proporciona las condiciones para la construcción de la función radial de soporte compacto.

Teorema 11. *Las funciones $\phi_{d,k}$ son estrictamente positivas definidas y radiales en \mathbb{R}^d y son de la forma*

$$\phi_{d,k} = \begin{cases} p_{d,k}(r), & r \in [0, 1], \\ 0, & r > 1, \end{cases}$$

con un polinomio univariado $p_{d,k}$ de grado $\lfloor d/2 \rfloor + 3k + 1$. Adicionalmente, $\phi_{d,k} \in C^{2k}(\mathbb{R})$ es única con excepción de un factor constante, y el grado del polinomio es mínimo con respecto de la dimensión del espacio d y diferenciabilidad $2k$.

Las funciones de soporte compacto de Wendland se pueden escribir de forma compacta

$$\Phi_{d,k} = (1 - r)_+^\ell p(r), \quad \text{para } k \geq 0,$$

donde $\ell = \lfloor d/2 \rfloor + k + 1$ denota exponenciación, $(\cdot)_+$ es la llamada función de corte determinada por: $(x)_+ = x$, para $x > 0$, y $(x)_+ = 0$, para $x \leq 0$. $p(r)$ es un polinomio, $2k$ corresponde al grado de diferenciabilidad de la función, $r = \|x - x_j\|$ con $x, x_j \in \mathbb{R}^d$.

Fijando una dimensión del espacio d (impar) se tiene una familia de funciones radiales de soporte compacto con distintos grados de diferenciabilidad, por ejemplo para $d = 1, 3$ se obtiene

$$d = 1 \quad \phi_{1,0}(r) = (1 - r)_+ \quad C^0$$

$$\phi_{1,1}(r) = (1 - r)_+^3 (3r - r) \quad C^2$$

$$\phi_{1,2}(r) = (1 - r)_+^5 (8r^3 + 5r + 1) \quad C^4$$

...

$$d = 3 \quad \phi_{3,0}(r) = (1 - r)_+^2 \quad C^0$$

$$\phi_{3,1}(r) = (1 - r)_+^4 (4r - r) \quad C^2$$

$$\phi_{3,2}(r) = (1 - r)_+^6 (35r^2 + 18r + 3) \quad C^4$$

...

El interpolante con las funciones de soporte compacto de Wendland queda definido como

$$s(x) = \sum_{j=1}^N \lambda_j \phi_{d,k} (\|x - x_j\|). \quad (1.12)$$

En las secciones anteriores se abordaron las funciones radiales de soporte global que generan un sistema algebraico denso, las funciones de soporte compacto generan un sistema esparcido de ecuaciones que puede ser resuelto de manera eficiente. Sin embargo, el factor de escalamiento δ degrada la exactitud en la interpolación.

1.3. Espacios Nativos

En esta sección definiremos el espacio nativo para las funciones estrictamente positivas definidas. En particular, asumiremos que contamos con un núcleo K y mostraremos la construcción del núcleo reproductor en espacios de Hilbert H_K tal que K es el núcleo reproductor en H .

Hacemos notar que todo espacio vectorial V está definido sobre un campo \mathbb{K} . Cuando el campo \mathbb{K} es el conjunto de los números reales \mathbb{R} se dice que V es un espacio vectorial real. Si \mathbb{K} es el conjunto de los números complejos \mathbb{C} diremos que V es un espacio vectorial complejo [56].

El producto interior en un espacio vectorial real V es una función que asocia a cada par elementos en V un número real. Es decir, es una función de la forma $\langle \cdot, \cdot \rangle: V \times V \rightarrow \mathbb{R}$ de manera que se verifican las siguientes propiedades:

- a) $\langle x, y \rangle = \langle y, x \rangle$
- b) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$
- c) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$
- d) $\langle x, x \rangle \geq 0$
- e) $\langle x, x \rangle = 0$ solamente cuando $x = 0$,

para todo $x, y, z \in V$ y $\alpha \in \mathbb{R}$.

Un espacio vectorial V cuando está provisto de un producto interior se denomina espacio vectorial con producto interior o preHilbert y se simboliza $(V, \langle \cdot, \cdot \rangle)$. Una consecuencia importante es que todo espacio con producto interior puede normarse [57]

$$\|x\| = \langle x, x \rangle^{1/2}, \quad x \in V,$$

por lo tanto, $(V, \langle \cdot, \cdot \rangle)$ también puede ser considerado como espacio normado con la norma natural inducida por el producto interior.

Un espacio métrico en donde cualquier sucesión de Cauchy converge es llamado *completo* [58]. Cuando el espacio normado es completo, se dice que es un

espacio de Hilbert, los parientes más próximos a los espacios euclídeos, y lo denotaremos con la letra \mathcal{H} . El producto interno en dicho espacio lo denotaremos como $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

A partir del espacio de Hilbert definiremos el núcleo reproductor K asociado a \mathcal{H} :

Definición 12. Sea \mathcal{H} un espacio vectorial real de Hilbert de las funciones continuas $f : \Omega \rightarrow \mathbb{R}$. Una función $K : \Omega \times \Omega \rightarrow \mathbb{R}$ es llamada núcleo reproductor para \mathcal{H} sí

1. $K(x, \cdot) \in \mathcal{H} \quad \forall x \in \Omega,$
2. $f(x) = \langle f, K(\cdot, x) \rangle_{\mathcal{H}} \quad \forall f \in \mathcal{H} \text{ y } \forall x \in \Omega.$

El núcleo reproductor K de un espacio de Hilbert es único y se satisface: $K(x, y) = K(y, x)$, $K(x, y) = \langle K(x, \cdot), K(\cdot, y) \rangle$ para todo $x, y \in \Omega$. Adicionalmente, la convergencia en un espacio de Hilbert normado implica convergencia puntual. La propiedad 2 es llamada propiedad reproductora (Aronszajn). La existencia del núcleo reproductor es equivalente a que para toda $f \in \mathcal{H}$ la evaluación puntual de f denotada por $\delta_x f := f(x)$ está acotada por funcionales lineales, es decir, sí existe $M = \|K_x\|_H$ tal que

$$|\delta_x f| = |\langle f, K(\cdot, x) \rangle| = |\langle f, K_x \rangle| \leq M \|f\|_H,$$

donde $\|\cdot\|_H$ denota la norma en el espacio de funciones de Hilbert. En los siguientes párrafos se muestra la construcción del espacio nativo para las funciones estrictamente positivas definidas.

De la definición de núcleo reproductor se deduce que \mathcal{H} contiene todas las funciones de la forma

$$\sum_{j=1}^N c_j K(x_j, \cdot), \quad x_j \in \Omega. \quad (1.13)$$

Sea f una función en \mathcal{H} , entonces el producto interior es la expresión

$$\langle f, f \rangle_{\mathcal{H}} = \sum_{j=1}^N \sum_{k=1}^N c_j c_k K(x_j, x_k).$$

Por lo tanto, definimos a $H_K(\Omega)$ como el espacio lineal generado por todas las funciones de la forma (1.13)

$$H_K(\Omega) = \text{gen}\{K(\cdot, y) : y \in \Omega\},$$

el cual tiene una forma bilineal asociada. Sea s y u cualquier función en $H_K(\Omega)$

$$s = \sum_{j=1}^N c_j K(x_j, \cdot), \quad u = \sum_{k=1}^N d_k K(y_k, \cdot),$$

la forma bilineal satisface

$$\langle s, u \rangle_K = \langle u, s \rangle_K, \quad \langle s, u \rangle_K = \sum_{j=1}^N \sum_{k=1}^N c_j d_k K(x_j, y_k).$$

A partir de los resultados anteriores, ahora podemos enunciar el siguiente resultado:

Teorema 13. *Sea $\Omega \subset \mathbb{R}^d$ un abierto y acotado. Si $K : \Omega \times \Omega \rightarrow \mathbb{R}$ es un núcleo simétrico y estrictamente positivo definido, entonces la forma bilineal $\langle \cdot, \cdot \rangle_K$ define un producto interior en $H_K(\Omega)$. Adicionalmente, $H_K(\Omega)$ es un espacio preHilbertiano que tiene como núcleo reproductor a K .*

Finalmente, el espacio nativo $\mathcal{N}_K(\Omega)$ de K lo definiremos como la completitud de $H_K(\Omega)$ con respecto de la K -norma $\|\cdot\|_K$ de donde $\|f\|_K = \|f\|_{\mathcal{N}_K(\Omega)}$ para toda función $f \in H_K(\Omega)$. El teorema 13 se puede generalizar para el espacio nativo de las funciones (radiales condicional) positivas definidas, dicha construcción se sale del ámbito de esta tesis y por ello no lo incluimos, ver [59, 60].

Ahora, determinaremos el espacio nativo, estudiado por Madych & Nelson, para las funciones estrictamente positivas definidas para el caso especial $\Omega = \mathbb{R}^d$ y a ϕ de la forma $\phi(x - y) = K(x, y)$, dicha caracterización del espacio nativo está expresada en términos de la transformada de Fourier.

Teorema 14. *Suponga que $\phi \in C(\mathbb{R}^d) \cap L_1(\mathbb{R}^d)$ es una función que toma valores en los reales estrictamente positiva definida. Definimos*

$$\mathcal{H}_\phi(\mathbb{R}^d) = \left\{ f \in L_2(\mathbb{R}^d) \cap C(\mathbb{R}^d) : \frac{\hat{f}}{\sqrt{\hat{\phi}}} \in L_2(\mathbb{R}^d) \right\},$$

y este espacio está equipado con la forma bilineal

$$\langle f, g \rangle_{\mathcal{H}_\phi(\mathbb{R}^d)} = \frac{1}{\sqrt{(2\pi)^d}} \left\langle \frac{\hat{f}}{\sqrt{\hat{\phi}}}, \frac{\hat{g}}{\sqrt{\hat{\phi}}} \right\rangle_{L_2(\mathbb{R}^d)} = \frac{1}{\sqrt{(2\pi)^d}} \int_{\mathbb{R}^d} \frac{\hat{f}(\omega) \overline{\hat{g}(\omega)}}{\hat{\phi}(\omega)} d\omega.$$

$\mathcal{H}_\phi(\mathbb{R}^d)$ es un espacio de Hilbert real con producto interno $\langle \cdot, \cdot \rangle_{\mathcal{H}_\phi(\mathbb{R}^d)}$ y núcleo reproductor $\phi(\cdot - \cdot)$. $\mathcal{H}_\phi(\mathbb{R}^d) = \mathcal{N}_\phi(\mathbb{R}^d)$ es el espacio nativo de ϕ con la norma canónica inducida por el producto interior.

En el teorema anterior, $\hat{f}(\omega)$ denota la transformada de fourier de $f \in L_1(\mathbb{R}^d)$, la barra superior en $\hat{g}(\omega)$ denota la conjugación compleja. Adicionalmente, el espacio de las funciones integrables requeridos son:

$$L_p(\mathbb{R}^d) = \left\{ f : \mathbb{R}^d \rightarrow \mathbb{C} \mid \int_{\mathbb{R}^d} |f(x)|^p dx < \infty \right\},$$

con $p = \{1, 2\}$. Si $f(x) \in L_1(\mathbb{R}^d)$, entonces la función $f(x)$ es absolutamente integrable, si $f(x) \in L_2(\mathbb{R}^d)$, entonces $f(x)$ es cuadrado integrable. En ambos casos la integral es finita.

El objetivo de esta sección fue la construcción del espacio nativo con base en la teoría de núcleos reproductores en el espacio de Hilbert. Esto es requerido posteriormente para poder definir la convergencia del interpolante.

1.3.1. Convergencia

Como es usual en el estudio de los métodos de aproximación, uno de los temas centrales es el análisis de convergencia. En esta sección determinaremos la estimación del error de aproximación para las funciones radiales estrictamente (condicional) positivas definidas.

En lo subsecuente es conveniente trabajar con la representación de Lagrange para el interpolante s , propuesta por Wu & Schaback [61]. Asumiendo que ϕ es estrictamente positiva definida, la forma cardinal del interpolante es

$$s(x) = \sum_{j=1}^N f(x_j) v_j^*(x), \quad x \in \mathbb{R}^d, \quad (1.14)$$

donde las funciones base de Lagrange $u_1^*(x), \dots, u_N^*(x)$ satisfacen

$$v_j^*(x_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} \quad 1 \leq j, k \leq N,$$

y se cumple la condición de interpolación $s(x_i) = f(x_i)$ para $i = 1, \dots, N$.

Dado un x fijo en \mathbb{R}^d el vector $v^*(x) = [v_1^*(x), \dots, v_N^*(x)]^T$ es la solución única del sistema lineal

$$Av^*(x) = b(x),$$

donde $A_{i,j} = \{\phi(\|x_i - x_j\|)\}_{1 \leq i, j \leq N}$ y $b(x) = [\phi(\|x - x_1\|), \dots, \phi(\|x - x_N\|)]^T$.

La extensión a funciones condicional positivas definidas de orden m se describe en el trabajo citado.

El otro elemento que requerimos para determinar la cota de error entre el interpolante y los datos es la llamada función potencia. Para ello consideremos que $\phi \in C(\Omega \times \Omega)$ con $\Omega \subset \mathbb{R}^d$, y para cualquier conjunto de puntos distintos $X = \{x_1, \dots, x_N\}$ contenidos en Ω , y para cualquier vector $v \in \mathbb{R}^N$, definimos la forma cuadrática

$$Q(v) = \phi(x, x) - 2 \sum_{j=1}^N v_j \phi(x, x_j) + \sum_{i=1}^N \sum_{j=1}^N v_i v_j \phi(x_i, x_j),$$

la cual es equivalente a la norma inducida por el producto interno en el espacio nativo $\mathcal{N}_\phi(\Omega)$

$$Q(v) = \|\phi(\cdot, x) - \sum_{j=1}^N v_j \phi(\cdot, x_j)\|_{\mathcal{N}_\phi(\Omega)}^2.$$

Ahora podemos definir la función potencia como:

Definición 15. *Asuma que $\Omega \subseteq \mathbb{R}^d$ y $\phi \in C(\Omega \times \Omega)$ es una función estrictamente positiva definida en \mathbb{R}^d . Para cualesquiera puntos distintos $X = \{x_1, \dots, x_N\} \subseteq \Omega$ la función potencia está definida por*

$$[\mathcal{P}_{\phi, X}(x)]^2 = Q(v^*(x)),$$

donde v^* es el vector de funciones cardinales.

La cota de error genérica entre el interpolante y los datos de la función está definida en términos del siguiente resultado.

Teorema 16. *Sea $\Omega \subset \mathbb{R}^d$, $\phi \in C(\Omega \times \Omega)$ una función estrictamente positiva definida en \mathbb{R}^d , y suponga que el conjunto de puntos $X = \{x_1, \dots, x_N\}$ son todos distintos. Denotando el interpolante $s(x)$ para $f \in \mathcal{N}_\phi(\Omega)$ en X , entonces para cualquier $x \in \Omega$*

$$|f(x) - s(x)| \leq \mathcal{P}_{\phi, X}(x) \|f\|_{\mathcal{N}_\phi(\Omega)}. \quad (1.15)$$

Un beneficio de este tipo de cota radica en que el error de interpolación está dividido en dos partes: a) la suavidad de los datos medidos en términos del espacio nativo normado $\|f\|_{\mathcal{N}_\phi(\Omega)}$, esto es independiente de la distribución de los centros, b) la contribución de la función ϕ seleccionada y de la distribución de los datos, lo cual está contenido en la función potencia la cual es independiente de los valores de f .

La cota de error encontrada es análoga a la estimación del error en la interpolación polinomial [62]

$$|f(x) - \mathcal{I}f(x)| \leq \mathcal{P}(x) \max_{x \in I} |f^n(x)|,$$

en donde la función $\mathcal{P}_{\phi, X}(x) \simeq \mathcal{P}(x)$ es de la forma

$$\mathcal{P}(x) = \frac{1}{n!} \prod_{j=1}^n |x - x_j|.$$

El acotamiento del residual $e = |f(x) - s(x)|$ está en función de la distribución de los nodos X , la función radial ϕ y la función a interpolar f . Para medir la distribución en los nodos distribuidos de manera no uniforme $X = \{x_1, \dots, x_N\}$, requerimos de la llamada distancia de relleno (*fill distance*), la cual está definida por

$$h = h_{X, \Omega} = \sup_{x \in \Omega} \min_{x_i \in X} \|x - x_i\|_2,$$

la cual mide la manera cómo los datos llenan el dominio, que corresponde a la distancia máxima de cualquier punto del dominio a su punto de referencia

más cercano. También conocida como el círculo contenido en X con el menor radio entre dos puntos cualesquiera. El siguiente paso es refinar la cota de error (1.15) para determinar como influye la distribución de los nodos X en términos de la distancia de relleno.

Teorema 17. *Suponga que $\Omega \subset \mathbb{R}^d$ es abierto y acotado y satisface la condición de cono interior. Suponga que $\phi \in C^{2k}(\Omega \times \Omega)$ es simétrica y estrictamente condicional positiva definida de orden m en \mathbb{R}^d . Denotando por s al interpolante de $f \in \mathcal{N}_\phi(\Omega)$ satisfaciendo $(m-1)$ -unisolvante en X . Para un valor fijo $\alpha \in \mathbb{N}_0^d$ con $|\alpha| \leq k$. Entonces existe una constante positiva h_0 y \hbar (independiente de x, f y ϕ) tal que*

$$|D^\alpha f(x) - D^\alpha s(x)| \leq \hbar \cdot \hbar_\phi(x)^{1/2} h_{X,\phi}^{k-|\alpha|} |f|_{\mathcal{N}_\phi(\Omega)},$$

donde $h_{X,\Omega} \leq h_0$ y D^α denota la derivada generalizada de orden α .

El teorema anterior es genérico, debido a que no toma en cuenta explícitamente la función ϕ usada en la interpolación. Dependiendo de la función radial ϕ , y bajo los supuestos anteriores es posible obtener una estimación del orden de convergencia en términos de la distancia de relleno (*fill distance*) y la función radial seleccionada.

Aplicando el teorema 17 a las funciones tales como: multicuádrico, inverso multicuádrico y Gaussiana, se obtiene una tasa de convergencia arbitrariamente alta. Es decir, para todo $\ell \in \mathbb{N}$ y $|\alpha| \leq \ell$ tenemos

$$|D^\alpha f(x) - D^\alpha s(x)| \leq \hbar_\ell \cdot h^{\ell-|\alpha|} |f|_{\mathcal{N}_\phi(\Omega)}.$$

Uno de los objetivos es definir la dependencia de la constante \hbar_ℓ para poder dar una cota de error más refinada que la mostrada en el teorema 17. En la tabla 1.2 mostramos la cota de error final de varias funciones radiales ϕ .

En la tabla 1.2 el valor de δ es positivo y es independiente del valor de la distancia de relleno h (*fill distance*). Como se observa en dicha tabla, para las funciones (inverso) multicuádrico y Gaussiana se tiene un orden de convergencia

Nombre	$\phi(r)$	L_∞ Cota de la función $P_{\phi,X}$
Multicuádrico	$(-1)^{\lceil\beta/2\rceil}(r^2 + c^2)^{\beta/2}$	$e^{-\delta/h}$
Gausiana	e^{-r^2c}	$e^{-\delta/h^2}$
Placa Delgada	$(-1)^{1+\beta/2}r^\beta \log r$	$h^{\beta/2}$
Potencias	$(-1)^{\lceil\beta/2\rceil}r^\beta$	$h^{\beta/2}$
Wendland	$(1 - r)_+^\beta (1 + 4r)$	$h^{3/2}$

Tabla 1.2: L_∞ Cotas de la función $P_{\phi,X}$ basados en el interpolante en la forma de Lagrange.

exponencial el cual es muy superior al orden de convergencia algebraico del resto de las funciones mostradas. La determinación de este orden de convergencia fue realizada por Madych & Nelson [63], el cual puede expresarse como $O(\lambda^{1/h})$ con $0 < \lambda < 1$. Para la función Gausiana, Wendland [64] refinó la cota a $O(\lambda^{\sqrt{1/h}})$. El orden de convergencia exponencial es la razón por lo cual en el estudio numérico realizado por Franke a principios de los 80's, concluye que con base al núcleo multicuádrico se obtiene la mayor exactitud numérica. Este resultado fue previo a la demostración de convergencia exponencial realizada por Madych & Nelson. Las funciones multicuádrica y Gausiana comparten el mismo orden de convergencia exponencial, sin embargo en la práctica se selecciona el núcleo multicuádrico debido a que la selección del parámetro es menos sensible. Adicionalmente, Madych [65] estimó una cota de error que incorpora el parámetro c para el núcleo multicuádrico

$$O(e^{ac}\lambda^{c/h}), \quad (1.16)$$

con $a > 0$ y $0 < \lambda < 1$. La implicación de este orden de convergencia es que existen dos maneras de reducir el error en la aproximación; manteniendo fijo c e incrementando el número de nodos para reducir h , o manteniendo fijo h e incrementar el parámetro c . Por lo tanto, es posible utilizar un esquema $h - c$ para reducir el error en la aproximación. La diferencia entre disminuir h o aumentar c radica en que al incrementar el número de nodos de manera uniforme se ob-

tiene un sistema de ecuaciones cada vez más grande y el costo computacional se incrementa considerablemente; con el incremento de c no sucede esto. Sin embargo, el precio que hay que pagar es usualmente el mal condicionamiento del sistema lineal de ecuaciones (1.3). Existe una dicotomía: conforme el error en la aproximación disminuye se incrementa el número de condición, sí uno busca un algoritmo estable se obtiene un error en la aproximación comparativamente mayor. Esto es conocido como principio de incertidumbre [66]. En el apéndice A se describe el número de condición. La selección del valor óptimo c de tal forma que se obtenga la mayor exactitud numérica en la interpolación es una pregunta de investigación abierta y de gran interés.

1.3.2. Ejemplo de Espacio Nativo

En esta sección mostraremos que la selección de la función ϕ puede ser vista como el núcleo reproductor del espacio nativo asociado. En particular para el espacio nativo (Beppo-Levi) de los splines poliarmónicos.

En el trabajo pionero de Duchon, el interpolante (función radial) está determinado a partir del espacio nativo ya construido [67, 68, 69]. La teoría para construir el espacio nativo a partir de la función radial fue desarrollada posteriormente, donde los autores principales son Madych & Nelson [70, 71] y R. Schaback [59, 60]. Observe que en el teorema 14 se construye el espacio nativo a partir de la función seleccionada.

Para poder describir, a grandes rasgos, el resultado de Duchon, primero definiremos la clase de funciones llamados splines poliarmónicos, los cuales están definidos como

$$\phi(r) = \begin{cases} r^{2k-d} \log r & \text{si } d \text{ es par} \\ r^{2k-d} & \text{si } d \text{ es impar} \end{cases}, \quad (1.17)$$

donde k debe satisfacer $2k > d$. Seleccionando $d = k = 2$ obtenemos la función radial placa delgada en dos dimensiones $\phi(r) = r^2 \log r$, el análogo al spline cúbico en una dimensión, y corresponde a la solución fundamental de la

ecuación biarmónica

$$\Delta\Delta\phi(\|x\|) = c\delta_x.$$

Dado un conjunto de puntos diferentes en \mathbb{R}^2 y no colineales, el interpolante de placa delgada es una función que minimiza la energía de deflexión de una placa en dos dimensiones. Esta energía de deflexión está definida como

$$I(u) = \int_{\mathbb{R}^2} \left(\frac{\partial^2 u}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 u}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 u}{\partial y^2} \right)^2 dx, \quad (1.18)$$

cuyo núcleo es el polinomio lineal en dos dimensiones, la selección de $u = u(x, y)$ está definida en el espacio Beppo-Levi

$$BL^2(\mathbb{R}^2) = \left\{ u \in C(\mathbb{R}^2) : \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2} \in L^2(\mathbb{R}^2) \right\},$$

satisfaciendo que u pase por los puntos a interpolar

$$s(x_i) = f(x_i), \quad i = 1, 2, \dots, n.$$

La solución del problema anterior es la función placa delgada en dos dimensiones más el polinomio lineal

$$r^2 \log r + a_0 + a_1 \xi_1 + a_2 \xi_2.$$

De manera similar a lo anterior, el interpolante s formado con las funciones radiales del tipo splines poliarmónicos -con el respectivo polinomio- minimiza la siguiente función de energía

$$|u|_{BL^k(\mathbb{R}^d)}^2 = \int_{\mathbb{R}^d} \sum_{|\alpha|=k} \binom{k}{\alpha} (D^\alpha u)^2 dx, \quad \binom{k}{\alpha} = \frac{k!}{\alpha_1! \cdots \alpha_d!}, \quad (1.19)$$

donde la selección de la función u está en el espacio Beppo-Levi

$$BL^k(\mathbb{R}^d) = \{u \in C(\mathbb{R}^d) : D^\alpha u \in L^2(\mathbb{R}^d) \forall |\alpha| = k\} \subset C(\mathbb{R}^d),$$

satisfaciendo que pase el interpolante buscado en los puntos a interpolar $u|_X = s|_X$. El espacio Beppo-Levi $BL^k(\mathbb{R}^d)$ está equipado con la semi-norma $|\cdot|_{BL^k(\mathbb{R}^d)}$ que tiene como espacio nulo a los polinomios P_k^d .

1.4. Parámetro c

El parámetro c requerido en las funciones radiales (inverso) multicuádrico y Gaussiana, está en función del número de puntos N , su distribución espacial X , la función radial ϕ , y la función f a interpolar. En la literatura se han propuesto distintos enfoques para seleccionar c , principalmente para el núcleo multicuádrico y datos en dos dimensiones. Hardy utiliza $c = 0.815d$ donde $d = (1/N) \sum_{i=1}^N d_i$ y d_i es la distancia entre el i -ésimo punto y el vecino más cercano [52]. Franke reemplaza d por D/N donde D es el diámetro del círculo mínimo conteniendo todos los datos y emplea $c = 1.25D/\sqrt{N}$ [51]. Rippa propone un algoritmo, basado en las ideas del método de validación cruzada utilizado ampliamente en estadística, para seleccionar el parámetro c en las funciones (inverso) multicuádrico y Gaussiana, una revisión de otros esquemas iterativos para determinar c se encuentra en el trabajo de Rippa [72].

En el contexto de la solución numérica de ecuaciones en derivadas parciales, Kansa et al. sugiere que el parámetro c dependa de la locación de los nodos

$$c_i^2 = a^2 \left(\frac{b^2}{a^2} \right)^{\frac{i-1}{N-1}},$$

donde i corresponde al índice del punto en la construcción de $\phi(\|x_i, \cdot\|)$, y a^2, b^2 son proporcionados por el usuario, $b > a$ [73]. Hon et al. también hace dependiente la c en función de la posición de los nodos

$$c_j = M_j + b, \quad j = 1, \dots, N,$$

donde M y b son seleccionados por el usuario [32]. Ling et al. utiliza $c^2 = 1/N$ en unión con el método de descomposición de dominio [74]. Cheng et al. con base en un problema elíptico, observa numéricamente que el error residual tomado en una muestra de los datos, se puede utilizar como un indicador de error para optimizar la selección de c . Adicionalmente, determinan numéricamente un orden de convergencia exponencial $\sim O(\lambda^{\sqrt{c}/h})$ en $h - c$, con nodos distribuidos uniformemente en dos dimensiones empleando la función multi-

cuádrica inversa [75]. El enfoque de emplear el error residual para seleccionar c fue posteriormente aplicado a problemas dependientes del tiempo [30].

El parámetro c juega un papel determinante en la aproximación para problemas de interpolación, así como en la solución numérica de ecuaciones diferenciales ordinarias y ecuaciones en derivadas parciales. El siguiente capítulo tiene como uno de sus objetivos el determinar numéricamente el orden de convergencia exponencial, con la función multicuádrica, para una ecuación diferencial parcial del tipo convección-difusión, con base en el esquema de colocación asimétrico.

1.5. Miscelanea

Para valores grandes de N , la memoria y el tiempo requeridos para construir y resolver el sistema lineal de ecuaciones representan un cuello de botella. La solución directa del sistema lineal de ecuaciones tiene una complejidad computacional $O(N^3)$ en tiempo y $O(N^2)$ en espacio. El costo de evaluar N puntos en el interpolante tiene una complejidad $O(N^2)$. En la solución del sistema lineal mediante métodos iterativos se requiere de una multiplicación matriz-vector $A\lambda$ con un costo $O(N^2)$ en tiempo.

Para reducir el costo de la multiplicación matriz-vector, cercano a $O(N \log N)$, se han desarrollado distintos esquemas con base en el método de multipolos rápidos. El principal autor en esta línea es Beatson y junto con sus colaboradores han reportado los siguientes trabajos: para la función placa delgada en dos dimensiones [76], multicuádrico para datos en \mathbb{R}^d [77], splines poliarmónicos en \mathbb{R}^ℓ , $\ell = 2, 3, 4$ [78, 79, 80]. En aplicaciones prácticas se tiene una distribución no equiespaciada de mediciones y lo que se requiere es interpolar a una superficie en una rejilla equiespaciada con m datos siendo $m \gg N$. El reducir el costo de la evaluación en los m nodos a un orden cercano a $O(m \log N)$ tiene un gran impacto en el tiempo requerido y hace factible la interpolación para grandes volúmenes de datos. Sin embargo, todavía tenemos el problema de re-

resolver el sistema de ecuaciones que es por lo general mal condicionado. Para reducir el mal número de condición el sistema debe ser preconditionado, y por consecuencia esto incrementará el desempeño computacional en los métodos iterativos [81].

La solución iterativa, basada en métodos de Krylov, ha sido abordada con base en preconditionadores usando funciones cardinales en un subconjunto de puntos de interpolación. Una revisión de los métodos de Krylov para la interpolación con funciones radiales se encuentra en el trabajo de Faul & Powell [82], el interés práctico de este esquema radica en que el número de iteraciones requeridas para converger es del orden de decenas. Varias maneras en la construcción de las funciones cardinales se han propuesto: selección de un subconjunto de puntos cercanos para cada nodo, una combinación de puntos cercanos y lejanos, y relajando la condición de la delta de Dirac mediante funciones cardinales con un decrecimiento cúbico [83, 84, 85]. La combinación de los métodos anteriores es requerida para abordar problemas con millones de datos.

Capítulo 2

Esquema de Colocación

Asimétrico

En este capítulo se describe en detalle el método de colocación asimétrico aplicado a ecuaciones en derivadas parciales estacionarias y dependientes del tiempo. Por medio de una ecuación de tipo convección-difusión analizamos numéricamente la tasa de convergencia del esquema propuesto a partir de la función radial multicuádrada. La influencia del número de Péclet en la tasa de convergencia es estudiada para valores de Péclet moderados encontrándose que su comportamiento es exponencial. Se observó, además, que el incremento en el número de Péclet produce una reducción en el coeficiente que determina el orden de convergencia exponencial; ambos resultados fueron determinados para un esquema implícito y explícito. En esta investigación, numéricamente se muestra que el valor óptimo del parámetro c decrece monótonicamente conforme el coeficiente difusivo es reducido.

2.1. Esquemas de Colocación

En esta sección se describe el método de colocación asimétrico (MCA) y el esquema de colocación simétrico (MCS). El objetivo es mostrar que a pesar de

que con el MCS se garantiza la no singularidad del sistema algebraico, en la práctica es preferible el uso del método de colocación asimétrico. En la siguiente sección se expone en detalle el esquema de colocación asimétrico.

2.1.1. Colocación Asimétrica

En 1990 E. J. Kansa [15, 16] propone un nuevo esquema numérico para aproximar la solución de ecuaciones en derivadas parciales. Este método está basado en la teoría de interpolación con funciones de base radial, donde la idea principal radica en que el interpolante con funciones radiales nos permite aproximar las derivadas parciales. Con fines de claridad, se muestra este esquema de colocación aplicado a un problema estacionario lineal.

Considere el siguiente problema de valor en la frontera

$$\begin{cases} \mathcal{L}u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega \subset \mathbb{R}^d \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \end{cases}, \quad (2.1)$$

donde $\partial\Omega$ denota la frontera del dominio Ω y \mathcal{L} es un operador diferencial lineal. Las funciones $f, g : \mathbb{R}^d \rightarrow \mathbb{R}$ se asume que son conocidas. En los esquemas de colocación simétrico y asimétrico se considera que el dominio Ω es discretizado mediante N distintos nodos de colocación, los cuales son divididos en nodos pertenecientes a la frontera $X_f = \{x_1, \dots, x_{N_f}\} \in \partial\Omega$ y los nodos que conforman el dominio Ω del problema analizado $X_I = \{x_{N_f+1}, \dots, x_N\}$, donde $N_I = N - N_f$. En la figura 2.1 se muestra dicha distribución de los nodos.

En general no es posible determinar la solución analítica de ecuaciones en derivadas parciales, es por ello la necesidad de aproximar numéricamente la solución exacta $u(\mathbf{x})$ de la ecuación diferencial parcial (2.1). Para determinar la solución aproximada $\tilde{u}(\mathbf{x})$ hacemos uso del interpolante radial (ver sección 1.2), donde $\tilde{u}(\mathbf{x})$ puede ser representada como una combinación lineal de funciones radiales

$$\tilde{u}(\mathbf{x}) = \sum_{j=1}^N \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \mathbb{R}^d, \quad (2.2)$$

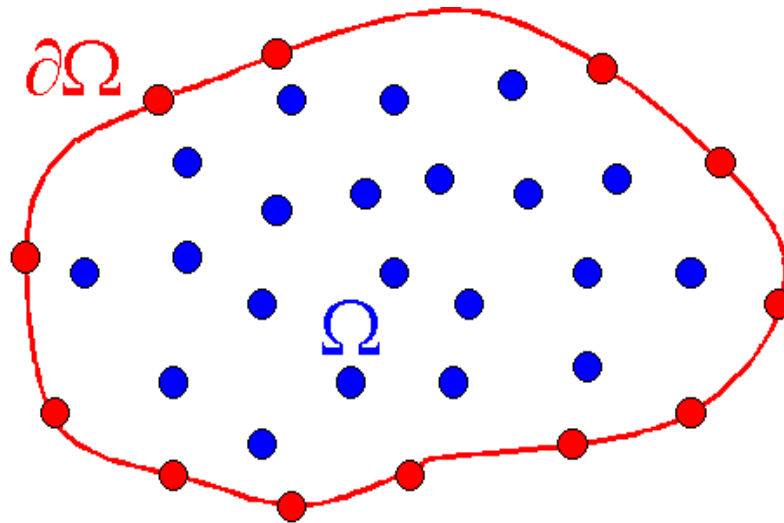


Figura 2.1: Distribución de los nodos en Ω y sobre la frontera $\partial\Omega$.

donde ϕ es una función radial, $\|\cdot\|$ es la distancia Euclidiana entre el par de vectores $\mathbf{x}, \mathbf{x}_j \in \mathbb{R}^d$ y $\lambda = [\lambda_1, \dots, \lambda_N]^T$ es el vector de coeficientes a ser determinado. Substituyendo la aproximación radial $\tilde{u}(\mathbf{x})$ en el problema (2.1) se obtiene

$$\begin{cases} \mathcal{L}\tilde{u}(\mathbf{x}_i) = \sum_{j=1}^N \lambda_j \mathcal{L}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = f(\mathbf{x}_i), & \mathbf{x}_i \in N_I, \\ \tilde{u}(\mathbf{x}_i) = \sum_{j=1}^N \lambda_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = g(\mathbf{x}_i), & \mathbf{x}_i \in N_f. \end{cases}$$

Escribiendo de una forma matricial el sistema anterior

$$\begin{bmatrix} \mathcal{L}\Phi \\ \Phi \end{bmatrix} \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (2.3)$$

obtenemos finalmente un sistema lineal de ecuaciones de $N \times N$ cuya incógnita es el vector $\lambda \in \mathbb{R}^d$ a ser determinado. El método descrito es llamado *colocación asimétrica* debido a la asimetría de la matriz resultante en la discretización $[\mathcal{L}\Phi \ \Phi]^T$. La solución del sistema anterior se puede expresar como

$$\lambda = \begin{bmatrix} \mathcal{L}\Phi \\ \Phi \end{bmatrix}^{-1} \begin{bmatrix} f \\ g \end{bmatrix}.$$

Con base en la solución obtenida λ podemos reconstruir la solución aproximada $\tilde{u}(\mathbf{x})$ mediante (2.2) de manera puntual en todo el dominio Ω . La facilidad

de codificación hacen de este método muy atractivo, debido a que el código programado para problemas en una dimensión se puede reutilizar en problemas de dimensiones superiores.

Para las funciones estrictamente positivas definidas o estrictamente condicional positivas definidas de orden m , sabemos que el sistema algebraico es no singular, ver teoremas 5 y 9. Por otra parte, la no singularidad del sistema lineal de ecuaciones (2.3) es un problema abierto y constituye uno de los mayores retos actuales.

2.1.2. Colocación Simétrica

En 1975 R. Hardy menciona la posibilidad de utilizar el núcleo multicuadrático en la interpolación de Hermite. Este problema fue desarrollado en 1992 por Wu [86] para la interpolación Hermite-Birkhoff con datos no equiespaciados en \mathbb{R}^d . La aplicación en la solución numérica de ecuaciones en derivadas es mostrada en [17], en este enfoque se considera que la solución aproximada $\tilde{u}(\mathbf{x})$ del problema (2.1) puede ser expresada como

$$\tilde{u}(\mathbf{x}) = \sum_{j=1}^{N_f} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{j=N_f+1}^N \lambda_j \overline{\mathcal{L}}\phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad (2.4)$$

donde $\overline{\mathcal{L}}$ denota la aplicación del operador lineal diferencial sobre el segundo argumento en la función $\phi(\cdot)$. Cuando se omite la barra superior en \mathcal{L} , el operador es aplicado con base en el primer argumento en $\phi(\cdot)$. Substituyendo (2.4) en el problema (2.1) se obtiene

$$\tilde{u}(\mathbf{x}_i) = \sum_{j=1}^{N_f} \lambda_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) + \sum_{j=N_f+1}^N \lambda_j \overline{\mathcal{L}}[\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)] = g(\mathbf{x}_i), \quad \mathbf{x}_i \in N_f,$$

$$\mathcal{L}\tilde{u}(\mathbf{x}_i) = \sum_{j=1}^{N_f} \lambda_j \mathcal{L}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) + \sum_{j=N_f+1}^N \lambda_j \mathcal{L}[\overline{\mathcal{L}}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)] = f(\mathbf{x}_i), \quad \mathbf{x}_i \in N_I.$$

Escribiendo de una forma matricial el sistema anterior obtenemos finalmente

el sistema lineal de ecuaciones cuya incógnita es el vector $\lambda \in \mathbb{R}^d$

$$\begin{bmatrix} \Phi & \bar{\mathcal{L}}\Phi \\ \mathcal{L}\Phi & \mathcal{L}[\bar{\mathcal{L}}\Phi] \end{bmatrix} \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \end{bmatrix}. \quad (2.5)$$

El sistema resultante es de la misma dimensión que lo obtenido con el método de colocación asimétrico. Este método es llamado *colocación simétrica* debido a la simetría en la matriz (2.5), excepto quizás por un signo diferente en $\bar{\mathcal{L}}\Phi$ con respecto de $\mathcal{L}\Phi$. La matriz obtenida es del mismo tipo que la obtenida con la interpolación de Hermite y por lo tanto el sistema es no singular.

Comparando el método de colocación simétrico y asimétrico en [87] muestran que el método de colocación simétrica es ligeramente superior al método de colocación asimétrico, para una ecuación de tipo Poisson en dos dimensiones. Adicionalmente, el número de condición en el MCS es por lo general menor al número condición obtenido con el MCA. Resaltan el hecho de que con el MCA se requieren menos derivadas y es más sencillo de implementar en comparación con el MCS. Posteriormente, con base en el núcleo multicuádrico y analizando 5 casos de prueba para una misma ecuación de Poisson [88], muestran que para el primer caso analizado el MCS supera en dos órdenes de magnitud al MCA, esto se debe a que el condicionamiento del MCS es mejor y permite ampliar el rango del parámetro c con lo cual se obtiene una mejor aproximación. Adicionalmente, con base en la integración de contorno Padé [89], concluyen que para los 5 casos analizados, cerca del óptimo existe poca diferencia entre ambos métodos de colocación. Recientemente, una comparación para una ecuación de tipo convección-difusión dependiente del tiempo muestra que el esquema simétrico es ligeramente superior al esquema asimétrico [30].

En la práctica, hasta donde tenemos conocimiento, el método de colocación asimétrico es preferible sobre el método de colocación simétrico [90]. Fasshauer quien es uno de los autores del esquema simétrico, concluye que el método asimétrico es preferible sobre el método simétrico a pesar de que no se ha demostrado la no singularidad del MCA [17].

2.2. Problemas Lineales Estacionarios

En esta sección se describe el método de colocación asimétrico, también llamado método de Kansa [15, 16], aplicado a ecuaciones en derivadas parciales lineales no dependientes del tiempo. Primero se expone el esquema general de aproximación, posteriormente con fines de claridad se describe la estructura de la matriz resultante de la discretización para un problema en particular.

Considere el siguiente problema de valor en la frontera

$$\begin{cases} \mathcal{L}u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega \subset \mathbb{R}^d \\ \mathcal{B}u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \end{cases}, \quad (2.6)$$

donde $\partial\Omega$ denota la frontera del dominio Ω , \mathcal{L} es un operador diferencial lineal y \mathcal{B} denota la aplicación de las condiciones de frontera de tipo Dirichlet, Neumann o tipo Robin. Las funciones $f, g : \mathbb{R}^d \rightarrow \mathbb{R}$ se asume que son conocidas.

En el esquema de colocación asimétrico se considera que el dominio Ω es discretizado con N distintos nodos de colocación, los cuales son divididos en nodos interiores $X_I = \{x_1, \dots, x_{N_I}\}$ y los nodos que conforman la frontera $X_f = \{x_{N_I+1}, \dots, x_N\} \in \partial\Omega$, con $N_f = N - N_I$. La solución $u(\mathbf{x})$ del problema (2.6) es aproximada mediante el esquema de interpolación con funciones radiales $\tilde{u}(\mathbf{x})$ mediante (1.5), substituyendo $\tilde{u}(\mathbf{x})$ en (2.6) se obtiene

$$\begin{cases} \mathcal{L}\tilde{u}(\mathbf{x}) = \sum_{j=1}^N \lambda_j \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{k=1}^M \mu_k \mathcal{L}p_k(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \in N_I, \\ \mathcal{B}\tilde{u}(\mathbf{x}) = \sum_{j=1}^N \lambda_j \mathcal{B}\phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{k=1}^M \mu_k \mathcal{B}p_k(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in N_f, \end{cases} \quad (2.7)$$

donde el polinomio requerido depende de la función radial utilizada, ver capítulo anterior. Observe que ambos operadores $\{\mathcal{L}, \mathcal{B}\}$ se aplican sobre la función radial ϕ y sobre el polinomio $p(x)$. Es decir, las derivadas parciales se obtienen derivando la función radial y el polinomio. Escribiendo de una forma matricial el sistema anterior

$$\begin{bmatrix} \mathcal{L}\Phi & \mathcal{L}P \\ \mathcal{B}\Phi & \mathcal{B}P \\ \mathcal{L}P^T & \mathcal{B}P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}, \quad (2.8)$$

donde $\mathcal{L}\Phi \in \mathbb{R}^{N_I \times N}$, $\mathcal{L}P \in \mathbb{R}^{N_I \times M}$, $\mathcal{B}\Phi \in \mathbb{R}^{N_f \times N}$, $\mathcal{B}P \in \mathbb{R}^{N_f \times M}$, $\lambda \in \mathbb{R}^N$, $\mu \in \mathbb{R}^M$, $f \in \mathbb{R}^{N_I}$ y $g \in \mathbb{R}^{N_f}$. El sistema lineal de ecuaciones anterior es de la forma $Ax = b$ con $A \in \mathbb{R}^{\bar{N} \times \bar{N}}$ y $x, b \in \mathbb{R}^{\bar{N}}$ con $\bar{N} = N + M$, el cual puede ser resuelto mediante una factorización LU para determinar las incógnitas. Los vectores de coeficientes obtenidos $\{\lambda, \mu\}$ son utilizados en la ecuación de interpolación (1.5) para reconstruir finalmente la solución aproximada $\tilde{u}(\mathbf{x})$.

En [90] muestran que para cierta configuración de nodos aleatorios el sistema es singular, sin embargo esto representa un caso muy particular. La contribución del trabajo citado es dar evidencia numérica de que una demostración general para la invertibilidad del esquema de colocación asimétrico no es posible. La existencia de casos particulares de singularidad no es una limitante seria para abandonar este esquema numérico, por el contrario, constituye un reto teórico y práctico el poder detectar estos casos patológicos.

La forma del sistema lineal de ecuaciones (2.8) se percibe complicada, para esclarecer este hecho en la siguiente sección se obtendrán las matrices resultantes de la discretización para un problema particular.

2.2.1. Problema de Poisson

Con fin de esclarecer los detalles omitidos en la sección anterior, se muestra la construcción de las matrices para la ecuación de Poisson en dos dimensiones

$$\begin{cases} \frac{\partial^2 u(\mathbf{x})}{\partial x^2} + \frac{\partial^2 u(\mathbf{x})}{\partial y^2} = f(\mathbf{x}), & \mathbf{x} \in \Omega \subset \mathbb{R}^2 \\ u(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \partial\Omega \end{cases}, \quad (2.9)$$

con $\mathbf{x} = (x, y)$. Para mantener la notación simple se consideró una frontera tipo Dirichlet. El problema (2.9) lo podemos enunciar como: dado los valores en la frontera y conociendo las relaciones (derivadas parciales) que deben de satisfacerse en el dominio Ω , queremos determinar la función desconocida $u(\mathbf{x})$ de tal forma que satisfaga las dos restricciones anteriores. Como se mencionó en la sección anterior, el dominio Ω es discretizado mediante N distintos nodos de colocación distribuidos en nodos al interior y los nodos que conforman la

frontera del problema: $X = X_I \cup X_f$ con $N = N_I + N_f$.

Para determinar la solución aproximada $\tilde{u}(\mathbf{x})$ del problema (2.9), hacemos uso del interpolante radial (2.2). Substituyendo (2.2) en (2.9) se obtiene el siguiente sistema lineal de ecuaciones

$$\begin{bmatrix} \mathcal{L}\Phi_a \\ \Phi_b \end{bmatrix} \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (2.10)$$

donde $\Phi_a \in \mathbb{R}^{N_I \times N}$, $\Phi_b \in \mathbb{R}^{N_f \times N}$, $\lambda \in \mathbb{R}^N$, $f = f(\mathbf{x})$ y $g = g(\mathbf{x})$. El operador \mathcal{L} es llamado Laplaciano, que es un operador diferencial elíptico de segundo orden $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Observe que la matriz Φ_b es una submatriz de la matriz de Gram $A = [\Phi_a \ \Phi_b]^T \in \mathbb{R}^{N \times N}$, donde $A_{i,j} = \{\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)\}_{1 \leq i, j \leq N}$. La solución del sistema lineal de ecuaciones anterior queda definida como

$$\lambda = \begin{bmatrix} \mathcal{L}\Phi_a \\ \Phi_b \end{bmatrix}^{-1} \begin{bmatrix} f \\ g \end{bmatrix}. \quad (2.11)$$

Observe que este esquema de discretización es independiente del espacio. El ejemplo mostrado es en dos dimensiones $d = 2$, de manera similar para $d \gg 2$ se obtiene un sistema lineal de ecuaciones.

La representación matricial del Laplaciano es expresada como

$$\nabla^2 \Phi_a = \frac{\partial^2 \Phi_a}{\partial x^2} + \frac{\partial^2 \Phi_a}{\partial y^2},$$

para obtener cada componente de la ecuación anterior se requiere obtener las segundas derivadas de la función radial y evaluar de manera puntual cada elemento en $\nabla^2 \Phi_a$, esto se realiza de la siguiente manera

$$\frac{\partial^2 \Phi_a}{\partial x^2} = \frac{\partial^2 \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)}{\partial x_i^2}, \quad \begin{matrix} i = 1, \dots, N_I \\ j = 1, \dots, N \end{matrix}$$

$$\frac{\partial^2 \Phi_a}{\partial y^2} = \frac{\partial^2 \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)}{\partial y_i^2}, \quad \begin{matrix} i = 1, \dots, N_I \\ j = 1, \dots, N. \end{matrix}$$

Finalmente, la estructura de la matriz Φ_b queda expresada de forma matricial como

$$\Phi_b = \begin{bmatrix} \phi(\|\mathbf{x}_{N_T+1} - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_{N_T+1} - \mathbf{x}_N\|) \\ \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix}.$$

En la literatura se han reportado excelentes resultados numéricos para el problema analizado [15, 16, 91, 33, 92, 93] y para el caso no lineal [94, 95]. Comparado con el esquema de elemento finito para una misma malla se ha observado que se obtiene una mejor aproximación [96]. De manera similar, se ha determinado numéricamente que con el método de colocación asimétrico se obtiene una mayor exactitud en comparación con el método de diferencias finitas y el método pseudospectral [88].

Recientemente se ha mostrado, vía experimentación numérica, una tasa de convergencia exponencial para un problema de Poisson en $2-d$ a partir del núcleo multicuádrico con el método de colocación asimétrico [75]. Cabe resaltar que la tasa de convergencia exponencial es significativamente superior a la tasa de convergencia algebraica obtenida con diferencias finitas o elemento finito. Este orden de convergencia exponencial, se traduce en un ahorro sustancial en tiempo de procesamiento en comparación con el tiempo requerido para obtener el mismo error empleando otros métodos numéricos con órdenes de convergencia algebraicos. En este capítulo queremos determinar si para ecuaciones lineales dependientes del tiempo es posible obtener una tasa de convergencia exponencial.

2.3. Problemas Dependientes del Tiempo

En esta sección se muestra el esquema de colocación asimétrico para ecuaciones diferenciales parciales lineales dependientes del tiempo. Posteriormente, se realiza su correspondiente análisis de estabilidad y se expone algorítmica-

mente la implementación de dicho esquema. Por último, se ejemplifica el método de Kansa para un caso particular correspondiente a la ecuación de tipo convección-difusión en una dimensión.

Considere la siguiente ecuación general dependiente del tiempo

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \mathcal{L}u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad (2.12)$$

sujeta a las condiciones de frontera e inicio

$$\mathcal{B}u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad x \in \partial\Omega, \quad (2.13)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad t = 0, \quad (2.14)$$

donde \mathcal{L} es un operador diferencial parcial y \mathcal{B} denota la aplicación de las condiciones de frontera que pueden ser de tipo Dirichlet, Newman o Robin. Las funciones $f(\mathbf{x}, t)$, $g(\mathbf{x}, t)$ y $u_0(\mathbf{x})$ son conocidas.

Sea $\tilde{u}(\mathbf{x}, t)$ la solución aproximada de la solución exacta $u(\mathbf{x}, t)$ del problema del valor inicial (2.12-2.14). La aproximación $\tilde{u}(\mathbf{x}, t)$ es definida como una combinación lineal de funciones radiales:

$$\tilde{u}(\mathbf{x}, t) = \sum_{j=1}^N \lambda_j(t) \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x}, \mathbf{x}_j \in \mathbb{R}^d, \quad (2.15)$$

donde $\lambda_j(t)$ son los coeficientes dependientes del tiempo a ser determinados, $\|\cdot\|$ es la distancia Euclidiana y ϕ denota la función de base radial. Dependiendo del tipo de función radial, es posible que se requiera incluir un polinomio $p(\mathbf{x})$ en (2.15) para garantizar la no singularidad en el problema de interpolación. Para mantener la notación simple se omitió el polinomio $p(\mathbf{x})$ en (2.15).

La diferencia con el problema estacionario planteado en la sección anterior, radica en la adición de la parte temporal en la ecuación diferencial parcial y por ende en la aproximación $\tilde{u}(\mathbf{x}, t)$.

La aproximación de la derivada temporal la definimos como:

$$\frac{\partial \tilde{u}(\mathbf{x}, t)}{\partial t} = \sum_{j=1}^N \frac{d\lambda_j(t)}{dt} \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad (2.16)$$

sobre los N distintos puntos de colocación. La aproximación de las derivadas espaciales quedan expresadas como:

$$\frac{\partial \tilde{u}(\mathbf{x}, t)}{\partial \mathbf{x}} = \sum_{j=1}^N \lambda_j(t) \frac{\partial \phi(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial \mathbf{x}}, \quad (2.17)$$

$$\frac{\partial^2 \tilde{u}(\mathbf{x}, t)}{\partial \mathbf{x}^2} = \sum_{j=1}^N \lambda_j(t) \frac{\partial^2 \phi(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial \mathbf{x}^2}, \dots \quad (2.18)$$

Como se observa arriba, en la aproximación de las derivadas espaciales el término dependiente del tiempo $\lambda(t)$ permanece invariante, por lo tanto el cálculo de las derivadas espaciales es similar a lo obtenido en la aproximación a problemas estacionarios.

Substituyendo (2.16) en la ecuación (2.12) y aplicando la condición de frontera (2.13), obtenemos un sistema algebraico con N incógnitas

$$\begin{cases} \sum_{j=1}^N \left[\frac{d\lambda_j(t)}{dt} \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) + \mathcal{L}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \right] \lambda_j(t) = f(\mathbf{x}_i, t), & i = 1, 2, \dots, N_I \\ \sum_{j=1}^N [\mathcal{B}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)] \lambda_j(t) = g(\mathbf{x}_i, t), & i = N_I + 1, \dots, N. \end{cases} \quad (2.19)$$

Con fines de claridad en la notación solo consideramos los términos que involucran las derivadas temporales y parciales definidas en Ω , los términos que afectan los nodos frontera son omitidos. Reformulando la ecuación (2.19) en una forma compacta

$$\Phi_a \dot{\lambda} + \mathcal{L}\Phi_a \lambda = f, \quad (2.20)$$

donde $\dot{\lambda} = \dot{\lambda}(t)$ es el vector de incógnitas dependiente del tiempo a ser determinado. Las matrices $\Phi_a \in \mathbb{R}^{N_I \times N}$ y $\mathcal{L}\Phi_a \in \mathbb{R}^{N_I \times N}$ son determinadas mediante $\Phi_a = \{\phi(\|x_i - x_j\|)\}$ y $\mathcal{L}\Phi_a = \{\mathcal{L}\phi(\|x_i - x_j\|)\}$ con $i = 1, 2, \dots, N_I$, $j = 1, \dots, N$.

Asumiendo la invertibilidad de Φ_a , la ecuación (2.20) se puede expresar como

$$\frac{d\lambda}{dt} = F(\lambda, t), \quad (2.21)$$

donde $F(\lambda, t) = -\Phi_a^{-1} \mathcal{L}\Phi_a \lambda + \Phi_a^{-1} f$. La expresión obtenida es un sistema de ecuaciones diferenciales ordinarias (EDO), en donde se tienen N incógnitas que

corresponden a las λ_i . El cálculo de la derivadas es realizado vía las funciones radiales y se deja libre el tiempo. Es decir, se tiene un sistema semi-discreto el cual puede ser resuelto empleando algún método eficiente de ecuaciones diferenciales ordinarias, por ejemplo los métodos de Runge-Kutta, métodos Adams o métodos multipaso [97, 98, 99]. En el presente trabajo abordamos el método clásico de discretización θ , la extensión a EDO será objeto de una investigación posterior.

2.3.1. Esquema Explícito e Implícito

En esta sección con base en el esquema clásico θ , $0 \leq \theta \leq 1$, se discretiza el problema de valor inicial analizado. Posteriormente, se realiza un análisis de estabilidad para determinar el criterio de estabilidad matricial para el esquema explícito e implícito.

Considere el problema de valor inicial definido por (2.12-2.14), discretizando (2.12) con respecto al tiempo y al espacio mediante el esquema standard θ , $0 \leq \theta \leq 1$, la ecuación resultante se puede expresar como

$$u(\mathbf{x}, t + \Delta t) - u(\mathbf{x}, t) = \Delta t f(\mathbf{x}, t + \Delta t) - \Delta t [\theta \mathcal{L}u(\mathbf{x}, t + \Delta t) + (1 - \theta) \mathcal{L}u(\mathbf{x}, t)],$$

$$\mathcal{B}u(\mathbf{x}, t + \Delta t) = g(\mathbf{x}, t + \Delta t),$$

donde Δt denota al tamaño del paso temporal. Empleando la notación $u(\mathbf{x}, t^n) = u^n$ y $t^n = t^{n-1} + \Delta t$, las dos ecuaciones anteriores se pueden representar como

$$u^{n+1} + \Delta t \theta \mathcal{L}u^{n+1} = u^n + \Delta t f^{n+1} - \Delta t (1 - \theta) \mathcal{L}u^n, \quad (2.22)$$

$$\mathcal{B}u^{n+1} = g^{n+1}. \quad (2.23)$$

Como se ha descrito anteriormente, se tienen N distintos nodos de colocación distribuidos en nodos al interior $X_I = \{\mathbf{x}_i\}_{i=1}^{N_I} \in \Omega$ y los nodos que conforman la frontera $X_f = \{\mathbf{x}_i\}_{i=N_I+1}^N \in \partial\Omega$ con $X = X_I \cup X_f$ y $N_f = N - N_I$. Para obtener la solución aproximada $\tilde{u}(\mathbf{x}, t)$ de la solución exacta $u(\mathbf{x}, t)$ del

problema de valor inicial, empleamos una combinación lineal de funciones radiales. Substituyendo (2.15) en (2.22) y aplicando las condiciones de frontera (2.23), obtenemos

$$\begin{bmatrix} \Phi_a + \theta \Delta t \mathcal{L} \Phi_a \\ \mathcal{B} \Phi_b \end{bmatrix} \lambda^{n+1} = \begin{bmatrix} \Phi_a - (1 - \theta) \Delta t \mathcal{L} \Phi_a \\ 0_b \end{bmatrix} \lambda^n + \begin{bmatrix} \Delta t f^{n+1} \\ g^{n+1} \end{bmatrix}, \quad (2.24)$$

donde $\lambda \in \mathbb{R}^N$ y $0_b \in \mathbb{R}^{N_f \times N}$ representa la matriz de ceros. La aplicación del operador diferencial \mathcal{L} sobre el conjunto de nodos interiores es la matriz $\mathcal{L} \Phi_a \in \mathbb{R}^{N_I \times N}$ donde la matriz Φ_a representa a una porción de la matriz de Gram $A = [\Phi_a \ \Phi_b]^T \in \mathbb{R}^{N \times N}$. La aplicación del operador diferencial \mathcal{B} sobre los nodos que conforman la frontera está definida por $\mathcal{B} \Phi_b \in \mathbb{R}^{N_f \times N}$. Por último, $f \in \mathbb{R}^N$ y $g \in \mathbb{R}^N$ corresponden a la funciones fuente y frontera. Esta última ecuación (2.24) se puede escribir de forma compacta

$$H_+ \lambda^{n+1} = H_- \lambda^n + F^{n+1}, \quad (2.25)$$

donde

$$H_+ = \begin{bmatrix} \Phi_a + \theta \Delta t \mathcal{L} \Phi_a \\ \mathcal{B} \Phi_b \end{bmatrix}, \quad H_- = \begin{bmatrix} \Phi_a - (1 - \theta) \Delta t \mathcal{L} \Phi_a \\ 0_b \end{bmatrix}$$

y

$$F^{n+1} = \begin{bmatrix} \Delta t f^{n+1} \\ g^{n+1} \end{bmatrix}.$$

El lado derecho en (2.25) representa la solución conocida al tiempo t_k , mientras que el lado izquierdo representa la solución desconocida al tiempo t_{k+1} . La ecuación (2.25) representa el sistema iterativo a resolver en cada paso del tiempo desde $n = 0$ hasta $n = n_{max}$ donde $n_{max} = t_{max}/\Delta t$. Este sistema iterativo corresponde a la aproximación numérica del problema de valor inicial empleando el método de colocación asimétrico.

Dependiendo del valor de θ en (2.25) se obtiene un método explícito $\theta = 0$ ó un método implícito $\theta = 0.5$ también conocido como esquema de Crank-Nicholson. En los siguientes párrafos determinaremos un criterio de estabilidad matricial para determinar el tamaño del paso Δt .

Análisis de Estabilidad

En esta sección se desarrolla un análisis de estabilidad matricial para el sistema iterativo (2.25). Este análisis fue tomado de los trabajos [27] y [100].

La solución aproximada discreta es representada como: $u^n = A\lambda^n$ siendo A la matriz de Gram, substituyendo la ecuación anterior en (2.25) con $F = 0$, obtenemos

$$u^{n+1} = Ku^n, \quad (2.26)$$

donde $K = AH_+^{-1}H_-A^{-1}$, considere ahora que en el tiempo n -ésimo se introduce una perturbación en (2.26); $u^n + \epsilon^n$, por lo tanto para $n + 1$ se tiene

$$\epsilon^{n+1} = K\epsilon^n, \quad (2.27)$$

considerando ausencia de errores de redondeo, el sistema es estable si $\epsilon^n \rightarrow 0$ conforme $n \rightarrow \infty$, esto puede ser garantizado si el radio espectral de K es menor que la unidad: $\rho(K) < 1$. Considerando $\mathcal{B} = I$ donde $I \in \mathbb{R}^{N \times N}$ representa la matriz identidad, la condición de estabilidad se puede expresar como: $\rho([I + \theta\Delta tM]^{-1}[I - (1 - \theta)\Delta tM]) < 1$ con $M = [\mathcal{L}\Phi_a \ 0]^T A^{-1}$, de manera equivalente se obtiene

$$\left| \frac{1 - (1 - \theta)\Delta t\lambda_M}{1 + \theta\Delta t\lambda_M} \right| < 1, \quad (2.28)$$

donde λ_M es el máximo eigenvalor asociado a la matriz M ; $\lambda_M = \max_{1 \leq i \leq N} \{\lambda_M^i\}$.

Los eigenvalores se pueden calcular resolviendo el problema generalizado de eigenvalores: $\mathcal{L}\Phi_a\{s\} = \lambda_M A\{s\}$ donde $\{s\}$ corresponde al eigenvector y λ_M al eigenvalor buscado.

Seleccionando $\theta = 0.5$ (método implícito) siempre se satisface (2.28) con $\lambda_M \geq 0$, por lo cual el método es incondicionalmente estable lo que implica que no hay una restricción en el tamaño del paso temporal Δt . Por otro lado, seleccionando $\theta = 0$ (método explícito) se obtiene

$$|1 - \Delta t\lambda_M| < 1, \quad (2.29)$$

el método es condicionalmente estable con $\lambda_M > 0$, el tamaño del paso Δt se

puede seleccionar como

$$\Delta t < 2/\lambda_M. \quad (2.30)$$

En ambos casos, método implícito y método explícito, se requiere que el sistema sea no singular y que esté bien condicionado. El buen condicionamiento del sistema algebraico depende del número de nodos y del parámetro utilizado en la función radial.

En el análisis de estabilidad desarrollado la hipótesis de trabajo fue una computadora con aritmética exacta. Es por ello que aún satisfaciendo (2.30) el error introducido puede no disminuir. Para tratar de atenuar dicha posibilidad es común ponderar el valor de Δt en (2.30) de la forma: $\Delta t = \nu(2/\lambda_M)$ con $0 < \nu < 1$.

2.3.2. Algoritmo

En esta sección se describe de forma algorítmica la implementación del sistema iterativo (2.25), posteriormente se realiza su análisis de complejidad.

Los pasos necesarios para determinar la solución al tiempo t_{max} del sistema iterativo (2.25) se muestran en el algoritmo 1. Como dato de entrada se tienen los N distintos nodos de colocación $X = X_I \cup X_f$, con base en dichos nodos se construye la matriz de Gram $\Phi = [\Phi_a \Phi_b]^T$ y las matrices que contienen la aplicación de los operadores diferenciales $\{\mathcal{L}\Phi_a, \mathcal{B}\Phi_b\}$. La construcción de H_+ es realizada con las matrices anteriormente obtenidas. El algoritmo es sencillo y es aplicable para el esquema de discretización implícito y explícito.

Como dato de salida se obtiene λ^{n+1} con lo cual se determina la solución aproximada $\tilde{u}(x, t_{max})$ para cualquier nodo contenido en el dominio Ω , mediante la ecuación de interpolación (2.15). Cabe destacar que el algoritmo presentado es independiente de la dimensión espacial (\mathbb{R}^d) en donde se encuentren definidos los nodos de colocación. Por lo tanto, desde el punto de vista de implementación se tiene reuso de código, lo cual es conveniente para reducir los tiempos de producción de software y reducir los errores en la codificación.

Algoritmo 1 Método Implícito y Explícito $t = 0$ Construir Φ_a , Φ_b , $\mathcal{L}\Phi_a$ y $\mathcal{B}\Phi_b$ Aproximar la condición inicial $\Phi\lambda^0 = u_0(\mathbf{x})$ **while** $t < t_{max}$ **do**Proyección: $P = [\Phi_a + (1 - \theta)\Delta t\mathcal{L}\Phi_a]\lambda^n + \Delta t f^{n+1}$ Corrección : $P_i = g(\mathbf{x}_i, t + \Delta t)$, $i = N_I + 1, \dots, N$ Solución : $H_+\lambda^{n+1} = P$ $t = t + \Delta t$ **end while****Análisis de Complejidad**

En esta sección se realiza el análisis de complejidad del algoritmo 1, el cual corresponde al esquema de discretización implícito y explícito a partir del esquema de colocación asimétrico.

La construcción de las cuatro matrices $\{\Phi_a, \Phi_b, \mathcal{L}\Phi_a, \mathcal{B}\Phi_b\}$ es realizada mediante dos ciclos iterativos, lo cual tienen complejidad en tiempo $O(N^2)$ y $O(N^2)$ en espacio.

El algoritmo 1 lo podemos dividir en dos bloques lógicos: a) aproximación de la condición inicial, b) el método iterativo temporal. En el bloque a) se requiere resolver un sistema lineal de ecuaciones de $N \times N$, para ello empleamos la factorización LU con un costo aproximado $O(N^3)$. Por otro lado, en el bloque lógico b) en cada iteración del tiempo se realizan tres pasos que llamaremos: Proyección-Corrección-Solución (PCS). El paso de Corrección no influye significativamente en el análisis y por lo tanto se omite.

En el paso de Proyección se requiere de una multiplicación matriz-vector con un costo $O(N_I \times N)$. Sabemos que el número de nodos interiores $N_I = N - N_f$ es comparativamente superior al número de nodos que conforman la frontera. Por lo tanto, $N_I \approx N$ y el costo de realizar la multiplicación matriz-vector es $O(N^2)$. El costo en el paso de Solución es $O(N^2)$, debido a que se realizan

$\tau = t_{max}/\Delta t$ iteraciones, la complejidad del bloque lógico b) es $O(\tau N^2)$.

Del análisis realizado, se deduce que la complejidad total del algoritmo 1 es

$$O(N^3) + O(\tau N^2).$$

De una manera natural podríamos tomar el grado más alto y decir que se requiere de $O(N^3)$ operaciones. Sin embargo, la constante τ en general tiene varios órdenes de magnitud y debe ser considerada. La condición que se debe cumplir para que $O(N^3) > O(\tau N^2)$, es $N > \tau$. Por lo tanto, la complejidad del algoritmo 1 se divide en dos casos

$$\begin{cases} O(N^3) & \text{si } N > \tau, \\ O(\tau N^2) & \text{en caso contrario.} \end{cases} \quad (2.31)$$

La relación obtenida está en función del número de nodos N , el tiempo máximo de simulación t_{max} y el tamaño del paso del tiempo Δt . De las simulaciones numéricas que hemos realizado en este trabajo, el mayor tiempo de procesamiento se realiza en el ciclo iterativo PCS. Es decir, por lo general se satisface $N < \tau$.

Notas

- Para disminuir el costo computacional en el algoritmo 1 se requiere disminuir el valor de τ . Es decir, tomar el paso temporal Δt lo más grande posible. Esto es posible hacerlo seleccionando $\theta = 0.5$, lo cual corresponde al método implícito que es incondicionalmente estable. Por otro lado, si \mathcal{L} es no lineal y seleccionando el esquema implícito, en cada paso del avance temporal se requiere resolver un sistema no lineal de ecuaciones de la forma $\mathbf{F}(\lambda_{n+1}) = 0$, lo cual no es una tarea trivial. Por lo tanto, la selección de un método implícito o un método explícito está en función de la estructura de la ecuación diferencial parcial a resolver.
- El algoritmo descrito en esta sección es independiente de la dimensión

del espacio, por lo cual es un método numérico general para la solución de problemas de valor inicial lineales.

En esta sección se trabajó con funciones de base radial estrictamente positivas definidas. La razón de esta elección radica en no requerir un polinomio $p(\cdot)$ en el esquema de interpolación radial. Esto nos permitió contar con una notación más clara para describir los distintos esquemas de discretización.

2.3.3. Caso Particular: Convección-Difusión

En esta sección se describe en detalle la construcción de las matrices del algoritmo 1 para una ecuación de tipo convección-difusión lineal en una dimensión, la cual es analizada en la sección 2.4.2. Con base en el núcleo multicuádrico y definiendo $r = x_i - x_j$ la determinación de las derivadas parciales está definida como:

$$\begin{aligned}\phi(r) &= (r^2 + c^2)^{1/2}, & \phi_x(r) &= r(r^2 + c^2)^{-1/2}, \\ \phi_{xx}(r) &= c^2(r^2 + c^2)^{-3/2}.\end{aligned}$$

Los subíndices en ϕ denotan la primera y segunda derivada. Las matrices requeridas para construir $\{H_+, H_-\}$ son definidas como:

$$\begin{aligned}\Phi_a &= \begin{bmatrix} \phi(\|x_1 - x_1\|) & \cdots & \phi(\|x_1 - x_N\|) \\ \vdots & \ddots & \vdots \\ \phi(\|x_{N_I} - x_N\|) & \cdots & \phi(\|x_{N_I} - x_N\|) \end{bmatrix}, \\ \Phi_b &= \begin{bmatrix} \phi(\|x_{N_I+1} - x_1\|) & \cdots & \phi(\|x_{N_I+1} - x_N\|) \\ \phi(\|x_N - x_1\|) & \cdots & \phi(\|x_N - x_N\|) \end{bmatrix}, \\ \Phi_a'' &= \begin{bmatrix} \phi_{xx}(\|x_1 - x_1\|) & \cdots & \phi_{xx}(\|x_1 - x_N\|) \\ \vdots & \ddots & \vdots \\ \phi_{xx}(\|x_{N_I} - x_N\|) & \cdots & \phi_{xx}(\|x_{N_I} - x_N\|) \end{bmatrix},\end{aligned}$$

$$\Phi'_a = \begin{bmatrix} \phi_x(\|x_1 - x_1\|) & \cdots & \phi_x(\|x_1 - x_N\|) \\ \vdots & \ddots & \vdots \\ \phi_x(\|x_{N_I} - x_N\|) & \cdots & \phi_x(\|x_{N_I} - x_N\|) \end{bmatrix}.$$

2.4. Convergencia para Convección-Difusión

En esta sección se investiga numéricamente la tasa de convergencia para una ecuación diferencial parcial dependiente del tiempo de tipo convección-difusión. Se consideran dos esquemas de discretización; implícito y explícito. El énfasis radica en la variación del número de Péclet y el número de nodos de colocación; h -refinamiento. El objetivo es mostrar que para números de Péclet moderados se obtiene una tasa de convergencia exponencial. Adicionalmente, se explora la relación entre el parámetro c del núcleo multicuádrico y los parámetros convectivo y difusivo de la ecuación diferencial parcial. Por último, se realiza una comparación para medir el desempeño entre el esquema de discretización explícito e implícito.

2.4.1. Convección-Difusión

La ecuación de convección-difusión sirve como un modelo simplificado de las ecuaciones de Navier-Stokes en dinámica de fluidos. Con base en esta ecuación diferencial parcial se tiene un modelo idealizado del fenómeno difusivo y convectivo en una sola ecuación. Los valores del término convectivo μ y difusivo β , caracterizan la ecuación diferencial parcial analizada. Definiendo el número Péclet $P_e = \mu/\beta$, se observa que cuando el término difusivo domina se obtiene un número Péclet bajo y la ecuación es parabólica. Cuando el coeficiente convectivo domina, se obtiene un número Péclet alto y la ecuación es predominantemente hiperbólica. Al pasar del tipo parabólico al hiperbólico, el gradiente en la solución analítica se va incrementando hasta llegar a una dis-

continuidad. La solución numérica de ecuaciones diferenciales parciales de tipo hiperbólico (predominantemente convectivas) representan uno de los mayores retos actualmente, debido principalmente a los efectos difusivos y dispersivos presentes en los métodos numéricos.

2.4.2. Planteamiento

Considere la siguiente ecuación de tipo convección-difusión en una dimensión [28, 30]

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} - \mu \frac{\partial u}{\partial x}, \quad 0 < x < 6, \quad (2.32)$$

con la condición de frontera tipo Dirichlet y condición inicial

$$u(0, t) = ae^{bt}, \quad u(6, t) = ae^{bt-6\gamma}, \quad t > 0, \quad (2.33)$$

$$u(x, 0) = ae^{-\gamma x}. \quad (2.34)$$

La solución analítica está determinada como

$$u(x, t) = ae^{bt-\gamma x}, \quad \gamma = \frac{-\mu \pm \sqrt{\mu^2 + 4\beta b}}{2\beta} > 0, \quad (2.35)$$

donde β es el parámetro difusivo mientras μ es el término convectivo. A lo largo de la sección se utilizarán los valores $a=107$ y $b=0.5$, dichos valores fueron seleccionados para obtener la forma de la solución analítica similar a lo mostrado en [28]. Por simplicidad el esquema explícito e implícito con multicuádracos serán referidos como EMQ e IMQ respectivamente.

El parámetro c del núcleo multicuádraco es seleccionado mediante la siguiente desigualdad

$$\|u(x, t_{max}, c_{i+1}) - \tilde{u}(x, t_{max}, c_{i+1})\|_2 < \|u(x, t_{max}, c_i) - \tilde{u}(x, t_{max}, c_i)\|_2, \quad (2.36)$$

donde u , \tilde{u} representan la solución analítica y numérica respectivamente, el parámetro c lo variamos por incrementos $c_i = 0.1 \cdot i$, $i \in \mathbb{N}$. Cuando se satisface (2.36) se incrementa el valor de c , en caso contrario seleccionamos el último valor de c que satisface (2.36). Este esquema para determinar el parámetro c es exhaustivo, y se basa en el error cuadrático medio entre la solución exacta $u(x, t)$ y la solución aproximada $\tilde{u}(x, t)$.

2.4.3. Método Implícito

Nuestro objetivo en esta sección es determinar experimentalmente la tasa de convergencia empleando IMQ. Para este propósito, los siguientes valores fueron seleccionados: $t_{max}=1$, $\Delta t=0.01$ y nodos equiespaciados definidos en el rango $N = \{10, 20, \dots, 50\}$.

Cuando restringimos el número de Péclet $=\mu/\beta \ll 30$ se tienen soluciones analíticas suaves. Por otro lado, conforme $P_e \rightarrow 30$ la solución analítica presenta un mayor gradiente cerca del origen $x = 0$. Fijando el coeficiente convectivo μ y variando el coeficiente difusivo β en el rango $\{0.1, 0.02, 0.01, 0.005, 0.0033\}$, se obtienen 5 casos en donde se analizará su tasa de convergencia. En cada variación del coeficiente difusivo β , el valor del parámetro c es determinado mediante (2.36) con $N = 50$ y $t_{max} = 1$.

En la figura 2.2 se muestra la tasa de convergencia, en escala semilog, para los cinco distintos casos de prueba: a) $\beta = 0.1$, $\mu = 0.1$ con $c = 1.3$, b) $\beta = 0.02$, $\mu = 0.1$ con $c = 1.3$, c) $\beta = 0.01$, $\mu = 0.1$ con $c = 1.2$, d) $\beta = 0.005$, $\mu = 0.1$ con $c = 1.3$ y e) $\beta = 0.0033$, $\mu = 0.1$ con $c = 1.2$. Las gráficas de esta figura muestran que manteniendo fijo el parámetro c e incrementando el número de nodos, el error de aproximación decrece de forma exponencial.

Cabe resaltar que la mayor exactitud obtenida corresponde a los números de Péclet bajos, lo que corresponde al caso predominantemente parabólico. Conforme incrementamos el número de Péclet; ver figura 2.2 de abajo hacia arriba, la pendiente de cada línea decrece, sin embargo se mantiene la convergencia exponencial.

Por completez, en la figura 2.3 se muestra la forma de la solución analítica y numérica de los casos analizados.

Como se menciona en Madych & Nelson [101] y Madych [65], se tienen dos formas de decrementar el error de aproximación para el caso de interpolación: incrementando el número de nodos (h -refinamiento) o incrementando el parámetro c (c -refinamiento). La figura 2.2 corresponde al caso de h -

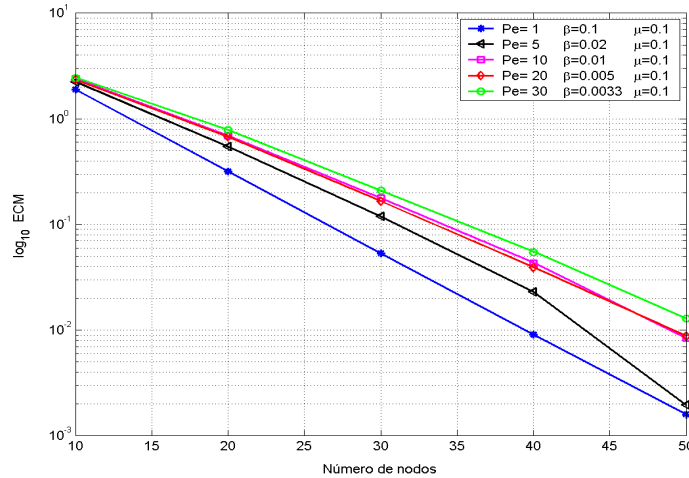


Figura 2.2: Convergencia exponencial para 5 distintos casos de prueba con el método implícito, al tiempo $t_{max} = 1$.

refinamiento, que es la manera tradicional de incrementar la exactitud en la solución numérica. Sin embargo, el esquema de h -refinamiento tiene la desventaja de incrementar el requerimiento en memoria así como el tiempo de procesamiento. Con base en el esquema de colocación asimétrico Cheng [75] muestra numéricamente que “incrementando el número de nodos N sin el correspondiente cambio en el parámetro c , es equivalente en mantener fijo N e incrementar el valor de c ”. Nuestra tarea ahora es investigar el esquema de c -convergencia para la ecuación de convección-difusión analizada. Para este propósito, considere el esquema IMQ con los siguientes parámetros: $N=30$, $\Delta t=0.001$, $\beta=0.9$, $\mu=0.4$ y $t_{max}=1$.

En la figura 2.4 se muestra en una escala semilog el error cuadrático medio para cada valor de c en el intervalo $[0, 2.5]$ con incrementos de 0.1. Se observa que conforme se incrementa el valor de c el error de aproximación desciende de forma exponencial. Para valores $c > 2.5$, se obtiene un desbordamiento en la solución numérica, esto es debido al mal condicionamiento del sistema algebraico.

De los resultados obtenidos, queda la impresión de que conforme la ecuación

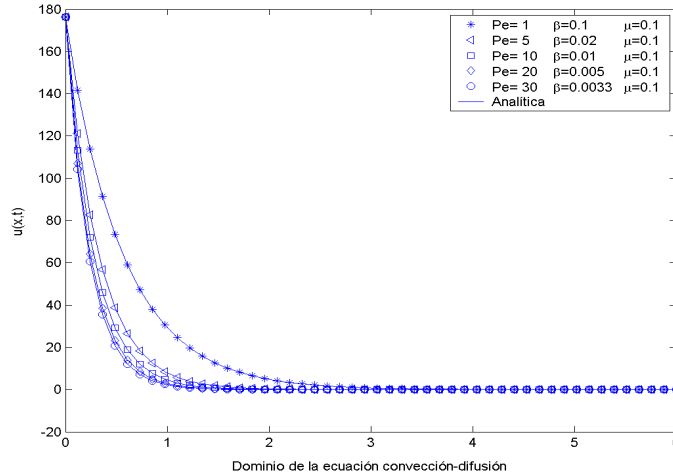


Figura 2.3: Solución analítica y numérica para distintos valores de β y μ , al tiempo

$$t_{max} = 1.$$

pasa de un comportamiento parabólico a uno hiperbólico es posible obtener una tasa de convergencia exponencial. Para esclarecer este punto, exploraremos el esquema de c -convergencia con los siguientes valores: $\mu = 0.01$, $\beta = 0.0002$, $t_{max} = 1$ y $\Delta t = 0.001$. El término convectivo μ domina al término difusivo β obteniendo un número de Péclet $P_e = 50$, lo cual ocasiona que en la solución analítica se presente una zona de alto gradiente cerca de $x = 0$. Para $N = 50$ no se obtuvo una buena aproximación numérica; $ECM = 0.2894$ con $c = 1.2$, y con una diferencia máxima entre el valor aproximado y el real $\max|\tilde{u}(x, t) - u(x, t)| = 2$, localizado en la zona de mayor gradiente. Fuera de la zona de alto gradiente la diferencia entre la solución analítica y numérica es del orden 10^{-2} . Para tratar de disminuir el error de aproximación incrementaremos el número de nodos.

Incrementando el número de nodos a $N = 200$, en la figura 2.5 se muestra el comportamiento del error cuadrático medio en función del incremento del parámetro c . En esta gráfica se observa una línea recta, lo cual indica que el decrecimiento del error es de carácter exponencial.

Analizando los datos mostrados en las figuras 2.4 y 2.5, las cuales corres-

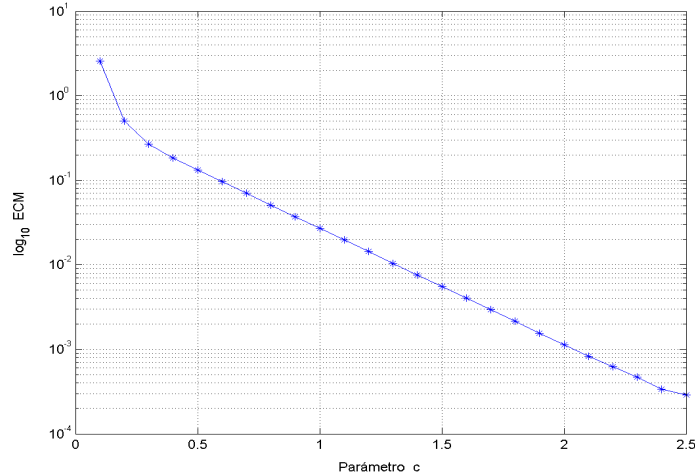


Figura 2.4: Convergencia exponencial variando c y manteniendo fijo $N = 30$ con el método implícito, con $t_{max} = 1$.

ponden al esquema de c -convergencia con $P_e = 0.4$ y $P_e = 50$ respectivamente, destaca el hecho de que en ambas gráficas se muestra un decrecimiento exponencial en el error de aproximación. Sin embargo, el rango del error ECM mostrado en la figura 2.4 es comparativamente superior al rango mostrado en la figura 2.5.

La diferencia máxima obtenida con $N = 200$ y $c = 0.25$ es $\max|\tilde{u}(x, t) - u(x, t)| = 0.56$, que es un orden de magnitud menor que lo obtenido con $N = 50$. Fuera de la zona de mayor gradiente, la diferencia entre la solución analítica y la aproximación numérica es del orden 10^{-4} , esto representa un decremento de dos órdenes de magnitud en comparación con lo obtenido con $N = 50$.

Observamos que conforme la ecuación pasa de una estructura parabólica a una estructura hiperbólica, el error de aproximación crecía en el esquema de c -convergencia. Además, observamos que el rango de búsqueda del parámetro c disminuía aproximadamente en un orden de magnitud, ocasionando que la búsqueda exhaustiva de c se convirtiera en una tarea costosa computacionalmente.

En principio es posible resolver numéricamente la ecuación de convección-

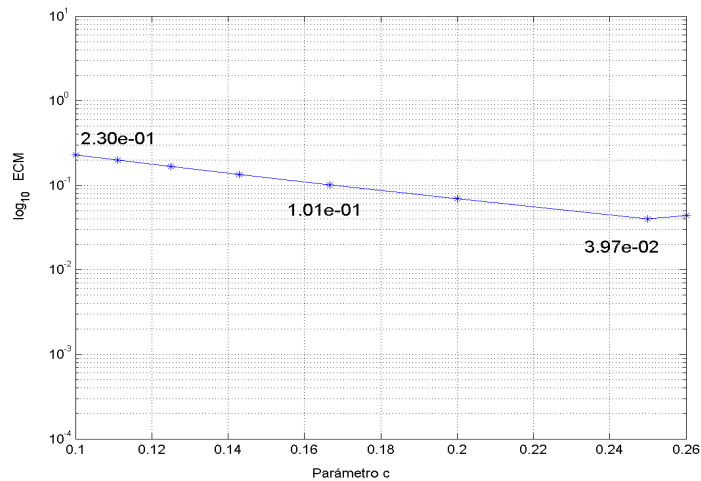


Figura 2.5: Convergencia exponencial variando c y manteniendo fijo $N = 200$ con el método implícito, con $t_{max} = 1$.

difusión para moderados números de Péclet. El problema estriba en lograr una buena aproximación en las zonas con alta variación espacial. Una estrategia consiste en incrementar el número de nodos, sin embargo, dado que el requerimiento en espacio se incrementa cuadráticamente, la complejidad se incrementa cúbicamente y el número de condición se deteriora; es necesario utilizar estrategias que nos ayuden a disminuir el mal condicionamiento y decrementar el costo computacional. En el capítulo 4 se analiza el problema cuando $Pe = 10^3$, lo cual corresponde a la predominancia del término convectivo sobre el término difusivo. Para capturar la zona de alta variación espacial se utiliza el esquema de adaptación local de nodos que ha mostrado ser una técnica computacionalmente eficiente [102, 103].

Por otra parte, se observó que el decrecimiento exponencial en el error con el esquema implícito está relacionado con el tamaño del paso en el tiempo Δt . Al aumentar Δt se incrementa el error de la aproximación pero todavía obtenemos la convergencia exponencial. El comportamiento anterior se muestra en la figura 2.6 en una escala semilog para diversos pasos temporales. Los siguientes parámetros fueron utilizados: $N=50$, $\beta=0.2$ y $\mu=1$.

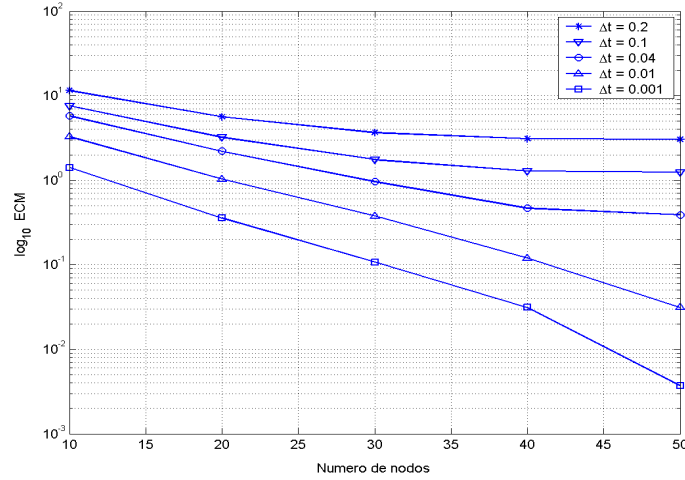


Figura 2.6: Convergencia para $\beta = 0.2$ y $\mu = 1$ con distintos pasos temporales Δt , $t_{max} = 1$ con el método implícito.

La relación entre el tamaño del paso temporal Δt y el parámetro c de la función multicuádrica se muestra en la tabla 2.1. Se observa que el valor de c se reduce conforme se incrementa Δt .

Δt	0.001	0.010	0.040	0.100	0.200
c	0.480	0.134	0.071	0.048	0.032

Tabla 2.1: Variación de c para distintos Δt , se mantiene fijo $\beta = 0.2$, $\mu = 1$ con $t_{max} = 1$.

2.4.4. Relación β vs. c

Nuestro objetivo ahora consiste en determinar una relación entre el parámetro c y el coeficiente difusivo β . El experimento numérico fue realizado con los siguientes parámetros: $N=50$, $\mu=1.2$ y β lo variamos en el rango $\{0.1, 0.2, \dots, 1\}$. Para cada valor de β el valor de c se determina con base en la desigualdad (2.36) con incrementos en c de 0.01 para $t_{max} = 1$.

En la figura 2.7 se muestra sobre el eje x la variación de β y sobre el eje y el valor del parámetro c obtenido. Se puede observar en la figura 2.7 que el

parámetro c desciende monótonicamente de $c = 1.2$ a 0.72 para $0.3 \leq \beta \leq 1$. Esta gráfica nos indica que conforme se disminuye el coeficiente difusivo β se incrementa el número Péclet, y el valor del parámetro c es reducido. Cuando $\beta < 0.3$ se nota un ligero incremento en el valor de c .

Una consecuencia práctica de la relación anterior, radica en que conforme disminuimos el coeficiente difusivo, el valor del parámetro c también es reducido. Lo cual nos acota el rango de búsqueda del parámetro c .

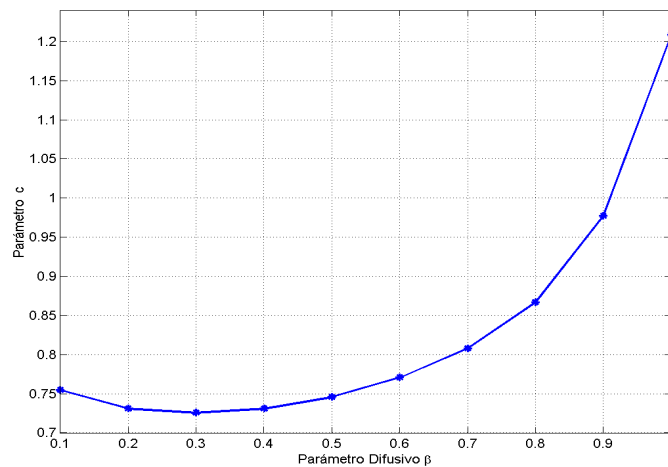


Figura 2.7: Variación del parámetro c con respecto del coeficiente β , fijo $\mu = 0.5$.

Observamos en este experimento numérico que al mantener fijos el número de nodos N , el paso del tiempo Δt , el término convectivo μ y el tiempo máximo t_{max} , el único coeficiente que queda libre es el coeficiente difusivo β , siendo este parámetro el que caracteriza la estructura de la ecuación diferencial parcial. Por lo tanto, la determinación del parámetro c está relacionado con la estructura de la ecuación diferencial parcial dependiente del tiempo de tipo convección-difusión.

2.4.5. Método Explícito

En una forma similar a lo realizado en la primera parte en la sección 2.4.3, ahora nuestro objetivo es determinar si con el método explícito con la función radial multicuádrica se obtiene una tasa de convergencia exponencial. Adicionalmente, se explora la influencia del parámetro c empleando h -refinamiento.

Fijamos el valor del término convectivo μ y variamos al término difusivo β de tal forma que obtengamos los 5 casos analizados con IMQ, empleamos un tamaño de paso $\Delta t=0.001$.

En la figura 2.8 se muestran los resultados obtenidos de los cinco casos de prueba. Para facilitar la interpretación de los resultados se despliegan en una escala semilog. Los cinco casos analizados son los siguientes: a) $\beta = 0.1$, $\mu = 0.1$ con $c = 1.3$, b) $\beta = 0.02$, $\mu = 0.1$ con $c = 1.2$, c) $\beta = 0.01$, $\mu = 0.1$ con $c = 1.1$, d) $\beta = 0.005$, $\mu = 0.1$ con $c = 1.2$ y e) $\beta = 0.0033$, $\mu = 0.1$ con $c = 1.2$. Se observa que el error de aproximación decrece de forma exponencial conforme aumentamos el número de nodos N , para los distintos números de Péclet.

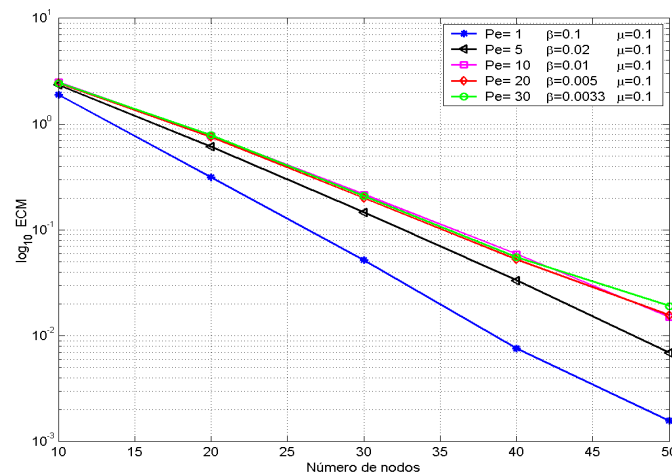


Figura 2.8: Convergencia exponencial para 5 distintos casos con el método explícito, al tiempo $t = 1$.

Conforme incrementamos el número de Péclet; ver figura 2.8 de abajo hacia arriba, la pendiente de cada línea decrece. Para números Péclet bajos se

observa una alta exactitud numérica, lo cual corresponde al caso predominantemente parabólico. Cabe mencionar que con el método implícito se obtuvo un comportamiento similar.

Note que el tamaño del paso en el tiempo Δt para EMQ es considerablemente inferior que el correspondiente Δt para IMQ. Por lo tanto, para IMQ se obtiene una reducción en el tiempo de procesamiento.

Con base en los resultados obtenidos con el método de colocación asimétrico para una ecuación diferencial parcial de tipo convección-difusión, podemos concluir que ambos esquemas, IMQ y EMQ, tienen una tasa de convergencia exponencial. La exactitud de la solución numérica es determinada por los coeficientes; difusivo β y convectivo μ de la EDP.

Los resultados obtenidos están basados fuertemente en la selección del parámetro c del núcleo multicuádrico. Es por ello que surgen las preguntas ¿Cuál es el efecto de utilizar otros valores de c ?, ¿Seguirá existiendo una tasa de convergencia exponencial?

En la figura 2.9 mostramos el efecto de variar el parámetro c en el esquema de h -refinamiento. El eje x corresponde al número de nodos $N = \{10, 20, 30, 40, 50\}$ y sobre el eje y se muestra el error cuadrático medio. Comenzado de arriba hacia abajo, los valores de c utilizados en los datos que conforman cada recta son: $\{0.1, 0.2, \dots, 1.4\}$.

Se observa en la figura 2.9 que para todos los valores de c seleccionados se obtuvo una línea recta en semilog, indicando una tasa de convergencia exponencial. La recta correspondiente a $c = 1.4$ solo llega hasta $N = 40$, esto se debe al mal condicionamiento ocasionado por el alto valor de c . Un resultado similar sucede con el esquema implícito.

Cabe resaltar que la variación del tamaño temporal Δt mostrado en la figura 2.6 para el método implícito produce un efecto similar al obtenido con la variación del parámetro c mostrado en esta sección, ver figura 2.9.

Como se menciona en la sección 2.3.1, el método explícito es condicional-

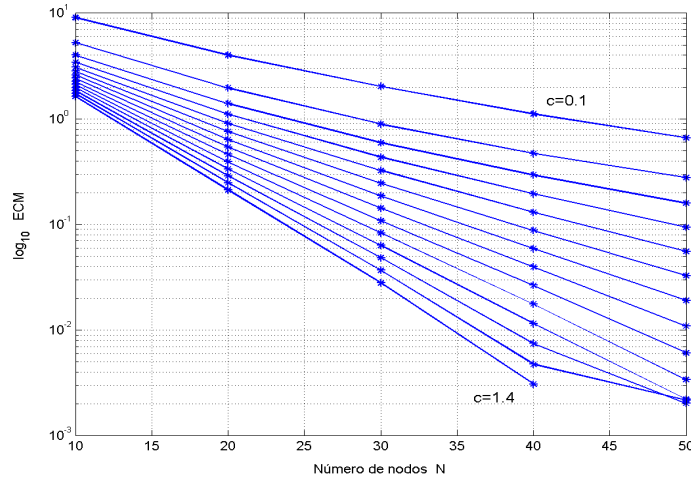


Figura 2.9: Variación del parámetro c , con $\beta = 0.1$, $\mu = 0.2$, al tiempo $t_{max} = 1$.

mente estable $\Delta t < 2/\lambda_M$ siendo λ_M el máximo eigenvalor asociado a la matriz $M = [\mathcal{L}\Phi_a \ 0]^T A^{-1}$, ahora nuestro objetivo es mostrar que conforme se incrementa el número de Péclet $P_e = \mu/\beta$ el tamaño del paso temporal Δt se reduce. Para ello considere que tenemos fijos los siguientes parámetros $\{h, c, \beta\}$ y lo que variaremos es el parámetro convectivo μ , tomando la norma matricial sobre el primer factor de M se observa

$$\|M_1\| \leq |\mu| \cdot \|\Phi'_a\| + |\beta| \cdot \|\Phi''_a\|,$$

incrementando el valor del término convectivo $|\mu + \varepsilon| > |\mu|$ se obtiene

$$\|M_2\| \leq |\mu + \varepsilon| \cdot \|\Phi'_a\| + |\beta| \cdot \|\Phi''_a\|,$$

de las dos relaciones anteriores se observa que $\|M_1\| < \|M_2\|$. Dado que el radio espectral es menor a igual que la norma de la matriz $\rho(M_1) \leq \|M_1\|$ y $\rho(M_2) \leq \|M_2\|$, implica que el tamaño del paso temporal para M_2 es menor que el tamaño del paso temporal para M_1

$$\frac{2}{\lambda_{M_2}} < \frac{2}{\lambda_{M_1}} \quad \text{ó} \quad \Delta t_2 < \Delta t_1,$$

por lo tanto, sí incrementamos el número de Péclet $P_e^2 > P_e^1$ es necesario reducir el tamaño del paso temporal Δt para satisfacer la condición de estabilidad

matricial. En el caso de que decrementemos el número de Péclet $P_e^2 < P_e^1$ se obtiene un incremento en el paso temporal $\Delta t_2 > \Delta t_1$.

2.4.6. Comparación Implícito y Explícito

En esta sección comparamos el método implícito y explícito para alcanzar una exactitud establecida.

En el experimento mantenemos fijos el número de nodos y variamos Δt para obtener el mismo error cuadrático medio. En cada variación del parámetro Δt , determinamos el valor del parámetro c mediante la desigualdad (2.36) con incrementos en c de 0.01. El resto de los parámetros utilizados son definidos como: $\beta=0.6$, $\mu=0.1$, $t_{max}=1$ y $N=20$. En la tabla 2.2 se muestran los resultados obtenidos. La primera columna corresponde a los esquemas de discretización, en la segunda columna indicamos el número de nodos N que permaneció constante. En la tercera y cuarta columna se despliega el tamaño del paso temporal y el error cuadrático medio respectivamente.

Esquema	N	c	Δt	ECM
IMQ	20	3.87	0.1	0.0097
EMQ	20	2.72	0.0048	0.0098

Tabla 2.2: Comparación entre IMQ y EMQ para alcanzar la misma precisión.

Se observa en la tabla 2.2, que el método explícito EMQ requiere un paso en el tiempo Δt menor que el requerido con IMQ para alcanzar la misma exactitud. Por lo tanto, se requiere un menor tiempo de procesamiento con el método implícito en comparación con el método explícito.

Modificando el experimento realizado anteriormente, si incrementamos el número de nodos a $N=40$ para ambos métodos y repetimos el procedimiento; lo que observamos fue una reducción del parámetro c aproximadamente a la mitad del valor obtenido con $N=20$, ver tabla 2.3. Con ambos métodos se alcanzó un ECM similar al mostrado en la tabla 2.2. Por lo tanto, el incremento

Esquema	N	c	Δt
IMQ	40	1.7	0.1
EMQ	40	1.03	0.004

Tabla 2.3: Variación del parámetro c para $N=40$ con IMQ y EMQ.

en el número de nodos manteniendo fijo el tamaño del paso temporal Δt no influye significativamente en la disminución del error cuadrático medio.

Capítulo 3

Descomposición de Dominio

En este capítulo mostramos el método clásico de Schwarz aplicado a problemas lineales temporales con base en el método de colocación asimétrico. Se realiza un estudio de la influencia que tiene el número de subdominios en el tiempo de procesamiento y la exactitud obtenida. Empleando el núcleo de placa delgada en un problema convectivo en dos dimensiones, se observó que conforme se incrementa el número de subdominios se reduce drásticamente el costo computacional, pero al mismo tiempo la exactitud disminuye. Se determinó un orden de complejidad lineal con respecto al número de subdominios y la ganancia en el tiempo de la simulación es de dos órdenes de magnitud cuando se utiliza el método de descomposición de dominio. Adicionalmente, se observó poca difusión numérica en comparación con el método de diferencias finitas. Los resultados obtenidos indican que es posible resolver problemas más complejos empleando descomposición de dominio para reducir el costo computacional significativamente.

3.1. Introducción

El término descomposición de dominio (DD) es dependiente del contexto. En lo subsecuente por método de descomposición de dominio se entenderá un

esquema iterativo para la solución numérica de ecuaciones en derivadas parciales basado en una descomposición espacial del dominio Ω . En lugar de resolver el problema completo (lo que involucra resolver un sistema lineal o no lineal de ecuaciones), el problema es dividido en varios subproblemas que se pueden resolver con un menor esfuerzo computacional. El método de descomposición de dominio es el encargado de unir la solución obtenida en cada subdominio en una forma correcta, de tal modo que la solución obtenida en muchos subdominios sea igual a la solución obtenida considerando el problema entero.

Las principales características que motivan usar descomposición de dominio son:

- El método de descomposición de dominio permite reducir el número de condición del sistema e incrementar el desempeño computacional.
- Al subdividir el dominio del problema en varios subdominios, es posible utilizar distintos modelos físicos en cada subdominio. Adicionalmente, se pueden utilizar diferentes esquemas de discretización y mallado en cada subdominio. Por ejemplo, en cada uno de ellos es posible utilizar un distinto núcleo radial.
- No es posible resolver numéricamente problemas reales en tres dimensiones con una sola computadora, debido a los requerimientos en tiempo y espacio. El método de descomposición de dominio combinado con un cluster de computadoras hace permisible la aproximación numérica a dichos problemas.
- La codificación en paralelo de esquemas basados en mallas equiespaciadas es relativamente fácil cuando se emplea descomposición de dominio.

Básicamente existen dos categorías para los métodos de descomposición de dominio que son con traslape y sin traslape. Los métodos de DD con traslape son conocidos con el nombre de Schwarz, que son métodos iterativos. Los métodos sin traslape se pueden dividir en métodos de substracción y métodos

de Schur. Los métodos con región de traslape y sin región de traslape difieren en que el último requiere resolver una ecuación adicional en la interface. En lo subsecuente trabajaremos exclusivamente con el método de Schwarz con coincidencia de nodos.

Una revisión de los métodos de descomposición de dominio se encuentra en los libros [104, 105, 106, 107]. Desde el punto de vista de implementación en [108] se describe la teoría y programación de ecuaciones diferenciales parciales con base en la biblioteca de interfaz de paso de mensajes (MPI [109]) como lenguaje de programación. Para mayor referencia de métodos de descomposición de dominio consulte la conferencia anual *International Conference on Domain Decomposition Methods* (<http://www.ddm.org>).

3.2. Formulación Clásica

El método de Schwarz data de 1870 [110] y fue originalmente diseñado como un método analítico para probar la existencia de funciones armónicas en regiones irregulares que son la unión de regiones con traslape.

Sea $\Omega \subset \mathbb{R}^2$ un dominio bidimensional y denotemos por $\partial\Omega$ la frontera del dominio. Se desea resolver la ecuación de Poisson

$$\begin{cases} \mathcal{L}u = f & \text{en } \Omega \\ u = g & \text{en } \partial\Omega \end{cases}, \quad (3.1)$$

donde \mathcal{L} es un operador diferencial lineal, en particular el Laplaciano ∇^2 .

Considere que Ω está dividido en dos subdominios Ω_1 y Ω_2 con una región de traslape de tal forma $\Omega = \Omega_1 \cup \Omega_2$; ver figura 3.1. Definiremos a $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ como la frontera artificial de Ω_1 , y $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$ como la frontera artificial de Ω_2 .

Observe que si los valores de la solución u en Γ_1 y Γ_2 son conocidos, entonces el problema (3.1) puede ser resuelto en cada subproblema definido en Ω_1 y Ω_2 independientemente.

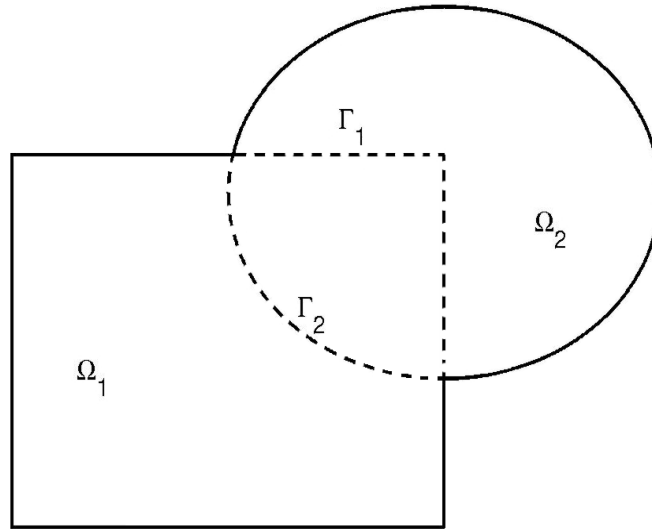


Figura 3.1: Dominio Ω particionado en dos subdominios con traslape.

El problema consiste en resolver la ecuación diferencial parcial (3.1) en cada región Ω_1 y Ω_2 , de tal modo que la aproximación numérica converja a la solución como si se considerara todo el dominio Ω . Para dar solución al planteamiento realizado, se muestra el método clásico de Schwarz también conocido como método multiplicativo.

$$\begin{cases} \mathcal{L}u_1^k = f & \text{en } \Omega_1 \\ u_1^k = g & \text{en } \partial\Omega_1 \setminus \Gamma_1 \\ u_1^k = u_2^{k-1} & \text{en } \Gamma_1 \end{cases}$$

y

(3.2)

$$\begin{cases} \mathcal{L}u_2^k = f & \text{en } \Omega_2 \\ u_2^k = g & \text{en } \partial\Omega_2 \setminus \Gamma_2 \\ u_2^k = u_1^k & \text{en } \Gamma_2 \end{cases}$$

El planteamiento anterior es un proceso iterativo: comenzando con un valor inicial dado $u_1^0 \in \Gamma_1$ resolvemos el problema en Ω_1 obteniendo la solución u_1^0 , con base en la solución obtenida actualizamos el valor en la frontera artificial Γ_2

para aproximar la solución en $u_2^0 \in \Omega_2$, con la solución obtenida actualizamos el valor en la frontera artificial Γ_1 . El proceso anterior se repite de forma iterativa hasta alcanzar la convergencia, cuando el error entre iteraciones consecutivas sea menor que un valor dado.

Este desarrollo fue previo a la aparición de la primera computadora electrónica en 1947 (Electronic Numerical Integrator And Calculator ENIAC), no fue pensado como un método para procesamiento en paralelo. Sin embargo, la influencia del trabajo de Schwarz es hoy en día uno de los pilares en la solución numérica de ecuaciones diferenciales en paralelo.

Dentro del campo de estudio de la solución numérica de ecuaciones diferenciales parciales con descomposición de dominio, mediante el método de colocación asimétrico (MCA), existen pocos trabajos realizados a la fecha. Los principales trabajos se muestran a continuación en forma de listado.

- Un trabajo previo se encuentra en Dubal (1994) [111], emplea descomposición de dominio con el núcleo multicuádrico para aproximar una EDP de segundo orden en $1-d$. Dubal concluye que con este enfoque se pueden resolver problemas elípticos de manera eficiente.
- La justificación teórica del algoritmo aditivo de Schwarz con funciones radiales de soporte compacto se desarrollo en Hon & Wu (2000) [112].
- Al dividir el problema en regiones más pequeñas el mal condicionamiento del sistema es reducido, con lo que es posible obtener una mejor aproximación numérica, y se incrementa el desempeño computacional; ver Kansa & Hon (2000) [43] y Wong et al. (1999) [44].
- La aplicación del algoritmo de Schwarz, con regiones de traslape en las variantes aditiva y multiplicativa, en la solución a un problema elíptico muestra las siguientes características: a) el número de iteraciones desciende conforme se incrementa el ancho en la región de traslape, b) el número de iteraciones para el método aditivo es más sensible al tamaño

de la región de traslape que lo obtenido con la versión multiplicativa, c) el número de iteraciones en la versión multiplicativa es aproximadamente la mitad del número de iteraciones requeridas en la versión aditiva para que el algoritmo converja, y d) conforme se incrementa el número de subdominios se decrementa la tasa de convergencia y se obtiene una solución numérica menos exacta; ver Zhou et al. (2003) [113]

- La versión aditiva del algoritmo de Schwarz aplicado a un problema de tipo convección-difusión estacionario, muestra numéricamente que las funciones radiales $r^6 \log r$, r^5 y r^7 superan en exactitud y son numéricamente más estables que lo obtenido con $r^4 \log r$ y $r^8 \log r$; ver Jichun et al. (2003) [114]
- El algoritmo clásico de Schwarz junto con un preconditionador basado en la aproximación con funciones cardinales, mejoran el desempeño computacional y el número de condición; ver Ling & Kansa (2004) [74].
- Un estudio numérico con la versión aditiva y multiplicativa (con correspondencia y sin correspondencia de nodos), muestra que cuando no existe correspondencia de puntos en las fronteras se obtiene un error similar a lo obtenido con correspondencia de puntos. Adicionalmente, para el problema descrito en el libro de Smith [104]-página 88, muestran que los resultados obtenidos son superiores a lo obtenido con diferencias finitas y elemento finito. Por último, el número de iteraciones para la versión multiplicativa en comparación con la aditiva es aproximadamente la mitad; ver Li & Hon (2004) [115].
- La convergencia en la solución para dos subdominios, es mostrada en Zhang (2006) [116]. Requiere que el sistema sea invertible, por lo tanto no es aplicable al esquema de colocación asimétrico.
- Para problemas dependientes del tiempo con el esquema de colocación asimétrico y descomposición de dominio, hasta donde tenemos conoci-

miento solo existen tres trabajos realizados a la fecha [44, 117, 100]. En Wong et al. (1999) [44], con base en el núcleo multicuádrico y con un método de descomposición con multiples-zonas aplicado a un problema de hidrodinámica lineal y no lineal en dos dimensiones, muestran una reducción entre el 40 % - 50 % en el tiempo de procesamiento, manteniendo un error de aproximación similar empleando para ello 5 y 7 particiones. Recientemente, Chinchapatnam et al. (2007) [100] aborda el esquema planteado en este capítulo y lo aplica a una ecuación de tipo convección-difusión dependiente del tiempo empleando el método θ . Realizan un estudio numérico para determinar la influencia que tiene el número de subdominios en el tiempo de procesamiento y la exactitud obtenida, concluyendo que conforme se incrementa el número de subdominios se reduce el costo computacional, pero al mismo tiempo la exactitud disminuye [100]. Este resultado fue también determinado en el presente trabajo en la sección 3.4 y fue publicado por nosotros en (2006) [117]. Es importante destacar que ambos trabajos fueron desarrollados de manera independiente y ambos dan evidencia de la factibilidad del esquema analizado.

En la siguiente sección se aplica el método de Schwarz a problemas lineales dependientes del tiempo. El énfasis radica en el uso del método de colocación asimétrico.

3.3. Lineal Temporal

En esta sección se expone el algoritmo de Schwarz con base en el método de colocación asimétrico para problemas lineales temporales [117, 100]. Posteriormente, se realiza su correspondiente análisis de complejidad y se ejemplifica el esquema desarrollado para un caso particular. Por último, se describe el esquema de partición de datos.

Considere la siguiente ecuación diferencial parcial dependiente del tiempo de la forma

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \mathcal{L}u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad (3.3)$$

con la condición de frontera e inicial

$$\mathcal{B}u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \quad (3.4)$$

$$u(\mathbf{x}, t) = u_0(\mathbf{x}), \quad t = 0, \quad (3.5)$$

donde Ω es el dominio en \mathbb{R}^d , $\partial\Omega$ denota la frontera de Ω . El operador diferencial lineal \mathcal{L} representa la aplicación de las derivadas parciales sobre $u(\mathbf{x}, t)$, \mathcal{B} es el operador en donde imponemos las condiciones de frontera de tipo Dirichlet, Newman o Robin. Las funciones $f(\mathbf{x}, t)$, $g(\mathbf{x}, t)$ y $u_0(\mathbf{x})$ son conocidas.

Discretizando hacia adelante en el tiempo y con el esquema clásico θ , $0 \leq \theta \leq 1$, las ecuaciones (3.3, 3.4) quedan expresadas

$$u(\mathbf{x}, t + \Delta t) - u(\mathbf{x}, t) = \Delta t f(\mathbf{x}, t + \Delta t) - \Delta t [\theta \mathcal{L}u(\mathbf{x}, t + \Delta t) + (1 - \theta) \mathcal{L}u(\mathbf{x}, t)],$$

$$\mathcal{B}u(\mathbf{x}, t + \Delta t) = g(\mathbf{x}, t + \Delta t),$$

donde Δt denota al tamaño del paso en el tiempo. Con base en la notación $u(\mathbf{x}, t^n) = u^n$ y $t^n = t^{n-1} + \Delta t$, las dos ecuaciones anteriores quedan definidas como

$$u^{n+1} + \Delta t \theta \mathcal{L}u^{n+1} = u^n + \Delta t f^{n+1} - \Delta t (1 - \theta) \mathcal{L}u^n, \quad (3.6)$$

$$\mathcal{B}u^{n+1} = g^{n+1}. \quad (3.7)$$

Dependiendo del valor del parámetro θ , se obtiene un método explícito $\theta = 0$, ó un esquema implícito $\theta = 0.5$.

La solución exacta $u(\mathbf{x}, t)$ del problema de valor inicial descrito anteriormente es aproximada $\tilde{u}(\mathbf{x}, t)$ mediante una combinación lineal de funciones radiales (estrictamente positiva definida), lo cual se puede expresar de forma matricial como

$$\tilde{u}(\mathbf{x}, t) = A \cdot \lambda(t), \quad (3.8)$$

siendo $\lambda(t)$ el vector de incógnitas y $A_{i,j} = \{\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)\}_{1 \leq i, j \leq N}$. Como se ha mencionado a lo largo de la tesis, el dominio Ω es discretizado mediante N nodos de colocación $X = \{\mathbf{x}_i\}_{i=1}^N$ los cuales están divididos en nodos al interior $X_I = \{\mathbf{x}_i\}_{i=1}^{N_I}$ y los nodos que conforman la frontera $X_f = \{\mathbf{x}_i\}_{i=N_I+1}^N$, siendo $X = X_I \cup X_f$ y $N_f = N - N_I$. Substituyendo la aproximación radial $\tilde{u}(\mathbf{x}, t)$ en (3.6, 3.7) obtenemos

$$\begin{bmatrix} \Phi_a + \theta \Delta t \mathcal{L} \Phi_a \\ \mathcal{B} \Phi_b \end{bmatrix} \lambda^{n+1} = \begin{bmatrix} \Phi_a - (1 - \theta) \Delta t \mathcal{L} \Phi_a \\ 0_b \end{bmatrix} \lambda^{n+1} + \begin{bmatrix} f^{n+1} \\ g^{n+1} \end{bmatrix}, \quad (3.9)$$

donde $\lambda \in \mathbb{R}^N$ y $0_b \in \mathbb{R}^{N_f \times N}$. La aplicación del operador diferencial \mathcal{L} sobre el conjunto de nodos interiores es la matriz $\mathcal{L} \Phi_a \in \mathbb{R}^{N_I \times N}$ donde la matriz $\Phi_a \in \mathbb{R}^{N_I \times N}$ representa a una porción de la matriz de Gram $A = [\Phi_a \ \Phi_b]^T \in \mathbb{R}^{N \times N}$. La aplicación del operador diferencial \mathcal{B} sobre los nodos que conforman la frontera está definida por $\mathcal{B} \Phi_b \in \mathbb{R}^{N_f \times N}$. Por último, $f \in \mathbb{R}^{N_I}$ y $g \in \mathbb{R}^{N_f}$ son las funciones fuente y frontera.

La ecuación anterior representa el esquema de colocación asimétrico, el cual como vimos en el capítulo 2 se puede resolver de manera iterativa desde $t = 0$ hasta t_{max} para determinar la solución $\tilde{u}(\mathbf{x}, t)$ en cualquier punto del dominio Ω .

Sin pérdida de generalidad, para explicar el método de Schwarz para problemas lineales temporales, consideraremos dos subdominios como los descritos en la sección 3.2. Con base en esta consideración ahora podemos aplicar el método de Schwarz en la versión aditiva a cada subdominio discretizado mediante (3.6, 3.7).

$$\begin{cases} u_1^{n+1} + \Delta t \theta \mathcal{L} u_1^{n+1} & = u_1^n - \Delta t (1 - \theta) \mathcal{L} u_1^n + \Delta t f^{n+1} & \text{en } \Omega_1 \\ \mathcal{B} u_1^{n+1} & = g^{n+1} & \text{en } \partial \Omega_1 \setminus \Gamma_1 \\ u_{1,s}^{n+1} & = u_{2,s-1}^{n+1} & \text{en } \Gamma_1 \end{cases}$$

y

$$\begin{cases} u_2^{n+1} + \Delta t \theta \mathcal{L}u_2^{n+1} &= u_2^n - \Delta t(1 - \theta)\mathcal{L}u_2^n + \Delta t f^{n+1} & \text{en } \Omega_2 \\ \mathcal{B}u_2^{n+1} &= g^{n+1} & \text{en } \partial\Omega_2 \setminus \Gamma_2 \\ u_{2,s}^{n+1} &= u_{1,s-1}^{n+1} & \text{en } \Gamma_2 \end{cases}$$

Manteniendo fijo n , la idea es resolver cada subdominio por separado y con la solución obtenida $u_{1,2,s}^{n+1}$ actualizamos los nodos que conforman la frontera artificial $\Gamma_{1,2}$ en la siguiente iteración sobre s . Este proceso se realiza de manera iterativa hasta que se satisfaga la condición de convergencia. La cual la definimos como:

$$\|u_{1,s}^{n+1} - u_{2,s-1}^{n+1}\|_\infty < \epsilon \quad \text{y} \quad \|u_{2,s}^{n+1} - u_{1,s-1}^{n+1}\|_\infty < \epsilon, \quad (3.10)$$

con $\epsilon > 0$, cuando se satisface la condición anterior se avanza en el tiempo $t = t + \Delta t$. Es importante resaltar que en cada paso del tiempo se aplica el método de Schwarz, y por ello el tiempo total de la simulación numérica depende de dos factores: el tamaño del paso temporal Δt y el número de iteraciones para que se satisfaga la condición de paro. Por lo tanto, para disminuir el tiempo total de la simulación es necesario reducir ambos factores. Esto es posible realizarlo con el esquema implícito en unión con la versión multiplicativa, debido a que el método implícito no tiene restricción en el tamaño del paso temporal y la versión multiplicativa requiere de aproximadamente la mitad de iteraciones en comparación con la versión aditiva. Por otro lado, la versión aditiva es más fácil de programar en un enfoque paralelo en comparación con la versión multiplicativa.

En los siguientes párrafos, describiremos la aplicación del método de Schwarz en unión con el método θ (3.9). Para ello, considere que tenemos dos subdominios como los descritos en la sección 3.2, y para el tiempo inicial $t = 0$ se tiene aproximada la condición inicial $A^1 \lambda_1^0 = u_0$ en Ω_1 y $A^2 \lambda_2^0 = u_0$ en Ω_2 , siendo A la matriz de Gram. La versión multiplicativa queda definida como:

$$H_1^+ \lambda_{1,i}^{n+1} = H_1^- \lambda_1^n + F_1^{n+1} + S_{2,i-1}^{n+1} \quad (3.11)$$

y

$$H_2^+ \lambda_{2,i}^{n+1} = H_2^- \lambda_2^n + F_2^{n+1} + S_{1,i}^{n+1}, \quad (3.12)$$

donde n denota el paso en la iteración temporal, el primer subíndice de cada matriz $k = \{1, 2\}$ representa el número del subdominio, el subíndice i denota la iteración del algoritmo de Schwarz en cada paso temporal, $\lambda_{k,i}^{n+1}$ es el vector de incógnitas a determinar en cada iteración. Las matrices H_k^+ , H_k^- y F_k^{n+1} están determinadas por

$$H_k^+ = \begin{bmatrix} \Phi_a^k + \theta \Delta t \mathcal{L} \Phi_a^k \\ \mathcal{B} \Phi_b^k \\ \Phi_c^k \end{bmatrix}, \quad H_k^- = \begin{bmatrix} \Phi_a^k - (1 - \theta) \Delta t \mathcal{L} \Phi_a^k \\ 0 \\ 0 \end{bmatrix}$$

y

$$F_k^{n+1} = \begin{bmatrix} \Delta t f_k^{n+1} \\ g_k^{n+1} \\ 0 \end{bmatrix},$$

donde Φ_a^k es la representación matricial radial de los nodos al interior de cada subdominio Ω_k , Φ_b^k corresponde a la matriz radial de los nodos que conforman la frontera de cada subdominio menos los elementos en la frontera artificial y Φ_c^k representa la matriz radial de los nodos que conforman la frontera artificial Γ_k . Finalmente las matrices $S_{1,i-1}^{n+1}$ y $S_{1,i}^{n+1}$ están definidas como:

$$S_{2,i-1}^{n+1} = \begin{bmatrix} 0 \\ 0 \\ \Phi_a^2 \end{bmatrix} \lambda_{2,i-1}^{n+1}, \quad S_{1,i}^{n+1} = \begin{bmatrix} 0 \\ 0 \\ \Phi_a^1 \end{bmatrix} \lambda_{1,i}^{n+1},$$

esto representa el proceso de interpolación para obtener la solución al interior de cada subdominio y sirve para actualizar los valores en la frontera artificial en el proceso iterativo de Schwarz.

El proceso iterativo del esquema clásico de Schwarz para problemas temporales con funciones de base radial lo podemos expresar como: al comienzo del proceso iterativo $i = 1$ obtenemos la solución en el primer subdominio (3.11),

esto corresponde a resolver un sistema lineal de ecuaciones para determinar $\lambda_{1,i}^{n+1}$, con base en la solución obtenida interpolamos $S_{1,i}^{n+1}$ para determinar el valor de la solución en los nodos que conforman la frontera artificial en Ω_2 , después aproximamos la solución en el segundo subdominio (3.12) obteniendo el vector $\lambda_{2,i}^{n+1}$, incrementamos el índice i y a partir de la solución obtenida $\lambda_{2,i}^{n+1}$ interpolamos la solución en el segundo subdominio $S_{2,i-1}^{n+1}$ para determinar la solución en los nodos que conforman la frontera artificial en el primer subdominio. El proceso anterior se itera mientras no se satisfaga la condición de paro (3.10) con $\epsilon = 5 \times 10^{-3}$ ó se alcance el número máximo de iteraciones $i \leq 300$. Una vez alcanzada la convergencia en la solución entre ambos subdominios avanzamos en el tiempo $t = t + \Delta t$ y volvemos a realizar el proceso iterativo anterior.

Note que las matrices H_k^+ permanecen constantes a lo largo de las iteraciones, por lo tanto es posible emplear la factorización LU al comienzo de la simulación. Con base en el esquema explícito $\theta = 0$ con el operador \mathcal{B} como la matriz identidad, se obtiene en H_k^+ la matriz de Gram.

La extensión del algoritmo de Schwarz para P subdominios es similar a lo mostrado anteriormente y queda expresado como

$$H_k^+ \lambda_{k,i}^{n+1} = H_k^- \lambda_k^n + F_k^{n+1} + S_{k,i-1}^{n+1}, \quad (3.13)$$

donde $k \in \{1, 2, \dots, P\}$ denota el índice del subdominio, n indica la iteración temporal que varía en el rango $n = \{0, 1, \dots, n_{max}\}$ con $n_{max} = t_{max}/\Delta t$. Las matrices H_k^+, H_k^-, F_k^{n+1} fueron descritas anteriormente. La matriz $S_{k,i-1}^{n+1}$ corresponde a la actualización del valor en la frontera artificial Γ_k

$$\Gamma_k \leftarrow \Upsilon_j, \quad j \in N_k, \quad (3.14)$$

donde N_k representa al conjunto de índices de los subdominios vecinos al subdominio k . Υ_j corresponde a los nodos en el interior de Ω_j que nos sirven para actualizar la frontera artificial de Γ_k . Observe que en cada avance temporal se modifican los valores en $\lambda_k^{n+1}, F_k^{n+1}$. Por otro lado, en cada iteración del algoritmo de Schwarz se mantienen fijas las matrices λ_k^n, F_k^{n+1} y lo que se varía es

el vector solución y la actualización del valor en la frontera $\lambda_{k,i}^{n+1}, S_{k,i-1}^{n+1}$, donde el índice $i = 0, 1, \dots$, indica la iteración del algoritmo de descomposición de dominio.

3.3.1. Análisis de Complejidad

En el análisis de complejidad nos interesa determinar como influye el número de particiones en el tiempo total de la simulación.

Para simplificar el problema se asume lo siguiente: se cuenta con un solo procesador en donde se corren todos los procesos, la convergencia del algoritmo de Schwarz es $O(1)$, el intercambio de información entre los subdominios es instantánea y el proceso de interpolación para determinar la solución en los nodos que conforman la frontera artificial de cada subdominio es de tiempo constante.

Considere que el dominio Ω es particionado en P subdominios y en cada subdominio se tiene aproximadamente el mismo número de nodos $|\Omega_i| \approx N/P$ con $i = 1, \dots, P$. En cada partición Ω_i se requiere resolver un sistema lineal de ecuaciones

$$H\lambda^{n+1} = P\lambda^n, \quad (3.15)$$

donde $H, P \in \mathbb{R}^{M \times M}$ y $\lambda \in \mathbb{R}^M$ con $M = N/P$. La factorización LU de la matriz H tiene una complejidad en tiempo $O(M^3)$ lo cual se realiza en una sola ocasión. Por otro lado, la multiplicación $P\lambda^n$ y la solución del sistema lineal de ecuaciones $[L, U]\lambda^{n+1} = P\lambda^n$, comparten el mismo orden de complejidad $O(M^2)$. Por lo tanto, para los P subdominios el orden de complejidad está determinado como:

$$O(PM^3) + O(PM^2),$$

substituyendo el valor de M se obtiene

$$O\left(\frac{N^3}{P^2}\right) + O\left(\frac{N^2}{P}\right),$$

dado que el análisis de complejidad es para una ecuación diferencial parcial dependiente del tiempo, introducimos el número de pasos requeridos en la simulación $\tau = t_{max}/\Delta t$ donde t_{max} denota el tiempo máximo y Δt el incremento temporal. Por lo anterior, el orden de complejidad queda expresado como:

$$O\left(\frac{N^3}{P^2}\right) + O\left(\tau \frac{N^2}{P}\right). \quad (3.16)$$

En el primer caso, lado izquierdo de (3.16), el tiempo de procesamiento descende cuadráticamente conforme se incrementa el número de particiones P , esto representa una gran diferencia con la complejidad $O(N^3)$ que corresponde a no considerar descomposición de dominio. Analizando el segundo caso, lado derecho de (3.16), su complejidad numérica descende de forma lineal conforme se incrementa el número de particiones.

Para determinar el orden de complejidad en función del número de subdominios P , manteniendo fijo el número de nodos N se tiene un decrecimiento lineal u orden de complejidad lineal en función del número de subdominios. Esto se debe a que el orden de complejidad en (3.16) está determinado por el exponente mínimo en P . En la sección 3.4.1 se utiliza $\tau \approx N$, substituyendo dicha constante en (3.16) se obtiene

$$O\left(\frac{N^3}{P}\right), \quad (3.17)$$

en donde claramente se observa que el tiempo de procesamiento total decrece más lentamente (linealmente) en función del número de subdominios P .

El requerimiento en espacio es $O\left(\frac{N^2}{P}\right)$, lo cual indica una disminución lineal en la memoria requerida conforme se incrementa el número de subdominios.

El análisis realizado sugiere elegir el número de particiones P lo más grande posible para disminuir la complejidad en el tiempo y en el espacio. Sin embargo, hay que considerar que se quiere resolver una ecuación diferencial parcial donde el dominio continuo es discretizado mediante una serie de nodos y es deseable que el número de nodos por partición sea elegido en función de las características de la ecuación diferencial parcial.

3.3.2. Caso Particular $P = 1$

En la sección 3.3 se describió la aplicación del método de Schwarz a problemas lineales temporales. Ahora, con fines de esclarecer los detalles omitidos, se desarrolla en esta sección la construcción de las matrices de las derivadas parciales y la matriz de interpolación radial para un caso particular.

Considere la ecuación diferencial parcial definida al comienzo de la sección 3.4.1, la aproximación $\tilde{u}(\mathbf{x}, t)$ de la solución exacta $u(\mathbf{x}, t)$ es expresada como:

$$\tilde{u}(\mathbf{x}, t) = \sum_{j=1}^N \lambda_j(t) \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{i=1}^M \mu_i(t) p_i(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2, \quad (3.18)$$

donde $\phi(r) = r^4 \log r$, p está en el espacio de todos los polinomios en dos dimensiones de orden 6, la base de dicho polinomio es $\pi_2^2 = \text{gen}\{1, x, y, xy, x^2, y^2\}$. Los vectores de incógnitas $\lambda(t) \in \mathbb{R}^N$ y $\mu(t) \in \mathbb{R}^M$ son dependientes del tiempo.

El dominio Ω es discretizado mediante N nodos de colocación divididos en nodos al interior X_I y los nodos que conforman la frontera X_f , donde $X = X_I \cup X_f$ y $N = N_I + N_f$. La aproximación de la condición inicial $u_0(\mathbf{x})$ del problema temporal analizado queda expresada en forma matricial

$$\begin{bmatrix} \Phi_a & P_a \\ \Phi_b & P_b \\ P_a^T P_b^T & 0 \end{bmatrix} \begin{bmatrix} \lambda^0 \\ \mu^0 \end{bmatrix} = \begin{bmatrix} u_0(\mathbf{x}) \\ u_0(\mathbf{x}) \\ 0 \end{bmatrix} \quad \begin{array}{l} \mathbf{x} \in \Omega \\ \mathbf{x} \in \partial\Omega \\ \end{array}, \quad (3.19)$$

donde $\Phi_a \in \mathbb{R}^{N_I \times N}$, $P_a \in \mathbb{R}^{N_I \times M}$, $\Phi_b \in \mathbb{R}^{N_f \times N}$ y $P_b \in \mathbb{R}^{N_f \times M}$. Observe que el sistema lineal de ecuaciones es un problema de interpolación con funciones de base radial el cual es invertible.

Antes de mostrar la representación con funciones radiales del esquema numérico para $t > 0$, describiremos la aproximación de las primeras derivadas parciales sobre $\tilde{u}(\cdot)$. A partir de (3.18) las primeras derivadas en x, y que actúan sobre los nodos que conforman el interior del dominio Ω quedan definidas como:

$$\frac{\partial \tilde{u}(\mathbf{x}, t)}{\partial x} = \sum_{j=1}^N \lambda_j(t) \frac{\partial \phi(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial x} + \sum_{i=1}^M \mu_i(t) \frac{\partial p(\mathbf{x})}{\partial x},$$

$$\frac{\tilde{u}(\mathbf{x}, t)}{\partial y} = \sum_{j=1}^N \lambda_j(t) \frac{\partial \phi(\|\mathbf{x} - \mathbf{x}_j\|)}{\partial y} + \sum_{i=1}^M \mu_i(t) \frac{\partial p(\mathbf{x})}{\partial y},$$

denotando con $p_x = \frac{\partial p(\mathbf{x})}{\partial x}$ y $p_y = \frac{\partial p(\mathbf{x})}{\partial y}$ ambas derivadas se pueden expresar en forma vectorial

$$p_x = [0, 1, 0, y, 2x, 0]^T \quad \text{y} \quad p_y = [0, 0, 1, x, 0, 2y]^T,$$

donde T denota la transpuesta. Las derivadas que actúan sobre $\phi(\cdot)$ son definidas como:

$$\frac{\partial \phi(\|\mathbf{x} - \mathbf{x}_i\|)}{\partial x} = \phi_x = (x - x_i) \cdot r \cdot (1 + 2 \cdot \log r),$$

$$\frac{\partial \phi(\|\mathbf{x} - \mathbf{x}_i\|)}{\partial y} = \phi_y = (y - y_i) \cdot r \cdot (1 + 2 \cdot \log r).$$

Con base en el esquema $\theta = 0$, con una discretización hacia adelante en el tiempo y definiendo $\mathcal{L} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ en (3.9), el esquema de colocación asimétrico aplicado al problema de valor inicial analizado queda expresado de forma matricial como:

$$H_+ \bar{\lambda}^{n+1} = (H_- - \Delta t H_-^x - \Delta t H_-^y) \bar{\lambda}^n, \quad (3.20)$$

donde $\bar{\lambda}^n = [\lambda^n, \mu^n]^T$, las matrices H_- , H_-^x y H_-^y están determinadas por

$$H_- = \begin{bmatrix} \Phi_a & Pa \\ 0_b & \\ 0 & \end{bmatrix}, \quad H_-^x = \begin{bmatrix} \Phi_{a,x} & P_{a,x} \\ 0_b & \\ 0 & \end{bmatrix}, \quad H_-^y = \begin{bmatrix} \Phi_{a,y} & P_{a,y} \\ 0_b & \\ 0 & \end{bmatrix},$$

la matriz H_+ corresponde a la matriz de Gram (3.19). Las matrices $\Phi_{a,x}$ y $P_{a,x}$ corresponden a la aproximación radial de la primera derivada en x del interpolante $\tilde{u}(\cdot)$, de manera similar para las matrices $\Phi_{a,y}$ y $P_{a,y}$. Las dos aproximaciones anteriores de las derivadas son calculadas en los nodos que conforman el interior del dominio Ω . La matriz de ceros $0_b \in \mathbb{R}^{N_f \times N}$ indica que el valor en la frontera $\partial\Omega$ es de un valor constante igual a cero. La otra matriz de ceros $0 \in \mathbb{R}^M$ es requerida en el problema de interpolación.

El sistema iterativo (3.20) es iterado desde $n = 0$ hasta $n = t_{max}/\Delta t$, obteniendo los vectores de incógnitas $\{\lambda(t), \mu(t)\}$. A partir de dichos vectores se puede reconstruir la solución mediante (3.18).

3.3.3. Partición de Datos

En esta sección se describe el esquema de partición de datos para nodos distribuidos de manera uniforme.

Dado un conjunto de N nodos distribuidos de manera uniforme, el esquema de partición de datos consiste en particionar el dominio Ω mediante una cuadrícula de $P = M \times M$, $M \geq 2$. En cada región se tiene aproximadamente el mismo número de nodos con una intersección vacía entre los subdominios. Posteriormente, se extiende la región de cada subdominio para conformar la zona traslape.

En la figura 3.2 se muestra la zona de traslape de 2 nodos para datos equiespaciados en una partición de 2×2 . Como se aprecia de la figura 3.2, el subdominio 1 tiene colindancia con los subdominios vecinos $\{\Omega_2, \Omega_3, \Omega_4\}$. En general, cada región Ω_i es colindante con sus 8 vecinos naturales; arriba, abajo, derecha, izquierda, arriba-derecha, arriba-izquierda, abajo-derecha y abajo-izquierda.

Una representación gráfica de las categorías que conforman cada subdominio se muestra en la figura 3.3. Esto corresponde a la región $(0, 0)$ de una partición de 5×5 en una rejilla equiespaciada de 75×75 nodos. En general cada subdominio está compuesto por:

$$\overline{\Omega}_i = \{\Omega_i, \partial\Omega_i, \Gamma_i, \Upsilon_i\},$$

donde Ω_i corresponde a los nodos que conforman la región en donde queremos conocer la solución del problema, $\partial\Omega_i$ denota los nodos que conforman la frontera real del problema ($\partial\Omega_i \in \partial\Omega$), Γ_i corresponde a los nodos que conforman la frontera artificial del problema, Υ_i son los nodos interiores que nos sirven para actualizar el valor de la frontera artificial en los subdominios vecinos.

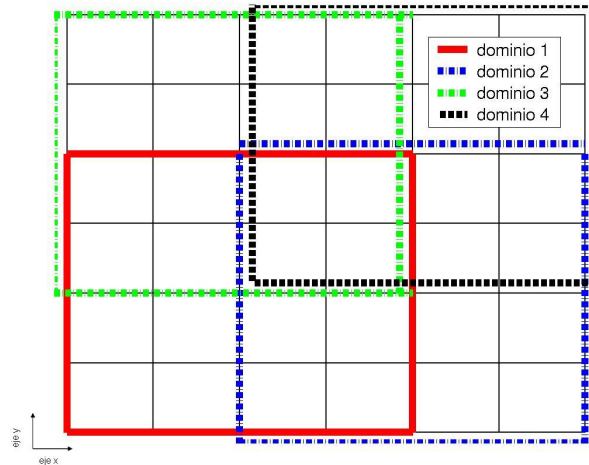


Figura 3.2: Zona de traslape de una partición 2×2 para datos equiespaciados.

La determinación de la zona de traslape es un requerimiento trivial cuando se trabaja con nodos distribuidos de manera uniforme en un dominio rectangular. En el capítulo 5 se propone un algoritmo eficiente para conformar las zonas de traslape con nodos distribuidos de manera no uniforme.

3.4. Resultados Numéricos

En esta sección investigamos el efecto que tiene el número de subdominios en el tiempo de procesamiento y la exactitud obtenida. Para ello, analizamos una ecuación diferencial parcial lineal dependiente del tiempo en dos dimensiones, discretizada mediante el esquema de colocación asimétrico.

La codificación se realizó en el lenguaje de programación C con el compilador GCC 3.3.1 con datos en precisión doble, en una computadora dedicada Intel Xeon a 3 GHz con 1 Gb en RAM bajo el ambiente Linux 2.4.21-144-smp4G, ubicada en el cluster virtual del INAOEP. Los tiempos reportados se obtuvieron con la instrucción ‘time ./ddm2d.o’ que incluye el proceso de generación de nodos, partición del dominio y la solución numérica de la ecuación diferencial

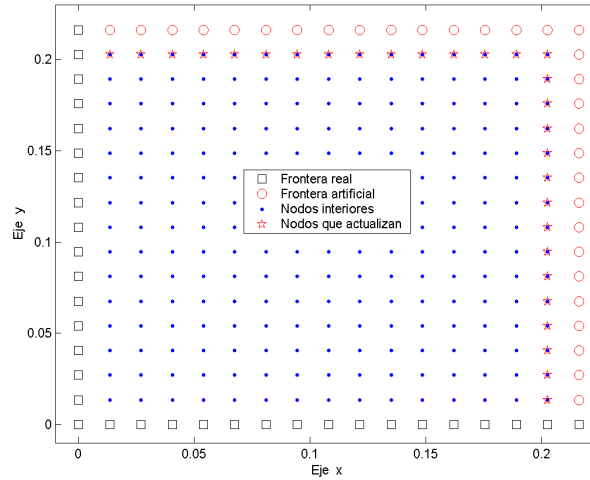


Figura 3.3: Categorías de nodos presentes en cada subdominio.

parcial.

3.4.1. Convección en 2-d

Considere la siguiente ecuación convectiva en dos dimensiones

$$\begin{aligned} \frac{\partial u(\mathbf{x}, t)}{\partial t} + v \cdot \nabla u(\mathbf{x}, t) &= 0, & \mathbf{x} \in \Omega \subset \mathbb{R}^2, \\ u(\mathbf{x}, t) &= 0, & \mathbf{x} \in \partial\Omega, \end{aligned} \quad (3.21)$$

definida en $[0, 1]^2$, donde $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ es el operador diferencial del gradiente, $v = (v_1, v_2)^T$ es el vector de velocidades, en particular seleccionamos el vector unitario. Como condición inicial se tiene

$$u(\mathbf{x}, t = 0) = e^{-\frac{r^2}{\sigma}}, \quad (3.22)$$

siendo $r^2 = (x - 0.3)^2 + (y - 0.3)^2$ con $\sigma = 0.01$, que corresponde a una función gaussiana centrada en $(0.3, 0.3)$. La solución analítica del problema anterior queda expresada en función de los datos iniciales $u_o(\mathbf{x}) = u(\mathbf{x}, t = 0)$

$$u(\mathbf{x}, t) = u_o(\mathbf{x} - t), \quad (3.23)$$

esto corresponde a una translación de la condición inicial con respecto del tiempo. Es decir, la solución se propaga a lo largo de las características [118].

En la figura 3.4 se muestra la forma de la condición inicial (3.22). El valor máximo de la función es alcanzado en el punto $(0.3, 0.3)$ con un valor de 2^0 y conforme nos alejamos de dicho punto la función tiende a cero.

Como datos iniciales se tiene una rejilla equiespaciada de 75×75 nodos, con lo cual se obtienen $N = 5625$ nodos distribuidos en $N_I = 5329$ nodos interiores y $N_f = 296$ como nodos frontera. La función de base radial seleccionada es placa delgada, el tiempo máximo de simulación es $t_{max} = 0.4$ con un tamaño en el paso temporal $\Delta t = 0.0001$, donde Δt satisface la restricción de estabilidad $\Delta t < \alpha(2/\lambda_M)$, con $\alpha = 0.01$ (ver sección 2.3.1). La elección de placa delgada se debe a que es una función radial libre de parámetros, y nos permite concentrarnos solamente en el número de particiones P como parámetro a investigar.

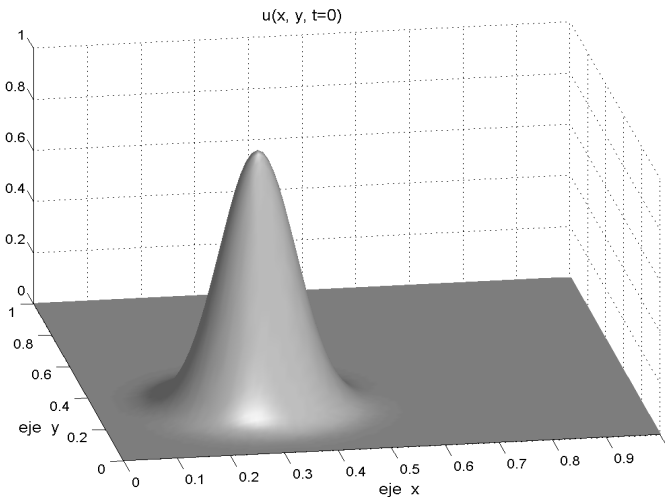


Figura 3.4: Condición inicial $u(\mathbf{x}, t = 0) = e^{-\frac{r^2}{\sigma}}$.

En la tabla 3.1 se muestra el resultado obtenido para $P = 1$, lo que corresponde a un esquema sin descomposición de dominio (ver sección 3.3.2). La máxima diferencia en valor absoluto entre la solución numérica y la exacta se ubica en el centro de la gaussiana al tiempo $t_{max} = 0.4$. Nótese que el error

cuadrático medio (ECM) es de orden 10^{-4} lo cual indica una alta exactitud en la aproximación numérica. El tiempo de procesamiento fue de 1 hora 45 minutos. El par de matrices requeridas son de dimensiones 5625×5625 , y dado que utilizamos precisión doble (8 bytes) se requirió de 506 Mb (Mega-bytes) en memoria.

ECM	0.000866
E_{max}	0.008063
Tiempo segs.	6335

Tabla 3.1: Error cometido sin descomposición de domino.

En este caso no se obtuvo un mal condicionamiento. Sin embargo, esto representa un caso particular. El tiempo de procesamiento y la memoria requerida limitan el uso directo del método de colocación asimétrico para problemas de gran escala.

Por lo expuesto en los párrafos anteriores, se requiere el uso de descomposición de dominio para mejorar el desempeño del método de colocación asimétrico. En particular, emplearemos la versión multiplicativa del algoritmo de Schwarz con una zona de traslape de 2 nodos coincidentes, esto corresponde a una zona con el mínimo traslape.

3.4.2. Incremento en el Número de Subdominios

En esta sección se investiga el efecto que tiene el número de particiones con respecto del tiempo de procesamiento y la exactitud de la solución obtenida.

La partición del dominio Ω está basada en una cuadrícula $P = M \times M$, donde P está en el rango $\{1, 2 \times 2, \dots, 12 \times 12\}$; ver sección 3.3.3. En la tabla 3.2 se muestran los resultados obtenidos, en la primera columna se exhibe el incremento en el número de subdominios, en la segunda y tercera columna se despliega el error en la norma infinita y el error cuadrático medio, ambas

medidas de error son calculadas sobre todo el dominio Ω . Finalmente, en la cuarta columna se indica el tiempo de procesamiento expresado en segundos.

Subdominios	E_{max}	ECM	Tiempo segs.
1	0.008063	0.000866	6335
2×2	0.006528	0.000711	739
3×3	0.007274	0.000922	298
4×4	0.006914	0.000850	181
5×5	0.005699	0.000863	121
6×6	0.005730	0.000928	101
7×7	0.006836	0.000895	71
8×8	0.007833	0.001032	69
9×9	0.008213	0.001249	56
10×10	0.009314	0.001330	49
11×11	0.009247	0.001257	45
12×12	0.023595	0.002028	43

Tabla 3.2: Comparación entre el número de particiones vs. tiempo y error.

El primer renglón en la tabla 3.2 corresponde al resultado obtenido anteriormente cuando se considero todo el dominio sin particiones, ver tabla 3.1.

Como se observa en la segunda y tercera columna en la tabla 3.2 ambas medidas de error se mantienen en el mismo orden de magnitud para $M \leq 7$, para valores posteriores el error comienza ligeramente a crecer. Sin embargo, el error de aproximación solo se incrementa un orden de magnitud lo cual para ciertas aplicaciones puede resultar suficiente.

El incremento en el número de subdominios ocasiona que a partir de un cierto valor P_o el error en la aproximación comience ligeramente a crecer, esto se aprecia claramente en la figura 3.5. Esta característica ha sido observada cuando se trabaja con elemento finito y descomposición de dominio en problemas estacionarios [104]. Un comportamiento similar, a partir del esquema

asimétrico, se ha observado en un problema elíptico [113] y en un problema convectivo-difusivo dependiente del tiempo [100].

En la tabla 3.2 en la cuarta columna se observa que conforme crece el número de subdominios el tiempo requerido en la simulación numérica desciende drásticamente. De hecho, el decrecimiento en el tiempo con respecto al número de subdominios tiene en teoría un comportamiento lineal, ver sección 3.3.1. En la figura 3.6 se muestra en una gráfica log-log el tiempo de procesamiento vs. particiones; en el eje x se indica el incremento en el número de subdominios $P = \{1, 2, 4, \dots, 100\}$, el tiempo de procesamiento se despliega en el eje y .

En la gráfica de la figura 3.6 se aprecia que el tiempo de procesamiento del algoritmo de discretización explícita con descomposición de dominio, para el problema de valor inicial lineal, tiene una disminución lineal en el tiempo de procesamiento en función del incremento en el número de particiones P .

La reducción lineal en el tiempo de procesamiento es un claro indicador de que es posible abordar problemas dependientes del tiempo de gran escala con base en el esquema de colocación asimétrico.

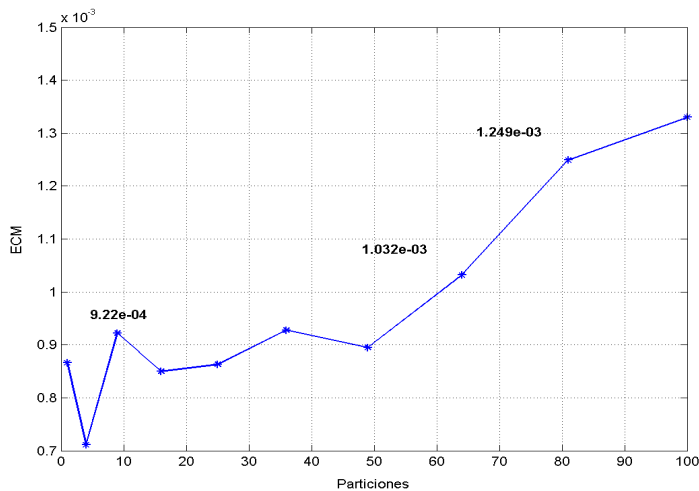


Figura 3.5: Decremento en la exactitud vs. incremento en las particiones P^2 .

En la figura 3.7 se muestra la solución numérica de la ecuación de advección en distintos instantes del tiempo $t = \{0.1, 0.2, 0.3, 0.4\}$ con una partición de

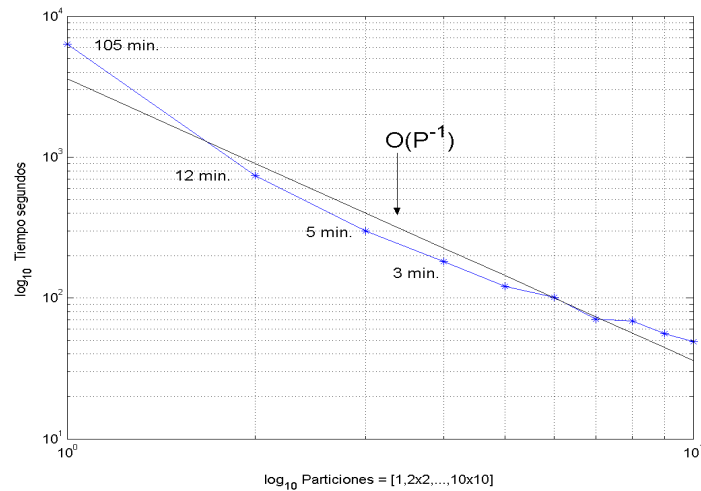


Figura 3.6: Log-log del decremento lineal del tiempo vs. incremento en las particiones P^2 .

5×5 . Esto corresponde al quinto renglón en la tabla 3.2. Se observa que se tiene un desplazamiento lineal de los datos iniciales en la dirección $(1, 1)^T$. Como se aprecia en la figura 3.7 se tiene dividido el dominio $[0, 1] \times [0, 1]$ en cuadriláteros marcados con las líneas en color oscuro en forma horizontal y vertical. Cada cuadrilátero corresponde a una región Ω_i en donde se resuelve de forma local la ecuación diferencial parcial. Se aprecia que la aproximación obtenida preserva la forma de los datos iniciales.

Con base en lo mostrado en esta sección se concluye que conforme se incrementa el número de subdominios se reduce drásticamente el costo computacional, pero al mismo tiempo se incrementa ligeramente el error en la aproximación.

3.4.3. Diferencias Finitas

Ahora investigaremos la exactitud numérica del problema advectivo analizado con el método de diferencias finitas.

Para el método de diferencias finitas empleamos una discretización de primer orden en el tiempo y en el espacio $O(k, h)$. La aproximación de las primeras

derivadas parciales $\{\partial u/\partial x, \partial u/\partial y\}$ se realiza en la dirección del flujo, obteniendo la aproximación del problema de valor inicial

$$U_{i,j}^{n+1} = U_{i,j}^n + \alpha \cdot (U_{i,j}^n - U_{i-1,j}^n + U_{i,j}^n - U_{i,j-1}^n), \quad (3.24)$$

donde $U = U(x, y)$ es la aproximación de la función desconocida $u(x, y)$, n denota el paso discreto temporal y los subíndices i, j representan la posición del nodo en la retícula equiespaciada. El paso del tiempo Δt satisface la restricción de estabilidad Courant-Friedrich-Lax $\alpha < 1$, con $\alpha = \Delta t/\Delta x$ y $\Delta x = \Delta y$.

Los resultados obtenidos se muestran en la figura 3.8. Las líneas punteadas corresponden a la solución obtenida con el método de diferencias finitas, la línea continua con círculos corresponde a la solución obtenida con el esquema de colocación asimétrico (sección transversal de la figura 3.7 al tiempo $t_{max} = 0.4$). Por último, la línea continua corresponde a la solución analítica.

Los resultados obtenidos con el método de colocación asimétrico son consistentes con respecto de la solución analítica. Cabe destacar, que la aproximación con el método de diferencias finitas presenta una gran difusión numérica [118], la que no se observó cuando utilizamos el método de colocación asimétrico con los datos iniciales descritos en la sección anterior.

El resultado anterior nos proporciona indicios de que el esquema de colocación asimétrico contiene poca difusión numérica. Sin embargo, hasta la fecha no se ha determinado la cantidad de dispersión y difusión numérica para problemas dependientes del tiempo mediante el esquema de colocación asimétrico con funciones de base radial.

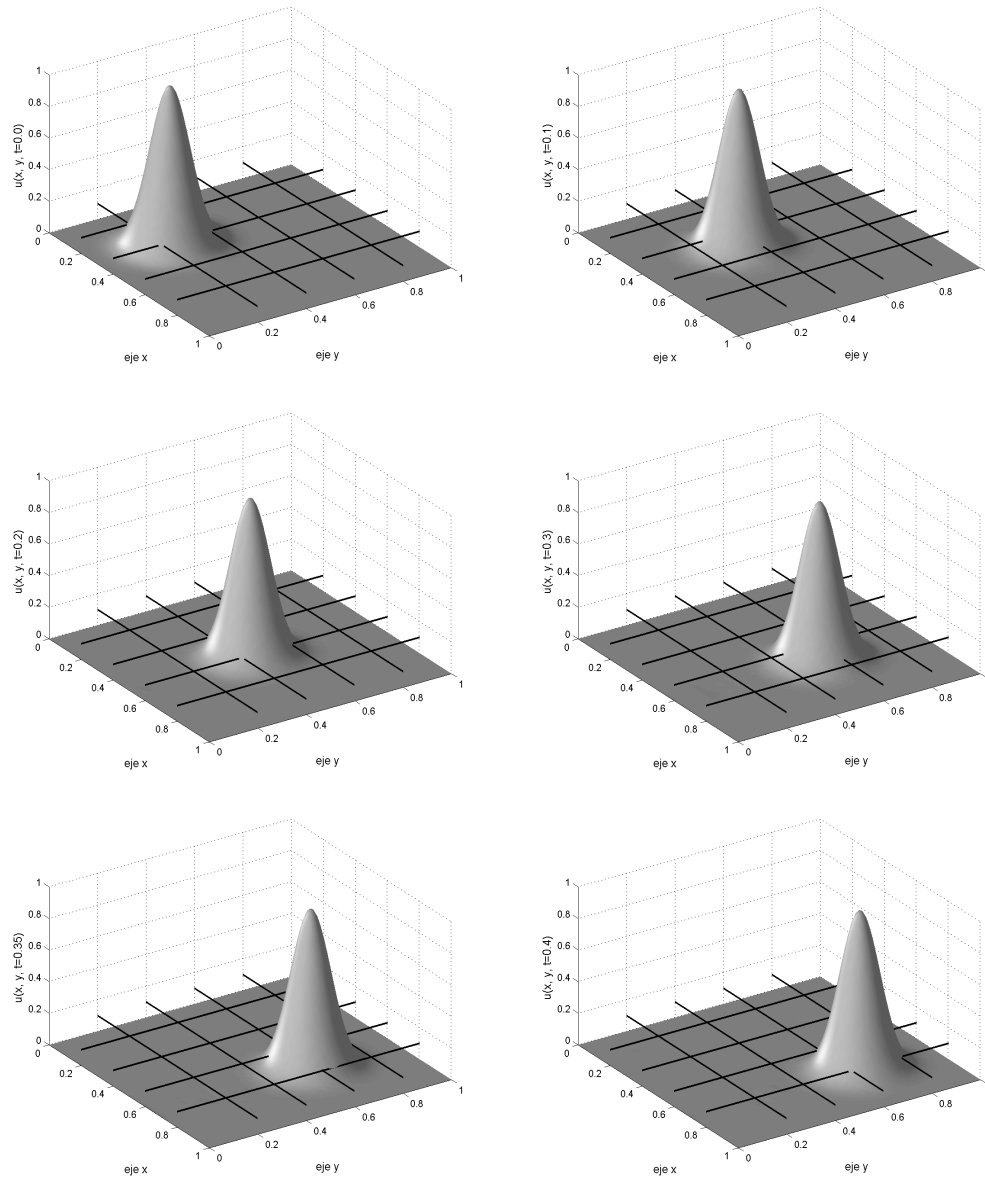


Figura 3.7: Solución numérica de la EDP: $\frac{\partial u(\mathbf{x}, t)}{\partial t} + v \cdot \nabla u(\mathbf{x}, t) = 0$, mostrada en distintos tiempos.

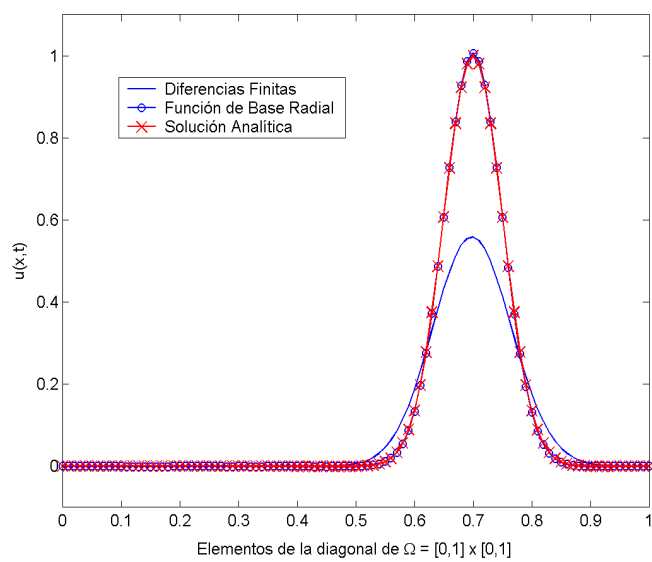


Figura 3.8: Comparación entre diferencias finitas y el método de colocación asimétrico, al tiempo $t_{max} = 0.4$.

Capítulo 4

Refinamiento Local de Nodos

En este capítulo se propone un nuevo algoritmo para el refinamiento local de nodos basado en el error de interpolación local [119] y en un esquema de celda×celda para la locación y el refinamiento de los nodos. El algoritmo propuesto está desarrollado para ecuaciones en derivadas parciales en una y dos dimensiones.

Con base en la función radial de placa delgada se estudia numéricamente el esquema de refinamiento local de nodos (RLN) con respecto del esquema clásico de refinamiento global de nodos (RGN). Para el problema en una dimensión se observa que con el RLN se obtiene una ganancia del 80–90 % en el número de nodos para alcanzar un error similar en la aproximación con respecto al RGN. En dos dimensiones se aborda un problema de Poisson en donde se muestra numéricamente la efectividad del esquema de refinamiento propuesto obteniendo una mejora en dos ordenes de magnitud en el error E_{max} en comparación con una distribución uniforme con el mismo número de nodos. Posteriormente, se extiende la funcionalidad del algoritmo propuesto hacia problemas dependientes del tiempo en 1- d . Se analiza un problema de tipo convección-difusión lineal con un número de Péclet $Pe = 10^3$, con base al esquema de refinamiento $h - c$ con el núcleo multicúbico, se observa que es posible capturar la zona de alta variación espacial. En todos los casos analizados se capturaron correc-

tamente las zonas que presentan cambios abruptos espaciales, densificando los nodos en las regiones donde se requiere de una mayor exactitud numérica.

4.1. Motivación

Diversos problemas físicos tienen variaciones de escala espacial o regiones en donde se conforman capas límites, discontinuidades u ondas de choque. A estas regiones les llamaremos regiones con altos gradientes. Cuando resolvemos numéricamente problemas con dichas características, se requiere de una alta resolución espacial para aproximar adecuadamente la solución. Una primera propuesta de solución es densificar los nodos globalmente sobre todo el dominio del problema. Esta primera propuesta no es la mejor manera de resolver el problema, debido a que el cómputo requerido resulta insuficiente en problemas reales en tres dimensiones. Adicionalmente, existen grandes porciones del dominio en donde no se requiere de una alta resolución espacial, dichas regiones corresponden a zonas con bajos gradientes. La idea del refinamiento local de nodos es refinar zonas solamente en donde sea necesario. El refinamiento local de nodos concentra una mayor cantidad de nodos cerca de las zonas con altos gradientes y una densidad menor de nodos en las zonas con bajos gradientes.

Otro enfoque para abordar problemas con cambios abruptos en la solución es emplear descomposición de dominio con un refinamiento global de nodos. A pesar de que con descomposición de dominio se disminuye el tiempo de procesamiento, el incremento en el espacio N^d hace inapropiado dicho enfoque en problemas prácticos.

4.2. Métodos Adaptivos

El método de refinamiento local de nodos (RLN) está inscrito en el contexto de funciones de base radial como esquema de discretización. En general, RLN pertenece a una clase más amplia llamada refinamiento de mallas adaptivas

(adaptive mesh refinement), siendo el método de elemento finito el nicho de dichos métodos de refinamiento. Los métodos adaptivos para problemas estacionarios se pueden clasificar en cinco categorías [120]:

- *h*-refinamiento: trata sobre el refinamiento espacial en el número de elementos. El esquema de inserción y remoción de elementos se ajusta dinámicamente a la estructura local del problema.
- *p*-refinamiento: se incrementa el grado del polinomio en la aproximación manteniendo fijo el número de elementos.
- *h-p*-refinamiento: es una combinación de los dos esquemas anteriores.
- *r*-refinamiento: se distribuyen los nodos para obtener una mejor resolución espacial manteniendo fijo el número de elementos y el grado del polinomio.
- *m*-refinamiento: dependiendo del comportamiento local de la solución se selecciona el modelo físico que satisface los requerimientos físicos del problema.

Para mayor referencia de técnicas de mallas adaptivas consultar el congreso anual International Meshing Roundtable. En el presente capítulo se aborda principalmente el esquema de *h*-refinamiento. En el área de funciones de base radial, de manera similar al esquema de *h-p*-refinamiento nativo del método de elemento finito, se tiene el esquema *h-c*-refinamiento. El esquema de *c*-refinamiento consiste en ajustar localmente el parámetro libre *c* con el fin de aumentar la exactitud numérica.

Los puntos generales que deben de satisfacer las mallas adaptivas son dados en Kallinderis [121] y son subrayados por los editores Thomson & Soni [122]:

- La adaptación debe dar como resultado una mejora de la precisión numérica.
- La adaptación debe ser automática y eficiente.

- La idea fundamental de la adaptación de la malla es reducir el error en la discretización espacial.
- El error temporal y de conservación deben ser preservados si es requerido.
- El error adicional introducido por los algoritmos adaptivos no debe reducir significativamente los beneficios obtenidos.

Los puntos generales mostrados son aplicables al diseño de algoritmos de nodos adaptivos. Tales puntos proveen criterios iniciales de evaluación para la comparación de algoritmos adaptivos.

En años recientes las mallas adaptivas han ganado mucha atención desde el punto de vista práctico y teórico. Una de las razones es el hecho de que las mallas adaptivas son requeridas para problemas en donde se dan cambios en la geometría del dominio de interés, como es la simulación de problemas de impacto, penetración, explosión, fragmentación y largas deformaciones [123]. La idea principal en mallas adaptivas es contar con una malla más fina en zonas de alto gradiente y una malla menos fina en regiones con bajo gradiente.

En problemas dependientes del tiempo empleando mallas adaptivas, el proceso de mallado se da cada cierto intervalo de tiempo Δt . Este proceso es la parte que consume la mayor cantidad de tiempo de cómputo, cerca del 70 por ciento del tiempo total [3]. Dado que el mallado es una tarea computacionalmente costosa, los nodos adaptivos no requieren de un proceso de mallado y por lo tanto ofrecen una ventaja computacional que debe ser aprovechada.

Aunque los métodos adaptivos nos pueden ayudar a disminuir el error cometido sin un excesivo número de nodos, la utilización de dichos métodos está lejos de ser trivial. Aunado a ello, el esquema de refinamiento local está estrechamente relacionado con el esquema de discretización utilizado. Para utilizar métodos adaptivos exitosamente necesitamos determinar: ¿Dónde? y ¿Cómo adaptar?, y ¿Cómo hacer el proceso eficiente? Patra et al. [124].

En el área de nodos adaptivos con funciones de base radial existen pocos resultados a la fecha, principalmente en 1- d .

- Para la ecuación de Burger viscosa con un número de Reynolds $R=10^4$ y con la función multicuádrada, Hon & Mao (1998) [32] determinan que con base a el movimiento de nodos se mejora el resultado numérico cerca de la discontinuidad.
- Con un esquema semi-Lagrangiano con el método de las características y el error de interpolación local como función indicadora de error, Behrens et al. (2003) [125] aproximan las ecuaciones no lineales; Burger viscosa con un número de Reynolds $R=2 \times 10^3$ y la ecuación de Buckley-Leverett, mostrando que la adaptividad de los nodos sigue la formación del frente de onda.
- Con base al cuasi-interpolante con multicuádricos el movimiento de los nodos es determinado a través de las características, logrando que los nodos se densifiquen cerca de la discontinuidad; ver Wu (2004) [126].
- A partir de un esquema de proyección local de los nodos, para ecuaciones temporales en 1- d , Sarra (2005) [127] expone que los resultados numéricos obtenidos son comparables con lo determinado con métodos espectrales.
- Posteriormente, para problemas de capas límites, Ling et al. (2006) [128] proponen un esquema de inserción de nodos basado en una modificación del indicador a -posteriori de Hon.
- Recientemente se propuso un esquema de remoción e inserción de nodos basado en el error residual $\|Lu - f\|_\infty$ calculado globalmente en una resolución más fina, con base en un esquema de tipo árbol (*quad-tree*) Driscoll et al. (2007) [103]. El esquema desarrollado es aplicado en diversos problemas: Poisson en dos dimensiones con fronteras irregulares y en ecuaciones temporales en una dimensión; lineal y no lineal. El método propuesto es

similar a lo desarrollado en el presente capítulo, la mayor diferencia radica en la función indicadora del error local para refinar los nodos, y en la composición del refinamiento local con el método de descomposición de dominio, ver capítulo 5. Lo cual nos permite abordar problemas más complejos con miles de nodos.

4.3. Aproximación Local

En esta sección, describimos el indicador de error desarrollado para la detección de discontinuidades sobre superficies $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ con datos no equiespaciados [129]. La función indicadora de error posteriormente fue utilizada en el contexto de mallas adaptivas en la solución numérica de ecuaciones en derivadas parciales [119, 130, 125, 131].

Primero mostraremos el error de interpolación local basado en el esquema de interpolación con el núcleo de placa delgada $\phi(r) = r^2 \log r$. Por compatibilidad utilizaremos la misma notación que en el artículo original [129]. Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ una función desconocida y sea $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^2$ un conjunto de $N \geq 3$ puntos distintos no colineales. Asumiendo que la función f tiene valores conocidos en los distintos puntos $\{f(x_1), f(x_2), \dots, f(x_N)\}$, el problema de interpolación radial consiste en determinar el interpolante $s : \mathbb{R}^2 \rightarrow \mathbb{R}$ satisfaciendo

$$f(x_i) = s(x_i), \quad 1 \leq i \leq N, \quad (4.1)$$

donde s es una combinación lineal de la forma

$$s(x) = \sum_{j=1}^N \lambda_j \phi(\|x - x_j\|_2) + \lambda_{N+1} + \lambda_{N+2}\xi + \lambda_{N+3}\eta, \quad (4.2)$$

donde $x = (\xi, \eta)^T$. Como se mostró en el capítulo 1, para garantizar la invertibilidad del sistema es necesario agregar las siguientes restricciones

$$\sum_{j=1}^N \lambda_j = 0 \quad \sum_{j=1}^N \lambda_j \xi_j = 0 \quad \sum_{j=1}^N \lambda_j \eta_j = 0. \quad (4.3)$$

Hasta este momento hemos descrito el problema de interpolación para un caso particular. Ahora, con base en la notación anterior, investigaremos el comportamiento local de la aproximación; [129]. Fijando un punto $x_0 \in \mathbb{R}^2$ y considerando ahora el problema de interpolación de la forma

$$f(x_0 + hx_i) = s^h(x_0 + hx_i), \quad 1 \leq i \leq N, \quad (4.4)$$

donde $h > 0$ es un factor de escalamiento. La familia de interpolantes $\{s^h\}_{h>0}$ generada por la selección de h , nos permite medir la calidad de la aproximación localmente alrededor de x_0 mediante el factor de escalamiento. Ahora, considerando a la función $f \in C^r$ con $r = \{1, 2\}$ finalmente definimos el orden de la aproximación local empleando el núcleo de placa delgada

$$|f(x_0 + hx) - s^h(x_0 + hx)| = O(h^r), \quad h \rightarrow 0. \quad (4.5)$$

Es decir, es de orden cuadrático para $f \in C^2$ y de orden lineal cuando f es diferenciable. Sí la función f es solo continua, entonces la aproximación local es al menos consistente

$$|f(x_0 + hx) - s^h(x_0 + hx)| \rightarrow 0, \quad h \rightarrow 0. \quad (4.6)$$

4.3.1. Indicador de Error Local

Dada la aproximación $\tilde{u}(x)$ obtenida como solución de una ecuación diferencial parcial estacionaria o dependiente del tiempo; $\tilde{u}(x) := \tilde{u}(x, t^n)$ siendo n el paso del tiempo discreto. Lo que queremos determinar es una medida del error local que determine la exactitud de la solución obtenida $\tilde{u}(x)$. Para satisfacer tal requisito se requiere obtener una función indicadora del error local, así como un algoritmo computacional que nos permita determinar el conjunto de vecinos cercanos a un punto dado $x \in \Omega$. La determinación de los vecinos es tratada en la siguiente sección, en esta sección nos enfocaremos en determinar la función indicadora del error local.

Podemos entender al indicador del error como una función $\eta : \Xi \rightarrow [0, \infty)$ con dominio en el sistema actual de nodos $\Xi := \Xi^t$ (al tiempo t). La función

$\eta(x)$ determina la calidad de la aproximación alrededor de un punto dado, con base en dicho valor definiremos en que regiones se requiere de una mayor exactitud numérica.

Sea Ξ el conjunto de nodos provenientes de la distribución inicial de los nodos. Para cada nodo $x \in \Xi$ seleccionamos un conjunto de nodos vecinos $N_x \setminus x \in \Xi$. Construimos un interpolante local $I\tilde{u}(x)$ basado en los nodos $N_x \setminus x$, para ello empleamos la ecuación de interpolación radial con el núcleo de placa delgada y un polinomio de primer grado.

Con base en la aproximación numérica $\tilde{u}(x)$ y el interpolante local $I\tilde{u}(x)$, definimos el indicador de error como

$$\eta(x) = |\tilde{u}(x) - I\tilde{u}(x)|, \quad (4.7)$$

que asigna un valor $\eta(x)$ a cada nodo $x \in \Xi$. Este valor de $\eta(x)$ es pequeño cuando los valores en la aproximación numérica alrededor de x son similares o cuando pueden ser representados por una función lineal. Por otra parte, un valor alto en $\eta(x)$ indica comúnmente que la aproximación numérica alrededor del nodo x tiene una alta variación, que corresponde por lo general a las regiones con alto gradiente.

La función indicadora de error (4.7) no depende del método numérico utilizado para obtener la solución del problema analizado. Con base en el indicador de error $\eta(x)$, definimos la regla para marcar los nodos que requieran ser refinados.

Definición A Sea $\eta^* = \max \eta(x)$ para todo x , y sea $\theta_{ref} \in (0, 1)$ un valor de umbral. Decimos que un nodo $x \in \Xi$ es marcado para ser refinado si $\eta(x) > \theta_{ref} \cdot \eta^*$.

Esta definición se puede entender como: los nodos marcados para refinar son aquellos que se encuentren por encima de un porcentaje θ_{ref} del máximo error local η^* . La estrategia de refinamiento mostrada anteriormente se aplica en un contexto iterativo, en donde en cada paso resolvemos la ecuación diferencial

parcial y con la aproximación numérica $\tilde{u}(x)$ obtenida, realizamos el esquema del refinamiento. Este algoritmo se describe en detalle en la sección 4.5.

4.4. Esquema de Celda×Celda

La discretización del dominio Ω es realizada en un esquema de celda×celda no traslapada como se muestra en la figura 4.1, el centro de cada celda contiene un nodo. La discretización basada en celdas pertenece a la clase de mallas sin estructura. En este trabajo los términos de refinamiento de celdas y refinamiento de nodos son utilizados indistintamente.

La diferencia con el trabajo de Behrens et al. [119, 130] radica en el esquema basado en celdas, en los citados trabajos emplean la triangulación de Delaunay y su dual Voronoi [132] para buscar e insertar los nodos respectivamente, la extensión a \mathbb{R}^d constituye un problema de frontera. Los esquemas basados en árboles o celdas son factibles de extenderlos a dimensiones superiores.

El esquema de celda×celda lo podemos pensar como un caso particular del refinamiento basado en bloques [133, 134, 135], en donde cada bloque puede contener varias decenas de nodos. Los esquemas de refinamiento basados en bloques han mostrado su efectividad en una amplia variedad de aplicaciones; propagación de ondas electromagnéticas [136], predicción del clima [137], ecuaciones de Navier Stokes [138, 139] y ecuaciones de Euler [133, 140, 141, 142]. Adicionalmente se han desarrollado distintas bibliotecas generales para refinamiento basado en bloques: BoxLib [143], Chombo [144], SAMRAI [145], FLASH [146], DAGH [147] y Racocon [148]. A pesar de las múltiples aportaciones en esta área y en particular de las bibliotecas desarrolladas, no se cuenta con información detallada de como emplear la estructura de datos nativa del refinamiento por bloques para otros esquemas de discretización. Por lo tanto, para esquemas libres de malla la presente tesis constituye un primer paso en dicha dirección.

4.4.1. Refinamiento de Celdas

A partir del indicador de error local $\eta(x)$ se marcan las celdas a ser refinadas, por cada celda marcada para refinar se insertan 4 nodos adicionales. En la figura 4.1 se ilustra este proceso de refinamiento de celdas. Observe que adicionalmente a las celdas marcadas para refinar (denotadas por R en la figura 4.1), las celdas vecinas de los elementos marcados también son refinadas. Esto corresponde a la restricción 2:1 (ver sección 4.4.2) impuesta en los niveles de refinamiento. Esta estrategia nos permite obtener una mayor densidad de nodos cerca de las zonas con altos gradientes.

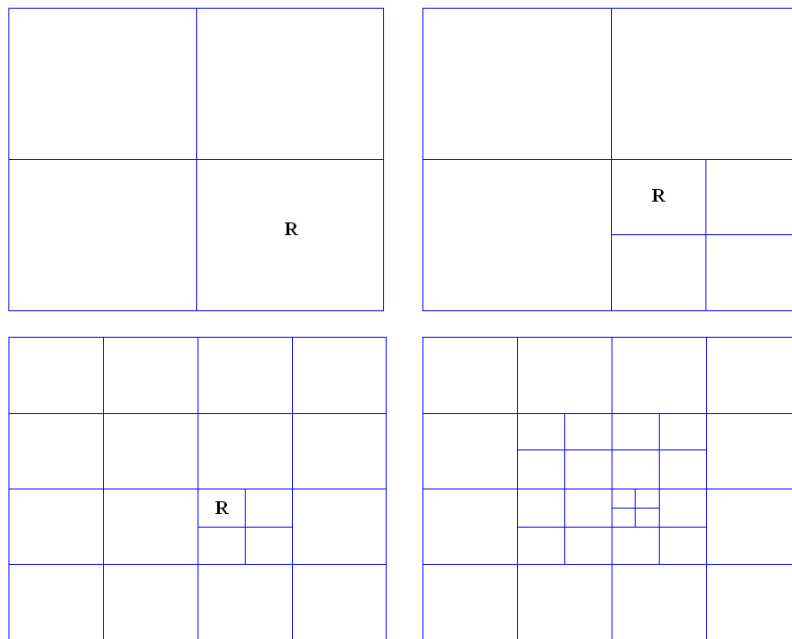


Figura 4.1: Ilustración de adaptación de celdas en dos dimensiones, se observa que las celdas vecinas son refinadas por la relación 2:1.

La estructura de datos utilizada almacena por cada celda un apuntador a sus 4-vecinos naturales: arriba, abajo, derecha e izquierda. Para construir el indicador de error local, el conjunto de nodos vecinos N_x de un punto x se obtiene a partir de los primeros 4-vecinos naturales y los nodos que conforman las 4-diagonales: arriba-derecha, arriba-izquierda, abajo-derecha y abajo-izquierda. Los nodos que conforman las diagonales son obtenidos a partir de los vecinos

naturales.

El guardar explícitamente la referencia a sus 4-vecinos naturales por cada celda se incrementa en 32 bytes el costo de almacenamiento en memoria. Sin embargo, la complejidad para buscar los nodos más cercanos es de orden constante $O(1)$.

En dos dimensiones con el esquema de celda×celda por cada nodo marcado para refinar se insertan/crean cuatro nuevas celdas. En la figura 4.2 se muestra el refinamiento para una celda marcada con x .

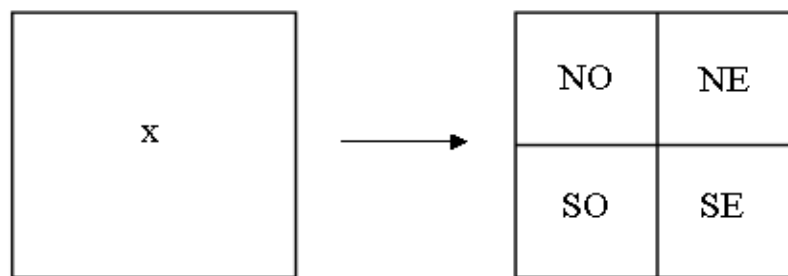


Figura 4.2: Ilustración de refinamiento para una celda en dos dimensiones.

Con base en las dimensiones de la celda marcada para refinar $\{x_{min}, x_{max}, y_{min}, y_{max}\}$, ver sección 4.4.3, las dimensiones de las nuevas cuatro celdas creadas son:

Tipo de Celda	x_{min}	x_{max}	y_{min}	y_{max}
SO	x_{min}	$x_{min} + \frac{1}{2}\Delta x$	y_{min}	$y_{min} + \frac{1}{2}\Delta y$
SE	$x_{min} + \frac{1}{2}\Delta x$	x_{max}	y_{min}	$y_{min} + \frac{1}{2}\Delta y$
NO	x_{min}	$x_{min} + \frac{1}{2}\Delta x$	$y_{min} + \frac{1}{2}\Delta y$	y_{max}
NE	$x_{min} + \frac{1}{2}\Delta x$	x_{max}	$y_{min} + \frac{1}{2}\Delta y$	y_{max}

Tabla 4.1: Dimensiones de las nuevas cuatro celdas.

donde $\Delta x = x_{max} - x_{min}$ y $\Delta y = y_{max} - y_{min}$. A partir de las dimensiones de cada nueva celda, podemos definir las coordenadas del nodo que se encuentra

definido al centro de cada celda

$$x_c = x_{min} + \frac{1}{2}(x_{max} - x_{min}), \quad y_c = y_{min} + \frac{1}{2}(y_{max} - y_{min}).$$

El proceso anterior se puede apreciar fácilmente si nos imaginamos una estructura de tipo árbol: comenzando con un nodo raíz con cuatro hijos, si queremos refinar a uno de los hijos se crean cuatro nuevos nodos que pasan a ser los hijos del nodo refinado. Con base en esta idea, las hojas del árbol corresponden a los nodos contenidos en el dominio Ω . En la codificación realizada la estructura de datos no es de tipo árbol (quad-tree), lo que se utiliza es un arreglo lineal con elementos de tipo celda, ver sección 4.4.3.

El esquema de refinamiento expuesto anteriormente es válido para todas las celdas que no tocan la frontera del dominio Ω , las celdas que conforman la frontera reciben un trato especial. Esto se debe a que el centro de cada celda contiene al nodo, al estar situado el nodo en la frontera, las dimensiones de la celdas exceden las dimensiones del dominio Ω , y por lo tanto al momento de refinar los nodos frontera se estarían creando nodos fuera del dominio.

En la práctica los nodos que conforman la frontera son celdas con una de sus componentes igual a cero. Es decir, los nodos que conforman la frontera paralela al eje x tienen un valor de cero en la componente Δy , de manera similar los nodos que conforman la frontera paralela al eje y tienen un valor de cero en la componente Δx . Adicionalmente, los nodos que conforman las esquinas reciben un trato diferente. En la distribución inicial de los nodos, las celdas que conforman las esquinas son marcadas para no ser refinadas.

Para problemas en una dimensión la estrategia del refinamiento de nodos consiste en la inserción de dos nodos por cada nodo marcado para ser refinado. Es decir, sea $x_0 < x_1 < \dots < x_N$ y sea x_i el nodo marcado para ser refinado. Por cada nodo marcado para ser refinado se insertan dos nodos adicionales; un nodo al lado izquierdo y un nodo al lado derecho de x_i

$$x_{izq} = x_i + \frac{1}{2}(x_{i+1} - x_i), \quad x_{der} = x_{i-1} + \frac{1}{2}(x_i - x_{i-1}),$$

y x_i es eliminada.

4.4.2. Restricción 2:1

La relación 2:1 indica que el nivel de refinamiento de una celda r_{ν_i} no debe exceder en un nivel de refinamiento con respecto a las celdas vecinas

$$|r_{\nu_i} - r_{\nu_j}| < 2, \quad (4.8)$$

para $i \neq j$ con $j = 1, \dots, K$, siendo K el número de celdas/nodos vecinos.

La distribución inicial de los nodos es equiespaciada y tiene un nivel de refinamiento igual a cero. Inicialmente cada celda tiene dimensiones $\Delta x \times \Delta y$ con $\Delta x = |x_i - x_{i-1}|$ y $\Delta y = |y_i - y_{i-1}|$, es claro que después de r -refinamientos sobre todo el dominio las dimensiones de cada celda serán

$$\frac{\Delta x}{2^r} \quad \text{y} \quad \frac{\Delta y}{2^r},$$

donde las dimensiones de cada celda descienden geoméricamente.

La codificación de la restricción 2:1 está basada en dos operaciones: búsqueda de vecinos más cercanos y el posible refinamiento de las celdas/vecinos más cercanos. Asumiendo que queremos refinar la celda ν_i , el procedimiento para realizar la refinación consiste en los siguientes pasos:

- Paso 1: Determinar los K -vecinos más cercanos de la celda ν_i .
- Paso 2: Para cada vecino determinar si satisfacen la restricción 2:1; ver ecuación (4.8).
- Paso 3: Para cada celda que no satisface la restricción 2:1 se itera recursivamente; ir al Paso 1, en caso contrario se refina la celda ν_i .

4.4.3. Estructura de Datos

La estructura de datos utilizada para representar cada nodo o celda se muestra a continuación

```
typedef struct nodo
{
    double  x,y;
    int     tipo;
    int     nivel;
    double  xmin,ymin,xmax,ymax;
    int     id;
    int     veci_arriba[2];
    int     veci_abajo[2];
    int     veci_derecha[2];
    int     veci_izquierda[2];
    double  sol;
    double  c;
    int     existe;
}nodo;
```

Dicha estructura de datos requiere de 120 bytes de almacenamiento en memoria, lo cual no es un parámetro crítico, en las simulaciones realizadas, que requiera ser optimizado. El centro de la celda que corresponde a las coordenadas del nodo son almacenadas en las variables x, y . La categoría del tipo de celda={interior, frontera, esquina}, es almacenada en la variable *tipo*. El nivel de refinamiento del nodo se almacena como una variable entera llamada *nivel*, con la cual se controla el nivel máximo de refinamiento y la restricción 2:1. Los límites que conforman la celda en dos dimensiones son almacenados en las variables: $xmin, ymin$ y $xmax, ymax$, las cuales corresponden a los límites inferior izquierdo y superior derecho respectivamente. El identificador único de la celda es almacenado en la variable *id*, este identificador único es requerido cuando trabajamos con descomposición de dominio; ver capítulo 5.

Las referencias para cada uno de los vecinos naturales: arriba, abajo, derecha e izquierda, son almacenadas en las variables: *veci-arriba*, *veci-abajo*, *veci-derecha* y *veci-izquierda*. Cada una de las variables es un vector de dimensión dos, debido a que cada cara de la celda puede tener como máximo dos vecinos colindantes.

La solución aproximada $\tilde{u}(x, y)$ se guarda en la variable *sol*. Cuando se utilice la función radial multicuádrica en la variable *c* se almacena el parámetro libre. Cada vez que una celda es refinada en la variable *existe* se asigna un valor de cero, lo cual indica que la celda fue eliminada.

Observación: Cabe resaltar que la estructura de datos utilizada se puede extender a tres dimensiones (octree [149]), y junto con el indicador de error local constituyen un verdadero esquema libre de malla para datos no equiespaciados en \mathbb{R}^3 .

4.5. Esquema General

En esta sección se integra el indicador de error $\eta(x)$ y el esquema basado en celdas para conformar un algoritmo que nos permita resolver numéricamente ecuaciones diferenciales parciales estacionarias con refinamiento local de nodos empleando funciones de base radial.

Los pasos requeridos del esquema propuesto se muestran en algoritmo 2.

Algoritmo 2 Refinamiento Local de Nodos

Inicializar con una distribución uniforme de nodos.

while *error* > ε **do**

- 1.- Resolver la EDP con el esquema de colocación asimétrico.
- 2.- Marcar los nodos a refinar.
- 3.- Realizar el refinamiento de nodos.

end while

Al comienzo del algoritmo se cuenta con una distribución uniforme de nodos, los nodos son generados con el esquema de celda×celda.

En este trabajo el criterio de convergencia está definido cuando el error en la norma infinita se encuentre por debajo de un cierto valor de umbral τ . En la siguiente sección se muestran los criterios de paro adicionales que son requeridos para garantizar que el algoritmo no entre en un ciclo infinito.

En cada iteración del algoritmo 2 se resuelve la ecuación diferencial parcial analizada mediante el esquema de colocación asimétrico. De esta manera, se obtiene la solución aproximada $\tilde{u}(x)$ para todos los nodos del dominio Ω . Con base en la solución obtenida, se construye el indicador de error $\eta(x)$ y se determinan los nodos que van a ser refinados. Finalmente, los nodos marcados para refinar son refinados utilizando un esquema de celda \times celda descrito en la sección 4.4.

4.5.1. Criterios de Paro

La solución numérica de ecuaciones en derivadas parciales con refinamiento local de nodos es un proceso iterativo, por lo tanto debemos contar con uno o varios criterios de paro. Los criterios de paro utilizados a lo largo de este trabajo son:

- $\|u_k - \tilde{u}_k\|_\infty < \varepsilon$.
- No se insertó ningún nodo adicional.
- Se alcanzó un número máximo de iteraciones.

La solución analítica en el paso de iteración k -ésimo está denotada por u_k y la aproximación numérica mediante \tilde{u}_k .

El primer criterio detiene el algoritmo 2 cuando la máxima diferencia entre la solución analítica y la solución numérica se encuentre por debajo de un valor de umbral ε . El cálculo es realizado sobre todos los nodos del dominio Ω incluyendo los nodos que conforman la frontera.

El proceso de inserción de nodos se realiza en dos pasos: primero se marcan los nodos que queremos refinar y después se refinan dichos nodos. El segundo criterio de paro corresponde a que a pesar de que hay nodos marcados para refinar no se logró refinar ningún nodo, debido a que se alcanzó el nivel máximo de refinamiento en cada nodo marcado. Por lo general, este nivel máximo de

refinamiento se establece con un valor de 15, lo cual corresponde a una distancia en valor absoluto entre nodos $\Delta x = 3 \times 10^{-5}$ en el intervalo $[0, 1]$. Es decir, el nivel de refinamiento indica la densidad de los nodos.

El tercer criterio de paro es para tener la certeza de que el programa no se quedará en un ciclo infinito.

Cuando no se cuenta con la solución analítica es posible establecer, como criterio de paro del algoritmo 2, el error cometido entre dos sucesivas aproximaciones numéricas: $\|\tilde{u}_k - \tilde{u}_{k-1}\|_l < \varepsilon$, donde l denota la norma vectorial utilizada y k el paso en la iteración. Esta medición del error en la aproximación es ampliamente utilizada en métodos de optimización [150] y recientemente en problemas estacionarios en $1-d$ con la función multicuádrica y métodos adaptivos [128].

4.6. Resultados Numéricos

El objetivo de esta sección es investigar el comportamiento del esquema de refinamiento local de nodos para problemas estacionarios en una y dos dimensiones. En particular, queremos mostrar que con el esquema de refinamiento local de nodos es posible resolver numéricamente problemas con regiones de alta variación espacial, densificando los nodos en donde se requiere de una mayor exactitud numérica. Además, queremos mostrar que el esquema de refinamiento propuesto es un método computacionalmente eficiente en comparación con el esquema clásico de refinamiento de nodos. Adicionalmente, extendemos el algoritmo 2 hacia problemas dependientes del tiempo en $1-d$. El esquema de refinamiento $h - c$ es aplicado a una ecuación de tipo convección-difusión con $Pe = 10^3$ mostrando numéricamente la factibilidad del método.

4.6.1. Ejemplo en 1-d

Considere la siguiente ecuación diferencial parcial en una dimensión

$$u_{xx}(x) = f(x), \quad x \in \Omega \subset \mathbb{R}, \quad (4.9)$$

donde $\Omega=[0,1]$, con una condición de frontera tipo Dirichlet

$$u(x) = g(x), \quad x \in \partial\Omega. \quad (4.10)$$

Los subíndices en u_{xx} denotan la segunda derivada parcial con respecto a x . La solución analítica está definida como:

$$u(x) = \tanh(\eta(x - 0.5)), \quad (4.11)$$

con $\eta > 0$. Con base en la solución analítica, la función $f(x)$ queda expresada

$$f(x) = -2\eta u(x)(\eta - \eta u(x)^2). \quad (4.12)$$

El parámetro η nos permite controlar el grado de suavidad en la solución analítica. Conforme incrementamos el valor del coeficiente η se va definiendo una discontinuidad en $x = 0.5$.

Como datos iniciales considere una distribución de 34 nodos equiespaciados, un porcentaje de refinamiento $\theta_{ref} = 0.1$, el criterio de paro se establece cuando $E_{max} < 10^{-3}$. La función radial utilizada es placa delgada $\phi(r) = r^4 \log r$. Seleccionamos el coeficiente $\eta=500$, con lo cual se obtiene un alto gradiente en la solución analítica sin llegar a la discontinuidad.

En la tabla 4.2 se muestran los resultados obtenidos. En la primera columna se indica el número de iteración del algoritmo 2. El número de nodos se despliega en la segunda columna. En la tercera y cuarta columna se despliegan el error en la norma infinita y el error cuadrático medio calculado sobre todo el dominio Ω .

Se observa en la tabla 4.2 que conforme iteramos el algoritmo de refinamiento local de nodos paulatinamente se va reduciendo el error cuadrático medio

Iteración	N	E_{max}	ECM
0	34	0.968643	0.559801
1	45	0.912644	0.533539
2	74	0.172233	0.084677
3	98	0.415363	0.249337
4	158	0.014681	0.008472
5	172	0.001712	0.000305
6	194	0.000352	0.000211

Tabla 4.2: Decrecimiento del error vs. iteraciones.

y el error en la norma infinita. Al mismo tiempo, como se observa en la segunda columna, el número de nodos se va incrementando hasta llegar a 194 nodos distribuidos no uniformemente. Ambas medidas del error en la aproximación numérica están en el orden de 10^{-4} , lo cual indica una alta exactitud numérica.

En la figura 4.3 se muestra la reconstrucción de la aproximación numérica. Se observa de esta gráfica que la zona con mayor gradiente fue capturada sin oscilaciones.

La distribución de los nodos en la figura 4.3 se muestra en la figura 4.4. De la última figura se puede observar que cerca del punto $x = 0.5$ se obtiene una mayor densidad de nodos y conforme nos alejamos de $x = 0.5$ la distribución de nodos disminuye. Esta disminución gradual de nodos se debe a la restricción 2:1 impuesta en el algoritmo. En las zonas planas se requirió de una densidad menor de nodos en comparación con los nodos requeridos en la zona de alto cambio espacial cercana al punto $x = 0.5$.

Adaptivo vs. no-Adaptivo

Ahora compararemos el esquema de refinamiento local de nodos con respecto a una distribución uniforme de nodos. Esto lo realizamos con el objetivo de

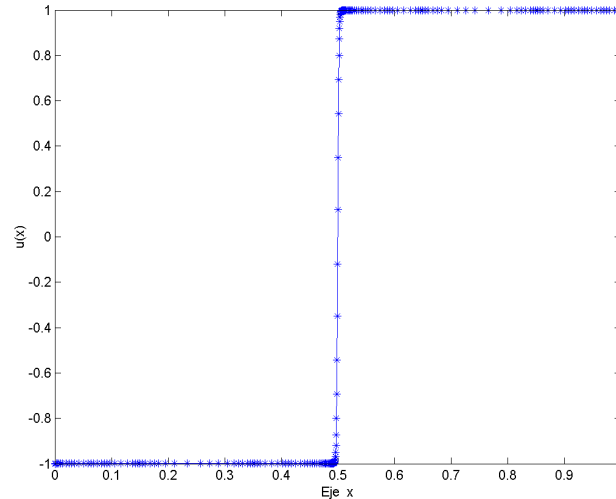


Figura 4.3: Reconstrucción de $u(x)$ con el esquema de refinamiento de nodos.

comparar el número de nodos requeridos con ambos esquemas para alcanzar el mismo error de aproximación.

El experimento numérico fue realizado utilizando un coeficiente $\eta=1000$ con refinamiento local de nodos y con un refinamiento global de nodos (RGN). Entendiendo como RGN al incremento en el número de nodos sobre todo el intervalo $[0, 1]$, donde el número de nodos pertenece al conjunto $\{100, 200, \dots, 1400\}$.

En la figura 4.5 se muestran dos gráficas en una escala log-log, sobre el eje x se indica el número de nodos y sobre el eje y se despliega el error cuadrático medio. Como se observa en la figura 4.5, para alcanzar el mismo error de aproximación, el esquema de refinamiento local requirió de menos nodos en comparación con el esquema de refinamiento global. Por lo tanto, el esquema de refinamiento local es más eficiente que el esquema de refinamiento global de nodos, lo cual se traduce en un menor tiempo de cómputo.

En la tabla 4.3 se muestra el error obtenido con el refinamiento local de nodos y el refinamiento global de nodos correspondiente a los nodos finales mostrados en la figura 4.5.

En la segunda y cuarta columna en la tabla 4.3 se observa que el esquema

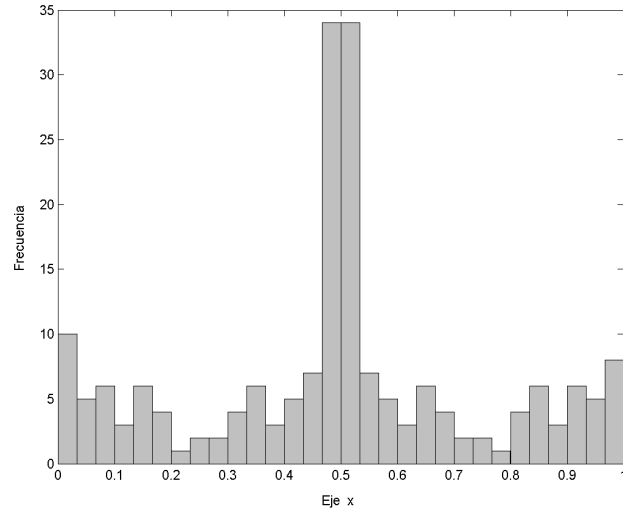


Figura 4.4: Histograma de la distribución de nodos.

Método	N	E_{max}	ECM
Refinamiento Local	249	8.151e-4	4.086e-4
Refinamiento Global	1400	3.546e-3	4.160e-4

Tabla 4.3: Comparación entre h -refinamiento global y h -refinamiento local.

de refinamiento local tiene una eficiencia del 82 % en el número de nodos para alcanzar el mismo error en comparación con el esquema de refinamiento global.

En la tabla 4.3 en la tercera columna se observa que el refinamiento local de nodos es un orden de magnitud menor en comparación con el refinamiento global de nodos. Para lograr alcanzar un error similar con el método de refinamiento global se requirió de 2700 nodos distribuidos uniformemente, obteniendo un error $E_{max}=8.281e-4$. Por lo tanto, con el refinamiento local de nodos se obtuvo una ganancia del 90 % en el número de nodos, esto equivale aproximadamente a 10 veces el número de nodos requeridos con el método local adaptivo. Esta gran diferencia en el número de nodos es dramática y se acentúa con la dimensión del problema. Es por ello que para aplicaciones reales con fuertes capas límites es conveniente el uso de refinamiento local de nodos.

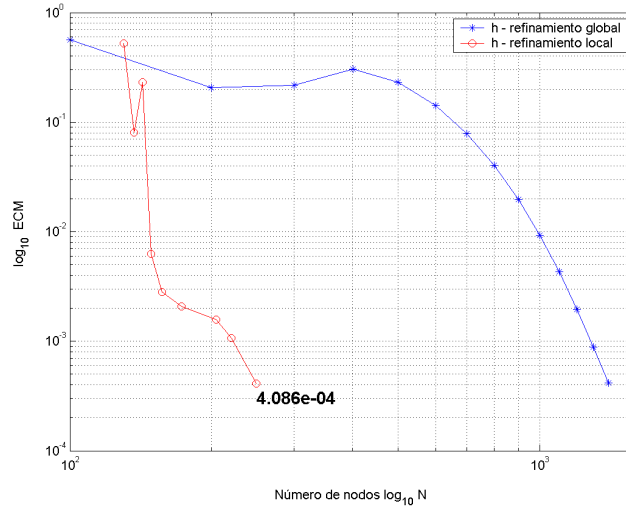


Figura 4.5: Comparación entre h -refinamiento local y h -refinamiento global.

4.6.2. Poisson en 2-d

En la sección anterior se mostró la ventaja de utilizar el refinamiento local de nodos para un problema estacionario en una dimensión. En esta sección se extiende el trabajo realizado a problemas estacionarios en dos dimensiones. Se considera un problema de Poisson con solución analítica, el cual contiene zonas con cambios abruptos espaciales.

Considere el siguiente problema de Poisson en dos dimensiones

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \quad \text{en } \Omega \subset \mathbb{R}^2, \quad (4.13)$$

$u = u(x, y)$, sujeta a las condiciones de frontera tipo Dirichlet

$$u = g \quad \text{en } \partial\Omega, \quad (4.14)$$

donde $\Omega = [0, 1]^2$. Las funciones $f, g : \mathbb{R}^2 \rightarrow \mathbb{R}$ son determinadas a partir de la solución analítica

$$u(x, y) = e^{-50x} + e^{-50y}. \quad (4.15)$$

En la figura 4.6 se muestra la forma de la solución analítica. Como se observa en dicha gráfica, cerca de los ejes $x = 0$ y $y = 0$, se tiene una zona de alto

gradiente semejante a una capa límite. Fuera de dicha zona se tiene una región sin variaciones con un valor constante cero. La determinación de una correcta aproximación numérica cerca de la frontera no es un problema trivial, debido a la fuerte variación espacial.

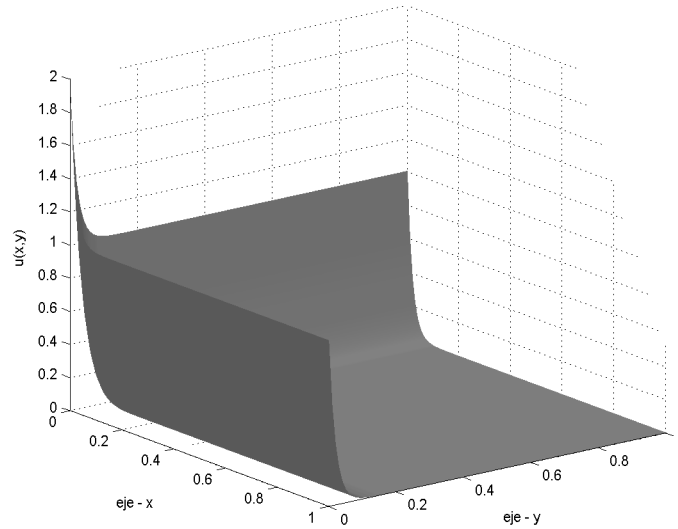


Figura 4.6: Solución analítica del problema de Poisson.

Considere una distribución inicial uniforme de nodos de nodos $N = 324$, un porcentaje de refinamiento $\theta_{ref} = 0.3$ y la función radial placa delgada $\phi(r) = r^4 \log r$. El criterio de paro se establece cuando $E_{max} < 5 \times 10^{-3}$.

En la tabla 4.4 se muestra el resultado obtenido. En la primera columna se despliega el número de la iteración del algoritmo 2, el incremento en el número de nodos se indica en la segunda columna. La tercera y cuarta columna corresponden a la medición del error en la aproximación con base en el error cuadrático medio y el error en la norma infinita respectivamente.

En la tercera y cuarta columna (tabla 4.4) se observa que con base al esquema de refinamiento local de nodos ambas medidas del error disminuyen. Del resultado obtenido se puede concluir que es posible reducir el error de aproximación conforme se continúe iterando. Sin embargo, para posteriores iteraciones,

Iteración	N	ECM	E_{max}
0	324	0.043148	0.141574
1	380	0.029619	0.088100
2	460	0.021939	0.076647
3	664	0.012453	0.059011
4	803	0.008232	0.041500
5	972	0.004799	0.040184
6	1111	0.003424	0.034594
7	1244	0.003762	0.018571
8	1463	0.002493	0.007224
9	1948	0.001703	0.007011
10	2646	0.001170	0.006585
11	3561	0.001020	0.007582
12	3776	0.000553	0.004722

Tabla 4.4: Decrecimiento del error vs. iteraciones para el problema de Poisson.

$iteraciones > 12$, el error residual $\|Lu - f\|_\infty$ y la función indicadora del error local $\eta(x)$ no coinciden en la localización del error. De hecho, el error residual se ubica cerca de los ejes $x = 1$ y $y = 1$, mientras que los valores máximos de $\eta(x)$ se ubican cerca de los ejes $x = 0$ y $y = 0$. Este problema no ha sido desarrollado en el actual trabajo y será objeto de una investigación posterior.

En la figura 4.7 se muestra la distribución de las celdas en las iteraciones $= \{0, 4, 8, 12\}$. Como se aprecia en dichas gráficas las celdas fueron refinadas de una manera incremental. Las zonas refinadas están cerca de los ejes $x = 0$ y $y = 0$, lo cual corresponde a las zonas con la mayor variación espacial en la solución analítica; ver figura 4.6. Por lo tanto, el indicador de error local $\eta(x)$ nos permite detectar regiones en donde exista una alta variación espacial y con base al refinamiento de nodos se obtiene una mayor densidad en dichas zonas logrando así una disminución en el error de aproximación.

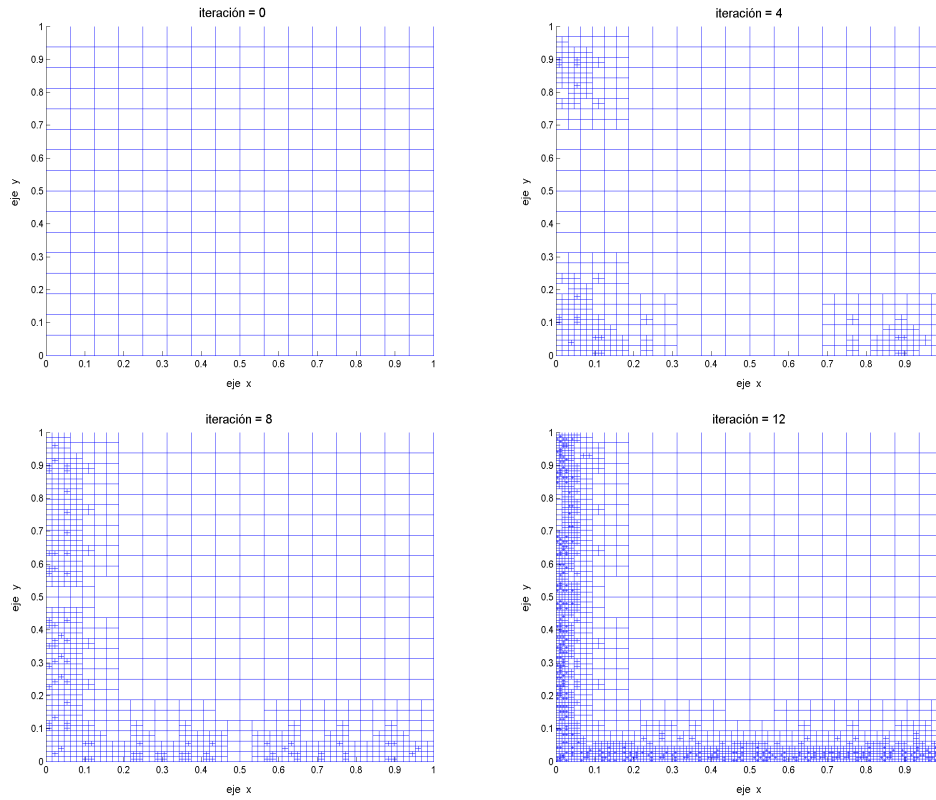


Figura 4.7: Adaptación de celdas para el problema en 2- d de Poisson en las iteraciones= $\{0,4,8,12\}$.

Ahora compararemos el resultado obtenido con una distribución equiespaciada de nodos empleando un similar número de nodos. Para ello aproximamos numéricamente el mismo problema de Poisson con $N = 3844$, los cuales son divididos en nodos interiores $N_I = 3600$ y nodos frontera $N_F = 244$. En la tabla 4.5 mostramos el resultado obtenido, no-adaptivo, con respecto de lo determinado con el esquema de nodos adaptivos. Para un similar número de nodos, con el esquema no-adaptivo, se observa una disminución en la exactitud numérica de dos órdenes de magnitud en ECM y un orden de magnitud en E_{max} . Por lo tanto, con un similar número de nodos la exactitud numérica obtenida con nodos equiespaciados es comparativamente inferior a la obtenida con el esquema de refinamiento local de nodos.

En las secciones 4.6.1 y 4.6.2 se mostró la eficiencia del algoritmo propuesto,

Esquema	N	ECM	E_{max}
adaptivo	3776	0.000553	0.004722
no-adaptivo	3844	0.015945	0.031331

Tabla 4.5: Comparación entre el método de nodos adaptivos y no-adaptivos para el problema de Poisson.

resaltando las características positivas. Con ánimos de ser crítico es conveniente indicar que adicionalmente a la inserción de nodos se requiere la remoción de nodos para datos en dos dimensiones.

4.6.3. Convección-Difusión en 1- d

Esta sección está vinculada con la sección 2.4 del capítulo 2, la cual trata sobre la tasa de convergencia exponencial empleando el núcleo multicuádrico para una ecuación de tipo convección-difusión. En particular, en esta sección se considera el caso cuando el coeficiente de velocidad μ domina al término difusivo β ; $Pe=1000$.

En el presente capítulo se abordó el tema de refinamiento local de nodos con un esquema de celda×celda con el método numérico de colocación asimétrica, el cual fue desarrollado para problemas estacionarios. Ahora, nuestro objetivo es extender la funcionalidad del algoritmo 2 hacia ecuaciones dependientes del tiempo en una dimensión.

Para abordar problemas dependientes del tiempo es necesario modificar la definición A con objeto de incluir el esquema de remoción de nodos. El criterio para insertar/remover nodos lo definimos como:

Definición B Decimos que un nodo $x \in \Omega$ es marcado para ser refinado si $\eta(x) > \theta_r$. En caso contrario, si $\eta(x) < \theta_c$ el nodo es marcado para ser removido, con $\theta_c < \theta_r$.

Observe que un nodo no puede ser refinado y removido al mismo tiempo. Por cada nodo x_i marcado para refinar se insertan dos nuevos nodos; ver sección 4.4. Cada nodo x_i marcado para remover es eliminado solo cuando los nodos vecinos $\{x_{i-1}, x_{i+1}\}$ también están marcados para remover. Esta regla evita la eliminación de nodos consecutivos. Cada vez que se efectúa la inserción o remoción de nodos es necesario volver a construir las matrices de derivadas Φ' , Φ'' y la matriz Φ . Adicionalmente, se requiere interpolar la solución $\tilde{u}(x, t^n)$ hacia el nuevo grid para así obtener el nuevo vector $\lambda(t^n)$.

La ecuación analizada de tipo convección-difusión es dependiente del tiempo, es por ello que tenemos que definir cada cuantos pasos temporales (τ) queremos refinar/remover los nodos. En nuestro caso seleccionamos $\tau=2$. Esto significa que cada dos pasos temporales preguntamos si la aproximación numérica basada en el sistema actual de nodos requiere ser refinada/removida. Para la condición inicial $u(x, t = 0)$ también aplica el esquema de inserción/remoción de nodos, esto es requerido debido a que con $Pe = 10^3$ se cuenta con una zona de alta variación espacial.

Es necesario adaptar localmente el parámetro c de la función radial multi-cuádrica [103]. Esto se realiza después de la remoción/inserción de nodos. En particular, el parámetro c es seleccionado como:

$$c_j = |x_{j+1} - x_{j-1}|,$$

conforme se incrementa el número de nodos el parámetro c tiende a cero. El parámetro c_j depende localmente de la densidad de los nodos. Esta función nos permite reducir localmente el parámetro c en las regiones donde se requiere de mayor exactitud, lo cual corresponde a las regiones con altos gradientes.

Con el esquema de refinamiento local de nodos para problemas dependientes del tiempo descrito en párrafos anteriores, ahora abordaremos la ecuación analizada de tipo convección-difusión; ver sección 2.4 del capítulo 2, con un valor alto en el número de Péclet; $Pe = 1000$, determinado por los valores $\beta=0.0001$ y $\mu=-0.1$. Para este propósito, los siguientes valores fueron selec-

cionados: $\theta_r=0.01$, $\theta_c=0.0005$, $\Delta t=0.01$, $t_{max}=1$ con una distribución inicial uniforme de nodos $N=601$.

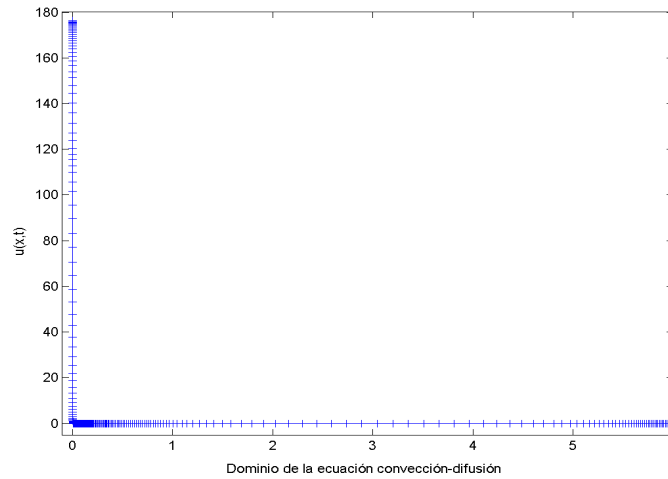


Figura 4.8: Solución numérica de la EDP convección-difusión con refinamiento local de nodos, $t_{max} = 1$.

En la figura 4.8 mostramos la reconstrucción de la aproximación numérica obtenida con el esquema de refinamiento local de nodos. Se observa de esta gráfica que la zona de alto gradiente es correctamente capturada, es decir, no se observan fuertes oscilaciones numéricas. El número de nodos finales es $N=248$ con un error $ECM=1.792 \times 10^{-2}$ y $E_{max}=5.329 \times 10^{-2}$. De las 50 veces que se ejecutó el esquema de RLN, solo 21 veces fue necesario refinar/remover los nodos.

Como se aprecia en la figura 4.8, el RLN logró detectar la zona de alta variación espacial de manera eficiente. Observe que cerca del centro de la gráfica ($x = 3$) se requirió una densidad menor de nodos y cerca de las fronteras la densidad de nodos se incrementó. Esta disminución gradual de nodos corresponde a la restricción 2:1 impuesta en los niveles de refinamiento.

Capítulo 5

Refinamiento Local de Nodos y Descomposición de Dominio

En este capítulo se integra en un solo algoritmo el método de descomposición de dominio (MDD) y el refinamiento local de nodos (RLN) para aproximar la solución numérica de ecuaciones en derivadas parciales estacionarias en dos dimensiones. Se describe en detalle un algoritmo eficiente para la creación de las zonas de traslape entre subdominios. El esquema propuesto es aplicado a dos problemas con gradientes abruptos; Poisson y Convección-Difusión, mostrando la efectividad del algoritmo propuesto.

5.1. Motivación

A través del proceso de refinamiento de nodos paulatinamente se van insertando nodos en las zonas donde se requiere de una mayor exactitud numérica. Este proceso de inserción de nodos, incrementa el número de nodos totales que cubren al dominio Ω . Mediante el esquema de colocación asimétrico, la discretización de la ecuación diferencial parcial genera un sistema denso, por lo general mal condicionado, y cuya solución tiene una complejidad $O(N^3)$ resultante de la factorización LU . Bajo estas premisas, el esquema de colocación

asimétrico sólo es viable para algunos cientos o miles de nodos. Por lo tanto, el uso de descomposición de dominio es recomendable para disminuir el costo computacional, así como para mejorar el número de condición del sistema algebraico.

5.2. Partición de Datos

El esquema de partición de datos consiste en dividir o particionar el dominio discreto Ω en P subdominios en donde cada subdominio contenga una cantidad similar de elementos.

$$\Omega = \bigcup_{i=1}^P \Omega_i, \quad |\Omega_i| \approx |\Omega_j| \quad \text{para } i \neq j, \quad (5.1)$$

con una intersección vacía entre subdominios $\Omega_i \cap \Omega_j = \emptyset$, $i \neq j$ con $i, j = 1, \dots, P$. Por lo general, la partición de los datos se realiza bajo algunas restricciones, como son: mantener el aspecto del radio de cada subdominio y minimizar los nodos colindantes entre las particiones. La minimización de nodos colindantes es una restricción importante ya que implícitamente lo que se busca es minimizar el tiempo de envío y recepción de datos entre procesadores. El particionar un dominio bajo las restricciones anteriores es un problema de naturaleza combinatorio, y por lo tanto en la práctica no es posible determinar el óptimo global, lo que se hace es determinar una partición de datos subóptima.

En general, los algoritmos dedicados a la partición de datos [151, 152, 153, 154] están enfocados a la partición de grafos o de mallas sin estructura. El esquema de discretización empleado en este trabajo no requiere de una malla de tipo Delaunay o un grafo para poder utilizar alguna aplicación de software para la partición de datos.

Para la partición de datos, empleamos el esquema de bisección recursiva basada en coordenadas (BRC) [155]. Este esquema de partición consiste en ordenar sobre el eje- x y dividir en dos subdominios, para cada subdominio obtenido se ordena sobre el eje- y y se vuelve a dividir en dos subdominios, este

proceso se repite recursivamente.

El esquema BRC produce subdominios con aproximadamente el mismo número de elementos. La complejidad computacional de este enfoque es $O(N \log N)$, debido a que se consideró el método de QuickSort para realizar el ordenamiento [156]. La ventaja de este tipo de partición de datos es la implementación sencilla y la eficiencia computacional.

El esquema de partición de datos utilizado es un método básico que no genera particiones óptimas en el sentido de minimizar los nodos colindantes entre particiones. Constituye un primer enfoque para abordar el tema de partición de datos.

5.3. Zonas de Traslape

La determinación de las zonas de traslape es requerida cuando se trabaja con el algoritmo de Schwarz con regiones de traslape con coincidencia o sin coincidencia de nodos. En esta sección, se propone un algoritmo eficiente para la creación de las zonas de traslape entre subdominios para nodos distribuidos de manera no uniforme en una y dos dimensiones.

La idea para conformar la región de traslape se tomó del libro [104]. Sin embargo, dicha idea es muy general y está enfocada a elemento finito. Por lo tanto, no es directamente aplicable al esquema numérico expuesto a lo largo de la tesis.

5.3.1. Descripción General

En lo subsecuente se considera que el dominio Ω se encuentra particionado en P subdominios, ver sección 5.2. Los subdominios obtenidos son el punto de partida para los siguientes algoritmos.

Para facilitar la exposición, consideraremos un solo subdominio Ω_j . Lo que se busca es dilatar el subdominio Ω_j , sin embargo, no conocemos cuáles elemen-

tos de Ω_j conforman la frontera. El algoritmo para crear la zona de traslape o la expansión de los nodos frontera, se puede caracterizar empleando la notación de conjuntos.

Considere ahora que deseamos expandir la frontera del subdominio Ω_j para determinar el conjunto de nodos expandidos Ω_j^e . Esta idea la podemos expresar como

$$\Omega_j^e = V(\Omega_j) - \Omega_j, \quad (5.2)$$

donde $V(\cdot)$ representa a la función que nos determina los vecinos más cercanos para cada nodo en Ω_j . La cardinalidad de $V(\Omega_j)$ es superior a la cardinalidad de Ω_j , de hecho el conjunto Ω_j está contenido en $V(\Omega_j)$. Con base en estas dos operaciones; búsqueda de vecinos más cercanos y resta de conjuntos, se tiene ya caracterizado el algoritmo de expansión de la frontera (dilatación) en Ω_j . Los elementos del conjunto de nodos expandidos Ω_j^e serán los nodos que conformarán la frontera artificial Γ .

La idea expuesta anteriormente se puede aplicar de manera recursiva para así crear zonas con una mayor región de traslape.

Resulta evidente que para lograr obtener un buen algoritmo computacional, es necesario poder realizar las operaciones de búsqueda de vecinos y la resta entre conjuntos de manera eficiente. En las siguientes secciones se describe de manera progresiva la determinación de dichos algoritmos, primero se exponen las ideas para datos en una dimensión, posteriormente con base en la experiencia adquirida se propone el algoritmo para datos en dos dimensiones.

5.3.2. Traslape en una Dimensión

En esta sección se describe un algoritmo para crear la zona de traslape entre dos subdominios para datos en una dimensión.

Considere al dominio $\Omega = [0, 1]$, que es discretizado por un conjunto de N nodos aleatorios $\{x_i\}_{i=1}^N \in \Omega$, los cuales están ordenados de forma ascendente, $x_0 < x_1 < \dots < x_N$. Considere ahora que empleando un esquema de partición

de datos el dominio Ω es particionado en dos subdominios $\{\Omega_1, \Omega_2\}$, obteniendo

$$\Omega_1 = \{x_0, x_1, \dots, x_k\},$$

$$\Omega_2 = \{x_{k+1}, x_{k+2}, \dots, x_N\},$$

donde la intersección es vacía $\Omega_1 \cap \Omega_2 = \emptyset$ y se tiene un mismo número de elementos en cada subdominio $|\Omega_1| = |\Omega_2|$.

Una vez determinados los subdominios $\{\Omega_1, \Omega_2\}$, nuestro siguiente paso es crear una zona de traslape entre ambos subdominios para poder aplicar el algoritmo de Schwarz.

Para conformar la zona de traslape entre Ω_1 y Ω_2 , la idea consiste en expandir/agregar el nodo más cercano a la frontera de cada subdominio. Al ser datos en una dimensión y estar ordenados de manera ascendente, es claro que los nodos que conforman la frontera de cada subdominio se encuentran al final e inicio del arreglo ordenado. Para ello, agregemos el nodo $x_{k+1} \in \Omega_2$ a Ω_1 y el nodo $x_k \in \Omega_1$ a Ω_2 , obteniendo los conjuntos expandidos

$$\Omega_1^E = \Omega_1 + \Omega_1^e = \{x_0, \dots, x_k, x_{k+1}\},$$

$$\Omega_2^E = \Omega_2 + \Omega_2^e = \{x_k, x_{k+1}, \dots, x_N\},$$

donde $\Omega_1^e = \{x_{k+1}\}$ y $\Omega_2^e = \{x_k\}$ corresponden a la zona de traslape entre ambos subdominios: $\{x_k, x_{k+1}\}$.

Lo mostrado anteriormente nos proporciona una idea de como construir la zona de traslape para datos en dos dimensiones, la cual consiste en expandir los nodos que conforman la frontera de cada subdominio. La restricción más importante que hay que considerar es que no contamos con una rejilla equiespaciada en dos dimensiones, y por lo tanto no es trivial determinar cuando un nodo se encuentra en la frontera de cada subdominio.

5.3.3. Traslape en dos Dimensiones

En esta sección, se describe y analizan los pasos requeridos para construir las zonas de traslape entre subdominios considerando datos distribuidos de

manera no uniforme en $2-d$.

Para construir el algoritmo de expansión/dilatación de los nodos que conforman la frontera en cada subdominio, requerimos realizar de manera eficiente las siguientes dos tareas:

- Encontrar los vecinos más cercanos de un nodo $x \in \Omega_j$.
- Dado un nodo $x \in \Omega$, determinar si x pertenece a Ω_j .

Por lo descrito en la sección 5.3.1, el primer punto mostrado arriba corresponde a la función $V(\cdot)$ que determina los vecinos más cercanos. El segundo punto corresponde a la resta entre dos conjuntos de datos $V(\Omega_j) - \Omega_j$.

La idea general la describimos como: para cada nodo contenido en Ω_j obtener sus K -vecinos más cercanos, para cada uno de los K -vecinos se determina si pertenecen a Ω_j . Si pertenece no lo agrego, en caso contrario el nodo es agregado a Ω_j y es marcado como nodo de tipo frontera artificial.

La complejidad computacional para determinar los K -vecinos más cercanos de un nodo $x \in \Omega_j$ es de orden constante $O(1)$, ver sección 4.4.1. A través de la tesis los K -vecinos serán referenciados con el símbolo K_v , con $K_v \in \Omega$.

La eficiencia del proceso de expansión/dilatación está determinada por la operación de pertenencia. Una primera idea para la operación de pertenencia consiste en que dado un nodo $x \in \Omega_j$ y sus K -vecinos, tomar cada nodo vecino y realizar una búsqueda lineal sobre el conjunto de nodos Ω_j para determinar si pertenece a Ω_j .

La complejidad computacional de esta primera propuesta resulta evidente debido a que la búsqueda lineal tiene una complejidad $O(M)$, siendo M el número de elementos en Ω_j . Esta búsqueda lineal se realiza K -veces, siendo K el número de nodos vecinos, y a su vez esto se realiza M -veces. Por lo tanto, se tiene una complejidad cuadrática para el primer algoritmo de pertenencia

$$O(KM^2). \tag{5.3}$$

Sabemos que $M \ll N$, siendo N el número de nodos en Ω y M el número de elementos en Ω_j . De este hecho se podría deducir que la complejidad cuadrática del algoritmo de pertenencia descrito anteriormente es aceptable. Sin embargo, en la práctica $700 \leq M \leq 3000$, lo cual resulta computacionalmente costoso debido a la complejidad cuadrática. Es por ello que se requiere disminuir el orden cuadrático del algoritmo de pertenencia.

Para disminuir la complejidad cuadrática $O(KM^2)$ del algoritmo de pertenencia, vamos a requerir que cada nodo en el dominio Ω esté definido con un identificador único o llave. La llave la podemos seleccionar de distintas maneras, mediante funciones de indexación tipo hash que actúen sobre la representación binaria de las coordenadas [157], o algún tipo de mezcla de bits al estilo de multipolos rápidos [158]. Ambas técnicas para la generación de llaves son muy generales y se pueden aplicar a un extenso rango de problemas. En este trabajo, basándonos en el dominio del problema y en la estructura de datos utilizada para representar los nodos, definiremos un tipo de llave de propósito particular que es fácil de implementar y no genera colisiones entre llaves.

La llave la definimos como un número entero que corresponde al índice del arreglo lineal en donde se almacenan los nodos. Así, cada vez que se agrega un nuevo nodo en Ω , al momento de insertarlo se le asigna la nueva llave

$$\text{llave}_{n+1} = \text{llave}_n + 1,$$

donde llave_n corresponde al valor de la última llave insertada. Con esta regla de asignación de llaves, se tiene un esquema de identificación único para cada nodo en el dominio Ω . La llave es incluida en la estructura de datos utilizada para representar a cada nodo en Ω , ver sección 4.4.3. Por lo tanto, particionando al dominio Ω en P -subdominios cada elemento de cada subdominio cuenta con un identificador único que hace referencia a la llave en Ω .

Con base en el esquema de llaves ahora definimos la función de pertenencia. El esquema propuesto consiste en una búsqueda binaria, consistiendo en los siguientes pasos:

- Obtener el conjunto de llaves $L = \{llave_i\}_{i=1}^M$ del subdominio Ω_j .
- Ordenar de forma ascendente el conjunto de llaves L .
- Para cada nodo en Ω_j realizar las siguientes tareas:
 - Determinar sus K -vecinos más cercanos y para cada vecino obtener su llave $L^v = \{ll_1, \dots, ll_k\}$.
 - Para cada elemento de L^v realizar una búsqueda binaria sobre el conjunto L . Con base en ello se determina si pertenece o no pertenece al conjunto L .

El último paso realizado nos permite determinar cuándo un nodo vecino de $x \in \Omega_j$ está contenido en el subdominio Ω_j . En caso de no estar contenido, entonces el nodo vecino referenciado corresponde a un nuevo elemento de la frontera expandida, el cual es agregado como un nodo de tipo frontera artificial (NFA). La inserción del nuevo nodo de tipo NFA se realiza después de que se recorrieron todos los nodos en Ω_j .

El algoritmo anterior genera el conjunto de nodos expandidos que corresponde a la frontera artificial de cada subdominio. Sin embargo, en dicho conjunto de nodos existen nodos repetidos. Para eliminar los nodos repetidos por cada nodo que no pertenece a Ω_j se realiza una búsqueda binaria sobre el conjunto Ω_j^e y con base en ello determinamos si incluimos o no incluimos el nodo que conformará la frontera expandida.

La complejidad computacional del algoritmo de pertenencia propuesto basado en llaves con una búsqueda binaria (APLB) es

$$O(KM \log(M)), \quad (5.4)$$

donde K es el número de nodos vecinos de un nodo $x \in \Omega_j$ y M es la cardinalidad del subdominio Ω_j . Considerando a K como un valor constante, se determina que la complejidad computacional de APLB es

$$O(M \log(M)), \quad (5.5)$$

este orden de complejidad es comparativamente inferior con respecto a la complejidad cuadrática $O(M^2)$ obtenida con el primer algoritmo propuesto.

5.3.4. Niveles de Expansión

El procedimiento de expansión de la frontera de cada subdominio se puede realizar en más de un nivel, para así crear una mayor zona de traslape. Es por ello que surge la pregunta: ¿Cuántas veces es necesario expandir la frontera de cada subdominio? Desafortunadamente no existe teoría al respecto que nos indique como contestar dicha pregunta, la respuesta encontrada fue determinada con respecto de múltiples simulaciones con distintas ecuaciones diferenciales parciales y variando los niveles en las zonas de traslape.

Cabe mencionar que este problema no se presenta con datos distribuidos de manera uniforme, la inclusión del refinamiento local de nodos con descomposición de dominio origina una clase distinta de problemas, debido principalmente a la distribución no uniforme de los datos.

La expansión de la frontera de cada subdominio se determinó mediante los siguientes tres pasos:

- 1.- Expandir la frontera y asignar los nodos expandidos al tipo nodo Interior.
- 2.- Crear un rectángulo con los límites del subdominio del paso 1 e insertar los nuevos nodos que estén contenidos en el rectángulo pero que no estén dentro del subdominio. El tipo de nodo asignado es Interior.
- 3.- Expandir la frontera y asignar los nodos expandidos al tipo de nodo Frontera Artificial (Γ).

La parte crucial del esquema propuesto de niveles de expansión corresponde al segundo punto. Lo que está implícito en el segundo punto es la creación de un subdominio de forma regular, es decir, un subdominio con nodos interiores que se asemejan a rectángulos.

Los niveles de expansión constituyen un punto crucial en el esquema de refinamiento local de nodos con descomposición de dominio.

5.3.5. Actualización de la Frontera Artificial

Una vez construida la región de traslape, el siguiente paso es determinar los nodos en Ω_i que actualizan el valor de la frontera artificial de algún subdominio Ω_j , para $j = 1, \dots, P$ con $i \neq j$. Esto es requerido en el algoritmo de Schwarz en la versión aditiva y multiplicativa con coincidencia de nodos (ver sección 3.2).

Es conveniente aclarar que cada nodo tiene asociado un identificador para el tipo de nodo (TN) definido como: $TN = \{\text{Interior, Frontera-Real, Frontera-Artificial, Interior-Actualiza}\}$. Cuando no trabajamos con descomposición de dominio, la señalización de los nodos se reduce a los dos primeros casos, es decir, la discretización de la ecuación diferencial parcial está conformada por los nodos que pertenecen al interior del dominio y los nodos que conforman la frontera real del problema. En nuestro caso estamos trabajando con descomposición de dominio y es por ello que en la sección anterior se marcaron los nodos que conforman la frontera artificial Γ del problema. Ahora, sólo resta marcar los nodos que son interiores a un subdominio y que son frontera artificial de otro subdominio (IA).

Para un subdominio Ω_j el proceso de marcado de los nodos que actualizan la frontera artificial está compuesto de los siguientes pasos

- $\forall x \in \Gamma$ de Ω_j realizar:
 - Para $i = 1, \dots, P$ con $i \neq j$
 - Si $x_k \in \Omega_i \Rightarrow$ sabemos que $x_k \doteq x_l \in \Omega_j$ y por lo tanto $x_k^{id} = IA$.

La eficiencia del algoritmo está determinada al inicio del tercer paso, para ello hacemos uso de las llaves únicas y mediante una búsqueda binaria determinamos ‘Si $x_k \in \Omega_i$ ’, en caso de ser afirmativa la respuesta, entonces obtenemos

el nodo $x_k \in \Omega_i$ que satisface la pregunta anterior y le asignamos el tipo de nodo Interior-Actualiza; $x_k^{id} = IA$, el superíndice id corresponde al identificador del nodo, IA es el acrónimo de Interior-Actualiza. Este proceso de marcado de nodos se realiza antes de llamar al algoritmo de Schwarz.

5.4. Esquema General

En esta sección se integra en un solo algoritmo el método de descomposición de dominio (MDD) y el refinamiento local de nodos (RLN) para ecuaciones en derivadas parciales estacionarias. El algoritmo propuesto puede ser aplicable cuando se trabaje con otros esquemas de discretización numérica.

En el algoritmo 3 se muestran los pasos que conforman la propuesta de integrar en un solo esquema el método de refinamiento local de nodos desarrollado en el capítulo 4, con el método de descomposición de dominio planteado en el capítulo 3. Cabe mencionar que la integración de MDD y RLN en un solo algoritmo no es una tarea trivial, debido a la no uniformidad en la distribución de los nodos, a la estructura de datos requerida, y a la creación de las zonas de traslape entre subdominios.

Algoritmo 3 Refinamiento Local de Nodos y Descomposición de Dominio (RLNDD)

Inicializar con una distribución uniforme de nodos.

while $error > \varepsilon$ **do**

- 1.- Particionar a Ω para crear los subdominios Ω_i , $i = 1, \dots, P$.
- 2.- Conformar las zonas de traslape.
- 3.- Resolver la EDP con el método de descomposición de dominio.
- 4.- Integrar la solución de cada subdominio al dominio Ω .
- 5.- Realizar el refinamiento local de nodos.

end while

Al inicio del algoritmo se cuenta con una distribución uniforme de nodos

que son generados considerando el esquema de celda×celda. El criterio de paro del algoritmo 3 está definido en la sección 4.5.1.

Para la partición de datos, ver paso 1, empleamos partición recursiva basada en coordenadas con restricción en la distribución de los nodos en cada subdominio, ver sección 5.2.

El segundo paso es la creación de zonas de traslape, ver sección 5.3. La idea básica consiste en expandir la frontera de cada subdominio $\{\Omega_i\}_{i=1}^P$ para crear así el traslape entre subdominios y poder aplicar el algoritmo de Schwarz.

El tercer paso del esquema propuesto trata sobre la aproximación de la solución mediante el esquema de colocación asimétrico con funciones de base radial y descomposición de dominio. El esquema de discretización no depende de la distribución de los nodos, es decir, los nodos pueden estar distribuidos de manera uniforme o no uniforme.

La diferencia con el estado del arte, radica en la combinación del refinamiento local de nodos con descomposición de dominio. Como hemos visto a lo largo de este capítulo, la distribución no uniforme de los nodos nos genera una clase distinta de problemas, como son: la estructura de datos para representar cada subdominio, la creación de las zonas de traslape y la actualización de los nodos que conforman la frontera artificial.

Una vez determinada la solución numérica $\tilde{u}(\cdot) \in \Omega_i$ con $i = 1, \dots, P$ en el paso 3, la siguiente tarea es unir de forma eficiente la solución de cada subdominio al dominio global Ω . Para implementar de manera eficiente la idea anterior, hacemos uso del identificador único o llave asociada a cada nodo del dominio Ω . El algoritmo consiste en cargar en memoria los nodos del dominio Ω y los nodos de cada subdominio Ω_i , recorreremos los nodos de cada subdominio y con base en el identificador único insertamos $\tilde{u}(\cdot)$ en Ω . Este proceso de inserción es de orden constante ya que el identificador único es en sí el índice de la lista dinámica de los nodos en Ω , por lo tanto el orden de complejidad es $O(N)$, siendo N el número de nodos en Ω . En las zonas de traslape se cuenta

con varias soluciones \tilde{u} provenientes de cada subdominio, el valor \tilde{u} asignado al dominio global Ω es el promedio de dichas aproximaciones.

Finalmente, el paso 5 correspondiente al refinamiento local de nodos se planteó en el capítulo 4. De manera general, este paso consiste en utilizar la solución obtenida $\tilde{u}(\cdot) \in \Omega$ para construir el error de interpolación $\eta(x)$ que refleja localmente que tan suave o pronunciada es la solución. Con base en dicho indicador, se refinan los nodos que estén por encima de un cierto porcentaje θ_{ref} obteniendo así una nueva distribución de nodos. Esta nueva distribución de nodos es el punto de partida para la siguiente iteración del algoritmo 3.

Nota : El algoritmo 3 está compuesto de 5 procedimientos separados, cada uno de ellos se encuentra programado en archivos ejecutables distintos. La comunicación entre cada ejecutable se realiza a través de archivos binarios. Este enfoque es modular y flexible, lo cual nos permitirá en un futuro expandir la funcionalidad del esquema propuesto sin tener que programar todos los módulos.

En el apéndice B indicamos brevemente las modificaciones requeridas en el algoritmo 3 para poder extenderlo a un esquema en paralelo.

5.4.1. Número de Particiones Adaptable

Conforme iteramos el algoritmo 3, de manera incremental se van insertando nodos en las regiones donde se requiere de una mayor exactitud numérica. Manteniendo fijo el número de particiones o subdominios P , el incremento en el número de nodos que cubren a Ω induce un incremento en el número de nodos en los subdominios

$$N^{k+1} > N^k \implies M^{k+1} > M^k \approx N^k / P, \quad (5.6)$$

donde k denota el índice en la iteración en el algoritmo 3, N^k corresponde a la cardinalidad de Ω en la iteración k -ésima y M^k es el número de nodos en cada subdominio.

A partir de un cierto paso de iteración $k > k_0$, el número de elementos en algún subdominio rebasará el número máximo de nodos permitido M_{max} . La restricción en el número máximo de nodos por subdominio está en función del tiempo de procesamiento, o bien por el número de condición del sistema, por lo general $M_{max} \in [700, 3000]$. Por lo tanto, es necesario satisfacer la restricción en el número de nodos contenido en cada subdominio

$$M^k \leq M_{max}. \quad (5.7)$$

Para satisfacer la restricción anterior es necesario incrementar el número de particiones P conforme se itera el algoritmo 3. El incremento en el número de particiones se puede realizar de manera manual, sin embargo, resulta conveniente realizarlo de manera automática. La selección automática del número de particiones es realizada en el algoritmo de bisección recursiva basada en coordenadas (BRC), ver sección 5.2. Lo que se condiciona es cuando particionar, al comienzo del esquema BRC preguntamos si el número de nodos del dominio Ω es menor que un cierto valor M_{max} , si se satisface la condición entonces no particionamos, en caso contrario llamamos a una función recursiva que divide al dominio Ω en dos subdominios sobre el eje- x , después sobre cada subdominio obtenido se invoca a la función recursiva para que particione sobre el eje contrario, este proceso se repite mientras el número de elementos en alguna partición sea mayor que el valor de M_{max} .

5.5. Resultados Numéricos

Para mostrar la efectividad del algoritmo desarrollado en este capítulo; Refinamiento Local de Nodos con Descomposición de Dominio (RLNDD). En esta sección resolvemos numéricamente dos problemas estacionarios en dos dimensiones: el primer problema es la ecuación de Poisson y el segundo problema corresponde a la ecuación de tipo Convección-Difusión. Ambas ecuaciones en derivadas parciales contienen regiones con cambios espaciales abruptos. Se ana-

liza numéricamente las características o parámetros libres del algoritmo propuesto.

Se determinó que la función radial multicuádrada requiere de una cantidad menor de nodos para alcanzar un error establecido en comparación con la función radial de tipo placa delgada. Además, la selección del parámetro de refinamiento no es un parámetro crítico que afecte la calidad de la aproximación, solo afecta al número de nodos y al número de las iteraciones en el algoritmo 3. Adicionalmente, en todos los casos analizados fué posible capturar correctamente las regiones con alta variación espacial, insertando nodos en las zonas donde se requiere de una mayor exactitud numérica y empleando descomposición de dominio para reducir el costo computacional.

5.5.1. Poisson en 2-d

El objetivo de esta sección es estudiar el efecto que tiene en la aproximación numérica el esquema de partición de datos, la selección de la función radial y el porcentaje de refinamiento.

Para tal efecto, consideraremos la extensión del problema en una dimensión mostrado en la sección 4.6.1 a dos dimensiones, el cual corresponde al siguiente problema de Poisson

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \quad \text{en } \Omega \subset \mathbb{R}^2, \quad (5.8)$$

$u = u(x, y)$, sujeta a las condiciones de frontera tipo Dirichlet

$$u = g \quad \text{en } \partial\Omega, \quad (5.9)$$

donde $\Omega = [-1, 1]^2$. La solución analítica está definida como

$$u(x, y) = \tanh(x + 50y), \quad (5.10)$$

a partir de $u(x, y)$ se obtiene el término fuente en (5.8)

$$f(x, y) = 5002 \cdot \tanh(x + 50y)(\tanh(x + 50y)^2 - 1). \quad (5.11)$$

En la figura 5.1 se muestra la forma de la solución analítica. Como se aprecia en dicha gráfica, paralelo al eje x y con la coordenada $y = 0$, se tiene una zona de alta variación espacial. Fuera de dicha zona se observa una región sin variaciones, con un valor constante -1 y 1 respectivamente. La aproximación numérica en la zona de alto gradiente no es un problema de fácil solución debido a la fuerte variación espacial, lo cual induce oscilaciones numéricas cerca de la -discontinuidad- (fenómeno de Gibbs).

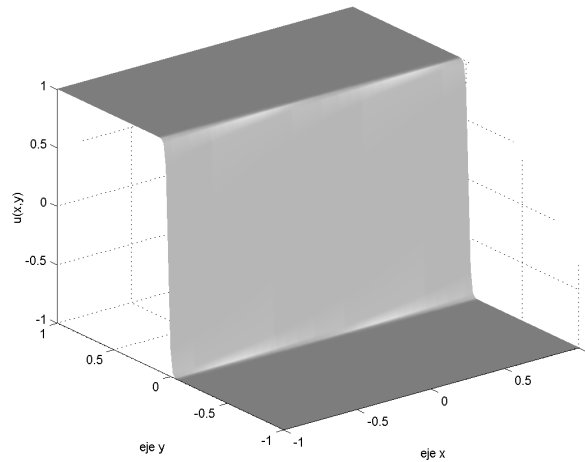


Figura 5.1: Solución analítica $u(x, y) = \tanh(x + 50y)$.

Partición Estática

Primero analizaremos el esquema estático para la partición de datos, y la selección de la función radial. Como funciones radiales seleccionamos placa delgada $\phi(r) = r^4 \log r$ y multicuádrica $\phi(r) = \sqrt{r^2 + c^2}$. Para la función multicuádrica el esquema de actualización del parámetro c se ajusta conforme se van insertando los nuevos nodos. El parámetro c se va decrementando por un factor de 0.5 [103]

$$c_j^{\text{nuevo}} = c_j^{\text{viejo}} / 2. \quad (5.12)$$

Como datos iniciales se tiene una distribución uniforme de nodos $N = 1156$, un umbral de refinamiento $\theta_{ref} = 0.2$ y $c_{ini} = 0.1$ para el núcleo multicuádrico.

El criterio de paro se establece como $E_{max} < 9 \times 10^{-3}$, el cual corresponde a la máxima diferencia en valor absoluto entre la solución analítica y la solución numérica calculada sobre todo el dominio Ω .

En la tabla 5.1 se muestran los resultados obtenidos. En la primera columna se despliega el tipo de función radial, la segunda columna corresponde al tipo de partición, el número de subdominios se indica en la tercera columna. En la cuarta columna se exhibe el número de nodos finales resultantes del proceso de refinamiento iterativo. Finalmente en la quinta y sexta columna se muestra el error cuadrático medio y el error en la norma infinita.

Función Radial	Esquema	Subdominios	Nodos	ECM	E_{max}
Placa delgada	Vertical	1×4	9587	0.001726	0.005178
Placa delgada	Caja	2×2	9862	0.002517	0.005853
Multicuádrico	Vertical	1×4	6085	0.001333	0.007920
Multicuádrico	Caja	2×2	5586	0.001006	0.007279

Tabla 5.1: Variación del núcleo radial y el esquema de partición de datos.

Se observa en la tabla 5.1 que en todos los casos analizados fue posible alcanzar el error deseado. La diferencia radica en el número de nodos requeridos, lo cual impacta en el tiempo de procesamiento y la memoria requerida. Por otra parte, como se indica en el trabajo de Driscoll [103], la incorrecta selección del parámetro de inicio c_{ini} para el núcleo multicuádrico nos puede ocasionar que se inserten más nodos de los necesarios. La manera en que seleccionamos el parámetro inicial c_{ini} fue restringiendo el número de condición de las matrices resultantes de la discretización, de tal forma se encuentren por debajo de un valor establecido 10^{15} .

Placa Delgada vs. Multicuádrico

En la sección 5.5.1 se mostró que con el refinamiento local de nodos y descomposición de dominio, es posible obtener una buena aproximación numérica

para el problema de Poisson analizado con las funciones radiales de tipo placa delgada y multicuádrlica. El resultado obtenido es válido, sin embargo se considero una partición estática de los datos. Es decir, seleccionamos a mano el número de particiones y la forma de cada partición, por lo tanto no es un método general.

Para desarrollar un método general, tenemos que dejar libre el número de particiones de tal forma que se vayan ajustando con respecto al número de nodos en cada iteración del algoritmo 3. En esta sección se muestra que con un esquema de partición dinámica de datos basada en coordenadas, la función radial de tipo multicuádrlica es comparativamente superior a lo obtenido con la función radial de tipo placa delgada, donde la ganancia se expresa en un menor número de nodos y un mejor número de condición del sistema.

Considere una distribución inicial de nodos $N = 1156$. El esquema de partición dinámica de datos está basado en coordenadas en donde el criterio de particionar el dominio depende del número de nodos máximos permitidos por subdominio, en particular $|\Omega_j| \leq 1300$, $j = 1, \dots, P$; siendo P el número de subdominios. Para la función radial multicuádrlica seleccionamos el parámetro inicial $c_{ini} = 0.1$, en donde el ajuste dinámico del parámetro c está definido por la ecuación 5.12. El parámetro que controla el refinamiento local de nodos lo definimos como $\theta_{ref} = 0.3$. El criterio de paro se establece cuando $E_{max} < 9 \times 10^{-3}$.

En la tabla 5.2 se muestra el resultado obtenido con una selección dinámica en el número de subdominios variando la función de base radial. Como se observa en la segunda columna, en ambos casos se requirió aproximadamente el mismo número de iteraciones para alcanzar el error de aproximación requerido. Sin embargo, el núcleo multicuádrlico requirió de un menor número de nodos y por consecuencia de subdominios. Esto se traduce en un menor tiempo de procesamiento y por lo tanto en una mejora en cuanto al desempeño computacional. Un resultado similar se obtuvo con un esquema de particiones fijas mostrado en la sección anterior.

Función Radial	Iteraciones	Subdominios	Nodos	ECM	E_{max}
Multicuádrico	20	6	5144	0.001829	0.007360
Placa delgada	22	10	8519	0.002035	0.008390

Tabla 5.2: Variación del núcleo radial con refinamiento local de nodos, descomposición de dominio y particiones dinámicas.

Como se observa en las dos últimas columnas en la tabla 5.2, en ambos casos fue posible reducir el error de aproximación logrando llegar al valor pre-establecido. Sin embargo, el número de condición final para placa delgada está en el rango de 10^{14} y el obtenido con el núcleo multicuádrico está en el rango 10^8 . Esto nos indica que con el parámetro c ajustado dinámicamente es posible mantener el número de condición comparativamente inferior al obtenido con placa delgada. Esto nos permitirá trabajar con subdominios con un mayor número de nodos, para así poder tener una mayor zona de traslape, lo cual resultaría conveniente para mantener acotado el número de iteraciones para converger en el método de Schwarz [104]. Debido a que a mayor zona de traslape el algoritmo de Schwarz requiere de un menor número de iteraciones para converger a la solución.

Porcentaje de Refinamiento

El algoritmo de refinamiento local de nodos tiene un parámetro libre θ_{ref} , el cual indica el porcentaje de nodos θ_{ref} , con respecto de η^* , que van a ser refinados con base en el error local de interpolación, ver sección 4.3.1. Este parámetro es seleccionado por el usuario. Es por ello que surge la pregunta: ¿Cómo seleccionar dicho parámetro de tal suerte que alcancemos el error deseado?

En los siguientes párrafos se muestra que la selección de dicho parámetro afecta al número de iteraciones requeridas así como al número de nodos insertados. En todos los casos se logró llegar al error requerido.

Considere el problema de Poisson analizado en la sección 5.5.1 con una distribución inicial de nodos $N = 1156$, empleando el núcleo multicuadrático con un parámetro inicial $c_{ini} = 0.1$. Como número máximo de elementos por subdominio se considera $|\Omega_i| \leq 1300$, con $i = 1, \dots, P$. El criterio de paro se establece cuando $E_{max} < 9 \times 10^{-3}$.

En la tabla 5.3 se muestran los resultados obtenidos. La primera columna corresponde a la variación del parámetro de refinamiento θ_{ref} . En la segunda columna se indica el número de iteraciones requeridas por el algoritmo 3 para alcanzar el error buscado. El número de particiones finales se despliega en la tercera columna, la cuarta columna corresponde al número de nodos finales. En la quinta y sexta columna se indica el error cuadrático medio y el error en la norma infinita calculado sobre todo el dominio Ω .

θ_{ref}	Iteraciones	Subdominios	Nodos	ECM	E_{max}
0.05	7	10	9458	0.000821	0.004332
0.1	11	6	5070	0.002244	0.006972
0.2	14	4	3819	0.002111	0.008438
0.3	20	6	5144	0.001829	0.007360
0.4	29	6	5189	0.000987	0.004563
0.5	38	5	4208	0.001051	0.003746

Tabla 5.3: Variación del parámetro de refinamiento θ_{ref} , con el núcleo multicuadrático.

Se observa en la tabla 5.3 que conforme incrementamos el parámetro θ_{ref} ocurren dos efectos:

- El número de iteraciones se incrementa.
- El número de nodos y subdominios disminuye.

En la última columna en la tabla 5.3 se observa que en todos los casos analizados se alcanzó el error requerido. Por lo tanto, la selección del parámetro

de refinamiento θ_{ref} no es un factor crítico que nos impida disminuir el error de aproximación.

Desde un punto de vista empírico el resultado obtenido es consistente con lo que se esperaba, debido a que si incrementamos θ_{ref} en cada iteración se insertan menos nodos y se requerirá de un mayor número de iteraciones para reducir el error de aproximación.

En la tabla 5.4 se muestra en detalle las iteraciones del algoritmo 3 para el caso $\theta_{ref} = 0.3$, lo cual corresponde al cuarto renglón en la tabla 5.3. El número en la iteración se indica en la primera columna, el número de subdominios requeridos se despliega en la segunda columna. El incremento paulatino en el número de nodos se exhibe en la tercera columna. Las mediciones del error cometido se señalan en la cuarta y quinta columna.

Se observa en la tabla 5.4 que conforme iteramos el algoritmo de refinamiento local de nodos con descomposición de dominio se va reduciendo paulatinamente el error cuadrático medio y el error en la norma infinita. Al mismo tiempo, como se observa en la segunda columna, el número de particiones se ajusta dinámicamente con respecto del número de nodos.

En la figura 5.2 se muestra la partición de datos y la aproximación numérica en las iteraciones $= \{0, 6, 13, 20\}$, correspondiente a los datos mostrados en la tabla 5.4. Como se observa en dichas gráficas, el refinamiento local de nodos se realiza en la zona de mayor gradiente logrando así reducir el error de aproximación. En la gráfica superior correspondiente a la iteración cero, se observa una mala aproximación numérica, sin embargo, conforme se van insertando los nodos la aproximación numérica va mejorando hasta alcanzar el error buscado.

Iteración	Subdominios	Nodos	ECM	E_{max}
0	1	1156	3.620	13.647
1	1	1174	3.369439	12.936703
2	1	1207	2.835797	11.227931
3	1	1238	2.314635	9.297117
4	1	1270	1.927845	7.481878
5	2	1324	1.703543	5.940608
6	2	1369	1.569170	5.944240
7	2	1396	1.237301	3.659237
8	2	1432	1.130226	3.571665
9	2	1456	1.053147	3.473156
10	2	1486	1.016876	3.451578
11	2	1516	0.999131	3.456837
12	2	1946	0.552101	2.246219
13	2	2045	0.434303	1.906522
14	2	2384	0.135675	0.808190
15	2	2519	0.118846	0.700411
16	4	2627	0.065679	0.368539
17	4	3309	0.013901	0.045664
18	4	3792	0.005627	0.018148
19	6	4763	0.003698	0.015959
20	6	5144	0.001829	0.007360

Tabla 5.4: Iteraciones en la adaptividad de los nodos con descomposición de dominio para el problema de Poisson, con el núcleo multicuadrado.

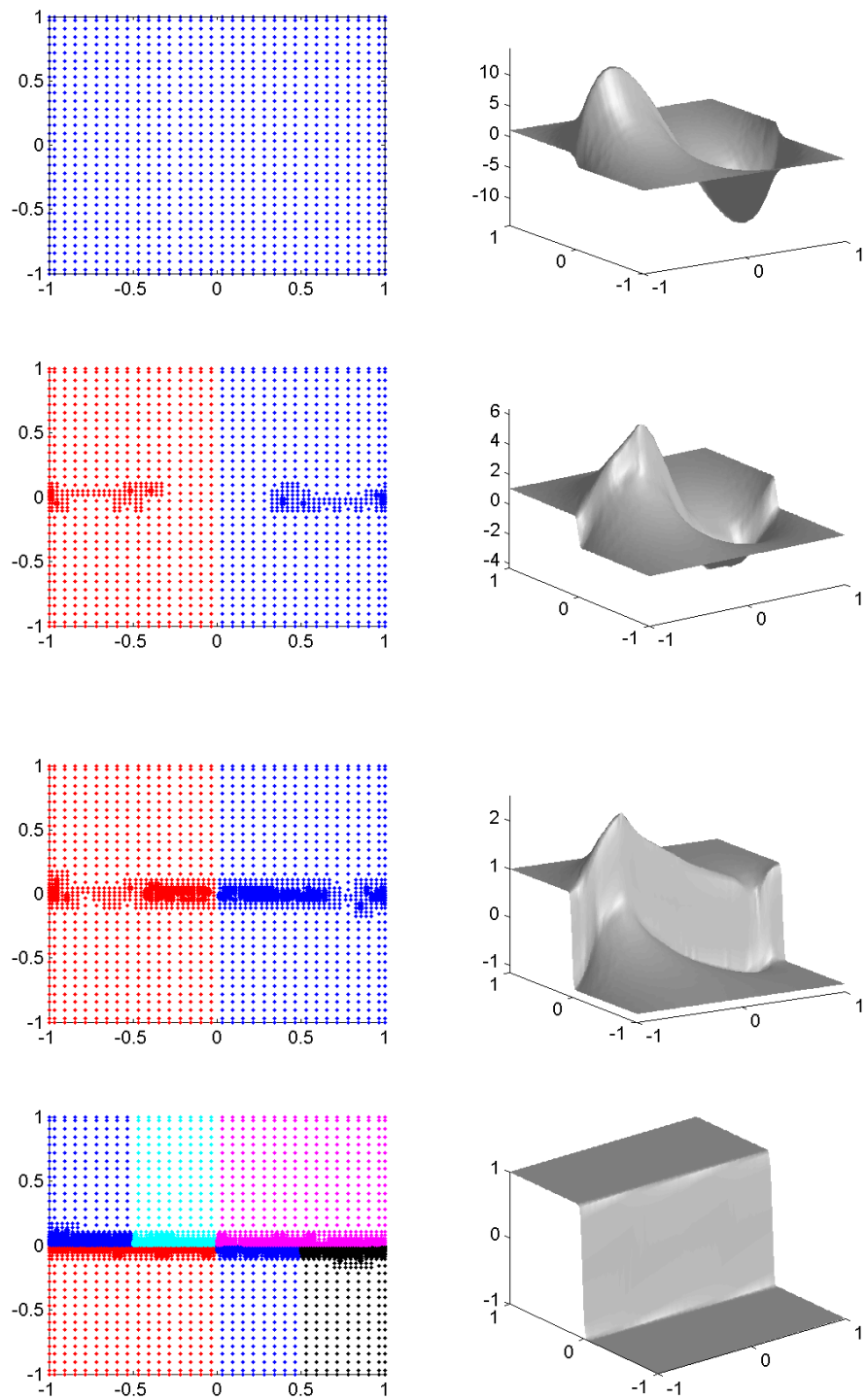


Figura 5.2: Refinamiento local de nodos con descomposición de dominio para el problema de Poisson, se muestran las iteraciones = $\{0, 6, 13, 20\}$. Función radial multicuádrlica.

5.5.2. Convección-Difusión en 2- d

En esta sección se analiza la ecuación de convección-difusión lineal estacionaria en dos dimensiones para distintos números de Péclet = $\{10^0, 10^1, 10^2, 10^3\}$. Nos interesa investigar la predominancia del término convectivo sobre el término difusivo y como afecta dicha selección en el error de aproximación empleando el núcleo multicuadrado.

Considere la siguiente ecuación diferencial parcial

$$\beta \nabla^2 u + \mathbf{v} \cdot \nabla u = 0, \quad \Omega \subset \mathbb{R}^2, \quad (5.13)$$

con $u = u(x, y)$, $\Omega = [0, 1]^2$ sujeta a las condiciones de frontera tipo Dirichlet

$$u = g \quad \text{en } \partial\Omega, \quad (5.14)$$

donde ∇ denota el operador del gradiente, β es el coeficiente de difusión, $\mathbf{v} = [v_x, v_y]^T$ es llamado el término convectivo (o coeficiente de velocidad), la función $g = g(x, y)$ se obtiene de la solución analítica, la cual está definida como

$$u(x, y) = a(e^{-c_x x} + e^{-c_y y}), \quad (5.15)$$

siendo $c_x = v_x/\beta$ y $c_y = v_y/\beta$. En particular trabajaremos con $v_x = v_y$, donde el número de Péclet queda definido: $Pe = v/\beta$.

Como datos iniciales se considera $a = 0.5$, $\beta = 1$ y el parámetro que variamos es el término convectivo para obtener los distintos números de Péclet. En todos los casos analizados se utiliza una distribución inicial uniforme de nodos $N = 1156$.

Péclet 10^0

Como primer caso consideremos un número de Péclet muy bajo: $Pe = 1$. El término convectivo es igual a término difusivo $\beta = v = 1$, en donde la forma de la solución aproximada es del tipo de una función rampa, ver figura 5.3. Los errores obtenidos fueron $ECM = 2 \times 10^{-7}$ y $E_{max} = 6.9 \times 10^{-7}$ con $c =$

0.2. Como era de esperarse, para un número de Péclet bajo la función analítica no presenta ninguna fuerte variación espacial y por lo tanto no es necesario emplear adaptividad local de nodos.

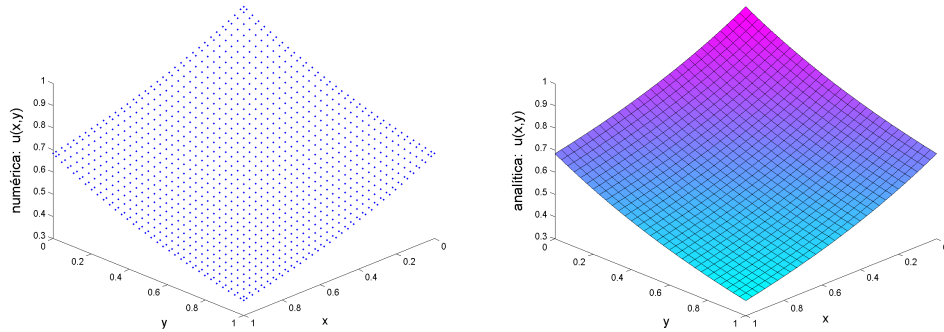


Figura 5.3: Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^0$.

Péclet 10^1

Ahora investigaremos el caso cuando el término convectivo es ligeramente superior al término difusivo: $Pe = 10$. Como se observa en la figura 5.4, la función aproximada no es completamente suave. Sin embargo, se obtuvo una buena aproximación numérica, donde los errores obtenidos fueron $ECM = 8 \times 10^{-6}$ y $E_{max} = 3.8 \times 10^{-5}$ con $c = 0.2$. En este caso tampoco fue necesario realizar el refinamiento local debido a la alta exactitud numérica.

Péclet 10^2

Como siguiente caso consideraremos que el término convectivo domina al término difusivo: $Pe = 100$. En este caso la solución presenta un gradiente pronunciado cerca de los ejes $x = 0$ y $y = 0$, fuera de dicha zona se tiene una región con un valor constante cero, ver figura 5.5. Con la distribución inicial de nodos $N = 1156$ fue posible capturar la zona con la variación espacial obteniendo una buena aproximación numérica; $ECM = 3 \times 10^{-3}$ y $E_{max} =$

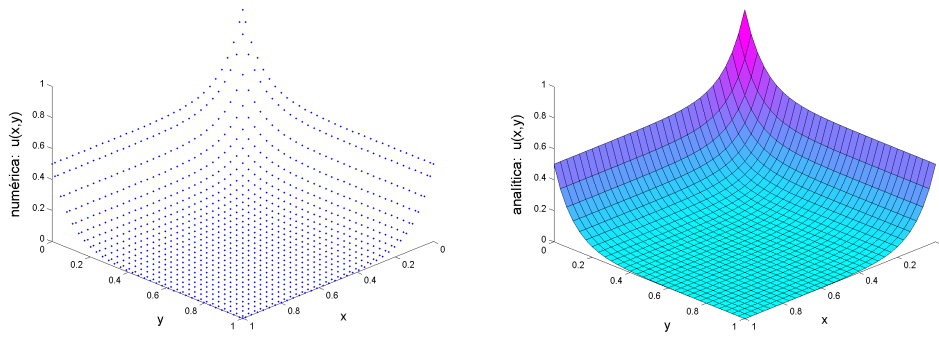


Figura 5.4: Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^1$.

4.1×10^{-2} con $c = 0.2$. Si cambiamos el número de nodos iniciales a $N = 324$ y seleccionando el parámetro c de acuerdo al error residual calculado en una muestra de los nodos [75], se obtiene $c = 0.47$ con un error máximo y un error cuadrático medio del orden de 10^{-2} . Esto nos indica que para lograr disminuir el error de aproximación no basta con el esquema de c -refinamiento, es necesario emplear también el esquema de h -refinamiento.

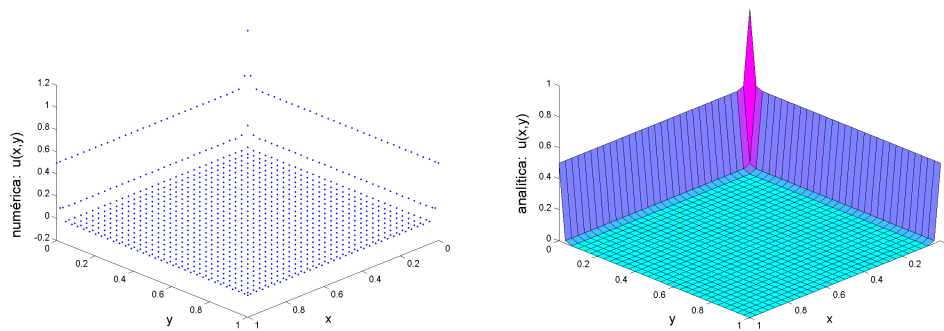


Figura 5.5: Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^2$.

El problema al seleccionar el parámetro c empleando el error residual sobre una muestra de los datos [75], radica en cómo tomar una muestra representativa de los nodos, y adicionalmente en cada paso del algoritmo 3 se requiere minimizar el residual $U(c) = \|Lu - f\|_2$. Sin embargo, este esquema nos per-

Pe	ECM	E_{max}
1	2×10^{-7}	6.9×10^{-7}
10	8×10^{-6}	3.8×10^{-5}
100	3×10^{-3}	4.1×10^{-2}

Tabla 5.5: Error obtenido conforme incrementamos el número de Péclet.

mite determinar el parámetro c de una manera automática y el algoritmo es de tipo general. Por otro lado, cuando trabajamos con descomposición de dominio, al inicio del algoritmo de Schwarz desconocemos el valor de la función en los nodos que conforman la frontera artificial Γ , así que para poder emplear el error residual $\|Lu - f\|_2$ para seleccionar el parámetro c es necesario minimizar dicho residual por lo menos dos veces en cada iteración del algoritmo de descomposición de dominio. La estrategia propuesta en Cheng [75] no nos resultó útil, debido al alto costo computacional y por el hecho de emplear descomposición de dominio con nodos adaptivos no es claro como obtener una muestra representativa de los nodos para cada subdominio.

De los tres casos analizados anteriormente, resulta evidente que conforme incrementamos el número de Péclet ($Pe = v/\beta$), el error cometido se incrementa drásticamente. En la tabla 5.5 se muestra el comportamiento del error cuadrático medio y el error máximo conforme incrementamos el término convectivo.

Un comportamiento similar a lo mostrado en la tabla 5.5 se observó para la ecuación de convección-difusión analizada, aumentándole la parte temporal y empleando el esquema de Crank-Nicholson con $\theta = 0.5$ [30].

Péclet 10^3

Finalmente, consideraremos el caso cuando el término convectivo $v = 10^3$ es muy superior al término difusivo $\beta = 1$ siendo el número de Péclet $Pe = 1000$. En este caso la solución analítica presenta un alto gradiente cerca de los ejes

$x = 0$ y $y = 0$, fuera de dicha zona se tiene una región con un valor constante cero. Con la distribución inicial de nodos $N = 1156$, no fue posible capturar correctamente la zona de alto gradiente, de tal modo que se obtuviera una buena aproximación numérica (similar a lo obtenido anteriormente). Es por ello que emplearemos refinamiento local de nodos con descomposición de dominio.

Como número de elementos máximo por subdominio se considera $|\Omega_i| \leq 2000$, con un porcentaje de refinamiento $\theta_{ref} = 0.07$ y como parámetro c inicial se tiene $c_{ini} = 0.01$.

El error numérico que deseamos obtener es $E_{max} \leq 0.05$, similar a lo obtenido con $Pe = 100$. Adicionalmente agregamos una restricción de 1 Gbyte a la cantidad de memoria utilizada. Esta restricción en la práctica es muy importante ya que si se excede la capacidad de memoria del sistema, el tiempo de procesamiento se dispara debido a la paginación en la memoria virtual [159].

En la tabla 5.6 se muestra el resultado obtenido por el algoritmo 3 para el problema de tipo convección-difusión analizado (5.13) y (5.14). En la primera columna se despliega el número de la iteración del algoritmo 3, en la segunda columna se indica el número de subdominios, en la tercera columna se muestra el número de nodos obtenidos del proceso de refinamiento local de nodos, en la cuarta y quinta columna se despliega el error cuadrático medio y el error en la norma infinita.

Como se observa en la tercera columna en la tabla 5.6, conforme se itera el algoritmo 3 se incrementa paulatinamente el número de nodos y por lo tanto se requiere del esquema dinámico de particiones. La selección dinámica de las particiones se muestra en la segunda columna. En las últimas dos columnas claramente se aprecia que conforme se itera el algoritmo 3 ambas medidas de error disminuyen; logrando así el objetivo de disminuir el error de aproximación mediante el esquema de refinamiento local de nodos con descomposición de dominio. En la figura 5.6 se muestra la solución aproximada $\tilde{u}(x, y)$ en la iteración 12, se observa que la zona de alta variación espacial fue capturada

Iteración	Subdominio	Nodos	ECM	E_{max}
0	1	1156	8.251532	136.243103
1	1	1363	1.236127	5.778277
2	1	1667	0.828661	4.830351
3	1	1991	0.652659	4.691801
4	2	2293	0.262702	2.196181
5	2	2817	0.161479	1.283982
6	3	4026	0.072166	0.917157
7	4	5310	0.044232	0.325591
8	6	6773	0.022449	0.299896
9	7	8803	0.021132	0.083681
10	9	13304	0.020352	0.068958
11	16	21127	0.020087	0.063680
12	16	22993	0.017400	0.051158

Tabla 5.6: Refinamiento local de nodos con descomposición de dominio para el problema de Convección-Difusión, con el núcleo multicuadrado.

correctamente.

El error de aproximación deseado lo habíamos establecido como $E_{max} \leq 0.05$, como se aprecia en la tabla 5.6 (ver último renglón, quinta columna), no fue posible obtener dicho error de aproximación, esto se debe a que después de la iteración 12 se llegó al límite de memoria en RAM permitida. Sin embargo, por el comportamiento decreciente del error de aproximación se infiere que con los suficientes recursos computacionales sería posible disminuir aún más el error de aproximación.

La partición dinámica de los datos en la iteración 12 se muestra en la parte superior de la figura 5.7, en la parte inferior de dicha figura se muestra la distribución final de las celdas. Como se observa en la partición de los datos, cerca del punto de origen $(0, 0)$ se tiene una disminución en el tamaño de los

rectángulos que conforman la partición, esto se debe a la alta densidad de nodos requerida para capturar el pico en el punto $(0, 0)$.

Por simple inspección se observan 12 particiones en distintos colores, las 4 particiones restantes se encuentran divididas en los intervalos $y = 0, x > 0.25$ y $x = 0, y > 0.25$. Es decir, cercanas a los ejes $y = 0$ y $x = 0$, esto es consistente con el hecho de que la solución analítica presenta una alta variación cercana a los ejes $y = 0$ y $x = 0$.

En la parte inferior de la figura 5.7, se aprecia que cerca de las zonas con cambios abruptos en la solución se tiene una concentración de las celdas y conforme nos alejamos de los ejes $x = 0$ y $y = 0$, se disminuye gradualmente la densidad de las celdas. Esta concentración local de las celdas/nodos nos permite capturar las regiones de alta variación espacial, concentrando nodos en las zonas de alto gradiente y manteniendo una densidad menor en las zonas de bajo gradiente.

La combinación de refinamiento local de nodos y descomposición de dominio conforman un algoritmo computacionalmente eficiente; donde la eficiencia estriba en el uso de descomposición de dominio para disminuir el tiempo de procesamiento y el refinamiento local de nodos es eficiente en cuanto al número de nodos requeridos para aproximar numéricamente la ecuación diferencial parcial.

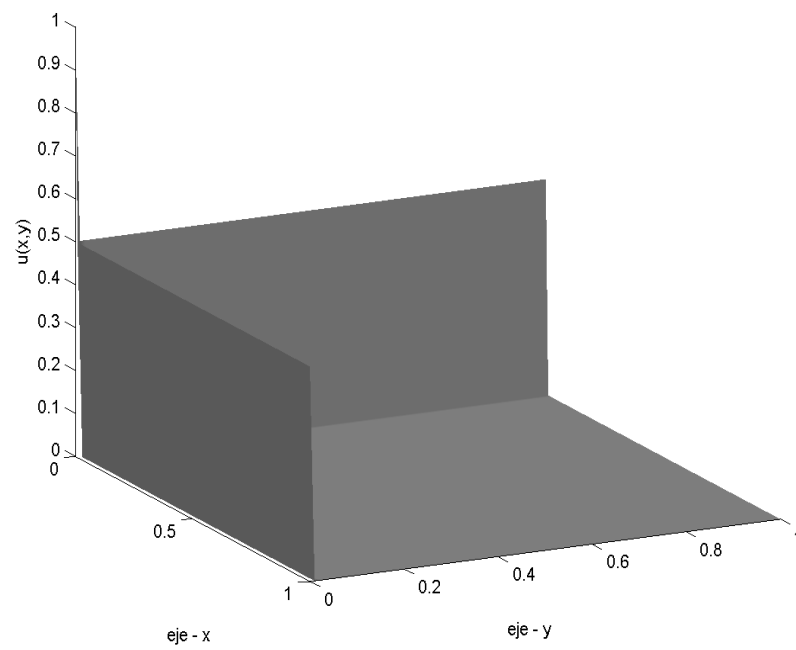


Figura 5.6: Solución numérica del problema de Convección-Difusión con $Pe = 10^3$. Función radial multicuádrada.

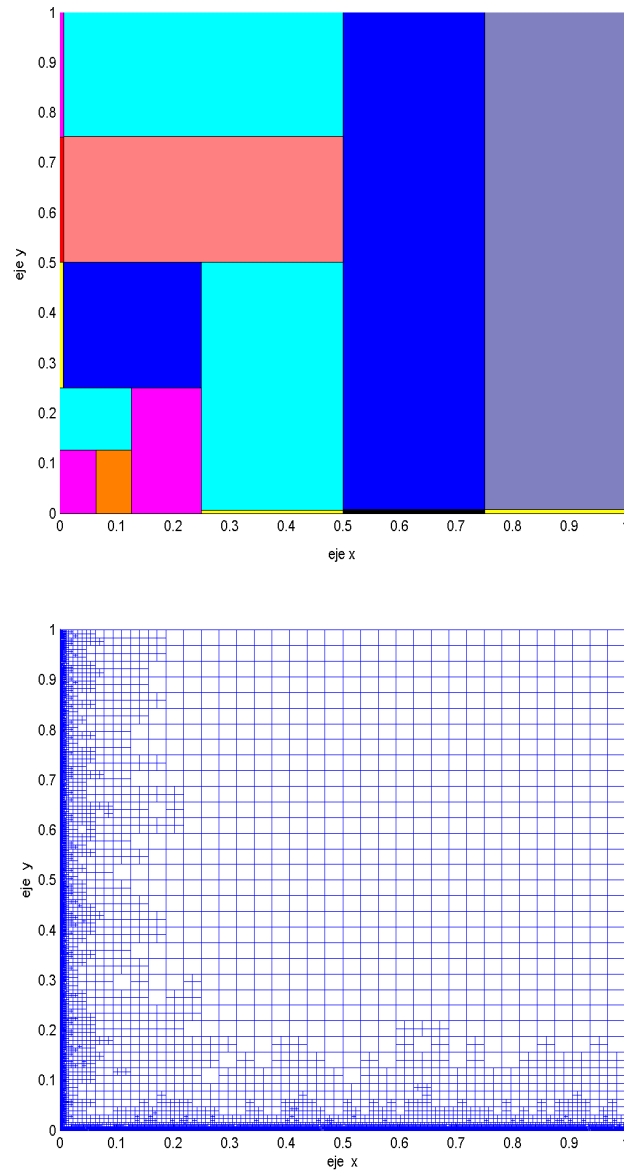


Figura 5.7: Arriba: Partición de datos en la iteración 12 del esquema RLNDD aplicado al problema de Convección-Difusión con $Pe = 10^3$. Abajo: Distribución de las celdas.

Capítulo 6

Conclusiones y Aportaciones

La presente disertación doctoral se enfocó en el desarrollo y aplicación del método de colocación asimétrica con funciones de base radial para la determinación de la solución numérica de ecuaciones en derivadas parciales lineales en una y dos dimensiones. Las principales aportaciones de este proyecto de investigación doctoral las podemos englobar en dos puntos:

- A partir de un problema temporal convectivo-difusivo en una dimensión, mostramos numéricamente que para valores bajos a moderados del coeficiente de Péclet se obtiene una tasa de convergencia exponencial (h -refinamiento). Con base en el núcleo multicuadrado y empleando un esquema de discretización implícito y explícito.
- Formulamos un nuevo algoritmo para la solución numérica de ecuaciones en derivadas parciales estacionarias en dos dimensiones con altos gradientes, el cual combina el refinamiento local de nodos con el método de descomposición de dominio. Este algoritmo constituye la primera formulación en el estado del arte con la cual es posible abordar problemas de gran escala empleando el método de colocación asimétrica.

La combinación de refinamiento local de nodos y descomposición de dominio conforman un algoritmo computacionalmente eficiente; donde la eficiencia

estriba en el uso de descomposición de dominio para disminuir el costo computacional y el refinamiento local de nodos es eficiente en cuanto al número de nodos requeridos para capturar las regiones con alta variación espacial. Con ambos métodos, se reduce el tiempo y la memoria requerida en la solución numérica de ecuaciones en derivadas parciales, lo cual nos permitirá abordar problemas de gran escala. Entendiendo por problemas de gran escala cuando el número de incógnitas es mayor a 10^6 .

Las aportaciones derivadas de los dos puntos mostrados anteriormente son las siguientes:

1.- Adicionalmente al orden de convergencia exponencial, determinamos numéricamente:

- La tasa de convergencia exponencial es también alcanzable con un esquema de c -refinamiento.
- El incremento en el número de Péclet produce una reducción en el coeficiente que determina el orden de convergencia exponencial.
- El valor óptimo del parámetro c decrece monotónicamente conforme el coeficiente difusivo es reducido.

2.- En problemas lineales temporales con descomposición de dominio se determinó:

- Un orden de complejidad lineal con respecto al número de subdominios. Para un problema convectivo en dos dimensiones observamos una ganancia en el tiempo de la simulación de dos órdenes de magnitud cuando se utiliza el método de descomposición de dominio.
- Al incrementar el número de subdominios la exactitud disminuye ligeramente.
- Difusión numérica baja.

3.- La extensión del esquema de refinamiento local de nodos basado en la triangulación de Delaunay, a un esquema basado en celda \times celda, con lo cual se obtuvo:

- Un esquema de refinamiento local de nodos (h -refinamiento) para problemas estacionarios en una y dos dimensiones. Con este esquema se logra refinar localmente en las regiones donde se requiere de una mayor exactitud.
- Para un problema en una dimensión determinamos numéricamente que el refinamiento local de nodos tiene una eficiencia del 90% en el número de nodos para alcanzar el mismo error en comparación con el refinamiento global de nodos, logrando así un ahorro en tiempo y espacio para capturar las regiones con cambios abruptos espaciales.
- La formulación de un esquema de remoción e inserción de nodos para problemas lineales temporales en una dimensión. En particular para la ecuación de tipo convección-difusión con un número de Péclet $Pe = 10^3$ fue posible capturar la zona de alta variación espacial.

4.- En la formulación del método de descomposición de dominio con el refinamiento local de nodos determinamos:

- Una partición dinámica de los datos con base en el método de bisección recursiva por coordenadas. El número de subdominios se adapta dinámicamente a la densidad de los nodos, logrando así abordar problemas en dos dimensiones con miles de nodos.
- Un algoritmo de orden logarítmico en tiempo para la creación de la zona de traslape requerida en el método de Schwarz con coincidencia de nodos.

En el último periodo del doctorado, extendimos el punto tres y cuatro hacia un esquema de representación de tipo árbol (quad-tree) para problemas estacionarios en dos dimensiones. Con ello podemos realizar la remoción e inserción de

nodos logrando así una mejora al algoritmo propuesto. Adicionalmente, la creación de la zona de traslape entre subdominios depende del nivel de refinamiento, a mayor nivel de refinamiento mayor es la zona de traslape; la formulación de los puntos anteriores fue desarrollada por el Dr. Pedro González-Casanova. Los resultados obtenidos se presentaron en el congreso *Multivariate Approximation: Theory and Applications*, así mismo se someterá el trabajo realizado en la revista *Numerical Methods for Partial Differential Equations*, ver siguiente sección.

6.1. Publicaciones

Las publicaciones derivadas de la investigación desarrollada son las siguientes:

- Publicadas
 - J. Antonio Muñoz-Gómez, P. González-Casanova, G. Rodríguez-Gómez, *Adaptive Node Refinement Collocation Method for Partial Differential Equations*. ENC-06 Sixth Mexican International Conference on Computer Sciences published by IEEE, pp. 70-80, September 2006.
 - J. Antonio Muñoz-Gómez, P. González-Casanova, G. Rodríguez-Gómez, *Domain Decomposition by Radial Basis Functions for Time Dependent Partial Differential Equations*. In the Proceedings of the IASTED: International Conference on Advances in Computer Science and Technology, pp. 105-109, January 2006.
- Sometida a la Revista de Ingeniería, Investigación y Tecnología en Noviembre del 2006.
 - J. Antonio Muñoz-Gómez, P. González-Casanova, G. Rodríguez-

Gómez, *Exponential Convergence of Multiquadric Collocation Method: a Numerical Study*.

- Por someter al Journal of Numerical Methods for Partial Differential Equations en Octubre del 2007.
- P. González-Casanova, J. Antonio Muñoz-Gómez, G. Rodríguez-Gómez, *Node Adaptive Domain Decomposition Method by Radial Basis Functions*.

6.2. Trabajo Futuro

Claramente se requiere de más investigación en esta área, tanto teórica como de nuevos algoritmos numéricos. Es de gran interés aplicar los resultados obtenidos a problemas definidos en tres dimensiones, por lo tanto, mostramos en forma de listado las metas a futuro:

- Investigar la difusión/dispersión numérica para problemas en una dimensión.
- Particiones no cartesianas que obedezcan al comportamiento de la ecuación diferencial parcial.
- Remoción e inserción de nodos para problemas dependientes del tiempo en dos y tres dimensiones.
- Extender los algoritmos propuestos a un esquema en paralelo.
- Patrones de diseño para el diseño y codificación de los puntos anteriores.

Adicionalmente, es de gran interés el desarrollo de una biblioteca numérica para la solución de ecuaciones en derivadas parciales con base en funciones de base radial.

Apéndice A

Número de Condición

En esta sección se define el significado de mal condicionamiento. Considere el problema de determinar la solución del siguiente sistema lineal de ecuaciones

$$Ax = b, \tag{A.1}$$

donde b es un vector en \mathbb{R}^n , A es una matriz real de dimensiones $n \times n$ y x es el vector de incógnitas en \mathbb{R}^n . Nuestro objetivo es determinar numéricamente el vector de incógnitas x , sin embargo primero explicaremos brevemente el punto de vista teórico.

Desde un punto de vista teórico existe solución única si y sólo si A es no singular, donde la singularidad se puede comprobar como: $\det(A) \neq 0$. Es decir, los vectores que conforman a la matriz A son linealmente independientes. En dicho caso el vector solución se determina como $x = A^{-1}b$, siendo A^{-1} la inversa de A . Por lo expuesto anteriormente se deduce que explícitamente es necesario calcular la inversa de A , esto en la práctica nunca se realiza debido al alto costo computacional y a los errores acumulativos de redondeo. Se selecciono un esquema de factorización LU basado en el método de Gauss con pivoteo parcial, el cual resulto conveniente para nuestros fines. Cabe aclarar que dicha rutina no fue codificada por nosotros, se utilizó la rutina disponible en el sitio web de Lawrence Shampine. Esto se debe a que codificar la factorización LU está lejos de ser un problema trivial, debido a que se deben considerar errores

de redondeo, errores de cancelación, precisión de la máquina, así como análisis teóricos en las perturbaciones de las matrices L, U [160].

Por cuestiones de claridad se dan algunas deficiones que son requeridas posteriormente.

La norma $\|\cdot\|$ de un vector en \mathbb{R}^n es una función $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ que satisface las siguientes propiedades para cualesquiera vectores $x, y \in \mathbb{R}^n$

- $\|x\| \geq 0$, $\|x\| = 0$ si y solo si $x = 0$
- $\|\alpha x\| = |\alpha| \cdot \|x\|$, con $\alpha \in \mathbb{R}$
- $\|x + y\| \leq \|x\| + \|y\|$

Las normas vectoriales más frecuentemente utilizadas son:

- $\|x\|_1 = |x_1| + \dots + |x_n|$
- $\|x\|_2 = (|x_1|^2 + \dots + |x_n|^2)^{\frac{1}{2}}$
- $\|x\|_\infty = \max |x_i|$, para $1 \leq i \leq n$

La primera condición nos dice que el tamaño de un vector no nulo es positivo, cuando no se satisface dicha condición se dice que la norma es semi-positiva.

El aspecto numérico que abordaremos será el problema de la estabilidad de los sistemas lineales de ecuaciones. Es decir, nos interesa saber cuanto influye en el resultado una pequeña perturbación en los datos. Para ello consideraremos la cuantificación del error absoluto y el error relativo de dichas perturbaciones.

Consideremos ahora la introducción de perturbaciones en x y b , siendo las perturbaciones $x + \delta x$ y $b + \delta b$. Introduciendo dichas perturbaciones en (A.1) se obtiene la siguiente igualdad

$$A(x + \delta x) = b + \delta b, \quad (\text{A.2})$$

expandiendo los términos y con base en una norma vectorial y su correspondiente norma inducida matricial; ver pp. 55 [160], se obtiene

$$\|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\|. \quad (\text{A.3})$$

Esta desigualdad acota el cambio absoluto de la solución en función del cambio absoluto del vector conocido b . Un inconveniente de utilizar el error absoluto es que no considera el orden en la magnitud del valor que se estima. Es por ello que se introduce el error relativo, el cual lo podemos expresar como un porcentaje del error cometido independientemente de las magnitudes que se estén midiendo.

Ahora, lo que nos interesa es comparar las cantidades del error relativo entre las perturbaciones de los vectores $x, b \in \mathbb{R}^d$, donde el error relativo se expresa

$$\frac{\|x - \delta x\|}{\|x\|}, \frac{\|b - \delta b\|}{\|b\|}. \quad (\text{A.4})$$

El error relativo en la norma infinita nos da un indicador acerca del número de las p cifras correctas significativas

$$\frac{\|x - \delta x\|_{\infty}}{\|x\|_{\infty}} \approx 10^{-p}, \quad (\text{A.5})$$

ver el libro de Golub et al. [160].

Perturbando el sistema lineal (A.1) se obtiene $(x - \delta x) = A^{-1}(b - \delta b)$, tomando la norma vectorial en el lado izquierdo y su correspondiente norma matricial subordinada en el lado derecho se tiene

$$\|x - \delta x\| \leq \|A^{-1}\| \cdot \|b - \delta b\|. \quad (\text{A.6})$$

Por otra parte, de (A.1) se obtiene $b = Ax$ lo cual implica $\|b\| = \|Ax\|$, y por lo expuesto anteriormente se obtiene $1/\|x\| \leq \frac{\|A\|}{\|b\|}$, lo que combinado con la ecuación anterior obtenemos el resultado buscado

$$\frac{\|x - \delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|b - \delta b\|}{\|b\|} \quad (\text{A.7})$$

El número $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ se llama número de condición de A , con el cual se tiene una cota superior. A menor número de condición decimos que el sistema está bien condicionado, a mayor número de condición el sistema

está mal condicionado y pequeños cambios relativos nos inducen un degradamiento en la solución. Por ejemplo, para $\kappa(A) > 10^{12}$ decimos que está mal condicionada la matriz A .

El número de condición tiene el siguiente significado práctico: cuando resolvemos el sistema lineal de ecuaciones $Ax = b$ el orden del error relativo será $\epsilon \cdot \kappa(A)$, donde ϵ corresponde al épsilon de la máquina. El épsilon de la máquina con redondeo es: $\frac{1}{2}\beta^{1-p}$, donde β es la base del sistema y p corresponde al número de bits en la mantisa en la representación de punto flotante del sistema [62]. En particular, trabajamos con números en precisión doble ($p = 52$) con una base binaria ($\beta = 2$), por lo que la unidad de redondeo corresponde a $\epsilon \approx 2.2 \times 10^{-16}$.

En la determinación del número de condición, con base en la norma $\|\cdot\|_\infty$, no se calcula la inversa de la matriz A . En términos generales se obtiene una cota inferior de $\|A^{-1}\|$ con base en la factorización LU , ver el libro de Golub [160].

Apéndice B

Extensión del Algoritmo 3

El algoritmo propuesto para aproximar la solución de ecuaciones en derivadas parciales empleando refinamiento local de nodos con descomposición de dominio; ver algoritmo 3, fue diseñado considerando que los algoritmos serían ejecutados en una sola computadora. Este diseño nos brinda la ventaja de poder concentrarnos en la eficiencia de los algoritmos y su posterior implementación, y no en la programación en paralelo de dichos algoritmos. Por otro lado, cuando en un futuro se requiera migrar a aplicaciones con cientos de miles o millones de nodos, claramente el procesamiento en una sola computadora resultará insuficiente, tanto por la capacidad de procesamiento como por el almacenamiento en memoria.

En general, los pasos que conforman el algoritmo propuesto son: partición de datos, solución numérica de la EDP con descomposición de dominio y refinamiento local de nodos. Con base en lo desarrollado en la presente tesis, el segundo paso puede ser extendido sin muchas modificaciones a un cluster de computadoras, esto se debe a que el algoritmo de descomposición de dominio utilizado requiere de módulos iguales en cada subdominio y la transferencia de datos entre subdominios requiere el uso correcto del envío y recepción de datos en paralelo, para lo cual sólo se requiere de una biblioteca que soporte dichas instrucciones. Las dos bibliotecas más utilizadas son: la interfaz de co-

municación en paralelo MPI [109] y la máquina virtual paralela PVM [161], una comparación de dichas librerías se describe en [162].

El primer paso del algoritmo 3 correspondiente a la partición de datos se ha abordado principalmente en el área de mallas sin estructura, por ejemplo triángulos o tetraedros. En la década pasada se generaron distintas aplicaciones de software [163, 164, 165, 166]. Con respecto a la adaptación de mallas en paralelo (Parallel Adaptive Mesh Refinement) de igual manera se concentró la atención en mallas sin estructura, en donde el manejo de estructura de datos y minimización de la latencia son incluidas como restricciones en el proceso de refinamiento en paralelo [148, 167, 168, 169, 170].

Existe un vínculo entre la partición de datos y el refinamiento de la malla, ambos en paralelo, este vínculo lo podemos establecer de la siguiente forma: asuma que para un tiempo inicial t_0 realizamos la partición de datos obteniendo un balanceo en la carga de trabajo y posteriormente dicha carga es asignada a cada procesador. Para un tiempo posterior t_1 , el proceso de refinamiento en paralelo modifica la carga de trabajo asignada a cada procesador obteniendo con ello un desbalanceo. El problema ahora es volver a distribuir en paralelo la nueva malla de tal forma que quede balanceado el sistema. Esto se conoce en la literatura como Balanceo Dinámico de Carga (Dynamic Load Balancing). Para ello se requiere de migración de información en paralelo entre los subdominios vecinos y constituye un enorme reto computacional, debido a que se requiere minimizar el número de pasos para volver a balancear la carga, y además es posible que cada procesador tenga un desempeño distinto y por lo tanto no basta con distribuir la carga de trabajo de forma equitativa; consultar [171, 172, 173, 174, 175].

Apéndice C

Acrónimos

EDP	Ecuación Diferencial Parcial
MDD	Método de Descomposición de Dominio
MCA	Método de Colocación Asimétrico
MCS	Método de Colocación Simétrico
EDO	Ecuaciones Diferenciales Ordinarias
PCS	Proyección-Corrección-Solución
IMQ	Esquema de Discretización Implícito con Multicuadrado
EMQ	Esquema de Discretización Explícito con Multicuadrado
ECM	Error Cuadrático Medio
DD	Descomposición de Dominio
RLN	Refinamiento Local de Nodos
RGN	Refinamiento Global de Nodos
BRC	Bisección Recursiva basada en Coordenadas
NFA	Nodo de tipo Frontera Artificial
RLNDD	Refinamiento Local de Nodos con Descomposición de Dominio

Apéndice D

Símbolos

u	Solución analítica
\tilde{u}	Solución numérica
$\ \cdot\ $	Norma Euclidea
\mathbb{R}	El conjunto de los números reales
\mathbb{C}	El conjunto de los números complejos
\mathbb{R}^d	El espacio vectorial de los d -vectores reales: $\mathbf{x} \in \mathbb{R}^d \Leftrightarrow \mathbf{x} = [x_1, \dots, x_d]^T$ con $x_i \in \mathbb{R}$
$\mathbb{R}^{N \times N}$	El espacio vectorial de las matrices reales $N \times N$: $A \in \mathbb{R}^{N \times N} \Leftrightarrow$ $A = (a_{ij}) = \begin{bmatrix} a_{11} & \dots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \dots & a_{NN} \end{bmatrix} \text{ con } a_{ij} \in \mathbb{R}$
A^T	Transpuesta de la matriz A
ϕ	Función de base radial
Φ	Matriz de Gramm
λ	Vector de incógnitas en la interpolación con funciones radiales
p	Polinomio requerido en la interpolación con funciones condicionalmente positivas definidas
Ω	Dominio en donde está definida la ecuación diferencial parcial
$\partial\Omega$	Frontera de Ω
N	Número de nodos de colocación $N = N_I + N_f$

N_I	Número de nodos en el dominio
N_f	Número de nodos en la frontera
\mathcal{L}	Operador diferencial parcial lineal definido en Ω
\mathcal{B}	Operador diferencial parcial lineal definido en $\partial\Omega$
∇u	Gradiente $\nabla u = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}$
∇^2	Laplaciano $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$
c	Parámetro en la función radial multicuádrada
c_{ini}	Parámetro inicial en la función radial multicuádrada requerido en el esquema de refinamiento local de nodos
$ \Omega $	Cardinalidad del dominio Ω
Ω_i	Subdominio o i -ésima partición en Ω
$ \Omega_i $	Cardinalidad del subdominio Ω_i
P	Número de subdominios Ω_i , $i = 1, \dots, P$
Γ_i	Frontera artificial del subdominio Ω_i
M_{max}	Máximo número de nodos en cada subdominio sin la región de traslape
θ_{ref}	Valor de umbral en el refinamiento local de nodos
t_{max}	Tiempo máximo requerido en los problemas dependientes del tiempo
Δt	Tamaño de paso temporal $0 < \Delta t < 1$
τ	Número de pasos para avanzar de $t = 0$ hasta t_{max}
β	Coefficiente difusivo en la ecuación de convección-difusión
μ	Coefficiente convectivo en la ecuación de convección-difusión
Pe	Coefficiente de Péclet = convectivo / difusivo
ECM	Error cuadrático medio: $ECM = \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i - \tilde{u}_i)^2}$
E_{max}	Error máximo $E_{max} = \ u - \tilde{u}\ _\infty$
$\ x\ _\infty$	Norma infinita, definida como: $\ \cdot\ _\infty = \max x_i $, $1 \leq i \leq n$

Índice de Figuras

2.1. Distribución de los nodos en Ω y sobre la frontera $\partial\Omega$	31
2.2. Convergencia exponencial para 5 distintos casos de prueba con el método implícito, al tiempo $t_{max} = 1$	50
2.3. Solución analítica y numérica para distintos valores de β y μ , al tiempo $t_{max} = 1$	51
2.4. Convergencia exponencial variando c y manteniendo fijo $N = 30$ con el método implícito, con $t_{max} = 1$	52
2.5. Convergencia exponencial variando c y manteniendo fijo $N = 200$ con el método implícito, con $t_{max} = 1$	53
2.6. Convergencia para $\beta = 0.2$ y $\mu = 1$ con distintos pasos temporales Δt , $t_{max} = 1$ con el método implícito.	54
2.7. Variación del parámetro c con respecto del coeficiente β , fijo $\mu = 0.5$	55
2.8. Convergencia exponencial para 5 distintos casos con el método explícito, al tiempo $t = 1$	56
2.9. Variación del parámetro c , con $\beta = 0.1$, $\mu = 0.2$, al tiempo $t_{max} = 1$	58
3.1. Dominio Ω particionado en dos subdominios con traslape.	64
3.2. Zona de traslape de una partición 2×2 para datos equiespaciados.	78
3.3. Categorías de nodos presentes en cada subdominio.	79
3.4. Condición inicial $u(\mathbf{x}, t = 0) = e^{-\frac{r^2}{\sigma}}$	80

3.5. Decremento en la exactitud vs. incremento en las particiones P^2 .	83
3.6. Log-log del decremento lineal del tiempo vs. incremento en las particiones P^2 .	84
3.7. Solución numérica de la EDP: $\frac{\partial u(\mathbf{x},t)}{\partial t} + v \cdot \nabla u(\mathbf{x},t) = 0$, mostrada en distintos tiempos.	86
3.8. Comparación entre diferencias finitas y el método de colocación asimétrico, al tiempo $t_{max} = 0.4$.	87
4.1. Ilustración de adaptación de celdas en dos dimensiones, se observa que las celdas vecinas son refinadas por la relación 2:1.	98
4.2. Ilustración de refinamiento para una celda en dos dimensiones.	99
4.3. Reconstrucción de $u(x)$ con el esquema de refinamiento de nodos.	108
4.4. Histograma de la distribución de nodos.	109
4.5. Comparación entre h -refinamiento local y h -refinamiento global.	110
4.6. Solución analítica del problema de Poisson.	111
4.7. Adaptación de celdas para el problema en $2-d$ de Poisson en las iteraciones= $\{0,4,8,12\}$.	113
4.8. Solución numérica de la EDP convección-difusión con refinamiento local de nodos, $t_{max} = 1$.	116
5.1. Solución analítica $u(x, y) = \tanh(x + 50y)$.	132
5.2. Refinamiento local de nodos con descomposición de dominio para el problema de Poisson, se muestran las iteraciones = $\{0, 6, 13, 20\}$. Función radial multicuádrada.	139
5.3. Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^0$.	141
5.4. Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^1$.	142

-
- 5.5. Solución numérica y analítica del problema de Convección-Difusión con $Pe = 10^2$ 142
- 5.6. Solución numérica del problema de Convección-Difusión con $Pe = 10^3$. Función radial multicuádrica. 147
- 5.7. Arriba: Partición de datos en la iteración 12 del esquema RLNDD aplicado al problema de Convección-Difusión con $Pe = 10^3$.
Abajo: Distribución de las celdas. 148

Índice de Tablas

1.1. Funciones de base radial frecuentemente utilizadas y grado del polinomio.	11
1.2. L_∞ Cotas de la función $P_{\phi,X}$ basados en el interpolante en la forma de Lagrange.	22
2.1. Variación de c para distintos Δt , se mantiene fijo $\beta = 0.2$, $\mu = 1$ con $t_{max} = 1$	54
2.2. Comparación entre IMQ y EMQ para alcanzar la misma precisión.	59
2.3. Variación del parámetro c para $N=40$ con IMQ y EMQ.	60
3.1. Error cometido sin descomposición de domino.	81
3.2. Comparación entre el número de particiones vs. tiempo y error.	82
4.1. Dimensiones de las nuevas cuatro celdas.	99
4.2. Decrecimiento del error vs. iteraciones.	107
4.3. Comparación entre h -refinamiento global y h -refinamiento local.	109
4.4. Decrecimiento del error vs. iteraciones para el problema de Poisson.	112
4.5. Comparación entre el método de nodos adaptivos y no-adaptivos para el problema de Poisson.	114
5.1. Variación del núcleo radial y el esquema de partición de datos. .	133

5.2. Variación del núcleo radial con refinamiento local de nodos, descomposición de dominio y particiones dinámicas.	135
5.3. Variación del parámetro de refinamiento θ_{ref} , con el núcleo multicuadrado.	136
5.4. Iteraciones en la adaptividad de los nodos con descomposición de dominio para el problema de Poisson, con el núcleo multicuadrado.	138
5.5. Error obtenido conforme incrementamos el número de Péclet. . .	143
5.6. Refinamiento local de nodos con descomposición de dominio para el problema de Convección-Difusión, con el núcleo multicuadrado.	145

Bibliografía

- [1] David E. Keyes. Four horizons for enhancing the performance of parallel simulations based on partial differential equations. In Arndt Bode, Thomas Ludwig 0002, Wolfgang Karl, and Roland Wismüller, editors, *Euro-Par*, volume 1900 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2000.
- [2] Michael Griebel, Thomas Dornseifer, and Tilman Beunhoeffler. *Numerical Simulation in Fluid Dynamics*. Society for Industrial and Applied Mathematics, 1997.
- [3] Michael Griebel and Marc Alexander Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic pdes. *SIAM Journal Scientific Computing*, 22(3):853–890, 2000.
- [4] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC Press, 2003.
- [5] J. Frank Bossen and Paul S. Heckbert. A pliant method for anisotropic mesh generation. In *Proceedings of the 5th International Meshing Roundtable*, pages 63–76, October 1996.
- [6] David R. White, Sunil Saigal, and Steven J. Owen. Meshing complexity of single part cad models. In *Proceedings of the 12th International Meshing Roundtable*, pages 121–134, September 2004.

- [7] Steven J. Owen. *Non-Simplicial Unstructured Mesh Generation*. PhD thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, 1999.
- [8] Steve Owen. A survey of unstructured mesh generation technology. In *Proceedings of the 7th International Meshing Roundtable*, pages 239–267, September 1998.
- [9] C. A. Duarte. A review of some meshless methods to solve partial differential equations. Technical report, TICAM University of Texas, 1995.
- [10] T. Belyschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.*, 139:3–47, 1996.
- [11] X. Zhang, K. Z. Song, M. W. Lu, and X. Liu. Meshless methods based on collocation with radial basis functions. *Computational Mechanics*, 26:333–343, 2000.
- [12] T. P. Fries and H. G. Matthies. Classification and overview of meshfree methods. Technical report, Technical University Braunschweig, Brunswick, Germany, Revised July 2004.
- [13] G. R. Liu and Y. T. Gu. *An Introduction to Meshfree Methods and their Programming*. Springer Netherlands, 2005.
- [14] A. I. Tolstykh and D. A. Shirobokov. On using radial basis functions in a finite difference mode with applications to elasticity problems. *Computational Mechanics*, 33(1):68–79, 2003.
- [15] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, 19(8-9):127–145, 1990.

- [16] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, 19(8-9):147–161, 1990.
- [17] G. E. Fasshauer. *Boundary Elements XXVII: Incorporating Electrical Engineering and Electromagnetics*, chapter RBF Collocation Methods and Pseudospectral Methods. WIT Press, Philadelphia, PA, USA, 2005.
- [18] M.A. Golberg and C.S. Chen. *Boundary Integral Methods - Numerical and Mathematical Aspects*, chapter The method of fundamental solutions for potential, Helmholtz and diffusion problems, pages 103–176. WIT Press, 1998.
- [19] Holger Wendland. Meshless Galerkin methods using radial basis functions. *Mathematics of Computation*, 68(228):1521–1531, 1999.
- [20] T. Yamada and L. C. Wrobel. Properties of gaussian radial basis functions in the dual reciprocity boundary element method. *Zeitschrift für Angewandte Mathematik und Physik*, 44(6):1054–1067, 1993.
- [21] M. A. Golberg and C. S. Chen. The theory of radial basis functions applied to the bem for inhomogeneous partial differential equations. *Boundary Elements Communications*, 5:57–61, 1994.
- [22] C. Shu, H. Ding, and K. S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(7-8):941–954, 2003.
- [23] W. Chen and M. Tanaka. New insights in boundary-only and domain-type rbf methods. *Int. J. Nonlinear Sci. Numer. Simul.*, 1:145–151, 2000.

- [24] W. Chen and M. Tanaka. A meshless, integration-free, and boundary-only rbf technique. *Computers and Mathematics with Applications*, 43:379–391, 2002.
- [25] J. G. Wang and G. R. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54:1623–1648, 2002.
- [26] M. Zerroukat, H. Power, and C. S. Chen. A numerical method for heat transfer problems using collocation and radial basis function. *International Journal for Numerical Methods in Engineering*, 4(7):1263–1278, 1998.
- [27] M. Zerroukat, K. Djidjeli, and A. Charafi. Explicit and implicit meshless methods for linear advection-diffusion-type partial differential equations. *International Journal of Numerical Methods in Engineering*, 48(1):19–35, 2000.
- [28] I. Boztosun and A. Charafi. An analysis of the linear advection-diffusion equation using mesh-free and mesh-dependent methods. *Engineering Analysis with Boundary Elements*, 26(10):889–895, 2002.
- [29] J. Li and C.S. Chen. Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems. *International Journal of Numerical Methods in Engineering*, 57(8):1085–1094, 2003.
- [30] P.P. Chinchapatnam, K. Djidjeli, and P.B. Nair. Unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation. *Computer Methods in Applied Mechanics and Engineering*, 195(19–22):2432–2453, 2006.

- [31] Yiu-Chung Hon and X. Z. Mao. A multiquadric interpolation method for solving initial value problems. *Journal of Scientific Computing*, 12(1):51–55, 1997.
- [32] Yiu-Chung Hon and X. Z. Mao. An efficient numerical scheme for burgers' equation. *Applied Mathematics and Computation*, 95(1):37–50, 1998.
- [33] M. Sharan, E. J. Kansa, and S. Gupta. Applications of the multiquadric method for numerical solutions of elliptic partial differential equations. *Applied Mathematics and Computation*, 84(2–3):275–302, 1997.
- [34] Y. C. Hon. A quasi-radial basis functions method for american options pricing. *Computers and Mathematics with Applications*, 43(3–5):513–524, 2002.
- [35] Yiu-Chung Hon and X. Z. Mao. A radial basis function method for solving options pricing models. *Financial Engineering*, 8(1):31–49, 1999.
- [36] Jichun Li, C. S. Chen, Darrel Pepper, and Yitung Chen. Mesh-free method for groundwater modeling. *Boundary Elements XXIV*, pages 115–124, 2002.
- [37] Yiu-Chung Hon, Kwok Fai Cheung, Xian-Zhong Mao, and Edward J. Kansa. Multiquadric solution for shallow water equations. *Journal of Hydraulic Engineering*, 125(5):524–533, 1999.
- [38] S. M. Wong, Y. C. Hon, and M. A. Golberg. Compactly supported radial basis functions for shallow water equations. *Applied Mathematics and Computation*, 127(1):79–101, 2002.
- [39] F. J. Hickernell and Y. C. Hon. Radial basis function approximation of the surface wind field from scattered data. *Intert. J. Appl. Sci. Comput.*, 4:221–247, 1998.

- [40] Yiu-Chung Hon, M. W. Lu, W. M. Xue, and Y.M. Zhu. Multiquadric method for the numerical solution of a biphasic mixture model. *Applied Mathematics and Computation*, 88(2-3):153–175, 1997.
- [41] Y. C. Hon, M. W. Lu, W. M. Xue, and X. Zhou. Numerical algorithm for triphasic model of charged and hydrated soft tissues. *Computational Mechanics*, 29(1):1–15, 2002.
- [42] D. L. Young, S. C. Jane, C. Y. Lin, C. L. Chiu, and K. C. Chen. Solution of 2d and 3d stokes laws using multiquadric method. *Engineering Analysis with Boundary Elements*, 28:1233–1243, 2004.
- [43] E. J. Kansa and Y. C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *Computers and Mathematics with Applications*, 39(7–8):123–137, 2000.
- [44] A. S. M. Wong, Y. C. Hon, T. S. Li, S. L. Chung, and E. J. Kansa. Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme. *Computers and Mathematics with Applications*, 37(8):23–43, 1999.
- [45] G.E. Fasshauer. *Handbook of Theoretical and Computational Nanotechnology*, chapter Meshfree Methods. American Scientific, 2005.
- [46] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics, 2003.
- [47] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.*, 2:11–12, 1986.
- [48] I. J. Schoenberg. Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.*, 44:522–536, 1938.

- [49] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39:811–841, 1938.
- [50] K. Guo, S. Hu, and X. Sun. Conditionally positive definite functions and laplace-stieltjes integrals. *J. Approx. Theory*, 74:249–265, 1993.
- [51] R. Franke. Scattered data interpolation: test of some methods. *In Mathematics of Computation*, 38:181–200, 1982.
- [52] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76(8):1905–1915, 1971.
- [53] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [54] Z. Wu. Compactly supported definite radial functions. *Adv. in Comput. Math.*, 4:283–292, 1995.
- [55] M. D. Buhmann. Radial functions on compact support. *Proceedings of the Edinburgh Mathematical Society*, 41:33–46, 1998.
- [56] Serge Lang. *Linear Algebra*. Addison-Wesley, 1969.
- [57] Walter Rudin. *Análisis Funcional*. Editorial Reverté, 1979.
- [58] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Kogakusha, Ltd., 1976.
- [59] Robert Schaback. Native hilbert spaces for radial basis functions i. *International Series of Numerical Mathematics, Birkhauser Verlag*, 132:255–282, 1999.
- [60] Robert Schaback. A unified theory of radial basis functions. native hilbert spaces for radial basis functions ii. *J. Comput Appl. Math.*, 121:165–177, 2000.

- [61] Zong-Min Wu and Robert Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA Journal of Numerical Analysis*, 13:13–27, 1993.
- [62] Lawrence F. Shampine, R. C. Allen, and S. Pruess. *Fundamentals of Numerical Computing*. Jhon Wiley & Sons, USA, 1997.
- [63] W. R. Madych and S. A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70(1):94–114, 1992.
- [64] H. Wendland. *Gaussian interpolation revisited*, pages 427–436. Vanderbilt University Press, 2001.
- [65] W. R. Madych. Miscellaneous error bounds for multiquadric and related interpolator. *Computers and Mathematics with Applications*, 24(12):121–138, 1992.
- [66] Robert Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.
- [67] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *R.A.I.R.O. Analyse Numeriques*, 10:5–12, 1976.
- [68] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive theory of functions of several variables*, (W. Schempp, K. Zeller, eds.), Springer-Verlag, pages 85–100, 1977.
- [69] J. Duchon. Sur l’erreur d’interpolation des fonctions de plusieurs variables par les d^m -splines. *R.A.I.R.O. Analyse Numeriques*, 12:325–334, 1978.
- [70] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions i. *Approx. Theory Appl.*, 4:77–89, 1988.

- [71] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions ii. *Math. Comp.*, 54:211–230, 1990.
- [72] Shmuel Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2–3):193–210, 1999.
- [73] E. J. Kansa and R. E. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameter. *Comput. Math. Appl.*, 24:99–120, 1992.
- [74] Leevan Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Mathematical and Computer Modelling*, 40(13):1413–1427, 2004.
- [75] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito. Exponential convergence and h - c multiquadric collocation method for partial differential equations. *Numerical Methods Partial Differential Equations*, 19(5):571–594, 2003.
- [76] R. K. Beatson and G.Ñ. Newsam. Fast evaluation of radial basis functions: I. *Computers Math. Applic.*, 24(12):7–19, 1992.
- [77] J. B. Cherrie, R. K. Beatson, and G.Ñ. Newsam. Fast evaluation of radial basis functions: Methods for generalized multiquadrics in r^n . *SIAM J. Sci. Comput.*, 23(5):1549–1571, 2002.
- [78] R. K. Beatson and W. A. Light. Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *IMA Journal of Numerical Analysis*, 17(3):343–371, 1997.
- [79] R. K. Beatson, M. J. D. Powell, and A. M. Tan. Fast evaluation of polyharmonic splines in three dimensions. *IMA Journal of Numerical Analysis*, 2006; doi:10.1093/imanum/drl027.

- [80] R. K. Beatson, J. B. Cherrie, and D. L. Ragozin. Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines. *SIAM J. Math. Anal.*, 32(6):1272–11310, 2001.
- [81] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995.
- [82] A.C. Faul and M.J.D. Powell. Krylov subspace methods for radial basis function interpolation. *Numerical Analysis*, pages 115–141, 1999.
- [83] R.K. Beatson and M.J.D. Powell. An iterative method for thin plate spline interpolation that employs approximations to lagrange functions. *Numerical Analysis, D. F. Griffiths & G. A. Watson, eds.*, pages 17–39, 1994.
- [84] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: methods based on preconditioned gmres iteration. *Advances in Comput. Math.*, 11:253–270, 1999.
- [85] Nail A. Gumerov and Ramani Duraiswami. Fast radial basis function interpolation via preconditioned krylov iteration. *accepted SIAM J. Scientific Computing*, 2006.
- [86] Z. Wu. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.*, 8:1–10, 1992.
- [87] G.E. Fasshauer. *Solving Partial Differential Equations by Collocation with Radial Basis Functions*, pages 131–138. Vanderbilt University Press, 1997.
- [88] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic pdes. *Computers and Mathematics with Applications*, 46(5–6):891–902, 2003.

- [89] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers and Mathematics with Applications*, 48(5-6):853–867, 2004.
- [90] Yiu-Chung Hon and Robert Schaback. On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation*, 119(2-3):177–186, 2001.
- [91] M. R. Dubal, S. R. Oliveira, and R. A. Matzner. Solution of Elliptic Equations in Numerical Relativity Using Multiquadrics. In R. D’Inverno, editor, *Approaches to Numerical Relativity*, pages 265+–, 1992.
- [92] G.E. Fasshauer. Solving differential equations with radial basis functions: multilevel methods and smoothing. *Advances in Computational Mathematics*, 11(2–3):139–159, 1999.
- [93] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via pde collocation on the boundary. *Computers and Mathematics with Applications*, 43(3-5):439–455, 2002.
- [94] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Continuation for nonlinear elliptic partial differential equations discretized by the multiquadric. *International Journal of Bifurcation and Chaos*, 10(2):481–492, 2000.
- [95] G.E. Fasshauer. Newton iteration with multiquadrics for the solution of nonlinear pdes. *Computers and Mathematics with Applications*, 43(3–5):423–438, 2002.
- [96] Jichun Li, H. D. Alexander Cheng, and Chen Ching-Shyang. A comparison of efficiency and error convergence of multiquadric collocation method and finite element method. *Engineering Analysis with Boundary Elements*, 27(3):251–257, 2003.

- [97] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems*. John Wiley, 1991.
- [98] Lawrence F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall Mathematics, New York, USA, 1994.
- [99] C. W. Gear and D. R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.
- [100] Phani P. Chinchapatnam, K. Djidjeli, and Prasanth B. Nair. Domain decomposition for time-dependent problems using radial based meshless methods. *Numerical Methods for Partial Differential Equations*, 23(1):38–59, 2007.
- [101] W.R. Madych and S.A. Nelson. Multivariate interpolation and conditionally positive definite functions. *Approximation Theory and its Applications*, 4:77–89, 1990.
- [102] José Antonio Muñoz-Gómez, Pedro González-Casanova, and Gustavo Rodríguez-Gómez. Adaptive node refinement collocation method for partial differential equations. In *ENC'06: Proceedings of the Seventh Mexican International Conference on Computer Science*, pages 70–80, Washington, DC, USA, September 2006. IEEE Computer Society.
- [103] T.A. Driscoll and A.R.H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Submitted to Comp. Math. with Applications*, 53(6):927–939, 2007.
- [104] Barry Smith, Petter Bjorstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [105] Alberto Valli and Alfio Quarteroni. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, 1999.

- [106] Barbara I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Springer Lecture Notes in Computational Science and Engineering, 2001.
- [107] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics, 2005.
- [108] George Em Karniadakis and Robert M. Kirby. *Parallel Scientific Computing in C++ and Mpi: A Seamless Approach to Parallel Algorithms and Their Implementation*. Cambridge University Press, 2003.
- [109] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel Computing*, 22(6):789–828, 1996.
- [110] H. A. Schwarz. Ueber einen grenzübergang durch alternirendes verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zurich*, 1870.
- [111] M.R. Dubal. Domain decomposition and local refinement for multiquadric approximations. I: Second-order equations in one-dimension. *J. Appl. Sci. Comp.*, 1(1):146–171, 1994.
- [112] Zongmin Wu and Y.C. Hon. Additive schwarz domain decomposition with radial basis approximation. *Int. J. Appl. Math.*, 4(1):81–98, 2000.
- [113] X. Zhou, Y. C. Hon, and Jichun Li. Overlapping domain decomposition method by radial basis functions. *Applied Numerical Mathematics*, 44(1–2):241–255, 2003.
- [114] Jichun Li and C.S. Chen. Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems. *International Journal for Numerical Methods in Engineering*, 57(8):1085–1094, 2003.

- [115] Jichun Li and Y. C. Hon. Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations*, 20(3):450–462, 2004.
- [116] Yunxin Zhang and Yongji Tan. Convergence of general meshless schwarz method using radial basis functions. *Applied Numerical Mathematics*, 56(7):916–936, 2006.
- [117] José Antonio Muñoz-Gómez, Pedro González-Casanova, and Gustavo Rodríguez-Gómez. Domain decomposition by radial basis functions for time dependent partial differential equations. In *ACST'06: Proceedings of the IASTED International Conference on Advances in Computer Science and Technology*, pages 105–109, Anaheim, CA, USA, January 2006. ACTA Press.
- [118] R. Leveque. *Numerical Methods for Conservation Laws*. Lectures in Mathematics ETH Zürich, Birkhäuser-Verlag, 1990.
- [119] J. Behrens, A. Iske, and St. Pohn. Effective node adaption for grid-free semi-lagrangian advection. In *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, pages 110–119, Berlin, 2001. Logos Verlag.
- [120] P. González-Casanova. Numerical methods for the solution of partial differential equations: A review. Technical report, Universidad Nacional Autónoma de México, DGSCA Unidad de Cómputo Aplicado, 2003.
- [121] Y. E. Kallinderis. Preface: Adaptive methods for compressible cfd. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1053, 2000.
- [122] Joe F. Thompson, Bharat K. Soni, and Nigel P. Weatherill, editors. *Handbook of grid generation*. CRC Press, 1999.
- [123] Shaofan Li and Wing Kam Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55(1):1–34, 2002.

- [124] A.K. Patra, A.C. Bauer, C.C. Nichita, E.B. Pitman, M.F. Sheridan, M. Bursik, B. Rupp, A. Webber, A.J. Stinton, L.M. Namikawa, and C.S. Renschle. Parallel adaptive numerical simulation of dry avalanches over natural terrain. *Journal of Volcanology and Geothermal Research*, 139(1-2):1–21, 2005.
- [125] J. Behrens, A. Iske, and M. Kaser. *Adaptive Meshfree Method of Backward Characteristics for Nonlinear Transport Equations*, pages 21–36. Springer Verlag, 2003.
- [126] Zongmin Wu. Dynamically knots setting in meshless method for solving time dependent propagations equation. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1221–1229, 2004.
- [127] Scott A. Sarra. Adaptive radial basis function methods for time dependent partial differential equation. *Applied Numerical Mathematics*, 54(1):79–94, 2005.
- [128] Leevan Ling and Manfred R. Trummer. Adaptive multiquadric collocation for boundary layer problem. *Journal of Computational and Applied Mathematics*, 188(2):265–282, 2006.
- [129] Tim Gutzmer and Armin Iske. Detection of discontinuities in scattered data approximation. *Numerical Algorithms*, 16(2):155–170, 1997.
- [130] J. Behrens and A. Iske. Grid-free adaptive semi-lagrangian advection using radial basis functions. *Computers and Mathematics with Applications*, 43(3-5):319–327, 2002.
- [131] Armin Iske and Martin Kaser. Two-phase flow simulation by ammoc, an adaptive meshfree method of characteristics. *Computer Modeling in Engineering and Sciences (CMES)*, 17(2):133–148, 2005.
- [132] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

- [133] M. J. Berger and A. Jameson. Automatic adaptive grid refinement for the euler equations. *American Institute of Aeronautics and Astronautics*, 23(4), 1985.
- [134] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64–84, 1989.
- [135] Quentin F. Stout, Darren L. De Zeeuw, Tamas I. Gombosi, Clinton P. T. Groth, Hal G. Marshall, and Kenneth G. Powell. Adaptive blocks: A high performance data structure. *ACM/IEEE SC Conference (SC'97)*, 1997.
- [136] Zdzislaw Meglicki, Stephen K. Gray, and Boyana Norris. Multigrid fdttd with chombo. *Computer Physics Communications*, 176(2):109–120, 2007.
- [137] Christiane Jablonowski. *Adaptive Grids in Weather and Climate Modeling*. PhD thesis, Department of Atmospheric, Oceanic and Space Sciences, Univ. of Michigan, 2004.
- [138] C. F. Lange, M. Schafer, and F. Durst. Local block refinement with a multigrid flow solver. *International Journal for Numerical Methods in Fluids*, 38(1):21–41, 2002.
- [139] Mike Barad. An embedded boundary adaptive mesh refinement method for environmental flows. In *In Proceedings of 9th Conference on Estuarine and Coastal Modeling*, November 2005.
- [140] Andreas E. Kanarachos Nikos G. Pantelelis. The parallel block adaptive multigrid method for the implicit solution of the euler equations. *International Journal for Numerical Methods in Fluids*, 22(5):411–428, 1996.
- [141] Rune Teigland and Inge K. Eliassen. A multiblock/multilevel mesh refinement procedure for cfd computations. *International Journal for Numerical Methods in Fluids*, 36(5):519–538, 2001.

- [142] W. A. Keats and F.S. Lien. Two-dimensional anisotropic cartesian mesh adaptation for the compressible euler equations. *International Journal for Numerical Methods in Fluids*, 46(11):1099–1125, 2004.
- [143] Center for computational sciences and engineering, lawrence berkeley national laboratory. <http://seesar.lbl.gov/ccse/>.
- [144] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, and D. B. Serafini B. Van Straalen. Chombo software package for amr applications. <http://seesar.lbl.gov/ANAG/chombo/>, 2003.
- [145] R. D. Hornung and S. R. Kohn. Managing application complexity in the samrai object-oriented framework. *Concurrency and Computation: Practice and Experience*, 14:347–368, 2002.
- [146] B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *The Astrophysical Journal Supplement Series*, 131:273–334, 2000.
- [147] Manish Parashar and James C. Browne. Dagh: Data-management for parallel adaptive mesh-refinement techniques. Technical report, <http://www.caip.rutgers.edu/parashar/DAGH/>, 1995.
- [148] Jurgen Dreher and Rainer Grauer. Raccoon: A parallel mesh-adaptive framework for hyperbolic conservation laws. *Parallel Computing*, 31(8-9):913–932, 2005.
- [149] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, 1990.
- [150] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 1999.

- [151] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.
- [152] Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. In *Conference on High Performance Networking and Computing*, 1995.
- [153] R. Preis and R. Diekmann. The party partitioning-library, user guide - version 1.1. Technical report, University of Paderborn, 1996.
- [154] F. Pellegrini and J. Roman. Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In *LNCS: Proceedings of HPCN'96*, pages 493–498, 1996.
- [155] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2:135–148, 1991.
- [156] Gaston H. Gonnet and Ricardo Baeza-Yates. *Handbook of Algorithms and Data Structures*. Addison-Wesley, 1990.
- [157] Robert Sedgewick. *Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching*. Addison-Wesley, 1999.
- [158] Nail A. Gumerov, Ramani Duraiswami, and Eugene A. Borovikov. Data structures, optimal choice of parameters, and complexity results for generalized multilevel fast multipole methods in d dimensions. Technical report, University of Maryland Institute for Advanced Computer Studies, 2003.
- [159] A.S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 2nd ed edition, 2001.
- [160] Charles F. Van Loan Gene H. Golub. *Matrix Computations*. Jhons Hopkins, 1996.

- [161] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994.
- [162] G. A. Geist, J. A. Kohla, and P. M. Papadopoulos. PVM and MPI: A Comparison of Features. *Calculateurs Paralleles*, 8(2):137–150, 1996.
- [163] Kirk Schloegel, George Karypis, and Vipin Kumar. Parallel static and dynamic multi-constraint graph partitioning. *Concurrency and Computation: Practice and Experience*, 14(3):219–240, 2002.
- [164] C. Walshaw and M. Cross. Parallel optimisation algorithms for multilevel mesh partitioning. *Parallel Comput.*, 26(12):1635–1660, 2000.
- [165] C. Walshaw, M. Cross, and M. G. Everett. Parallel Unstructured Mesh Partitioning. In *Domain Decomposition Methods in Sciences and Engineering*, pages 647–654. DDM.org, www.ddm.org, 1998. (Proc. DDM '96, Bergen, Norway, 1996).
- [166] Chao-Wei Ou and Sanjay Ranka. Parallel incremental graph partitioning. *IEEE Transactions on Parallel and Distributed Systems*, 08(8):884–896, 1997.
- [167] Weicheng Huang and Danesh K. Tafti. A parallel adaptive mesh refinement algorithm for solving nonlinear dynamical systems. *Int. J. High Perform. Comput. Appl.*, 18(2):171–181, 2004.
- [168] Mark T. Jones and Paul E. Plassmann. Parallel algorithms for adaptive mesh refinement. *SIAM Journal on Scientific Computing*, 18(3):686–708, 1997.
- [169] M. Parashar and J. Browne. Distributed dynamic data-structures for parallel adaptive meshrefinement, 1995.

- [170] C. Ozturan, H. deCougny, M. Shephard, and J. Flaherty. Parallel adaptive mesh refinement and redistribution on distributed memory computers, 1993.
- [171] Saeed Iqbal and Graham F. Carey. Performance analysis of dynamic load balancing algorithms with variable number of processors. *Journal of Parallel and Distributed Computing*, 65(8):934–948, 2005.
- [172] A. Bourgeade and B. Ñkonga. Dynamic load balancing computation of pulses propagating in a nonlinear medium. *J. Supercomput.*, 28(3):279–294, 2004.
- [173] Bruce Hendrickson and Karen Devine. Dynamic load balancing in computational mechanics. *Comp. Meth. Applied Mechanics and Engineering*, 184(2-4):485–500, 2000.
- [174] Tommy Minyard and Yannis Kallinderis. Parallel load balancing for dynamic execution environments. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1295–1309, 2000.
- [175] Roy D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Pract. Exper.*, 3(5):457–481, 1991.