



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS
COMPUTACIÓN CIENTÍFICA

Cómputo paralelo para la solución de flujo en medios porosos
aplicando funciones de base radial

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
GERMÁN STREMLER CHÁVEZ

Director de Tesis:
Dr. Luis Miguel De la Cruz Salas
Instituto de Geofísica



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Este trabajo ha nacido del deseo de ver más allá de donde alcanza mi mirada, y de aportar a la sociedad lo descubierto. He tenido para ello, el privilegio de contar con el invaluable apoyo de muchas personas e instituciones sin el cual no habría sido posible.

Primeramente quiero agradecer a mi director de tesis, el Dr. Luis Miguel de la Cruz Salas, por guiarme con su sabiduría en terrenos desconocidos, por alentarme a lograr un provechoso descubrimiento y por inspirarme la investigación.

A todos mis profesores del posgrado que por compartir lo que saben, tengo herramientas para concretar este trabajo y grandes alternativas de aplicación de mis conocimientos. Agradezco que me ayudaran a ampliar mi visión.

A mi jurado de examen de grado: Dr. Ismael Herrera Revilla, Dr. Luis Miguel De la Cruz Salas, Dr. Demetrio Fabián García Nocetti, Dr. Guillermo Hernández García y al Dr. Pedro González Casanova Henríquez, cuyos comentarios y correcciones mejoraron este trabajo rumbo a su objetivo.

También quiero agradecer a la Universidad Nacional Autónoma de México, al posgrado de Ciencia e Ingeniería de la Computación y al Instituto de Geofísica, así como a su personal académico y administrativo que me apoyaron durante mis estudios de posgrado. Al programa nacional de Becas de Posgrado del CONACyT cuyo apoyo económico lo tuve durante dos años.

Finalmente quiero dar mi más profundo agradecimiento a mi esposa, que me ha dado su infatigable mano en el arduo camino; a mi hijo, por su tierna paciencia; a mis padres por la disposición de sus manos y su corazón y a mis hermanos, por apoyarme en momentos cruciales.

A todos, ¡ gracias por favorecerme con sus obras !

Dedicatoria

A mi esposa

La que llegó de mis sueños para acompañarme en el duro camino hacia ellos.

A mi hijo

El que los habita y con infinita ternura los acaricia.

A mis padres

Las poderosas raíces amorosas de donde vienen.

A mis hermanos

Mis amados espejos, que los abrazan cuando nos vemos.

Índice General

Índice General	5
Índice de Figuras	8
1 Introducción	15
1.1 Modelación de la recuperación de hidrocarburos	15
1.2 Modelación matemática y computacional	17
1.3 Objetivos y metas	17
1.4 Organización de la tesis	18
2 Modelo conceptual	21
2.1 Definición del problema	21
3 Modelo matemático	23
3.1 Propiedades intensivas	23
3.2 Propiedades extensivas	24
3.3 Ecuaciones de balance de propiedades extensivas e intensivas	24
3.3.1 Ecuaciones de balance global	24
3.3.2 Ecuaciones de balance local	25
3.4 Forma conservativa de las ecuaciones de balance	25
3.5 Modelo matemático de flujo en sistemas multifásicos	26
3.6 Formulación presión-saturación	27
3.6.1 Ecuación de presión	27
3.6.2 Ecuación de saturación	28
3.6.3 Formulación presión-saturación para flujo en dos fases	28
3.7 Modelo Buckley-Leverett	29
3.8 Permeabilidad relativa	29
4 Modelo numérico	31
4.1 Método de Líneas de Corriente SLS	32
4.1.1 Líneas de corriente (Streamlines)	32
4.1.2 Tiempo de vuelo (Time Of Flight)	33
4.1.3 Simulación con líneas de corriente	34

4.2	Método de Funciones de Base Radial RBF	36
4.2.1	Conceptos fundamentales	36
4.2.1.1	Interpolación	36
4.2.1.2	Desarrollo del método multicuadrado	37
4.2.2	El método RBF	40
4.2.2.1	Interpolación con el método RBF básico	40
4.2.2.2	Interpolación con el método RBF aumentado	41
4.2.2.3	Solución de PDEs con el método RBF básico	42
4.2.2.4	Solución de PDEs con el método RBF aumentado	43
4.3	SLS y RBF para el modelo Buckley-Leverett	44
4.3.1	Solución de la ecuación de la presión	44
4.3.2	Cálculo del campo de velocidad	49
4.3.3	Trazado de las líneas de corriente	50
4.3.4	Cálculo del tiempo de vuelo τ	51
4.3.5	Solución de las ecuaciones de transporte con el método RBF	51
4.3.6	Transformación de la solución de coordenadas TOF a coordenadas espaciales	54
5	Modelo computacional	57
5.1	Herramientas del modelo computacional	57
5.1.1	Cómputo paralelo	57
5.1.2	Computadora paralela de memoria compartida	59
5.1.3	Computadora paralela de memoria distribuida	60
5.2	Algoritmo computacional	61
6	Resultados	67
6.1	Discretización del espacio	67
6.2	Solución del problema usando FVM	68
6.3	Solución del problema usando SLS-RBF	70
6.3.1	Cálculo de la presión con RBF básico	70
6.3.2	Cálculo de la presión con RBF aumentado	71
6.3.3	Construcción de las líneas de corriente	72
6.3.4	Cálculo de la saturación con Forward Euler	72
6.3.5	Cálculo de la saturación con Backward Euler	73
6.4	Análisis de resultados	74
6.4.1	Resultados con FVM	74
6.4.2	Resultados con FVM y con la presión fuera del IMPES	75
6.4.3	Resultados con RBF y con la presión fuera del IMPES	75
6.4.4	Resultados con RBF, la presión fuera del IMPES y cómputo paralelo	76
6.4.5	Resumen comparativo	76
7	Conclusiones	79
7.1	Conclusiones	79
7.2	Direcciones futuras	80

Apéndices	81
A Ecuación de presión	83
B Inyección de agua	85
C Interpolación de S	87
D Derivadas de RBF	89
E Cálculo analítico de $\frac{\partial F_w}{\partial S}$	91
Bibliografía	93

Índice de Figuras

1.1	Etapas del proceso de modelado y simulación	18
2.1	Medio homogéneo inicialmente saturado con aceite	22
3.1	Volumen representativo del medio continuo	25
4.1	A la izquierda el campo de velocidad en un dominio dado. A la derecha, las líneas de corriente tangentes al campo de velocidad local.	33
4.2	Tiempo de vuelo τ para una partícula de prueba.	33
4.3	Interpolación de algunos datos dispersos usando la ecuación (4.10). Los puntos fuente están marcados con círculos.	38
4.4	Interpolación de algunos datos dispersos usando la ecuación (4.11) con $c = 2$. Los puntos fuente están marcados con círculos.	39
4.5	Puntos esparcidos en el dominio.	46
5.1	Arquitectura de John von Neumann	58
5.2	Arquitectura de un multiprocesador centralizado genérico.	60
5.3	Arquitectura de un computadora multiprocesador de memoria distribuida.	60
5.4	Algoritmo computacional	62
5.5	Datos introducidos por el usuario.	63
5.6	Creación del objeto points de la clase BoxDots, para construir los puntos en los conjuntos NB, NI y obtener el total N de ellos, distribuyéndolos de manera uniforme o aleatoria según el usuario indique en los datos de entrada.	64
5.7	Interpolación de la saturación.	64
5.8	Construcción de las coordenadas iniciales de las partículas de prueba.	65
5.9	Traza de las trayectorias de las líneas de corriente usando el método <i>Forward Euler</i>	66
5.10	Cálculo de la saturación en cada línea.	66
6.1	Discretización del dominio	67
6.2	Función de la presión obtenida con FVM.	68
6.3	Función de la saturación obtenida con FVM.	68
6.4	Error de la presión para cada tiempo de simulación.	69

6.5	Función de la saturación deseada.	70
6.6	Trazo de las líneas de corriente.	72
6.7	Saturación en líneas de corriente usando el método RBF básico con base en Forward Euler.	73
6.8	Saturción en líneas de corriente usando el método RBF básico con base en Backward Euler.	73
6.9	Error de la saturación dependiendo del número de puntos en el dominio.	74
6.10	Función de la saturación obtenida con RBF.	76
6.11	Función de la saturación obtenida con FVM, RBF y Analítica.	77

Abstract

In order to support in a quick and accurate way, the characterization of the Secondary Oil Recovery in a reservoir, this work presents a simulation of the oil displacement by the water through a homogeneous porous medium. "The Radial Basis Function (RBF) method is used to solve both the transport Partial Differential Equations (PDEs) which are derived from the Streamline Simulation (SLS) method and the pressure PDEs".

The flow is two phase immiscible and incompressible, with water in one phase and oil in the other; the effects of capillary pressure is not taken into account; there is no existence of source and sink and finally, the gravity effects are neglected. All of this considerations are taken into account throughout this work. Buckley-Leverett two phase fluid is the mathematical model obtained with the axiomatic formulation that Ismael Herrera has introduced. This model consists of a non linear PDEs set where they are coupled strongly. The Implicit Pressure Explicit Saturation (IMPES) method is used to solve them. The RBF method is used to solve the governing equations of pressure and on the other hand, the governing equations of transport are translated into the individual streamlines first and then are solved with the RBF method too.

The saturation PDEs along their respective streamlines don't depend on each other, so that, they can be solved in a parallel way. In order to perform this technique, an algorithm based on a shared memory parallel computer is constructed, where one needs the OpenMP API to distribute the tasks and communicate the different processes.

At the end, after application of RBF method that is used to obtain the solution, the results are analyzed and compared with traditional methods in order to show, how this one in addition to overcome by its high accuracy and speed of convergence, it increases its efficiency using the parallel computing.

Resumen

Para ayudar de manera más rápida y precisa a la caracterización del comportamiento de la recuperación secundaria de hidrocarburos en un yacimiento petrolero, este trabajo presenta una simulación del desplazamiento de aceite con agua en un medio poroso homogéneo, resolviendo con Funciones de Base Radial RBFs (del inglés *Radial Basis Functions*) tanto las Ecuaciones Diferenciales Parciales PDEs (del inglés *Partial Differential Equations*) gobernantes de transporte formuladas con el método Simulación de Líneas de Corriente SLS (del inglés *Streamline Simulation*), como las PDEs de la presión.

Los dos fluidos considerados son inmiscibles e incompresibles, no se toman en cuenta los efectos de la presión capilar, no hay fuentes ni sumideros y no se considera la fuerza de gravedad. El modelo matemático es el del flujo bifásico Buckley-Leverett, al cual se llega por medio de la formulación axiomática. Este modelo consiste de un conjunto de PDEs no lineales fuertemente acopladas que se resuelven utilizando el método IMPES (del inglés *IMplicit Pressure EXplicit Saturation*) mejorado, donde las ecuaciones gobernantes de presión, son resueltas aplicando RBF y las ecuaciones gobernantes de transporte, luego de ser transformadas a líneas de corriente individuales, se linealizan para ser resueltas de la misma forma.

Las PDEs a lo largo de las líneas de corriente son independientes y por tanto, pueden ser resueltas paralelamente. Para ello se implementa el algoritmo con base en la filosofía de computadora paralela de memoria compartida, donde para distribuir las tareas y comunicar a los procesos se usa la interfaz de programación OpenMP.

Finalmente, de la aplicación de RBF en la solución, se analizan los resultados y se hace una comparación contra métodos tradicionales para exponer cómo este método además de superarlos por su alta precisión y rapidez de convergencia, incrementa su eficiencia aún más con la paralelización del cómputo.

1

Introducción

Un yacimiento, depósito o reservorio petrolífero es una acumulación natural de hidrocarburos en el subsuelo contenidos en rocas porosas o fracturadas. El principal objetivo de la simulación de yacimientos es predecir la tasa de recuperación de hidrocarburos y encontrar la forma de optimizarla bajo diversas condiciones de operación. Un proyecto de recuperación usualmente involucra un capital de investigación muy costoso, por lo que el riesgo asociado a las estrategias de producción y desarrollo debe ser evaluado y minimizado. Este riesgo incluye factores importantes como la complejidad de la naturaleza de la roca y los fluidos que contiene el yacimiento, la complejidad de los mecanismos de recuperación y la aplicabilidad de los métodos predictivos. Estos factores pueden tomarse en cuenta en la simulación y de acuerdo con los resultados obtenidos en ella, estimar la caracterización real de recuperación.

1.1. Modelación de la recuperación de hidrocarburos

Generalmente, hay tres etapas de recuperación que son identificadas en la vida productiva de un yacimiento petrolero: primaria, secundaria y terciaria. La *recuperación primaria* se refiere a la producción que es obtenida usando la energía inherente en el yacimiento debido al gas bajo presión o al movimiento natural del agua. En una etapa muy temprana, el yacimiento contiene un solo fluido tal como el gas o el aceite (la presencia del agua puede usualmente ser despreciada) y frecuentemente la presión es tan alta que el aceite ó el gas es producido sin la necesidad de pozos inyectores. La *recuperación primaria* termina cuando el campo de aceite y la atmósfera alcanzan el equilibrio en la presión. La recuperación total obtenida en esta etapa es generalmente de 12-15% de los hidrocarburos contenidos en el yacimiento.

La técnica de inyección de agua, es considerada como el *método de recuperación*

secundaria. En ese enfoque, el agua es inyectada en algunos pozos (de inyección) para mantener el campo de presión y las tasas de flujo, mientras que el petróleo es producido a través de otros pozos (de producción). En este tipo de recuperación el flujo es bifásico inmiscible donde el agua es una fase y el aceite la otra. En este caso no hay intercambio de masa entre las fases. La recuperación secundaria genera una adicional del 15-20 % de los hidrocarburos contenidos en el yacimiento.

Después de que se ha completado la recuperación secundaria, 50 % ó más de los hidrocarburos frecuentemente permanecen en el yacimiento. Las técnicas más avanzadas que han sido desarrolladas para recuperar el valioso volumen de hidrocarburos restantes son conocidas como *técnicas de recuperación terciaria ó mejorada*[1].

En la recuperación de hidrocarburos contenidos en un yacimiento petrolero, existe el riesgo de tener pérdidas de cientos de millones de dólares debido a los procesos complejos que están involucrados y al gran capital que se invierte. Dada la necesidad de minimizar ese riesgo, la industria petrolera se ha ido proveyendo de métodos que le permiten predecir el futuro desempeño del yacimiento. Algunos métodos clásicos predictivos son los analógicos, los experimentales y los matemáticos. Los *métodos analógicos* utilizan características de yacimientos maduros que son análogos al que está bajo estudio. Los *métodos experimentales* miden las propiedades físicas como la presión y la saturación en los laboratorios y luego las escalan a toda la acumulación del hidrocarburo. Finalmente los *métodos matemáticos* por medio de las ecuaciones de un modelo pronostican el desempeño del yacimiento [2] . Estos métodos matemáticos se engloban en lo que hoy en día se conoce como la Modelación Matemática y Computacional (MMC), véase por ejemplo [3].

En las últimas cuatro décadas, la simulación numérica en MMC ha sido dominada por los métodos de Diferencias Finitas (FDM por sus siglas en inglés), de Volumen Finito (FVM por sus siglas en inglés) y de Elemento Finito (FEM por sus siglas en inglés), los cuales requieren de una malla para proporcionar soluciones aproximadas. Sin embargo, con estos métodos al poligonizar el espacio cuando se trata de problemas que involucran resolución alta y geometrías complejas, se producen mas variables de las que pueden manejarse tanto por el método como por el equipo de cómputo actualmente existente. Los recientes desarrollos en las técnicas de Simulación con Líneas de Corriente (SLS por sus siglas en inglés) atacan este tipo de problemas proporcionando herramientas para la simulación rápida de yacimientos a escalas finas [4]. La evolución de los frentes de inyección y su interacción con las heterogeneidades del yacimiento pueden ser visualizadas fácil y rápidamente. Los fundamentos de estas técnicas se remontan a mediados del siglo XIX, desde entonces ha habido un constante desarrollo de este tipo de métodos [5].

El método SLS es una herramienta que proporciona rapidez y eficiencia en la simulación de yacimientos.

En la década pasada, se han estudiado intensamente métodos para resolver ecuaciones diferenciales parciales que no incluyan la técnica del mallado, porque de esta

forma es posible tratar con dominios más complejos. De entre los métodos más populares está el de Funciones de Base Radial (RBF por sus siglas en inglés) [6, 7, 8], que tiene la característica importante de convergencia exponencial a la solución con un pequeño número de puntos en el dominio de estudio. Es decir, el método aproxima de manera más precisa a la solución con el manejo de un menor número de variables, y en principio también con menor tiempo de cómputo.

Simular el yacimiento combinando SLS y RBF, implicaría hacerlo con mayor rapidez y precisión numérica que cuando se hace con métodos tradicionales. Adicionalmente, una característica que tienen las líneas de corriente es que son independientes entre ellas, lo que permite hacer el cómputo en paralelo de la solución repartiendo grupos de líneas de corriente a diferentes procesos y en ellos obtener las soluciones de transporte para cada uno de los grupos [9].

1.2. Modelación matemática y computacional

El uso de la MMC en los procesos industriales tiene beneficios como la reducción del costo del producto y del desarrollo del proceso, la optimización de actividades, la reducción de experimentación física y la mejora en la confiabilidad del diseño [10]. La MMC consta de cuatro pasos principales: (1) Modelo Conceptual, donde se toman en cuenta todos los procesos que se desean estudiar durante una simulación, (2) Modelo matemático, donde un conjunto de ecuaciones matemáticas expresan el comportamiento del fenómeno bajo estudio, (3) Modelo numérico, el cual utiliza métodos numéricos para aproximar las soluciones bajo diferentes contextos y (4) Modelo computacional, donde se programan los recursos de una computadora para hacer los cálculos con una enorme cantidad de datos y visualizar los resultados de manera entendible. Las etapas del proceso de modelado pueden verse en el diagrama de flujo de la figura 1.1.

1.3. Objetivos y metas

El objetivo general de este trabajo es evaluar, en términos de precisión y eficiencia, los métodos SLS y RBF en la solución de un problema de inyección de agua en un yacimiento petrolero simple.

Para lograr este objetivo general se plantean las siguientes metas:

- ❶ Definir el modelo conceptual de un problema de inyección de agua en un yacimiento para un dominio de estudio sencillo. Esto incluye la definición de condiciones iniciales y de frontera.
- ❷ Definir un modelo matemático que tome en cuenta todos los procesos descritos en el modelo conceptual. Para hacer uso de las técnicas SLS, el modelo matemático

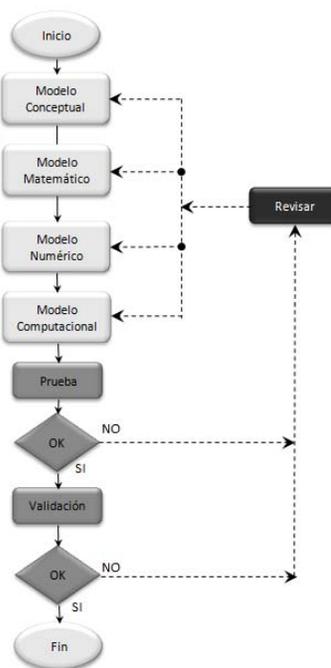


Figura 1.1: Etapas del proceso de modelado y simulación

se hará en términos de una formulación presión-saturación. Además de estas ecuaciones, que son diferenciales parciales no lineales y acopladas, se definirán relaciones constitutivas para algunos parámetros físicos.

- 3 Definir un modelo numérico en donde se describirá con detalle la aplicación del método SLS. En resumen, se hará uso de un método que desacople las ecuaciones: método IMPES, y se enfatizarán los lugares de dicho método donde el SLS y RBF son implementados.
- 4 Definir un modelo computacional en el cual se explicarán los pormenores de la implementación en un código de los métodos numéricos. Adicionalmente, se explicará la manera en que el método SLS puede paralelizarse.

1.4. Organización de la tesis

En el **capítulo 2** se aborda el modelo conceptual, donde se define el problema especificando todos los procesos que se van estudiar y las condiciones bajo las cuales van a desempeñarse durante la simulación.

En el **capítulo 3** se hace un desarrollo de los modelos matemáticos que describen el desplazamiento de un flujo bifásico en dos dimensiones en un medio poroso. Dicho modelo matemático consta de ecuaciones diferenciales parciales que se deducen a partir de la formulación axiomática [11, 1].

En el **capítulo 4** se presentan tres métodos numéricos que se utilizan para aproximar la solución del modelo matemático. Los métodos son: algoritmo Presión Implícita Saturación Explícita, Simulación con Líneas de Corriente y Funciones de Base Radial. Se plantea la implementación del primero con la combinación de los dos últimos para resolver el problema.

En el **capítulo 5** se hace la codificación de los algoritmos de la sección anterior con el uso de herramientas computacionales tales como el lenguaje de programación C++, los paradigmas de programación orientada a objetos y programación genérica, la biblioteca Eigen para operaciones de álgebra lineal, OpenMP para la paralelización del programa y Gnuplot para la graficación de las funciones resultantes.

En el **capítulo 6** se presentan tablas y gráficas de los resultados obtenidos de la ejecución del programa computacional. Los resultados incluyen número de puntos usados en la discretización del dominio, valores de algunos parámetros de entrada, precisión de la soluciones obtenidas y consumo de tiempo de CPU para los cálculos realizados.

Finalmente, el **capítulo 7** consta de las conclusiones a las que se llegó en este trabajo y de algunas posibles direcciones futuras que se tienen respecto a él.

2

Modelo conceptual

En este trabajo, la producción de aceite se lleva a cabo en el yacimiento por medio de su desplazamiento con agua; la cantidad de recuperación es limitada a la extensión en que el agua desplazante se acumula. La facilidad con que el aceite y el agua fluirán depende de las características de la roca, del fluido y de las cantidades relativas de los fluidos en la roca. Conceptos sobre *propiedades de la roca*, *propiedades de los fluidos* y *propiedades roca-fluidos* pueden verse en [2]. El mecanismo de desplazamiento de aceite con agua, consiste básicamente en que el fluido desplazante se mueve de una región de alta saturación a una de baja saturación, y por tanto, remueve al aceite convirtiendo la región invadida en una de alta saturación del fluido desplazante. Esto no quiere decir sin embargo, que se trate de un pistón en que el fluido desplazante esté expulsando aceite de los poros y ocupando el espacio vaciado, sino que en todos los casos, tanto el aceite como el agua fluyen juntos y simultáneamente a través de los mismos poros. El desplazamiento en consecuencia, jamás puede ser completo.

2.1. Definición del problema

Se tiene un dominio bidimensional como el mostrado en la figura 2.1 inicialmente saturado con aceite. Se inyecta agua a una razón de flujo constante en la cara A del dominio, la cual desplazará el aceite hacia la cara B donde la presión se mantiene constante. Además se hacen las siguientes suposiciones:

- No se consideran los efectos de la presión capilar.
- No hay fuentes ni sumideros.
- Los fluidos son incompresibles e inmiscibles.
- El medio poroso es homogéneo.

- No se toman en cuenta los efectos de la fuerza de gravedad.

Las condiciones iniciales y de frontera son las siguientes:

Condiciones	Presión	Saturación
Iniciales	$p(t_0) = 1e + 07Pa$	$S(t_0) = 0$
De frontera	$(g_p^{in})_A = 3,4722e - 07m/s$ $(P^{out})_B = 1e + 07Pa$	$(S^{in})_A = 0,8$ $(S^{out})_B = 0$

Tabla 2.1: Condiciones iniciales y de frontera.



Figura 2.1: Medio homogéneo inicialmente saturado con aceite

Nótese que las condiciones de frontera para la saturación son de tipo Dirichlet, mientras que para la presión se tiene una condición Dirichlet a la salida y una condición tipo Neumann a la entrada dada en términos de la velocidad de inyección.

3

Modelo matemático

Para formular los modelos matemáticos que nos permitirán predecir el comportamiento del sistema, se usa la formulación axiomática desarrollada e introducida por Herrera *et al.* [11, 1].

Existen dos enfoques para estudiar la materia y su movimiento: el *enfoque microscópico*, el cual estudia moléculas, átomos y partículas elementales; y el *enfoque macroscópico*, el cual estudia sistemas con un número muy grande de partículas y que en su conjunto se consideran sistemas continuos. La predicción del comportamiento de estos sistemas, como el de un yacimiento petrolero ó la atmósfera, los cuales contienen un número muy grande de moléculas y átomos, es un objetivo inalcanzable usando el enfoque microscópico. Por tanto, para problemas de importancia práctica en ciencia e ingeniería, un enfoque macroscópico es el apropiado. Para este último, el cual es el propósito de la *formulación axiomática*, la mecánica del medio continuo es la que proporciona los fundamentos teóricos. En este enfoque, los individuos microscópicos son ignorados; en su lugar, las poblaciones se asumen como distribuciones continuas en todo el espacio que ellas ocupan.

En la teoría de sistemas continuos, un cuerpo es un conjunto de partículas que en cualquier momento dado, ocupa un dominio (en el sentido en que esta palabra es usada en matemáticas) del espacio físico. El conjunto de estas partículas es denotado por \mathcal{B} , y el dominio que ocupa en el tiempo t será denotado por $B(t)$, donde $-\infty < t < \infty$, aunque en la mayoría de los estudios de sistemas físicos, el período de interés es finito.

3.1. Propiedades intensivas

En la mecánica continua hay una clase de funciones como la densidad o temperatura las cuales cuando son estudiadas, están definidas para cada *partícula* de un *cuerpo* y para cada tiempo. En general, cualquier función como estas se conoce como *propiedad intensiva*.

Existen dos representaciones para estas propiedades dependiendo de la referencia

que se tome para su observación: si se considera el valor de la propiedad intensiva tomando como referencia la partícula bajo estudio en cualquier tiempo, se le llama *Representación Lagrangiana* y si se hace tomando como referencia el espacio físico en cualquier tiempo, se le llama *Representación Euleriana* [1].

En el problema de este trabajo, que es la solución de flujo en medios porosos, se usa la segunda representación. Se plantean las ecuaciones desde el punto de vista de las coordenadas del espacio físico, con el fin de obtener el valor de las propiedades intensivas para las partículas que pasen por los puntos indicados.

3.2. Propiedades extensivas

Se consideran funciones (es decir, propiedades) que son *definidas para cada cuerpo de un sistema continuo*. Este concepto contrasta con el de la sección anterior, el cual fue definido para cada partícula material. La definición de una propiedad extensiva: $E(\mathcal{B}, t)$ se dice que es una *propiedad extensiva* cuando para cada \mathcal{B} puede ser expresada como una integral de una propiedad intensiva sobre el cuerpo; esto es:

$$E(\mathcal{B}, t) = E(B(t), t) = \int_{B(t)} \psi(\underline{x}, t) d\underline{x} \quad (3.1)$$

El integrando de la ecuación anterior, define una función $\psi(\underline{x}, t)$ la cual es una propiedad intensiva Euleriana. La ecuación (3.1) establece una correspondencia biunívoca entre las propiedades intensivas y extensivas, porque el integrando de la función puede siempre ser tomado como la definición de la representación Euleriana de una propiedad intensiva. Inversamente, dada una propiedad intensiva, su integral sobre cada cuerpo define una propiedad extensiva.

3.3. Ecuaciones de balance de propiedades extensivas e intensivas

Los *modelos matemáticos básicos* de sistemas continuos son formulados vía ecuaciones de balance de propiedades intensivas y extensivas.

Para calcular los balances, es necesario identificar tales propiedades que sean relevantes para el problema bajo estudio.

3.3.1. Ecuaciones de balance global

En general el cambio de una propiedad extensiva E con el paso del tiempo, se debe a que ésta se produce en el interior del sistema o a que entra y/o sale por la frontera. Matemáticamente este balance se escribe como

$$\frac{dE}{dt}(t) = \frac{d}{dt} \int_{B(t)} \psi(\underline{x}, t) d\underline{x} = \int_{B(t)} q(\underline{x}, t) d\underline{x} + \int_{\partial B(t)} \underline{\tau}(\underline{x}, t) \cdot \underline{n}(\underline{x}, t) d\underline{x} \quad (3.2)$$



Figura 3.1: Volumen representativo del medio continuo

esta ecuación es referida generalmente como *ecuación general de balance global* donde, $q(\underline{x}, t)$ es la fuente interna o externa de la propiedad extensiva, por unidad de volumen y por unidad de tiempo. Representa la cantidad de la propiedad extensiva que entra al cuerpo en el punto \underline{x} en el tiempo t , por unidad de volumen. $\underline{\tau}(\underline{x}, t) \cdot \underline{n}(\underline{x}, t)$ representa la cantidad de propiedad extensiva que entra al cuerpo a través de su frontera en el punto \underline{x} , en el tiempo t por unidad de área. $B(t)$ es un volumen representativo como el que se muestra en la figura 3.1, y \underline{n} es el vector normal a la superficie $\partial B(t)$ la cual envuelve a $B(t)$.

3.3.2. Ecuaciones de balance local

Un resultado puramente matemático, véase [1], muestra que la ecuación (3.2) se satisface para cada cuerpo $B(t)$ de un sistema continuo, si y solo si, la siguiente condición de balance se cumple

$$\int_{B(t)} \left\{ \frac{\partial \psi}{\partial t} + \nabla \cdot (\underline{v}\psi) \right\} d\underline{x} = \int_{B(t)} q d\underline{x} + \int_{B(t)} \nabla \cdot \underline{\tau} d\underline{x} \quad (3.3)$$

Lo anterior produce la siguiente *ecuación diferencial de balance local* para la propiedad intensiva ψ

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\underline{v}\psi) = q + \nabla \cdot \underline{\tau}. \quad (3.4)$$

Para abordar detalladamente el análisis matemático de la formulación axiomática, véase [11, 1].

3.4. Forma conservativa de las ecuaciones de balance

Las ecuaciones de balance antes descritas se pueden escribir en forma conservativa si se define la siguiente *función de flujo* $\underline{\mathcal{F}} = \underline{v}\psi - \underline{\tau}$. Con esta definición es posible escribir (3.3) como

$$\int_{B(t)} \frac{\partial \psi}{\partial t} + \int_{B(t)} \nabla \cdot \underline{\mathcal{F}} d\underline{x} = \int_{B(t)} q d\underline{x}, \quad (3.5)$$

lo que produce la siguiente ecuación diferencial

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \underline{\mathcal{F}} = q \quad (3.6)$$

Las ecuaciones (3.5) y (3.6) se conocen como la forma conservativa de las ecuaciones (3.3) y (3.4) respectivamente, véase [12].

Adicionalmente, aplicando el teorema de Gauss a la ecuación (3.5) se obtiene

$$\int_{B(t)} \frac{\partial \psi}{\partial t} d\underline{x} + \int_{\partial B} \underline{\mathcal{F}} \cdot \underline{n} d\underline{x} = \int_{B(t)} q d\underline{x}. \quad (3.7)$$

3.5. Modelo matemático de flujo en sistemas multifásicos

Aplicando el análisis axiomático, se presenta el modelo matemático para flujo en sistemas multifásicos en un medio poroso. En este caso se considera la porosidad del medio ϕ , la densidad y la saturación de la fase α , ρ_α y S_α respectivamente. La masa de fluido de la fase α es una propiedad extensiva que se escribe como

$$E = \int_{B(t)} \phi \rho_\alpha S_\alpha d\underline{x},$$

y es la única propiedad extensiva necesaria para describir este sistema. De acuerdo con la formulación axiomática, la correspondiente propiedad intensiva en este caso es

$$\psi_\alpha = \phi \rho_\alpha S_\alpha.$$

Suponiendo que no hay difusión, es decir $\tau = 0$, y aplicando la ecuación (3.4), la ecuación de balance local para la masa de fluido de la fase α se escribe como

$$\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot (\underline{v} \phi \rho_\alpha S_\alpha) = q_\alpha.$$

Tomando en cuenta que la velocidad de Darcy de la fase α se escribe como $\underline{u}_\alpha = \underline{v} \phi S_\alpha$, la ecuación diferencial de balance de masa para la fase α , que forma parte de un sistema multifásico, se escribe como

$$\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \underline{u}_\alpha) = q_\alpha. \quad (3.8)$$

La ley de Darcy para un sistema multifásico se expresa como

$$\underline{u}_\alpha = -\frac{k k_{r\alpha}}{\mu_\alpha} (\nabla p_\alpha - \rho_\alpha \underline{g}), \quad (3.9)$$

donde $k_{r\alpha}$ es la permeabilidad relativa a la fase α . La viscosidad, la presión y la densidad se denotan por μ_α , p_α y ρ_α respectivamente.

Sustituyendo (3.9) en (3.8), y definiendo la movilidad de la fase α como

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}, \quad (3.10)$$

se obtiene

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} - \nabla \cdot (\rho_\alpha \lambda_\alpha \underline{k}(\nabla p_\alpha - \rho_\alpha \underline{g})) = q_\alpha. \quad (3.11)$$

La ecuación (3.11) representa un conjunto de Ecuaciones Diferenciales Parciales (PDE por sus siglas en inglés) de segundo orden y en general no lineales. Estas ecuaciones se completan con una serie de fórmulas constitutivas las cuales describen la dependencia de una variable con respecto de las otras e integran el conocimiento científico y tecnológico del problema bajo estudio, véase [1]. En un sistema multifásico, generalmente se usan las siguientes ecuaciones constitutivas

$$\sum_{\alpha=1}^N S_\alpha = 1 \quad (3.12)$$

$$p_{c\alpha_1\alpha_2} = p_{\alpha_1} - p_{\alpha_2}, \alpha_1 \neq \alpha_2 \quad (3.13)$$

donde N es el número de fases, α_1 y α_2 representan una fase no mojadora (*non-wetting*) y una fase mojadora (*wetting*), respectivamente.

La combinación de las ecuaciones (3.11), (3.12) y (3.13) produce un sistema de PDEs de segundo orden generalmente no lineales y fuertemente acopladas (*fully coupled*). Para tratar este problema, se debe usar una estrategia de linealización. En este trabajo se usa el método IMPES, véase [13].

3.6. Formulación presión-saturación

Para poder aplicar el método IMPES en la solución del sistema de PDEs no lineales y fuertemente acopladas de la sección 3.5, es necesario hacer una formulación donde la ecuación de presión se resuelva de manera aislada y los resultados obtenidos se usen para resolver las ecuaciones de transporte (saturación). Posteriormente, los resultados de la solución de las ecuaciones de saturación, se insertan en la ecuación de presión para obtener la solución en el paso de tiempo siguiente. Este proceso es iterativo y se repite durante un número de pasos de tiempo previamente definidos. En las secciones siguientes se muestra cómo obtener una formulación presión-saturación para un sistema multifásico.

3.6.1. Ecuación de presión

Por medio de operaciones algebraicas, de sumar sobre todas las fases, de desarrollar las derivadas, de usar ecuaciones constitutivas como la (3.12) y de la definición de la ley

de Darcy para flujo multifásico y del uso de una función de flujo fraccional, se obtiene una ecuación para la presión que se escribe como sigue

$$\frac{\partial \phi}{\partial t} - \nabla \cdot \lambda \underline{k} \left[\sum_{\alpha} f_{\alpha} \nabla p_{\alpha} - \sum_{\alpha} f_{\alpha} \rho_{\alpha} \underline{g} \right] + \sum_{\alpha} \frac{1}{\rho_{\alpha}} \left[\phi S_{\alpha} \frac{\partial \rho_{\alpha}}{\partial t} + \underline{u}_{\alpha} \cdot \nabla \rho_{\alpha} \right] - \sum_{\alpha} \frac{q_{\alpha}}{\rho_{\alpha}} = 0. \quad (3.14)$$

En el apéndice A se muestra cómo se llega a la ecuación (3.14).

3.6.2. Ecuación de saturación

La solución de la ecuación (3.14), produce la presión necesaria para obtener la velocidad que se requiere para resolver la ecuación de saturación. La saturación puede calcularse directamente de la ecuación de balance de masa (3.8). Por ejemplo para un sistema de dos fases agua y aceite, en este trabajo se usaría la presión del aceite para generar una ecuación de presión, mientras que la ecuación para la saturación del agua se escribiría como

$$\frac{\partial(\phi \rho_w S_w)}{\partial t} + \nabla \cdot (\rho_w \underline{u}_w) = q_w. \quad (3.15)$$

Las ecuaciones (3.14) y (3.15) se han formulado en términos de una presión de fase (la del aceite por ejemplo). Sin embargo, también es posible hacer una formulación en términos de una presión global. El apéndice B muestra una manera de definir una presión global y la forma que toman las ecuaciones.

3.6.3. Formulación presión-saturación para flujo en dos fases

En esta sección se obtiene un modelo matemático para flujo bifásico en un medio poroso haciendo las siguientes suposiciones:

- Se tiene un sistema de dos fases, agua (w) y aceite (o),
- Las fases se consideran incompresibles,
- La porosidad de la roca es constante.
- No se consideran los efectos de la fuerza de gravedad.

Dado lo anterior y a partir de la ecuación (3.14) se llega a

$$-\nabla \cdot \lambda \underline{k} [f_w \nabla p_w + f_o \nabla p_o] - (Q_w + Q_o) = 0 \quad (3.16)$$

donde $Q_{\alpha} = q_{\alpha}/\rho_{\alpha}$, para $\alpha = w, o$.

En este trabajo se obtendrá una ecuación en términos de la presión del aceite. Entonces usando que la presión capilar $p_c \equiv p_{cow} = p_o - p_w$ para eliminar p_w en la ecuación (3.16) se obtiene

$$-\nabla \cdot (\underline{k} \lambda \nabla p_o) + \nabla \cdot (\underline{k} \lambda_w \nabla p_c) - (Q_w + Q_o) = 0.$$

En muchas ocasiones, la presión capilar depende de la saturación del agua, de tal manera que el gradiente de la presión capilar se puede expresar como

$$\nabla p_c(S_w) = \frac{dp_c}{dS_w} \nabla S_w. \quad (3.17)$$

Usando esta última relación para la presión capilar, se obtiene la siguiente ecuación para la presión del aceite

$$-\nabla \cdot (\underline{k}\lambda \nabla p_o) + \nabla \cdot \left(\underline{k}\lambda_w \frac{dp_c}{dS_w} \nabla S_w \right) = Q_w + Q_o. \quad (3.18)$$

Para obtener una ecuación para la saturación del agua, se parte de la ecuación (3.15) y tomando en cuenta las suposiciones hechas al inicio de esta sección se llega a

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \underline{u}_w = Q_w. \quad (3.19)$$

Ahora, sustituyendo la ley de Darcy para la velocidad de la fase acuosa, la definición de la presión capilar para eliminar p_w y la ecuación (3.17), en la ecuación (3.19) se obtiene

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot (\underline{k}\lambda_w \nabla p_o) + \nabla \cdot \left(\underline{k}\lambda_w \frac{dp_c}{dS_w} \nabla S_w \right) = Q_w. \quad (3.20)$$

3.7. Modelo Buckley-Leverett

En el modelo Buckley-Leverett [14, 15], no se toma en cuenta la presión capilar y tampoco se consideran fuentes ni sumideros. Por lo tanto, el modelo matemático se escribe como sigue:

$$-\nabla \cdot (\underline{k}\lambda \nabla p) = 0 \quad (3.21)$$

y

$$\phi \frac{\partial S}{\partial t} - \nabla \cdot (\underline{k}\lambda_w \nabla p) = 0 \quad (3.22)$$

donde se ha usado la notación $p_o \equiv p$ y $S_w = S$.

3.8. Permeabilidad relativa

La permeabilidad relativa que aparece en las ecuaciones (3.21) y (3.22) se debe aproximar mediante correlaciones o en su caso, se puede obtener esta información mediante tablas de datos experimentales. En este trabajo se usarán las siguientes relaciones

$$K_{rw} = S_{ef}^\sigma \quad y \quad K_{ro} = (1 - S_{ef})^\sigma, \quad \text{con } \sigma = 1$$

donde S_{ef} representa a una saturación efectiva que está dada por

$$S_{ef} = \frac{S - S_{rw}}{1 - S_{rw} - S_{ro}} \quad (3.23)$$

donde S_{rw} es la saturación residual del agua y S_{ro} es la saturación residual del aceite.

4

Modelo numérico

El método IMPES fue originalmente desarrollado por Sheldon *et al.* (1959) y Stone y Gardner (1961) y es ampliamente usado en la industria petrolera. La idea básica de este método clásico para resolver un sistema de PDEs acoplado, correspondiente a un flujo bifásico en un medio poroso, es separar el cálculo de la presión y el de la saturación. Es decir, este sistema acoplado se divide en una ecuación de presión y una de saturación y se resuelven con respecto al tiempo usando un esquema de aproximación implícito para la primera y uno explícito para la segunda [16]. Este método es sencillo de implementar y requiere menor memoria de computadora comparado con otros métodos de solución de sistemas de ecuaciones simultáneas. El algoritmo que lo describe es el siguiente

Algoritmo 1 IMPES

- 1: Definir condiciones iniciales y de frontera.
 - 2: **Mientras** $t < T_{max}$ **hacer**
 - 3: Resolver la ecuación de presión de manera implícita usando valores de la saturación del paso anterior.
 - 4: Resolver la ecuación de saturación de manera explícita usando los valores de la presión calculados en el paso 3.
 - 5: $t \leftarrow t + \Delta t$
 - 6: **Fin Mientras**
-

El algoritmo 1 representa el método IMPES clásico. En este trabajo se hará una adecuación para hacer uso de los métodos SLS y RBF, los cuales nos pueden dar mayor precisión en la solución y velocidad de cálculo.

4.1. Método de Líneas de Corriente SLS

El modelado de yacimientos con base en líneas de corriente ha sido usado en la industria del petróleo desde los años 50's [17]. Recientemente se ha incrementado el interés en estas técnicas debido al desarrollo de nuevas formas de caracterizar los yacimientos [18]. El método fue introducido para simular procesos de desplazamiento en medios heterogéneos por Muskat [19]. Después Higgins y Leighton [20], Martin [21] *et al.*, Lake [22] *et al.* y Thiele [23] *et al.*, extendieron este método y ampliaron su estudio y aplicación. Usarlo en lugar de los métodos FDM, FVM y FEM en la simulación de yacimientos a gran escala, tiene ciertas ventajas como mayor rapidez y menor uso de memoria de cómputo.

Aunque este método ahora es aceptado como una herramienta rápida de simulación, es todavía una tecnología joven y es aplicable solo a cierto rango limitado de problemas. Por el momento, ningún simulador comercial basado en SLS ofrece la habilidad de usar mallas no estructuradas, o coeficientes de tensor completo [24]. A la fecha se sigue investigando SLS en el campo de la modelación de flujo.

Los métodos de líneas de corriente usan conceptos de seguimiento de partículas para definir trayectorias de líneas en un espacio tridimensional [25, 17]. El enfoque es adecuado para modelar flujo y transporte en tres dimensiones [26, 27, 9], y es de gran facilidad debido a la introducción del concepto del tiempo de vuelo (TOF por sus siglas en inglés) como una variable espacial [17]. El tiempo de vuelo es simplemente el tiempo que tarda una partícula neutral en viajar de un punto A a un punto B a lo largo de una línea de corriente. La formulación TOF desacopla los efectos de las heterogeneidades geológicas de los cálculos del transporte. Este desacoplamiento es llevado a cabo por la transformación de las ecuaciones de saturación del espacio físico a las coordenadas TOF de la línea de corriente. Las ecuaciones de saturación multidimensionales se reducen ahora a series de cálculos en una dimensión a lo largo de las líneas de corriente.

4.1.1. Líneas de corriente (Streamlines)

Las líneas de corriente son líneas instantáneas tangentes en cualquier punto al campo de velocidad. Para la aplicación desarrollada en este trabajo, la velocidad total será la velocidad total multifásica de Darcy dividida por la porosidad. No se consideran líneas de corriente para diferentes fases o para diferentes componentes. Si la velocidad varía con el tiempo, entonces se piensa en términos de una velocidad instantánea para definir las líneas de corriente. Una vez que son calculadas, las líneas de corriente definen una discretización espacial del campo de flujo. Esta forma de discretización produce una resolución mayor de líneas de corriente en regiones donde el flujo es más rápido, como sucede análogamente en el refinamiento de la malla local en simulaciones de diferencias finitas. La figura 4.1 muestra un conjunto de líneas de corriente para un campo de velocidades dado.

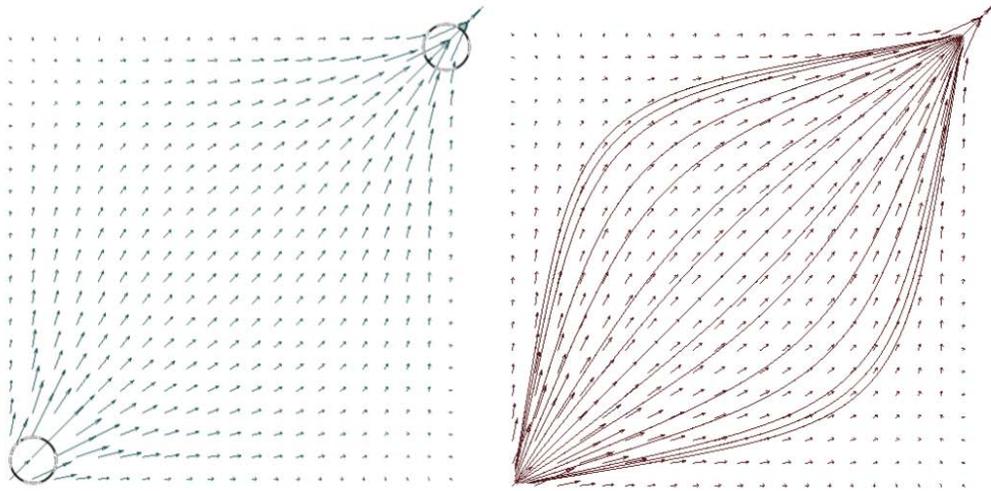


Figura 4.1: A la izquierda el campo de velocidad en un dominio dado. A la derecha, las líneas de corriente tangentes al campo de velocidad local.

4.1.2. Tiempo de vuelo (Time Of Flight)

El TOF es el tiempo de viaje de una partícula neutral a lo largo de una línea de corriente, o la distancia a lo largo de la línea de corriente dividida por la velocidad de la partícula. Debido a que la velocidad de la partícula puede cambiar a lo largo de la línea de corriente, más generalmente está dada por la integral de la distancia sobre la velocidad. El tiempo de vuelo $\tau(x, y, z)$ se usa como coordenada espacial, es decir, la distancia desde la entrada de un sistema se mide por este tiempo, no por la distancia Euclidiana, véase la figura 4.2 . En simulación con líneas de corriente, usar el tiempo de vuelo como coordenada espacial es fundamental.

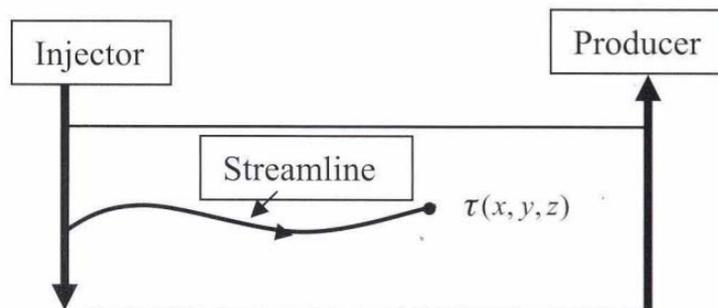


Figura 4.2: Tiempo de vuelo τ para una partícula de prueba.

4.1.3. Simulación con líneas de corriente

La técnica SLS se basa en el enfoque IMPES para resolver las ecuaciones gobernantes de conservación y consta de siete pasos principales:

- ❶ Resolver la ecuación de presión.
 - Inicialmente la ecuación de presión (3.21) es resuelta.
- ❷ Calcular la velocidad total aplicando la ley de Darcy.
 - La velocidad total es entonces calculada usando la ley de Darcy, la cual para flujo multifásico sin considerar la fuerza de gravedad es

$$\underline{u} = \sum_{\alpha} \underline{u}_{\alpha} = \sum_{\alpha} \left[-\underline{k} \lambda_{\alpha} (\nabla p_{\alpha}) \right].$$

- ❸ Trazar las líneas de corriente con base en la velocidad total del fluido.
 - Las líneas de corriente, son trazadas con el método *Forward Euler* [28], con el cual se describe la trayectoria de una partícula dentro del campo de velocidad.
- ❹ Calcular el TOF a lo largo de las líneas de corriente.
 - El TOF a lo largo de una línea de corriente trazada desde una fuente a un sumidero está dado por (véase [17])

$$\tau = \int \frac{\phi(\xi)}{|\underline{u}(\xi)|} d\xi \quad (4.1)$$

donde la partícula de prueba se mueve con una velocidad intersticial \underline{u}/ϕ , y ξ es la distancia espacial a lo largo de la línea de corriente.

La ecuación (4.1) puede ser reescrita como la siguiente relación diferencial

$$\underline{u} \cdot \nabla \tau = \phi \quad (4.2)$$

o de forma más intuitiva

$$\frac{u}{\phi} = \frac{\Delta \xi}{\Delta \tau}. \quad (4.3)$$

Para transformar coordenadas espaciales a coordenadas de TOF, se usan funciones de corriente y el jacobiano que relaciona los elementos en ambos espacios (véase [17]), de tal forma que se llega a la siguiente ecuación

$$\underline{u} \cdot \nabla = \phi \frac{\partial}{\partial \tau}. \quad (4.4)$$

Esta última relación se usará para transformar ecuaciones del espacio físico a coordenadas de TOF.

Tomando en cuenta el flujo fraccional $F_w = \lambda_w/\lambda$ y la velocidad total $\underline{u} = -\underline{k}\lambda(\nabla p)$ entonces (3.22) puede escribirse como

$$\phi \frac{\partial S}{\partial t} - \nabla \cdot (F_w \underline{u}) = 0. \quad (4.5)$$

Aplicando algunas identidades diferenciales se obtiene

$$\phi \frac{\partial S}{\partial t} - (\underline{u} \cdot \nabla F_w + F_w \nabla \cdot \underline{u}) = 0. \quad (4.6)$$

Dado que en este problema se toma en cuenta la condición de no compresibilidad ($\nabla \cdot \underline{u} = 0$), entonces, (4.6) puede escribirse como

$$\phi \frac{\partial S}{\partial t} - \underline{u} \cdot \nabla F_w = 0 \quad (4.7)$$

y de acuerdo con (4.4) se obtiene

$$\frac{\partial S_w}{\partial t} + \frac{\partial F_w}{\partial \tau} = 0, \quad (4.8)$$

que es la ecuación de saturación para flujo inmisible e incompresible en el espacio de coordenadas TOF.

Esta transformación de coordenadas, descompone el flujo del fluido bidimensional en series de ecuaciones 1D para la saturación a lo largo de las líneas de corriente. Esta ecuación es válida para una, dos y tres dimensiones, así como para medios homogéneos y heterogéneos. Lo que se requiere para la implementación es el campo de velocidad y el cálculo de la integral de línea en la ecuación (4.1).

- 5 Resolver las ecuaciones de transporte (saturación y concentración) a lo largo de las líneas de corriente. Los cálculos de transporte se realizan en términos de las coordenadas TOF.
 - Se resuelve la ecuación (4.8).
- 6 Actualizar periódicamente las líneas de corriente.
 - Si hay cambios en las condiciones de movilidad, se actualizan las líneas de corriente. Una vez que las líneas son regeneradas, se repiten los pasos 1, 2, 3, 4 y 5.
- 7 Notar que cada vez que las líneas de corriente son actualizadas, se necesita un mecanismo para mapear las saturaciones del conjunto anterior de líneas de corriente al nuevo.

Para mayores detalles sobre el método véase [17, 9].

4.2. Método de Funciones de Base Radial RBF

En la última década se ha hecho un gran esfuerzo en el desarrollo de los métodos llamados libres de malla para la solución de PDEs multivariadas. La proliferación de un conjunto de problemas que involucran no linealidad, grandes gradientes, grandes deformaciones y movimientos de fronteras tales como simulación de fluidos multifásicos o interacción de sólidos y fluidos entre otros, han sido la principal motivación para el desarrollo de tales métodos. Métodos tradicionales como FDM, FVM y FEM, incluyen la generación de una malla para aproximar la solución. La poligonización del dominio puede consumir grandes cantidades de tiempo y esfuerzo computacional en caso de que los problemas involucren dimensiones altas o variables y geometrías complejas. Cerca de 20 métodos distintos libres de malla han sido desarrollados desde 1977. El Smoothed Particle Hydrodynamics Method (SPH) [29], el Diffusive Element Method (DEM) [30], el Element-Free Galerkin Method (EFGM) [31], el Reproducing Kernel Particle Method (RKPM) [32] y el RBF [33, 34, 35, 6, 7] son algunos de ellos.

En 1982, Frank [35] comparó 29 métodos de interpolación con funciones analíticas de prueba bidimensionales. De acuerdo con ello, uno de los métodos más poderosos es el *multicuádrico diferenciable continuo* MQ-RBF (RBF que usa como función de base radial el multicuádrico), descubierto por Hardy [33, 34]. Ha sido mostrado por varios autores que el MQ-RBF, para el problema de interpolación, posee algunas propiedades muy poderosas. Madych y Nelson [36] probaron que la interpolación con el MQ-RBF es exponencialmente convergente.

En 1990, Kansa [6, 7] modificó el método MQ de Hardy [34] para resolver PDEs. Desde entonces, resolver PDEs usando RBFs ha sido usado para diferentes tipos de aplicaciones. Fedoseyev *et al.* [37] demostró que las soluciones de PDEs elípticas convergen exponencialmente requiriendo menor número de puntos y operaciones que FDM, FVM y FEM. Las principales ventajas del esquema MQ-RBF sobre los métodos tradicionales, es que posee ordenes de convergencia mayor, requiere menos puntos y es fácil de implementar en más de una dimensión. Por otro lado, la principal desventaja de aplicar MQ-RBF a sistemas de PDEs, es que la matriz resultante de coeficientes puede llegar a ser mal condicionada tanto como el rango de la matriz crezca.

El costo de incrementar la precisión en RBF, es generalmente el mal condicionamiento del sistema lineal asociado que necesita ser resuelto: *a mayor condicionamiento, se tiene menor precisión y a peor condicionamiento, mayor precisión*. Se han hecho diferentes propuestas para superar estas dificultades, véase por ejemplo [38, 39, 40, 41, 42, 43, 44, 45, 46].

4.2.1. Conceptos fundamentales

4.2.1.1. Interpolación

El método RBF es una herramienta para interpolar datos dispersos multidimensionales. La facilidad con que permite manejar datos dispersos en un espacio de varias dimensiones, y para proporcionar precisión espectral, lo han hecho particularmente popular en varios tipos de aplicaciones. Algunas de las más recientes son cartografía,

redes neuronales, imágenes médicas y solución numérica de PDEs [47]. En este trabajo se aborda la interpolación y la solución de PDEs.

Un problema común en muchas disciplinas de la ciencia y la ingeniería es la interpolación de datos dispersos. Cuando se trata de datos en una dimensión, existen muchos métodos para resolver el problema. La mayoría de ellos (e.g. las interpolaciones polinomial y de Fourier) tienen base en la misma idea: para un conjunto de n puntos $\{x_j\}_{j=1}^n$ y sus valores correspondientes $\{f_j\}_{j=1}^n$, se elige un conjunto de funciones base $\{\psi_j(x)\}_{j=1}^n$ tal que una combinación lineal de estas funciones satisface las condiciones de interpolación. Para ser más específicos, se busca una función $u(x)$ de la siguiente forma

$$u(x) = \sum_{j=1}^n w_j \psi_j(x) \quad (4.9)$$

donde $u(x_i) = f_i$ para $i = 1, \dots, n$. Las condiciones de interpolación conducen a un sistema lineal de ecuaciones el cual determina los coeficientes de expansión w_j . Para muchas opciones de funciones base $\psi_j(x)$, se garantiza que el sistema lineal sea no-singular, siempre que los puntos $\{x_j\}_{j=1}^n$ sean distintos.

Para datos en más de una dimensión (i.e. x y x_j no son escalares sino vectores $\underline{x}, \underline{x}_j \in R^{d>1}$), el enfoque arriba descrito entonces ya no funciona. Puede mostrarse que para cualquier conjunto de funciones base $\{\psi_j(\underline{x})\}_{j=1}^n$ ($n > 1$) (independiente de los puntos dados), existen conjuntos de distintos puntos $\{\underline{x}_i\}_{i=1}^n$ tales que el sistema lineal de ecuaciones para determinar la expansión de los coeficientes llega a ser singular (i.e. no existe un interpolante de la forma (4.9)). Este resultado es generalmente referido como el teorema de Haar [48].

Este problema de la no-singularidad puede ser evadido siguiendo un enfoque diferente para crear una función de interpolación. En lugar de tomar un conjunto de combinaciones lineales de un conjunto de funciones base que son independientes de los puntos, se toma una combinación lineal de mapeos de una sola función base que es (*radialmente*) simétrica con respecto a su centro. Este enfoque del cual fue pionero R. L. Hardy [33] es conocido como el método RBF.

4.2.1.2. Desarrollo del método multicuádrico

El método RBF es una versión generalizada del método multicuádrico (MQ) desarrollado en 1968 por Hardy [33]. Inicialmente él propuso para un conjunto n distinto de puntos fuente esparcidos $\{x_j\}_{j=1}^n$ y sus valores correspondientes $\{f_j\}_{j=1}^n$ la siguiente función de interpolación

$$u(x) = \sum_{j=1}^n w_j |x - x_j|, \quad (4.10)$$

donde los coeficientes w_j son determinados por colocación, i.e. $u(x_i) = f_i, i = 1, 2, \dots, n$. Geométricamente, esto corresponde a interpolar los datos por una combinación lineal de n mapeos del valor absoluto de la función base $|x|$, donde el vértice de cada función base es centrado en uno de los puntos fuente. La figura 4.3 muestra una función con $n = 13$ puntos usando esta técnica.

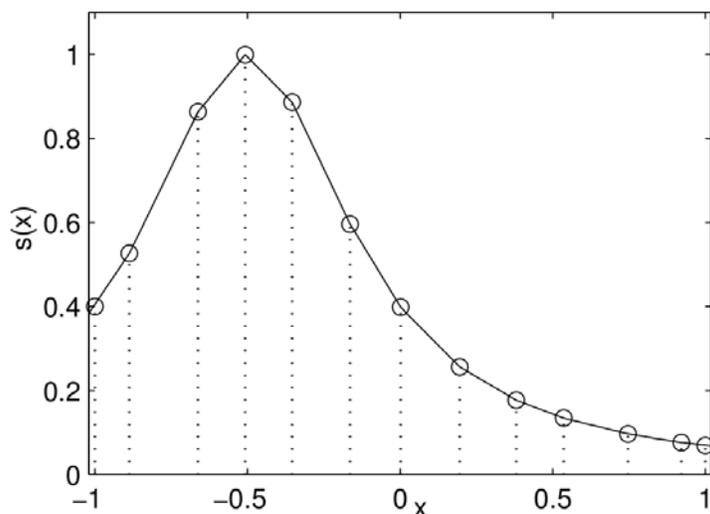


Figura 4.3: Interpolación de algunos datos dispersos usando la ecuación (4.10). Los puntos fuente están marcados con círculos.

El problema con esta representación es que la función tiene un salto en la primera derivada en cada punto. Mientras esto puede modelar precisamente el comportamiento en una fractura, no permite usar las técnicas del cálculo para encontrar máximos y mínimos. Hardy resolvió este problema reemplazando la función base de valor absoluto (4.10) por una continuamente diferenciable. Él propuso usar la función base $\sqrt{c^2 + x^2}$ (i. e. la curva superior de una hipérbola), donde c es alguna constante arbitraria diferente de cero, por su similitud con la función de valor absoluto [49]. La nueva función de interpolación entonces es

$$u(x) = \sum_{j=1}^n w_j \sqrt{c^2 + (x - x_j)^2}, \quad (4.11)$$

donde w_j son nuevamente determinados por colocación. Para $c \neq 0$, (4.11) conduce a un interpolante continuamente diferenciable, y para $c = 0$, (4.11) es equivalente a (4.10) y la misma interpretación aplica para este nuevo método. Aunque, en lugar de la función base del valor absoluto, este método corresponde a tomar una combinación lineal de mapeos de la función base hipérbola. La figura 4.4 muestra una función con los mismos datos de la figura 4.3 usando (4.11) con $c = 2$.

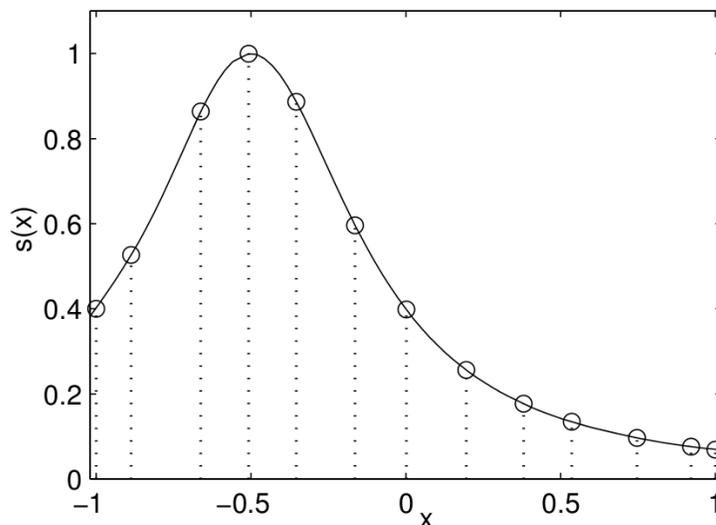


Figura 4.4: Interpolación de algunos datos dispersos usando la ecuación (4.11) con $c = 2$. Los puntos fuente están marcados con círculos.

Hardy encontró que este método no sólo proporciona una representación precisa, sino que las técnicas del cálculo podían ser fácilmente aplicadas a él. Por ejemplo la interpolación de los datos en la figura 4.4 podía ahora ser usada para encontrar los puntos máximos y de inflexión de una función.

La clave principal del nuevo enfoque de Hardy, es que puede aplicarse a más de una dimensión, lo que conduce a la siguiente definición del método MQ general para interpolar datos dispersos multidimensionales:

Definición 1 (Método MQ) Dado un conjunto de n datos distintos (ó puntos fuente) $\{\underline{x}_j\}_{j=1}^n$ en R^d (i.e. $\underline{x}_j = (x_1^{(j)}, x_2^{(j)}, \dots, x_d^{(j)})$), y sus correspondientes valores escalares $\{f_j\}_{j=1}^n$, el interpolante MQ de los datos está dado por

$$u(\underline{x}) = \sum_{j=1}^n w_j \sqrt{c^2 + \|\underline{x} - \underline{x}_j\|^2} \quad (4.12)$$

donde $\|\cdot\|$ es la norma euclidiana. Los coeficientes w_j son determinados por las condiciones de interpolación $u(\underline{x}_i) = f_i, i = 1, \dots, n$ lo cual conduce al siguiente sistema lineal de ecuaciones

$$\begin{pmatrix} A \end{pmatrix} \begin{pmatrix} w \end{pmatrix} = \begin{pmatrix} f \end{pmatrix} \quad (4.13)$$

donde las entradas de A están dadas por $a_{i,j} = \sqrt{c^2 + \|\underline{x}_i - \underline{x}_j\|^2}$

De acuerdo a un número extensivo de experimentos en 1979, Franke proporcionó evidencia empírica de que el método MQ era *no-singular* (y que la matriz A de $n \times n$ en (4.13) satisfacía la relación $(-1)^n \det(A) > 0$) [50]. Pero no es sino hasta 1986

cuando Micchelli [51] prueba la conjetura de Franke. Fundamenta matemáticamente las condiciones suficientes para garantizar la *no-singularidad* del método. Micchelli no solo mostró que el método era incondicionalmente *no-singular*, sino que esa *no-singularidad* podía ser garantizada cuando otras funciones base fueran usadas. Consecuentemente, el método MQ fue reconocido como un ejemplo específico de un método más general. La idea principal detrás de este método general es usar el mapeo de una sola función base $\phi(r)$ que dependa solamente de la distancia euclidiana desde su centro para crear un interpolante multidimensional. Una función con esta sola dependencia es *radialmente* simétrica con respecto a su centro, por tanto estas $\phi(r)$ llegaron a ser conocidas como “*funciones radiales*” ó, como “*funciones de base radial*” y el método general fue nombrado “*método RBF*”.

4.2.2. El método RBF

Con base en el trabajo de tesis de Grady B. Wright [47], se separan dos versiones del método RBF. Uno se denota como el método RBF básico y el otro como el método RBF aumentado.

4.2.2.1. Interpolación con el método RBF básico

Dado un conjunto de n puntos distintos $\{\underline{x}_j\}_{j=1}^n$ y sus valores correspondientes $\{f_j\}_{j=1}^n$, el interpolante RBF básico está dado por

$$u(x) = \sum_{j=1}^n w_j \varphi(\|\underline{x} - \underline{x}_j\|), \quad (4.14)$$

donde $\varphi(r)$, $r \geq 0$, es alguna función radial. Para encontrar los coeficientes w_j escribimos la ecuación (4.14) para todos los puntos fijos donde el valor de $u(\underline{x}_i)$ es conocido, esto es:

$$u(\underline{x}_i) = u_i = \sum_{j=1}^n w_j \varphi_j(r_i), \quad \text{para } i = 1, \dots, n, \quad (4.15)$$

donde $\varphi_j(r_i) = \varphi_j(\|\underline{x}_i - \underline{x}_j\|)$ y $r_i = \|\underline{x}_i - \underline{x}_j\|$ es la distancia entre \underline{x}_i y \underline{x}_j . Esta ecuación puede ser escrita en forma matricial como sigue:

$$\begin{pmatrix} & & & \\ & A & & \\ & & & \\ & & & \end{pmatrix} \begin{pmatrix} w \\ \\ \\ \end{pmatrix} = \begin{pmatrix} f \\ \\ \\ \end{pmatrix} \quad (4.16)$$

donde las entradas de A están dadas por $a_{i,j} = \varphi_j(r_i) = \varphi_j(\|\underline{x}_i - \underline{x}_j\|)$.

Existen varias posibles opciones de RBFs, la tabla 4.1 enlista las más usadas. El parámetro c que aparece en varias de ellas es conocido como el parámetro de forma:

RBF	$\varphi(\mathbf{r})(\mathbf{r} \geq \mathbf{0})$
RBFs continuas	
Multiquadric (MQ)	$\varphi(r, c) = (r^2 + c^2)^{\beta/2}, \beta = 1, 3, 5, \dots, 2n + 1, \dots$
Inverse Multiquadric (IMQ)	$\varphi(r, c) = (r^2 + c^2)^{-\beta/2}, \beta = 1, 3, 5, \dots, 2n + 1, \dots$
Inverse Quadratic (IQ)	$\varphi(r, c) = (r^2 + c^2)^{-1}$
Gaussian (GA)	$\varphi(r, c) = \exp^{-c^2 r^2}$
RBFs discontinuos	
Smooth Splines (SS)	$\varphi(r) = r^\beta, \beta = 1, 3, 5, \dots, 2n + 1, \dots$
Thin-Plate Splines (TPS)	$\varphi(r) = r^\beta \log(r), \beta = 2, 4, 6, \dots, 2n, \dots$

Tabla 4.1: Algunas funciones de base radial comúnmente usadas.

Observar que $\varphi(r)$ conduce a una solución única para las primeras cuatro funciones de la tabla 4.1, pero para las últimas dos, la matriz de coeficientes correspondiente a estas RBFs, puede fácilmente hacerse singular. Afortunadamente Micchelli fue capaz de incluir este tipo de RBFs en su solución agregando algunas restricciones extras que conducen al método RBF aumentado.

4.2.2.2. Interpolación con el método RBF aumentado

Antes de dar una definición para el método RBF aumentado, es necesario hacer una definición:

Definición: Sea $\Pi_m(R^d)$ el espacio de todos los polinomios de grado menor o igual a m . M denota la dimensión de $\Pi_m(R^d)$, entonces, $M = \binom{m+d}{m}$.

Método RBF aumentado: Dado un conjunto de n puntos distintos $\{\underline{x}_j\}_{j=1}^n$ y sus correspondientes valores $\{f_j\}_{j=1}^n$, el interpolante RBF aumentado está dado por

$$u(\underline{x}) = \sum_{j=1}^n w_j \varphi(\|\underline{x} - \underline{x}_j\|) + \sum_{k=1}^M \gamma_k p_k(\underline{x}) \quad \underline{x} \in R^d \quad (4.17)$$

donde $\{p_k(\underline{x})\}_{k=1}^M$ es una base para $\Pi_m(R^d)$ y $\varphi(r), r \geq 0$, es alguna función radial. Para tomar en cuenta las condiciones de los términos polinomiales adicionales, las siguientes restricciones son impuestas:

$$\sum_{j=1}^n w_j p_k(\underline{x}_j) = 0, \quad k = 1, 2, \dots, M. \quad (4.18)$$

La expansión de los coeficientes w_j y γ_k son entonces determinados por las condiciones de interpolación y las restricciones (4.18), lo cual conduce al siguiente sistema lineal simétrico

$$\left(\begin{array}{c|c} A & P \\ \hline P^T & 0 \end{array} \right) \begin{pmatrix} w \\ \gamma \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (4.19)$$

A es la matriz en (4.16) y P es la matriz de $n \times M$ con entradas $p_k(\underline{x}_j)$ para $j = 1, \dots, n$ y $k = 1, \dots, M$.

Como puede observarse, al agregar las restricciones polinomiales al método básico, este método aumentado siempre proporciona las condiciones suficientes para garantizar la solución única.

4.2.2.3. Solución de PDEs con el método RBF básico

Los primeros intentos para resolver PDEs con el método RBF, fueron realizados por Kansa [6, 7] en 1990. Desde entonces, se han hecho varios estudios para investigar la aplicabilidad de este método en aproximar numéricamente la solución de PDEs de diferentes campos de estudio. En contraste con los métodos numéricos estándares como FDM, FVM y FEM, en RBF los operadores diferenciales de una PDE nunca son discretizados, sino que cada uno de ellos es aplicado a las funciones base directamente [46].

Considerar la siguiente forma general de un problema de valor en la frontera:

$$\begin{aligned}\mathcal{L}u(\underline{x}) &= f(\underline{x}) \text{ in } \Omega \in \mathbb{R}^d \\ \mathcal{B}u(\underline{x}) &= h(\underline{x}) \text{ on } \partial\Omega,\end{aligned}\tag{4.20}$$

donde $\underline{x} \in \Omega$, \mathcal{L} y \mathcal{B} son operadores diferenciales arbitrarios en el dominio Ω con frontera $\partial\Omega$, respectivamente.

Asumir que existe un número n total puntos de colocación $X = \{\underline{x}_j\}_{j=1}^n \in \bar{\Omega}$, donde $\bar{\Omega} = \Omega \cup \partial\Omega$ es conocido como la cerradura del dominio. Sea n_I el número de puntos interiores en Ω y n_B el número de puntos en la frontera $\partial\Omega$, de tal forma que $n = n_I + n_B$. Sustituyendo la ecuación (4.15) en la ecuación (4.20), el problema de valor en la frontera puede ser escrito como sigue:

$$\begin{aligned}\sum_{j=1}^n w_j \mathcal{L}\varphi_j(\underline{x}_i) &= f(\underline{x}_i) = f_i, & \text{para } i = 1, \dots, n_I \\ \sum_{j=1}^n w_j \mathcal{B}\varphi_j(\underline{x}_i) &= h(\underline{x}_i) = h_i, & \text{para } i = n_I + 1, \dots, n.\end{aligned}\tag{4.21}$$

Para encontrar los coeficientes w_j es necesario resolver el siguiente sistema lineal de $n \times n$:

$$\left(\begin{array}{ccc|ccc} \mathcal{L}\varphi_1(r_1) & \cdots & \mathcal{L}\varphi_{n_I}(r_1) & \mathcal{L}\varphi_{n_I+1}(r_1) & \cdots & \mathcal{L}\varphi_n(r_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}\varphi_1(r_{n_I}) & \cdots & \mathcal{L}\varphi_{n_I}(r_{n_I}) & \mathcal{L}\varphi_{n_I+1}(r_{n_I}) & \cdots & \mathcal{L}\varphi_n(r_{n_I}) \\ \hline \mathcal{B}\varphi_1(r_{n_I+1}) & \cdots & \mathcal{B}\varphi_{n_I}(r_{n_I+1}) & \mathcal{B}\varphi_{n_I+1}(r_{n_I+1}) & \cdots & \mathcal{B}\varphi_n(r_{n_I+1}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}\varphi_1(r_n) & \cdots & \mathcal{B}\varphi_{n_I}(r_n) & \mathcal{B}\varphi_{n_I+1}(r_n) & \cdots & \mathcal{B}\varphi_n(r_n) \end{array} \right) \begin{pmatrix} w_1 \\ \vdots \\ w_{n_I} \\ w_{n_I+1} \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_{n_I} \\ h_{n_I+1} \\ \vdots \\ h_n \end{pmatrix}\tag{4.22}$$

Definiendo las submatrices L_{11} , L_{12} , B_{21} y B_{22} donde

$$\begin{aligned}
L_{11} & \text{ con los elementos } \mathcal{L}\varphi_j(r_i), \text{ para } i, j = 1, \dots, n_I, \\
L_{12} & \text{ con los elementos } \mathcal{L}\varphi_j(r_i), \text{ para } i = 1, \dots, n_I, j = n_I + 1, \dots, n, \\
B_{21} & \text{ con los elementos } \mathcal{B}\varphi_j(r_i), \text{ para } i = n_I + 1, \dots, n, j = 1, \dots, n_I, \\
B_{22} & \text{ con los elementos } \mathcal{B}\varphi_j(r_i), \text{ para } i = n_I + 1, \dots, n, j = n_I + 1, \dots, n, \\
\Lambda & = [w_1, \dots, w_n]^T, \\
F & = [f_1, \dots, f_{n_I}]^T, \\
H & = [h_{n_I+1}, \dots, h_n]^T \text{ y} \\
G & = \begin{bmatrix} L_{11} & L_{12} \\ B_{21} & B_{22} \end{bmatrix}
\end{aligned}$$

El sistema (4.22) puede escribirse como:

$$\begin{bmatrix} G \end{bmatrix} \begin{bmatrix} \Lambda \end{bmatrix} = \begin{bmatrix} F \\ H \end{bmatrix} \quad (4.23)$$

donde G es conocida como la matriz de Gramm.

La solución del sistema (4.23) proporciona los coeficientes w_j que son requeridos para aproximar u usando (4.14). En principio, con esta técnica es posible encontrar el valor de u en cualquier lugar dentro del dominio de Ω . El método de colocación con RBFs es muy simple y sin necesidad de generar una malla. Puede ser directamente aplicado a una, dos, tres ó mas dimensiones.

4.2.2.4. Solución de PDEs con el método RBF aumentado

El método RBF aumentado agrega restricciones polinomiales para garantizar la solución única de (4.17), si se sustituye esta ecuación en (4.20), resulta

$$\begin{aligned}
\sum_{j=1}^n w_j \mathcal{L}\varphi_j(\underline{x}_i) + \sum_{k=1}^M a_k \mathcal{L}p_k(\underline{x}_i) & = f(\underline{x}_i) = f_i, \quad \text{para } i = 1, \dots, n_I \\
\sum_{j=1}^n w_j \mathcal{B}\varphi_j(\underline{x}_i) + \sum_{k=1}^M a_k \mathcal{B}p_k(\underline{x}_i) & = h(\underline{x}_i) = h_i, \quad \text{para } i = n_I + 1, \dots, n.
\end{aligned} \quad (4.24)$$

El sistema junto con las restricciones polinomiales puede ser escrito en forma matricial como sigue

$$\begin{bmatrix} WL & PL \\ WB & PB \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Lambda \\ A \end{bmatrix} = \begin{bmatrix} F \\ H \\ \underline{0} \end{bmatrix} \quad (4.25)$$

donde $A = [a_1, \dots, a_M]^T$ es el vector de coeficientes para el polinomio, $\mathbf{0}$ es una matriz de $M \times M$ cuyos elementos son cero, $\underline{0}$ es un vector de M elementos igual a cero y

WL	es una matriz de $n_I \times n$ con elementos $\mathcal{L}\varphi_j(r_i)$,	para $i = 1, \dots, n_I$,	$j = 1, \dots, n$,
WB	es una matriz de $n_B \times n$ con elementos $\mathcal{B}\varphi_j(r_i)$,	para $i = n_I + 1, \dots, n$,	$j = 1, \dots, n$,
PL	es una matriz de $n_I \times M$ con elementos $\mathcal{L}p_k(\underline{x}_i)$,	para $i = 1, \dots, n_I$,	$k = 1, \dots, M$,
PB	es una matriz de $n_B \times M$ con elementos $\mathcal{B}p_k(\underline{x}_i)$,	para $i = n_I + 1, \dots, n$,	$k = 1, \dots, M$,
P^T	es una matriz de $M \times N$ con elementos $p_k(\underline{x}_i)$,	para $i = 1, \dots, n$	$k = 1, \dots, M$.

4.3. SLS y RBF para el modelo Buckley-Leverett

Aplicando el algoritmo IMPES presentado al inicio del capítulo con los pasos de la *simulación de líneas de corriente* de la sección 4.1.3 y, resolviendo con el método RBF las ecuaciones planteadas, se obtiene el algoritmo 2 que integra los tres conceptos y produce de manera más rápida y precisa, la simulación del desplazamiento de aceite por medio de agua en un yacimiento petrolero.

Algoritmo 2 SLS-RBF

- 1: Definir condiciones iniciales y de frontera del problema.
 - 2: **Mientras** $t < T_{max}$ **hacer**
 - 3: Resolver la ecuación de la presión implícitamente con RBF, tomando como entrada los valores de saturación del paso anterior.
 - 4: Calcular el campo de velocidad usando la ley de Darcy y la presión previamente calculada. Aquí se usan los coeficientes de RBF obtenidos en el paso 3.
 - 5: Trazar las líneas de corriente usando el método de Euler hacia adelante con la velocidad total del fluido.
 - 6: Calcular el tiempo de vuelo τ a lo largo de las líneas de corriente.
 - 7: Resolver las ecuaciones de transporte (saturación) a lo largo de las líneas de corriente. Estos cálculos se realizan en términos de las coordenadas TOF y se aplica RBF nuevamente.
 - 8: Transformar la solución de la saturación de coordenadas TOF, a coordenadas espaciales.
 - 9: $t \leftarrow t + \Delta t$
 - 10: **Fin Mientras**
-

4.3.1. Solución de la ecuación de la presión

La ecuación de presión a resolver es la (3.21). Definimos la siguiente función de flujo

$$\underline{F} = -\underline{k}\lambda\nabla p = - \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} \lambda \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} = \left(-k_{11}\lambda \frac{\partial p}{\partial x}, -k_{22}\lambda \frac{\partial p}{\partial y} \right). \quad (4.26)$$

Haciendo un poco de álgebra se obtiene

$$\begin{aligned}
\nabla \cdot (-\underline{k}\lambda\nabla p) &= \nabla \cdot (\underline{F}) = 0 \\
&= \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = 0 \\
&= -k_{11} \left[\frac{\partial}{\partial x} \left(\lambda \frac{\partial p}{\partial x} \right) \right] - k_{22} \left[\frac{\partial}{\partial y} \left(\lambda \frac{\partial p}{\partial y} \right) \right] = 0 \\
&= -k_{11} \left[\lambda \frac{\partial^2 p}{\partial x^2} + \frac{\partial p}{\partial x} \frac{\partial \lambda}{\partial x} \right] - k_{22} \left[\lambda \frac{\partial^2 p}{\partial y^2} + \frac{\partial p}{\partial y} \frac{\partial \lambda}{\partial y} \right] = 0 \\
&= -k_{11} \left[\lambda \frac{\partial^2 p}{\partial x^2} + \frac{\partial p}{\partial x} \left(\frac{\partial \lambda}{\partial S} \right) \left(\frac{\partial S}{\partial x} \right) \right] - k_{22} \left[\lambda \frac{\partial^2 p}{\partial y^2} + \frac{\partial p}{\partial y} \left(\frac{\partial \lambda}{\partial S} \right) \left(\frac{\partial S}{\partial y} \right) \right] = 0
\end{aligned} \tag{4.27}$$

Dada la forma de las permeabilidades relativas se obtiene la siguiente fórmula para λ :

$$\lambda = \frac{1}{1 - S_{rw} - S_{ro}} \left[\frac{S - S_{rw}}{\mu_w} + \frac{1 - S_{ro} - S}{\mu_o} \right] \tag{4.28}$$

Por lo tanto su derivada es

$$\frac{\partial \lambda}{\partial S} = \frac{1}{1 - S_{rw} - S_{ro}} \left[\frac{1}{\mu_w} - \frac{1}{\mu_o} \right] = C. \tag{4.29}$$

Dado que todos los parámetros en la ecuación (4.29) son constantes, resulta que esta ecuación es constante y sustituyéndola en la ecuación (4.27) se obtiene

$$\nabla \cdot (-\underline{k}\lambda\nabla p) = -k_{11} \left[\underbrace{\lambda}_{n-1} \underbrace{\frac{\partial^2 p}{\partial x^2}}_n + C \underbrace{\frac{\partial p}{\partial x}}_n \underbrace{\frac{\partial S}{\partial x}}_{n-1} \right] - k_{22} \left[\underbrace{\lambda}_{n-1} \underbrace{\frac{\partial^2 p}{\partial y^2}}_n + C \underbrace{\frac{\partial p}{\partial y}}_n \underbrace{\frac{\partial S}{\partial y}}_{n-1} \right] = 0. \tag{4.30}$$

Para esta última ecuación, la presión se calculará en un tiempo n , mientras que la saturación se supone conocida de un tiempo anterior $n - 1$. Esto permite el cálculo de λ , de $\frac{\partial S(x_i)}{\partial x}$ y de $\frac{\partial S(x_i)}{\partial y}$, y por lo tanto la ecuación se linealiza, teniendo como incógnita únicamente a la presión.

Para aplicar el método RBF a la solución de la PDE (4.30) se realizan los siguientes pasos:

- ❶ Se discretiza el dominio del problema esparciendo puntos en él. Se tiene un número N total de puntos de colocación $X = \{\underline{x}_j\}_{j=1}^N \in \bar{\Omega}$, donde $\bar{\Omega} = \Omega \cup \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3 \cup \partial\Omega_4$. Ω es el conjunto de puntos en el interior del dominio y $\partial\Omega_i$ para $i = 1, 2, 3, 4$, es el conjunto de puntos contenidos en las fronteras 1, 2, 3, 4 respectivamente como se muestra en la figura 4.5.

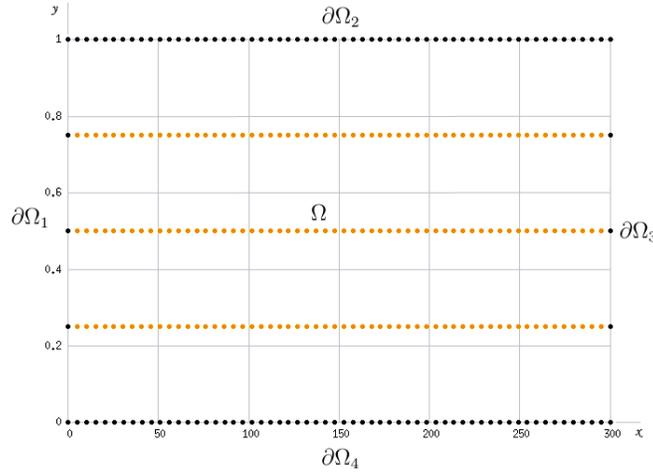


Figura 4.5: Puntos esparcidos en el dominio.

2 Se determina el número de puntos que hay en cada conjunto:

- Sea N_I el número de puntos contenidos en el interior Ω del dominio.
- Sea N_{B_1} el número de puntos contenidos en la frontera $\partial\Omega_1$ del dominio.
- Sea N_{B_2} el número de puntos contenidos en la frontera $\partial\Omega_2$ del dominio.
- Sea N_{B_3} el número de puntos contenidos en la frontera $\partial\Omega_3$ del dominio.
- Sea N_{B_4} el número de puntos contenidos en la frontera $\partial\Omega_4$ del dominio.

3 Se plantea el problema de valor en la frontera

$$\begin{aligned}
 \mathcal{L}u(\underline{x}) &= f(\underline{x}) \text{ in } \Omega \in \mathbb{R}^2 \\
 \mathcal{B}_1u(\underline{x}) &= h_1(\underline{x}) \text{ on } \partial\Omega_1, \\
 \mathcal{B}_2u(\underline{x}) &= h_2(\underline{x}) \text{ on } \partial\Omega_2, \\
 \mathcal{B}_3u(\underline{x}) &= h_3(\underline{x}) \text{ on } \partial\Omega_3, \\
 \mathcal{B}_4u(\underline{x}) &= h_4(\underline{x}) \text{ on } \partial\Omega_4,
 \end{aligned} \tag{4.31}$$

de acuerdo con la ecuación (4.30), se tiene que

$$\mathcal{L} = -k_{11} \left[\lambda \frac{\partial^2}{\partial x^2} + C \frac{\partial}{\partial x} \frac{\partial S}{\partial x} \right] - k_{22} \left[\lambda \frac{\partial^2}{\partial y^2} + C \frac{\partial}{\partial y} \frac{\partial S}{\partial y} \right], \quad u(\underline{x}) = p \quad \text{y} \quad f(\underline{x}) = 0. \tag{4.32}$$

Las condiciones de frontera que se imponen para este problema son:

- Frontera $\partial\Omega_1$.
Se impone una razón de inyección de agua constante $(g_p^{in})_A = 3,4722e - 07m/s$, por tanto se tiene una condición tipo Neumann

$$-\lambda k_{11} \frac{\partial p}{\partial x} = g_p^{in}, \tag{4.33}$$

para esta ecuación

$$\mathcal{B}_1 = -\lambda k_{11} \frac{\partial}{\partial x}, \quad u(\underline{x}) = p \quad y \quad h_1(\underline{x}) = g_p^{in}, \quad (4.34)$$

en el apéndice B se muestra cómo llegar a la ecuación (4.33).

- Frontera $\partial\Omega_2$ y $\partial\Omega_4$.

No hay flujo: $\frac{\partial p}{\partial y} = 0$, por tanto se tienen también condiciones tipo Neumann, entonces

$$\mathcal{B}_2 = \frac{\partial}{\partial y} = 0, \quad u(\underline{x}) = p \quad y \quad h_2(\underline{x}) = 0, \quad (4.35)$$

$$\mathcal{B}_4 = \frac{\partial}{\partial y} = 0, \quad u(\underline{x}) = p \quad y \quad h_4(\underline{x}) = 0, \quad (4.36)$$

- Frontera $\partial\Omega_3$.

Se impone una presión constante $P_o = 1e + 07$ pa, se tiene una condición tipo Dirichlet, entonces

$$\mathcal{B}_3 = I, \quad u(\underline{x}) = p \quad y \quad h_3(\underline{x}) = 1e + 07 \text{ pa}, \quad (4.37)$$

- ④ Se aplican los operadores diferenciales a la función de base radial directamente.

- Aplicando el operador diferencial \mathcal{L} de la ecuación (4.32) se tiene

$$\begin{aligned} & -k_{11} \left[\lambda(S(\underline{x}_i)) \left(\sum_{j=1}^N w_j \frac{\partial^2 \varphi_j(r_i)}{\partial x^2} \right) + C \left(\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x} \right) \left(\frac{\partial S(\underline{x}_i)}{\partial x} \right) \right] + \\ & -k_{22} \left[\lambda(S(\underline{x}_i)) \left(\sum_{j=1}^N w_j \frac{\partial^2 \varphi_j(r_i)}{\partial y^2} \right) + C \left(\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial y} \right) \left(\frac{\partial S(\underline{x}_i)}{\partial y} \right) \right] = 0. \end{aligned} \quad (4.38)$$

Aquí se supone que la saturación S se conoce en cada uno de los puntos dispersos en el dominio en el paso de tiempo anterior, por lo tanto se usa el método RBF para encontrar una ecuación de S que los interpole. Esta ecuación nos permitirá a su vez, conocer su derivada en cada uno de ellos de tal forma que

$$\frac{\partial S(\underline{x}_i)}{\partial x} = \sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial x} \quad y \quad \frac{\partial S(\underline{x}_i)}{\partial y} = \sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial y} \quad \text{para } i = 1, \dots, N, \quad (4.39)$$

al sustituirlas en la ecuación (4.38) se obtiene

$$\begin{aligned} & -k_{11} \left[\lambda(S(x_i)) \left(\sum_{j=1}^N w_j \frac{\partial^2 \varphi_j(r_i)}{\partial x^2} \right) + C \left(\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x} \right) \left(\sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial x} \right) \right] + \\ & -k_{22} \left[\lambda(S(x_i)) \left(\sum_{j=1}^N w_j \frac{\partial^2 \varphi_j(r_i)}{\partial y^2} \right) + C \left(\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial y} \right) \left(\sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial y} \right) \right] = 0, \end{aligned} \quad (4.40)$$

en el apéndice C se muestra como llegar a la ecuación (4.39).

Con operaciones algebraicas, la ecuación (4.40) puede escribirse finalmente como sigue

$$\sum_{j=1}^N w_j \left\{ \begin{aligned} & -k_{11} \left(\lambda(S(x_i)) \frac{\partial^2 \varphi_j(r_i)}{\partial x^2} + C \frac{\partial \varphi_j(r_i)}{\partial x} \sum_{s_j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial x} \right) \\ & -k_{22} \left(\lambda(S(x_i)) \frac{\partial^2 \varphi_j(r_i)}{\partial y^2} + C \frac{\partial \varphi_j(r_i)}{\partial y} \sum_{s_j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial y} \right) \end{aligned} \right\} = 0 \quad \text{para } i = 1, \dots, N_I, \quad (4.41)$$

donde observamos que las incógnitas son los coeficientes w_j los cuales se encontrarán al resolver el sistema lineal.

- Aplicando el operador diferencial \mathcal{B}_1 de la ecuación (4.34) se tiene

$$-k_{11} \left[\lambda(S(x_i)) \sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x} \right] = g_p^{in} \quad \text{para } i = N_I + 1, \dots, N_I + N_{B_1}. \quad (4.42)$$

- Aplicando el operador diferencial \mathcal{B}_2 de la ecuación (4.35) se tiene

$$\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x} = 0 \quad \text{para } i = N_I + N_{B_1} + 1, \dots, N_I + N_{B_1} + N_{B_2}. \quad (4.43)$$

- Aplicando el operador diferencial \mathcal{B}_3 de la ecuación (4.37) se tiene

$$\sum_{j=1}^N w_j \varphi_j(r_i) = 1e+07 \quad \text{para } i = N_I + N_{B_1} + N_{B_2} + 1, \dots, N_I + N_{B_1} + N_{B_2} + N_{B_3}. \quad (4.44)$$

- Aplicando el operador diferencial \mathcal{B}_4 de la ecuación (4.36) se tiene

$$\sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x} = 0 \quad \text{para } i = N_I + N_{B_1} + N_{B_2} + N_{B_3} + 1, \dots, N. \quad (4.45)$$

- ⑤ Se construye el sistema lineal (4.23).

Al expandir las sumatorias del paso 4, se construye el siguiente sistema lineal

$$\begin{pmatrix} \mathcal{L}\varphi_1(r_{\alpha_1}) & \cdots & \mathcal{L}\varphi_N(r_{\alpha_1}) \\ \vdots & \ddots & \vdots \\ \mathcal{L}\varphi_1(r_{\alpha_2}) & \cdots & \mathcal{L}\varphi_N(r_{\alpha_2}) \\ \mathcal{B}_1\varphi_1(r_{\beta_1}) & \cdots & \mathcal{B}_1\varphi_N(r_{\beta_1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_1\varphi_1(r_{\beta_2}) & \cdots & \mathcal{B}_1\varphi_N(r_{\beta_2}) \\ \mathcal{B}_2\varphi_1(r_{\gamma_1}) & \cdots & \mathcal{B}_2\varphi_N(r_{\gamma_1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_2\varphi_1(r_{\gamma_2}) & \cdots & \mathcal{B}_2\varphi_N(r_{\gamma_2}) \\ \mathcal{B}_3\varphi_1(r_{\delta_1}) & \cdots & \mathcal{B}_3\varphi_N(r_{\delta_1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_3\varphi_1(r_{\delta_2}) & \cdots & \mathcal{B}_3\varphi_N(r_{\delta_2}) \\ \mathcal{B}_4\varphi_1(r_{\epsilon_1}) & \cdots & \mathcal{B}_4\varphi_N(r_{\epsilon_1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_4\varphi_1(r_{\epsilon_2}) & \cdots & \mathcal{B}_4\varphi_N(r_{\epsilon_2}) \end{pmatrix} \begin{pmatrix} w_{\alpha_1} \\ \vdots \\ w_{\alpha_2} \\ w_{\beta_1} \\ \vdots \\ w_{\beta_2} \\ w_{\gamma_1} \\ \vdots \\ w_{\gamma_2} \\ w_{\delta_1} \\ \vdots \\ w_{\delta_2} \\ w_{\epsilon_1} \\ \vdots \\ w_{\epsilon_2} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_p^{in} \\ \vdots \\ g_p^{in} \\ 0 \\ \vdots \\ 0 \\ 1e+07 \\ \vdots \\ 1e+07 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.46)$$

donde

$$\begin{array}{ll} \alpha_1 = 1 & y \quad \alpha_2 = N_I, \\ \beta_1 = N_I + 1 & y \quad \beta_2 = N_I + N_{B_1}, \\ \gamma_1 = N_I + N_{B_1} + 1 & y \quad \gamma_2 = N_I + N_{B_1} + N_{B_2}, \\ \delta_1 = N_I + N_{B_1} + N_{B_2} + 1 & y \quad \delta_2 = N_I + N_{B_1} + N_{B_2} + N_{B_3}, \\ \epsilon_1 = N_I + N_{B_1} + N_{B_2} + N_{B_3} + 1 & y \quad \epsilon_2 = N_I + N_{B_1} + N_{B_2} + N_{B_3} + N_{B_4} = N. \end{array}$$

Resolviendo el sistema lineal (4.46), se obtienen los coeficientes de expansión w_j . Con estos valores, es posible entonces conocer el valor de la presión del aceite en cualquiera de los puntos del dominio por medio de su siguiente ecuación de interpolación

$$p(\underline{x}_i) = \sum_{j=1}^N w_j \varphi_j(r_i) \quad \text{para } i = 1, \dots, N. \quad (4.47)$$

Se obtiene así la solución de la ecuación (3.21) con el método RBF. En este caso la función base que se elige es MQ.

4.3.2. Cálculo del campo de velocidad

Se calcula la velocidad para cada punto del dominio donde la presión ya se conoce según la ecuación (4.47).

De la definición de la ley de Darcy para flujo multifásico (véase ecuación A.5) se tiene que

$$\underline{u} = \sum_{\alpha} \underline{u}_{\alpha} = \sum_{\alpha} \left[-\underline{k} \lambda_{\alpha} (\nabla p_{\alpha} - \rho_{\alpha} \underline{g}) \right].$$

El sistema de este problema consiste de dos fases: agua (w) y aceite (o), por lo tanto la velocidad total es:

$$\underline{u} = \underline{u}_w + \underline{u}_o. \quad (4.48)$$

Sustituyendo la ecuación (A.5) en la (4.48) y tomando en cuenta que $g = 0$, se obtiene

$$\underline{u} = -\underline{k}\lambda_w(\nabla p_w) - \underline{k}\lambda_o(\nabla p_o). \quad (4.49)$$

Dado que no se consideran los efectos de la presión capilar: $p_c = p_o - p_w = 0 \Rightarrow p_o = p_w$, por tanto la ecuación (4.49) puede escribirse como

$$\underline{u} = -\underline{k}(\lambda_w + \lambda_o)(\nabla p_o). \quad (4.50)$$

La movilidad total $\lambda = \lambda_w + \lambda_o$, sustituyéndola en la ecuación (4.50) se obtiene

$$\underline{u} = -\underline{k}\lambda(\nabla p) \quad (4.51)$$

donde $p \equiv p_o$.

La ecuación (4.51) puede escribirse de la siguiente forma

$$\underline{u} = -\lambda \begin{pmatrix} k_{11} & 0 \\ 0 & k_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{pmatrix} = -\lambda \begin{pmatrix} k_{11} \frac{\partial p}{\partial x} \\ k_{22} \frac{\partial p}{\partial y} \end{pmatrix}. \quad (4.52)$$

Usando las derivadas del interpolante RBF (véase el apéndice D), y sustituyéndolas en la ecuación (4.52) se obtiene la relación (4.53), que permite conocer la velocidad en cada punto del dominio y por tanto el campo de velocidad.

$$\underline{u}(x_i) = -\lambda \left(k_{11} \sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial x}, \quad k_{22} \sum_{j=1}^N w_j \frac{\partial \varphi_j(r_i)}{\partial y} \right)^T \quad (4.53)$$

4.3.3. Trazado de las líneas de corriente

Se usa el método de *Euler hacia adelante* [28], el cual tiene por objeto obtener una aproximación de un problema bien planteado de valor inicial

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha. \quad (4.54)$$

En la práctica, no se obtendrá una aproximación continua a la solución $y(t)$; por el contrario, se generarán aproximaciones a esa solución en varios valores, llamados **puntos de red**, en el intervalo $[a, b]$. Una vez obtenida la aproximación en los puntos, podemos obtener por interpolación la solución aproximada en otros puntos del intervalo.

El método de Euler construye $\tilde{y}_i \approx y(t_i)$ para cada $i = 1, 2, \dots, N$. Por tanto,

$$\begin{aligned} \tilde{y}_0 &= \alpha, \\ \tilde{y}_{i+1} &= \tilde{y}_i + hf(t_i, y_i), \quad \text{para cada } i = 0, 1, \dots, N-1. \end{aligned} \quad (4.55)$$

Así, si se ponen partículas de prueba en la frontera $\partial\Omega_1$, se puede encontrar la trayectoria que describe cada una en el campo de velocidad.

Se sabe que el tiempo de vuelo τ de una partícula a lo largo de una línea de corriente trazada desde una fuente a un sumidero está dado por la ecuación (4.1)

$$\tau = \int_0^s \frac{\phi(\xi)}{|\underline{u}(\xi)|} d\xi.$$

De esta ecuación se tiene que una partícula de prueba se mueve a la velocidad intersticial de

$$\frac{\underline{u}(\xi)}{\phi(\xi)} = \frac{d\xi}{d\tau} \quad (4.56)$$

donde τ es el tiempo de vuelo y ξ es la distancia espacial a lo largo de la línea de corriente.

Ahora, resolviendo con el método de *Euler hacia adelante* como en (4.55) se tiene

$$\xi(\tau_{i+1}) = \xi(\tau_i) + hf(\tau_i, \xi(\tau_i)) \quad (4.57)$$

donde si $f(\tau_i, \xi(\tau_i)) = \frac{d\xi}{d\tau} = \frac{\underline{u}(\xi)}{\phi(\xi)}$, la ecuación (4.57) puede escribirse como

$$\xi(\tau_{i+1}) = \xi(\tau_i) + h \frac{\underline{u}(\xi)}{\phi(\xi)} \quad (4.58)$$

Tomando en cuenta que el problema se resuelve en dos dimensiones, la velocidad en cada punto tiene una componente en x y otra en y . Entonces, la aproximación de los puntos en que se va moviendo la partícula de prueba con ambas componentes, se obtiene con

$$\xi^{[x]}(\tau_{i+1}) = \xi^{[x]}(\tau_i) + h \frac{u^{[x]}(\xi)}{\phi(\xi)} \quad y \quad \xi^{[y]}(\tau_{i+1}) = \xi^{[y]}(\tau_i) + h \frac{u^{[y]}(\xi)}{\phi(\xi)} \quad (4.59)$$

4.3.4. Cálculo del tiempo de vuelo τ

Una vez trazadas las líneas de corriente con el método de Euler hacia adelante (4.59), se conoce el valor de $\xi(\tau)$, donde τ es el tiempo de vuelo total que necesita la partícula de prueba para viajar de la frontera $\partial\Omega_1$ a la frontera $\partial\Omega_3$

4.3.5. Solución de las ecuaciones de transporte con el método RBF

Resolver las ecuaciones de transporte (saturación) a lo largo de las líneas de corriente, implica resolver la ecuación (4.8)

$$\frac{\partial S_w}{\partial t} + \frac{\partial F_w}{\partial \tau} = 0$$

para cada línea de corriente, donde se ha hecho la transformación de coordenadas espaciales a coordenadas en términos de TOF.

Por la regla de la cadena y haciendo $S \equiv S_w$, la ecuación (4.8) puede escribirse como

$$\frac{\partial S}{\partial t} + \frac{\partial F_w}{\partial S} \frac{\partial S}{\partial \tau} = 0. \quad (4.60)$$

De forma analítica

$$\frac{\partial F_w}{\partial S} = \frac{\mu_w}{\mu_o} \frac{1 - S_{ro} - S_{rw}}{\left(S - S_{rw} + \frac{\mu_w}{\mu_o} (1 - S_{ro} - S) \right)^2} ; \quad (E.5)$$

los detalles de cómo se llega a esta relación, pueden verse en el apéndice E.

Para construir la solución de (4.60) en términos de RBF, la función base MQ y sus derivadas se asumen fijas y los coeficientes varían en el tiempo, esto es $\psi_j = \psi_j(t)$.

- ❶ En este caso el dominio es unidimensional sobre cada línea de corriente. La discretización de este dominio es la partición del intervalo $[0, s]$ en términos del TOF. Se tiene un número n total de puntos de colocación $T = \{\tau_j\}_{j=1}^n \in \overline{\Omega}_\tau$, donde $\overline{\Omega}_\tau = \Omega_\tau \cup \Omega_{\partial\tau_1} \cup \Omega_{\partial\tau_2}$. Las fronteras en este caso son los puntos extremos de la línea de corriente.
- ❷ Se determina el número de puntos que hay en cada conjunto
 - Sea n_I el número de puntos contenidos en el interior Ω_τ del dominio.
 - Sea n_{B_1} el número de puntos contenidos en la frontera $\partial\Omega_{\tau_1}$ del dominio.
 $n_{B_1} = 1$ dado que el dominio del problema es una línea.
 - Sea n_{B_2} el número de puntos contenidos en la frontera $\partial\Omega_{\tau_2}$ del dominio.
 $n_{B_2} = 1$ dado que el dominio del problema es una línea.

- ❸ Se plantea el problema de valor en la frontera

$$\begin{aligned} \mathcal{L}_\tau u(\tau) &= f(\tau) \text{ in } \Omega_\tau \in \mathbb{R} \\ \mathcal{B}_{\tau_1} u(\tau) &= h_1(\tau) \text{ on } \partial\Omega_{\tau_1}, \\ \mathcal{B}_{\tau_2} u(\tau) &= h_2(\tau) \text{ on } \partial\Omega_{\tau_2}, \end{aligned} \quad (4.61)$$

En la ecuación (4.60), sea $S^k \approx S(x, k\Delta t)$, donde se usa un esquema de diferencias de *Euler hacia atrás* [28] para aproximar la derivada temporal:

$$\frac{\partial S}{\partial t} = \frac{S^k - S^{k-1}}{\Delta t}. \quad (4.62)$$

Usando (4.62) en (4.60) se tiene

$$S^k + \Delta t \left[\frac{\partial F_w}{\partial S}(S^{k-1}) \frac{\partial S^k}{\partial \tau} \right] = S^{k-1}, \quad (4.63)$$

para esta ecuación,

$$\mathcal{L}_\tau = I + \Delta t \left[\frac{\partial F_w(S^{k-1})}{\partial S} \frac{\partial}{\partial \tau} \right], \quad u(\tau) = S^k \quad \text{y} \quad f(\tau) = S^{k-1}. \quad (4.64)$$

Las condiciones de frontera que se imponen para este problema son:

- Frontera $\partial\Omega_{\tau_1}$.
En cuanto empieza la saturación de agua en esta frontera, $S = 0.8$ de forma constante, por tanto se tiene una condición tipo Dirichlet y

$$\mathcal{B}_{\tau_1} = I, \quad u(\tau) = S^k \quad \text{y} \quad h_1(\tau) = 0.8 \quad (4.65)$$

- Frontera $\partial\Omega_{\tau_2}$.
Se impone una saturación constante $S = 0$, se tiene una condición tipo Dirichlet, entonces

$$\mathcal{B}_{\tau_2} = I, \quad u(\tau) = S^k \quad \text{y} \quad h_2(\tau) = 0 \quad (4.66)$$

④ Se aplican los operadores diferenciales a la función directamente.

- Aplicando el operador diferencial \mathcal{L}_τ de la ecuación (4.64) se tiene

$$\sum_{j=1}^n \psi_j^k \varphi_j(\tau_i) + \Delta t \left[\frac{\partial F_w(S^{k-1})}{\partial S} \right] \sum_{j=1}^n \psi_j^k \frac{\partial \varphi_j(\tau_i)}{\partial \tau} \quad \text{para } i = 1, \dots, n_I. \quad (4.67)$$

- Aplicando el operador diferencial \mathcal{B}_{τ_1} de la ecuación (4.65) se tiene

$$\sum_{j=1}^n \varphi_j(\tau_i) = 0.8 \quad \text{para } i = n_I + 1, \dots, n_I + n_{B_1}. \quad (4.68)$$

- Aplicando el operador diferencial \mathcal{B}_{τ_2} de la ecuación (4.66) se tiene

$$\sum_{j=1}^n \varphi_j(\tau_i) = 0 \quad \text{para } i = n_I + n_{B_1} + 1, \dots, n. \quad (4.69)$$

- 5 Se construye el sistema lineal (4.23).
Al expandir las sumatorias del paso 4, se construye el siguiente sistema lineal

$$\begin{pmatrix} \mathcal{L}_{\tau_1}\varphi_1(r_1) & \cdots & \mathcal{L}_{\tau_1}\varphi_n(r_1) \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{\tau_1}\varphi_1(r_{n_I}) & \cdots & \mathcal{L}_{\tau_1}\varphi_n(r_{n_I}) \\ \hline \mathcal{B}_{\tau_1}\varphi_1(r_{n_I+1}) & \cdots & \mathcal{B}_{\tau_1}\varphi_n(r_{n_I+1}) \\ \mathcal{B}_{\tau_2}\varphi_1(r_{n_I+2}) & \cdots & \mathcal{B}_{\tau_2}\varphi_n(r_{n_I+2}) \end{pmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{n_I} \\ \psi_{n_I+1} \\ \psi_{n_I+2} \end{pmatrix} = \begin{pmatrix} S^{k-1} \\ \vdots \\ S^{k-1} \\ 0.8 \\ 0 \end{pmatrix} \quad (4.70)$$

Resolviendo el sistema lineal (4.70) se obtienen los coeficientes de expansión ψ_j . Con estos valores, es posible entonces conocer el valor de la saturación en cualquiera de los puntos del dominio por medio de su siguiente ecuación de interpolación

$$S(\tau_i) = \sum_{j=1}^n \psi_j^n \varphi_j(\tau_i) \quad \text{para } i = 1, \dots, n. \quad (4.71)$$

4.3.6. Transformación de la solución de coordenadas TOF a coordenadas espaciales

La solución para la saturación obtenida con la ecuación (4.71), está en el espacio τ . Es necesario transformar el resultado a los valores que le corresponden en coordenadas espaciales. Es decir

$$S(\tau) \longrightarrow S(\underline{x}) \quad (4.72)$$

En la ecuación (4.59), para cada línea $\xi^{[x]}(\tau)$ y $\xi^{[y]}(\tau)$, dan las coordenadas de los puntos que describen el movimiento de las partículas y están en el espacio correspondiente a las coordenadas espaciales (x, y) , entonces de acuerdo con (4.1), (4.72) puede escribirse

$$S(\tau) \longrightarrow S(\underline{\xi}_k) = S\left(\tau \frac{u(\underline{\xi}_k)}{\phi(\underline{\xi}_k)}\right) \quad (4.73)$$

donde $\underline{\xi}(\tau_k) = \underline{\xi}_k$.

De acuerdo con los resultados obtenidos en el cómputo de $\underline{\xi}_k$, su componente $\xi_k^{[y]}$ se mantiene prácticamente constante para cada k ; la que varía incrementalmente es su componente $\xi_k^{[x]}$, debido a que se tiene un desplazamiento horizontal por la partícula de prueba. Sin embargo, $\xi_k^{[x]}$ para todas las k , no coincide con x_i para todas las i donde se ha calculado la presión, aún más, $i \neq k$.

Para poder insertar los valores de la saturación S en las ecuaciones de la presión p en la siguiente iteración, es necesario encontrar $S(x_i)$ donde para cada x_i se está haciendo el cálculo de la presión $p(x_i)$. Ello se logra encontrando una ecuación de S que interpole todos los $S(\underline{\xi}_k)$ de todas las líneas con el método RBF.

Ahora, si

$\{\xi_k\}_{k=1}^\theta$ es el número de θ puntos distintos que describen la trayectoria de una partícula de prueba (línea de corriente), donde se ha calculado la saturación y

$\{\{\xi_k\}_{k=1}^\theta\}_{l=1}^\eta$ es el número total $M = (\eta)(\theta)$ de puntos del conjunto de partículas de prueba (conjunto de líneas de corriente), donde se ha calculado la saturación,

entonces

$$S(\underline{x}_i) = \sum_{j=1}^M \gamma_j \varphi(r_i) \quad \text{para } i = 1, \dots, N. \quad (4.74)$$

Para detalles del método de interpolación RBF, véase el apéndice C.

5

Modelo computacional

Para obtener soluciones numéricas con buena precisión y de manera eficiente, de los modelos matemáticos y numéricos descritos en las secciones anteriores, es necesaria la codificación de un conjunto de algoritmos. Para ello, en este trabajo se hace uso de: el lenguaje C++ aplicando los paradigmas de programación orientada a objetos (POO) y programación genérica(PG); de la biblioteca Eigen para operaciones óptimas entre vectores y matrices; del cómputo paralelo con la programación paralela de memoria compartida por medio de OpenMP, para paralelizar las partes del código que más tiempo consumen; y de Gnuplot, para la graficación de las funciones y los datos.

5.1. Herramientas del modelo computacional

Se eligió el lenguaje C++ porque permite desarrollar programas usando la POO y la PG; aunque no contiene operaciones ni tipos de datos básicos de muy alto nivel, permite definir tipos de datos abstractos de alto nivel de manera que el código expresa clara y concisamente las matemáticas que implementa: se pueden separar conceptos en clases y módulos, lo cual promueve la reutilización del código; permite obtener eficiencia de ejecución y de uso de memoria en cada uno de sus estilos de programación; además de que trabaja en conjunto con las directivas de compilación de la biblioteca de funciones de OpenMP[52, 53, 54].

En este trabajo se usó Eigen [55] que es una biblioteca genérica para álgebra lineal. Permite la operación con matrices y vectores, solución de sistemas lineales y algoritmos relacionados. Soporta desde matrices pequeñas hasta matrices densas y matrices dispersas; incluye tipos de datos estándares numéricos, complejos y tipos numéricos personalizados.

5.1.1. Cómputo paralelo

Remontándose a la década de los 40's, las computadoras secuenciales tradicionales se basan en el modelo introducido por John von Neumann, el cual consiste de una unidad central de procesamiento (CPU por sus siglas en inglés), una memoria principal para

almacenar información, un camino (BUS) sobre el cual fluyen los datos y un mecanismo de sincronización (reloj).

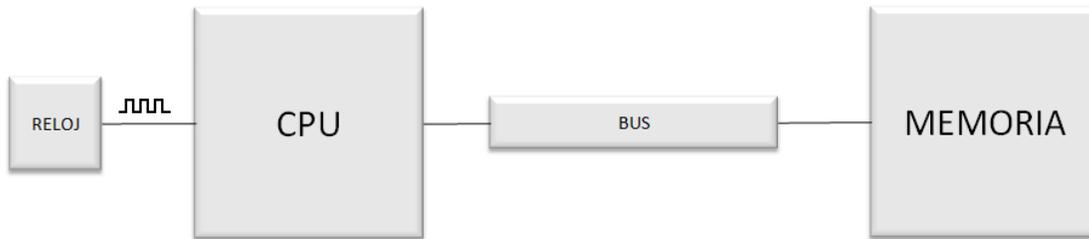


Figura 5.1: Arquitectura de John von Neumann

El número de ciclos de reloj por segundo medidos en Hertz (Hz), caracteriza la velocidad a la cual se ejecutan las instrucciones en la computadora. La velocidad en este tipo de arquitectura, está limitada por dos factores:

- razón de ejecución de las instrucciones (ciclo de reloj) y
- la velocidad a la cual se intercambia la información entre la memoria y el CPU (ancho de banda del bus de memoria),

En el primer factor, la frecuencia no puede incrementarse indefinidamente: el consumo de energía de un chip está dado por la ecuación $P = C_L * V_{dd}^2 * f$ donde P es la potencia, C_L el cambio de capacitancia por ciclo de reloj (proporcional al número de transistores cuyas entradas cambian), V_{dd} es la tensión de alimentación y f es la frecuencia del reloj. Un aumento en la frecuencia aumenta la cantidad de energía utilizada en un procesador [56]; aún si no se considera el consumo de energía, la máxima velocidad a la cual puede viajar la información a lo largo de las conexiones entre los componentes, es a la velocidad de la luz ($3 \times 10^8 m/s$), en este aspecto, reducir el tiempo de ejecución implicaría acortar la longitud de las conexiones, las cuales también tienen un límite.

En el segundo factor, la velocidad está limitada por el ancho de banda del bus de datos. Para mejorarla existen varias técnicas como incrementar el número de canales sobre los cuales el CPU accede a la memoria (*Memory interleaving*), incrementar la razón de intercambio de información mediante el uso de una memoria relativamente pequeña y muy rápida (*memoria caché*) y el uso de segmentación encauzada (*pipeline*), donde múltiples instrucciones pueden ejecutarse simultáneamente, no significando esto, que se reduzca el tiempo que tarda una instrucción en ejecutarse, sino solamente incrementando el número de instrucciones que se ejecutan al mismo tiempo[57].

Todas estas técnicas tienen limitaciones físicas y económicas. Una manera alterna de incrementar la razón de ejecución de instrucciones, es usando múltiples procesadores y unidades de memoria conectadas a ellos de alguna manera, es decir, una computadora paralela.

Una computadora paralela es un sistema de cómputo multiprocesador que soporta la programación paralela. De ella puede hacerse una clasificación en dos categorías importantes: multicomputadoras y multiprocesadores centralizados [54]. Como su nombre lo implica, una multicomputadora es una computadora paralela construída de múltiples computadoras y una red de interconexión. Los procesadores en diferentes computadoras interactúan pasándose mensajes entre ellos; esta clasificación es comúnmente conocida como *computadora paralela de memoria distribuida*. En contraste, un multiprocesador centralizado, también llamado multiprocesador simétrico (SMP por sus siglas en inglés), es un sistema más altamente integrado, en el cual todos los CPUs comparten el espacio físico de la memoria global, el que a la vez, es su medio de comunicación. De ahí que a esta clasificación se le conoce como *computadora paralela de memoria compartida* [54].

5.1.2. Computadora paralela de memoria compartida

Un multiprocesador centralizado es una extensión del uniprocador de John von Neumann. CPUs adicionales se adjuntan al bus y todos los procesadores comparten la misma memoria principal (Figura 5.2). Esta arquitectura es también llamada SMP, porque toda la memoria está en un solo lugar y tiene el mismo tiempo de acceso para cada procesador. Los **datos privados**, son aquellos que sólo un procesador usa, mientras que los **datos compartidos** son aquellos que múltiples procesadores usan. En un procesador centralizado, los procesadores se comunican entre ellos a través de los datos compartidos. Los diseñadores de este tipo de arquitectura, deben tener en cuenta dos problemas asociados a los datos compartidos: *cache-coherency* y la *sincronización* [54], donde en el primero, al haber agregado la memoria caché para reducir la contención entre los procesadores para los datos compartidos y dado que el caché es una vista de la memoria principal, deben asegurarse que los diferentes procesadores no tengan valores distintos de la misma localidad de memoria principal; en el segundo problema, deben asegurar la exclusión mutua: un solo proceso excluye temporalmente a todos los demás para usar la(s) localidad(es) de memoria compartida de forma que garantice la integridad de la actividad específica.

Para programar la computadora de memoria compartida, se usa OpenMP, la cual es una interfaz de programación de aplicaciones (API por sus siglas en inglés) que consta de un conjunto de directivas de compilador.

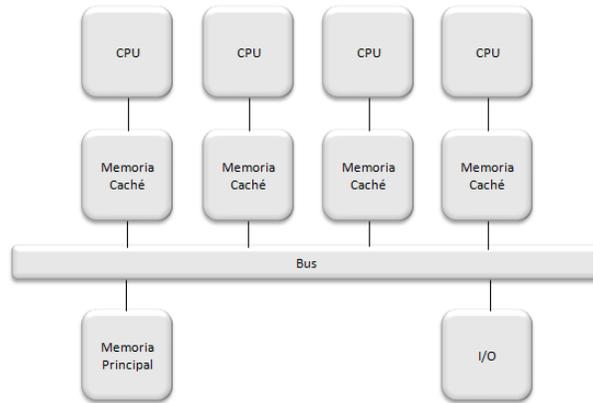


Figura 5.2: Arquitectura de un multiprocesador centralizado genérico.

5.1.3. Computadora paralela de memoria distribuida

En este tipo de arquitectura, cada procesador tiene su propia memoria local, lo que conlleva las siguientes ventajas: que no exista bus de memoria compartida (evita problemas de ancho de banda y contención); que no haya límite para el número de procesadores (depende de la red de interconexión) y que no haya problemas de *cache-coherency*. Sin embargo, se tiene la desventaja de que las tareas que se ejecutan en cada procesador solo operan sobre datos locales, por lo que si se requieren datos remotos, se debe realizar una comunicación con otros procesadores, esto implica, mayor tiempo de consumo para construir y enviar un mensaje, así como para recibirlo y desempaquetarlo. La arquitectura de una computadora de memoria distribuida aparece en la figura 5.3. Si la colección distribuida de memorias forma un espacio lógico de direcciones, la computadora paralela es llamada un multiprocesador distribuido, pero si los espacios locales de direcciones son disjuntos, es llamada una multicomputadora.

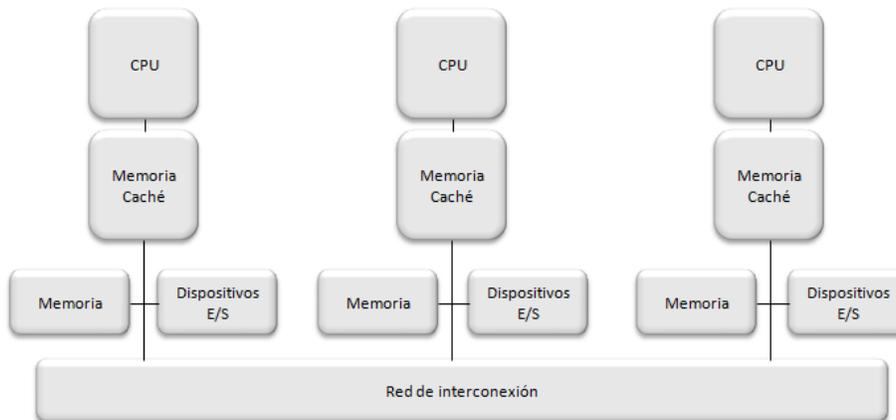


Figura 5.3: Arquitectura de un computador multiprocesador de memoria distribuida.

Una forma perfectamente satisfactoria para comunicar a los procesadores que se encuentran en una computadora paralela de memoria distribuida, es la interfaz de paso de mensajes (MPI por sus siglas en inglés), la cual es una especificación estándar que comprende un conjunto de rutinas para el manejo de procesos e intercambio de mensajes.

5.2. Algoritmo computacional

El algoritmo computacional que se ha construido para poder llevar a cabo el algoritmo SLS-RBF de la sección 4.3, se describe a continuación con un diagrama de flujo (figura 5.4). El objetivo es mostrar las abstracciones esenciales con las que se ha programado la computadora para resolver el problema de flujo bifásico en un medio poroso, y no alejar al lector de un panorama claro de la solución. Si él desea a detalle, revisar el código en lenguaje C++ del algoritmo, puede hacerlo en

<https://drive.google.com/folderview?id=0B-2gJ745cSu7M2NnRTA3dEJEM00&usp=sharing>

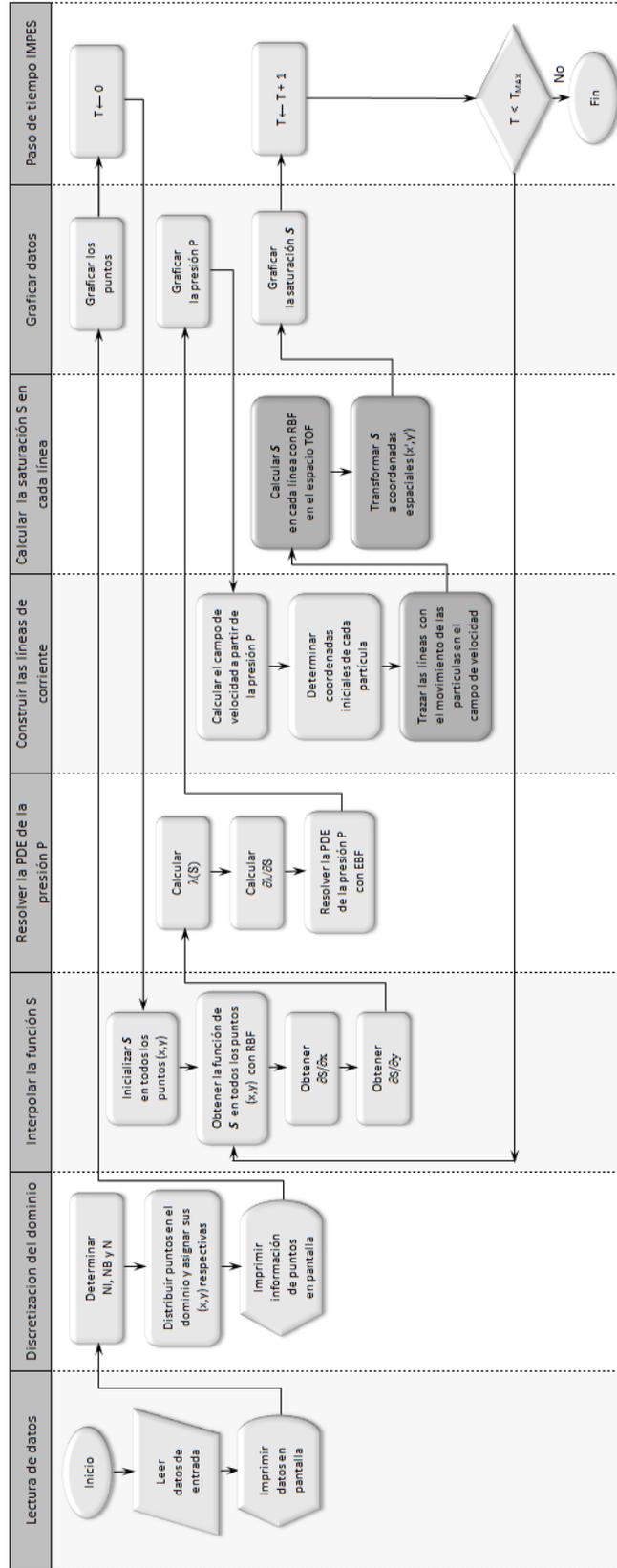


Figura 5.4: Algoritmo computacional

Lectura de datos

Se creó una clase llamada *InputData*, la cual lee los datos introducidos por el usuario en un archivo de texto y los imprime en pantalla. Los datos que el usuario del sistema introduce, son los mostrados en la figura 5.5.

```

16
17 class InputData
18 {
19     private:
20
21         double hx;           // Length in X axis.
22         double hy;           // Length in Y axis.
23
24         int Nx;              // Number of boundary points in X axis.
25         int Ny;              // Number of boundary points in Y axis.
26
27         int rtype;           // Random's type: 0-Uniform, 1-Random, 2-Random "uniform".
28         int layer;           // Randomness of the interior points.
29         int beta;            // The power of the multiquadric.
30         int fsol;            // Solver: 1 Gauss, 2 GMRES, 3 GMRES-Jacobi, 4 GMRES-ACBF, 5-coLPivHouseholderQr(Eigen)
31         double ep;           // Randomness of the interior points (0-1.0)
32         double c;            // Shape parameter
33         double cl;           // Setting the shape parameter value for computing the saturation in the line with the RBF method.
34         double k11;           // Permeability in x-direction
35         double k22;           // Permeability in y-direction
36         double Srw;          // Residual water saturation
37         double Sro;          // Residual oil saturation
38         double Mw;           // Viscosity of water
39         double Mo;           // Viscosity of oil
40         double p_t0;         // P(t0), pressure initial condition
41         double s_t0;         // S(t0), saturation initial condition
42         double gp_in;        // Constant water injection at face 1, boundary condition
43         double p_out;        // P(t,X), Constant pressure at face 3, boundary condition
44         double s_in;         // S(t,X), Constant saturation at face 1, boundary condition
45         double s_out;        // S(t,X), Constant saturation at face 3, boundary condition
46         double phi;          // Porosity
47
48         int Nx_d;            // Number of desired boundary points to interpolate in X axis.
49         int Ny_d;            // Number of desired boundary points to interpolate in Y axis.
50         int rtype_d;         // Random's type of the points to interpolate: 0-Uniform, 1-Random, 2-Random "uniform".
51
52         int nParticles;      // Number of particles that will draw the pathlines
53         double xIniParticle; // The x coordinate of all particles
54         int typeLineDistribution; // Type line distribution: 0 - Uniform, 1 - Random
55         double ymargin;      // The margin y's value
56         int TMaxLine;        // Period of duration of the line
57         double dt;           // Delta tau value
58
59         double DT;           // Delta t value for computing the saturation in the IMPES method.
60         double TMax;         // TMax limit for computing the saturation in the IMPES method.
61

```

Figura 5.5: Datos introducidos por el usuario.

Discretización del dominio

Se creó una clase llamada *BoxDots*, la cual calcula el número total N de puntos en el dominio, el número NI de puntos en el interior del dominio y el número NB de puntos en las fronteras del dominio. Este cálculo se hace a partir del número de puntos Nx y Ny que el usuario proporciona al programa (figura 5.6). Otra funcionalidad que esta clase tiene, es la de distribuir los puntos en todo el dominio de forma uniforme ó aleatoria.

```

//-----
//--The domain's point generation (foundation points)

BoxDots points(inputData); // Create the number of points: NB, NI, N
points.constructDots(); // Distribute the points and assign x-y coordinates to them
points.print(); // Print the dots information in the screen
points.writeToFilexy("xy.dat"); // Write to a file the xy-coordinates of the points
points.writeToFilexy_xnyn("xyxnyn.dat"); // Write to a file the xy-xnyn-coordinates of the points
Plot::plotData("xy.dat"); // Plot the constructed points.

```

Figura 5.6: Creación del objeto `points` de la clase `BoxDots`, para construir los puntos en los conjuntos NB, NI y obtener el total N de ellos, distribuyéndolos de manera uniforme o aleatoria según el usuario indique en los datos de entrada.

Interpolación de la función de saturación S

Con la clase `FunctionTools`, se asigna el valor inicial de la saturación S en cada uno de los puntos del dominio; con la clase `RBFInterpolator` se encuentra la función de la saturación S usando el método RBF, el cual consiste en: 1) expresar matricialmente la expansión de la sumatoria de la ecuación (C.1) y en conjunto con la inicialización de la saturación ya realizada en cada punto del dominio, 2) resolver el sistema lineal de ecuaciones (C.2). Con la función de saturación S encontrada, se obtienen $\frac{\partial S}{\partial x}$ y $\frac{\partial S}{\partial y}$.

```

//-----
//--Interpolate the saturation equation with RBF through the system [Gs][Ws]=[S] (foundation points)

int N = points.get_N(); // Get the total points
Mat intervals = points.get_intervals(); // Get the matrix that stores the each edge's intervals of the domain
Vec x(N), y(N), S(N); // Get the x, y coordinates and the value of the function S, of each point
x = points.getCoordinatePoints(X);
y = points.getCoordinatePoints(Y);
S = FunctionTools::initialSaturation(N, intervals); // Initializing the saturation array.

Vec dS_dx(N), dS_dy(N), Ws(N); // The saturation's derivative vector with respect to x and y.
RBFInterpolator saturation(inputData, N, x, y, S, intervals); // Interpolate the saturation function
dS_dx = saturation.get_dF_dx(x,y); // Get the 1st derivative of the saturation function with respect to x
dS_dy = saturation.get_dF_dy(x,y); // Get the 1st derivative of the saturation function with respect to y
Ws = saturation.get_W(); // Get the saturation's expansion coefficients

```

Figura 5.7: Interpolación de la saturación.

Solución de la PDE de la presión p

Se usan los métodos `lambda` y `dlambda_dS` de la clase `FunctionTools` para calcular $\lambda(S)$ y $\frac{\partial \lambda}{\partial S}$ respectivamente. Luego se resuelve la ecuación de la presión (4.41), la cual se ha expresado en términos del método RBF.

En este paso del algoritmo, se presenta la dificultad de tener el valor adecuado del parámetro de forma c . Encontrar c para que la solución que se obtiene con el método RBF refleje la realidad del fenómeno, es uno de los retos que en la actualidad se tienen. Este parámetro debe variarse hasta que su valor contribuya a una buena solución.

Construir las líneas de corriente

Se crea el objeto `pathline` de la clase `PathLine`, donde con el método `constructIniCoord`, se construyen las coordenadas iniciales de las partículas de prueba (el número de ellas,

las proporciona el usuario en los datos de entrada).

Debido a que hay una independencia de datos entre las líneas de corriente y a que, es la parte de cómputo que más tiempo se consume para la solución (el cálculo de la presión consume un tiempo CPU de 0.1 seg; el cálculo de la saturación consume 3.51 seg), en este paso inicia la codificación paralela con OpenMP. Se distribuye entre el número de hilos indicado por el usuario en el comando de ejecución, el cálculo de las coordenadas iniciales de las partículas de prueba (véase la figura 5.8).

```

=====
// Construct the initial coordinates of each particle
void constructIniCoord()
{
    x0y0 = new XY[nParticles];           //Array that has the initial coordinates of each particle
    if (typeLineDistribution==0)         //If the line distribution is uniform
    {
        #pragma omp parallel for
        for(int i=0; i<nParticles; ++i)
        {
            //std::cout << "iteracion " << i << ", hilo " << omp_get_thread_num() << "\n";
            x0y0[i].x_val = xIniParticle;
            x0y0[i].y_val = (hy * (i+1)) / (nParticles + 1);
        }
    }
}

```

Figura 5.8: Construcción de las coordenadas iniciales de las partículas de prueba.

Con la misma filosofía de codificación paralela, usando el método *Forward Euler*, se trazan las trayectorias de las líneas que pertenecen al movimiento de las partículas en el campo de velocidad. En este cálculo, se insertan los valores de las velocidades de cada punto obtenidas con la solución de la ecuación (4.52) y se generan las coordenadas del espacio TOF (véase la figura 5.9).

Calcular la saturación S en cada línea

Continuando la codificación paralela, se usa el método *computeSaturation* de la misma clase *PathLine* para obtener la solución de la saturación en cada línea. Se incrementa el paso de tiempo IMPES ΔT y se repite el proceso hasta alcanzar el tiempo límite T_{MAX} definido por el usuario, véase la figura 5.10.

```

#pragma omp for
for(int i=0; i<nParticles; ++i)
{
    tmp1[0].x_val = x0y0[i].x_val;           // The origin of the line
    tmp1[0].y_val = x0y0[i].y_val;
    tSteps[i]=0;
    accTime = 0;                             // Accumulated time
    computedX = 0;                           // Computed variable x

    while (accTime <= TMaxLine && computedX <= hx)
    {
        // These actions expand the size of the array every time
        tmp2 = new XY[tSteps[i]+2];         // Create a new bigger array
        for(int k=0; k<tSteps[i]+1; ++k)     // Copy values to new array
            tmp2[k] = tmp1[k];
        delete [] tmp1;                     // Free old array memory
        tmp1 = tmp2;                         // Now tmp1 points to new array

        // Forward Euler's Method

        double sat = 0;                     // Compute the velocity in one point
        for(int a=0; a<N; ++a) (....)
        double lambdaInOnePoint = (1/(1-Srw-Sro)) * ((sat-Srw) / Mw + (1 - Sro - sat) / Mo);

        velocity.x_val = 0;
        velocity.y_val = 0;

        for(int a=0; a<N; ++a) (....)

        tmp1[tSteps[i]+1].x_val = dt * (1/phi) * velocity.x_val + tmp1[tSteps[i]].x_val;
        tmp1[tSteps[i]+1].y_val = dt * (1/phi) * velocity.y_val + tmp1[tSteps[i]].y_val;
        computedX = tmp1[tSteps[i]+1].x_val;
        tSteps[i]++;
        accTime += dt;
    }

    xyLine[i] = new XY[tSteps[i]+1];       // Sizing the number of columns (steps) for the i line but
    // near the left boundary, one assigns the good Y value fo
    // So, the number of steps for an i line, will be tSte

    xyLine[i][0].x_val = 0;
    xyLine[i][0].y_val = tmp1[0].y_val;
    tSteps[i]++;
}

```

Figura 5.9: Trazo de las trayectorias de las líneas de corriente usando el método *Forward Euler*.

```

#pragma omp parallel for
for(int nLine=0; nLine<nParticles; ++nLine)
{
    SIL[nLine].saturation.resize(tSteps[nLine]); // Resizing the matrices
    SIL[nLine].psi.resize(tSteps[nLine]);
    SIL[nLine].G.resize(tSteps[nLine],tSteps[nLine]);

    for(int i=0; i<tSteps[nLine]; ++i) // Initializing the matrix saturation with the values at t = 0 (....)
    {
        SIL[nLine].psi.setZero(); // Setting the psi vector to zero.
        SIL[nLine].G.setZero(); // Setting the G matrix to zero.

        double t = 0.0; // Setting the t value to zero.
        while(t < TMax) // the period while the computation is performed
        {
            //-----
            // Compute the G Matrix
            int n = tSteps[nLine]; // Number of interior points
            int ni = n-2;

            for(int a=0; a<ni; ++a) // Colocation of interior points in omega domain (....)
                for(int b=0; b<n; ++b)

            int a = ni; // Colocation of the boundary points in domega_1 domain (....)
            for(int b=0; b<n; ++b)

            a = ni+1; // Colocation of the boundary points in domega_2 domain (....)
            for(int b=0; b<n; ++b)

            //-----
            // Compute the Psi vector
            SIL[nLine].psi = SIL[nLine].G.fullPivHouseholderQr().solve(SIL[nLine].saturation);

            //-----
            // Compute the saturation vector
            SIL[nLine].saturation.setZero();
            for(int a=0; a<n; ++a)
                for(int b=0; b<n; ++b) (....)
            SIL[nLine].saturation(n-2)=0.8;
            SIL[nLine].saturation(n-1)=0;

            t += DT; // Increasing the t value
        }
    }

    #pragma omp parallel for // Order the values of SIL (....)
    for(int nLine=0; nLine<nParticles; ++nLine)

    #pragma omp parallel for (....)
    for(int nLine=0; nLine<nParticles; ++nLine)

```

Figura 5.10: Cálculo de la saturación en cada línea.

6

Resultados

6.1. Discretización del espacio

Para la presión, el dominio de estudio es un rectángulo con longitud $L_x = 300$ m y altura $L_y = 1$ m. La discretización se hace uniforme. Esta distribución consiste en colocar los puntos en las coordenadas formadas por segmentos en x con longitud igual a L_x/N_x y por segmentos en y con longitud igual a L_y/N_y , donde N_x es el número de puntos en L_x , y N_y es el número de puntos en L_y .

La figura 6.1 muestra la distribución de puntos usada.

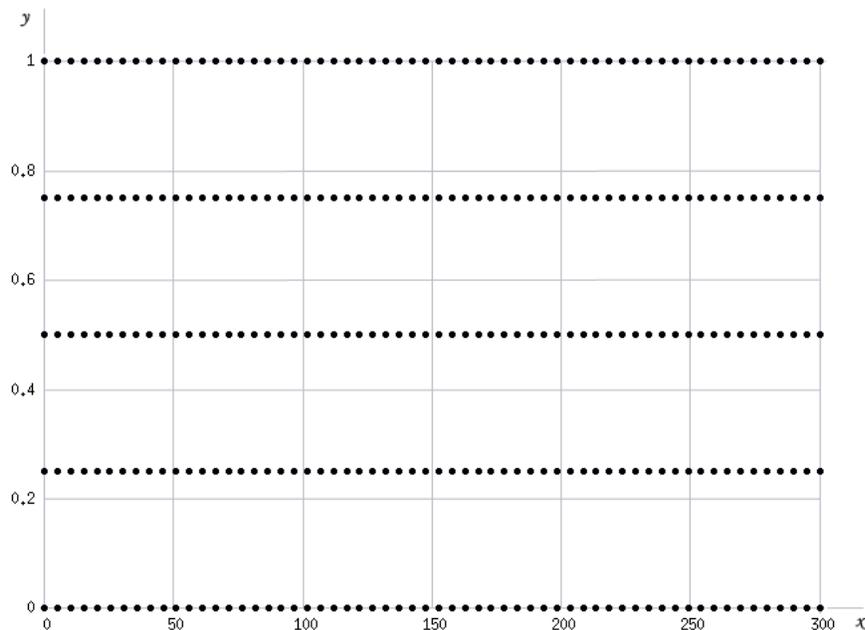


Figura 6.1: Discretización del dominio

Para la saturación, el dominio son las líneas de corriente que inician en $x = 0$ y terminan en $x = L_x$. La discretización se hace esparciendo puntos en ellas de manera uniforme, esto es, en cada una de ellas se colocan puntos cada $\Delta x = L_x/N_L$, donde N_L es el número de puntos sobre cada línea de corriente.

6.2. Solución del problema usando FVM

En el trabajo de De la Cruz [58] puede verse detalladamente cómo para la solución del problema usó el algoritmo IMPES y aplicó el FVM en la solución de las PDEs de la presión y la saturación.

En este trabajo, haciendo uso de [58], se discretizó el espacio con 2500 volúmenes de control, distribuyendo 500 a lo largo por 5 de ancho y se hizo una simulación para 300 días. Las funciones de presión y saturación obtenidas son las que se muestran en las figuras 6.2 y 6.3.

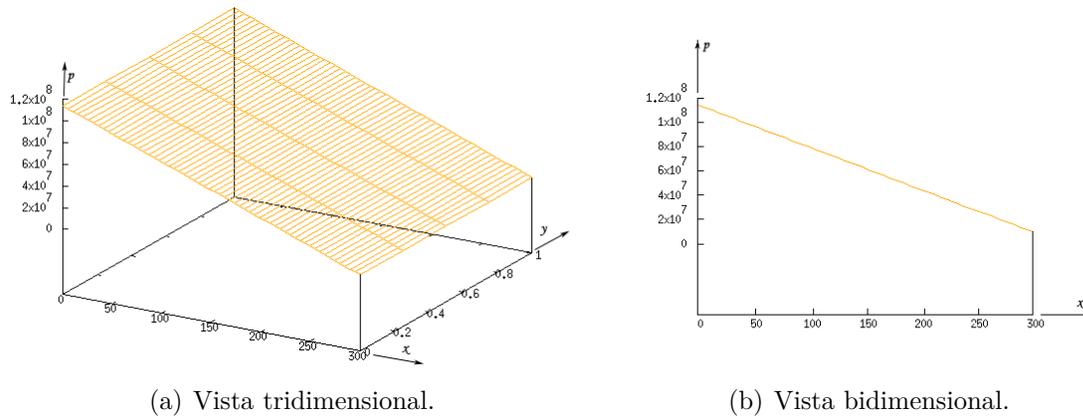


Figura 6.2: Función de la presión obtenida con FVM.

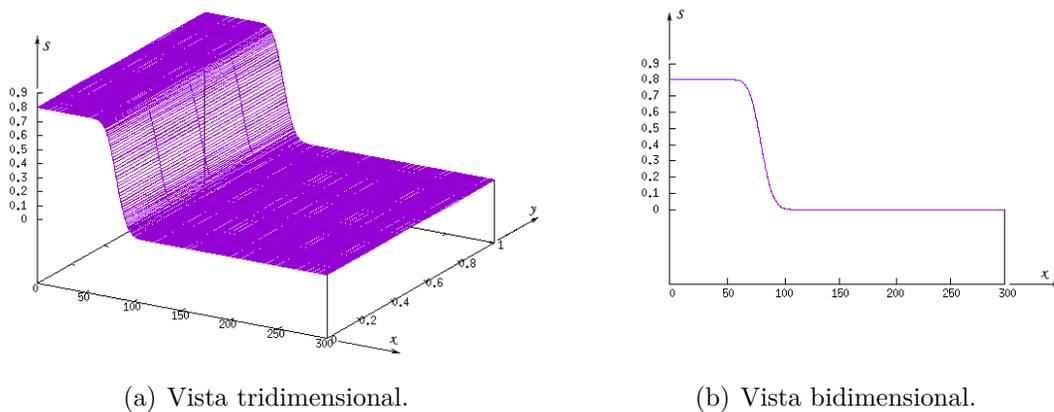


Figura 6.3: Función de la saturación obtenida con FVM.

Presión

Para cada incremento de tiempo en el ciclo IMPES, se observó que la función de la presión converge a la superficie mostrada en la figura 6.2. Este fenómeno pudo observarse evaluando el error de la presión con la siguiente fórmula

$$\|E^p\| = \frac{1}{N} \sum_i^N |p_i^{T+1} - p_i^T| \quad (6.1)$$

donde E^p es el error de la presión, N es el número de puntos en los que se evalúa la presión y T es el número de iteración que corresponde a cada incremento de tiempo en el ciclo IMPES.

Si $T \rightarrow \infty$, $\|E^p\| \rightarrow 0$. Este hecho permite excluir el cálculo de la presión en cada iteración del ciclo IMPES, una vez encontrada la función de presión a la cual converge.

El comportamiento del error de la presión puede visualizarse en la figura 6.4.

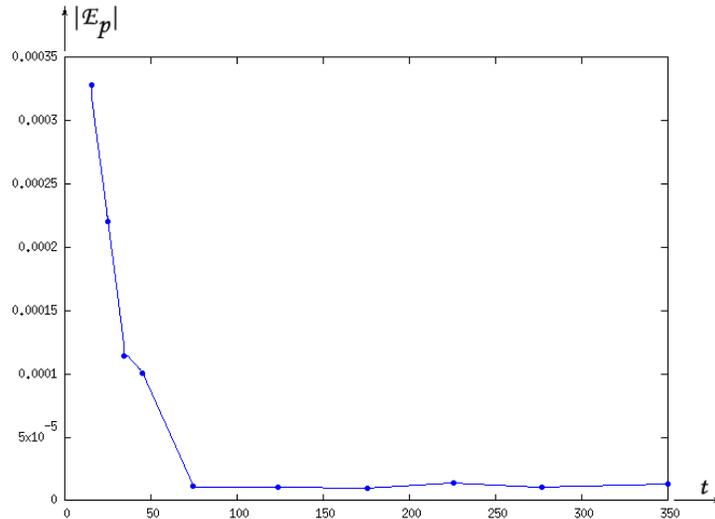


Figura 6.4: Error de la presión para cada tiempo de simulación.

Saturación

La función de saturación deseada en este problema, es una función escalón cuyo frente de onda avanza conforme lo hace el tiempo de simulación (véase la figura 6.5). Se obtiene de forma analítica y se toma como referencia para medir el grado de precisión que tiene la función de saturación obtenida (figura 6.3) con FVM usando la fórmula

$$\|E^s\| = \frac{1}{N} \sum_i^N |s_i^{FVM} - s_i^{Analitica}|, \quad (6.2)$$

donde E^s es el error de la saturación y N es el número de puntos en los que se evalúa la saturación. El error de la función de saturación obtenida con FVM con respecto a la analítica es $\|E^s\| = 0.0180$.

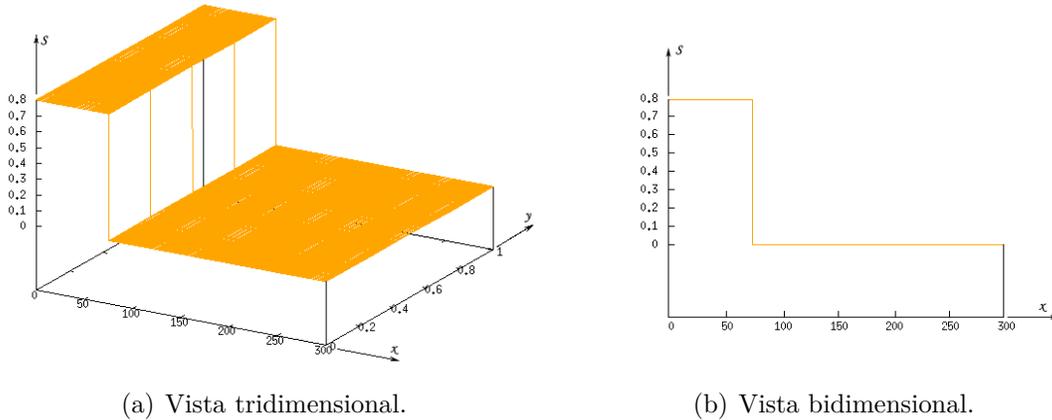


Figura 6.5: Función de la saturación deseada.

6.3. Solución del problema usando SLS-RBF

6.3.1. Cálculo de la presión con RBF básico

Para la ecuación de presión (4.30), se discretizó el dominio esparciendo puntos uniformemente con $N_x = 60$ y $N_y = 5$, y se hizo una simulación para 300 días.

Usando del método RBF básico, se derivó un sistema lineal de ecuaciones que se resolvió con dos métodos de solución de la biblioteca Eigen. Primero con el método *fullPivHouseholderQR* y posteriormente con el *fullPivLU*.

El método *fullPivHouseholderQR* calcula la descomposición QR de una matriz. Es decir, hace la descomposición de una matriz A en matrices P , Q y R tal que $AP = QR$ usando transformaciones Householder. Aquí P es una matriz de permutaciones, Q es una matriz identidad y R una matriz triangular superior. La descomposición hace el pivoteo completo [55].

Usando este método y variando el parámetro de forma c (mismo para todo el dominio) como se indica en la tabla 6.1, no fue posible obtener una solución cercana a la obtenida por el FVM (figura 6.2), la cual se usó como referencia. La siguiente tabla muestra los resultados obtenidos.

c	Error vs FVM	Tiempo CPU [s]
0.0000009	57043305.96	0.13
1.0000000	57032521.53	0.08
100.0000000	56808179.83	0.07
300.0000000	52131789.66	0.05
800.0000000	52019708.33	0.05
1200.0000000	52062258.50	0.05

Tabla 6.1: Presión obtenida con RBF básico usando *fullPivHouseholderQR*.

El grado de aproximación de la presión obtenida por medio de este método con respecto a la obtenida con FVM, se evaluó calculando el error usando la siguiente fórmula

$$\|Error\| = \frac{1}{N} \sum_i^N |p_i^{FVM} - p_i^{RBF}|. \quad (6.3)$$

El método *fullPivLU* calcula la descomposición LU de una matriz con pivoteo completo: la matriz A es descompuesta como $A = P^{-1}LUQ^{-1}$ donde L es una matriz triangular inferior unitaria, U es una matriz triangular superior, y P y Q son matrices de permutación. Los eigenvalores (coeficientes de la diagonal) de U son almacenados de tal forma que no resulten ceros al final. Esta descomposición LU es muy estable y bien provada con matrices densas [55].

Con este método se obtuvieron soluciones cercanas a la obtenida por el FVM (figura 6.2) que se usó como referencia. Con base en [59], se varió el valor de c asignándole valores como se describe en la tabla 6.2, y el grado de aproximación se evaluó usando la fórmula (6.3). También se muestra el tiempo de CPU que se requirió para realizar los cálculos de cada caso. Aquí, c es el valor que se usó en los cálculos para los puntos pertenecientes al interior del dominio Ω , c_1 para $\partial\Omega_1$, c_2 para $\partial\Omega_2$, c_3 para $\partial\Omega_3$ y c_4 para $\partial\Omega_4$.

c	c_1	c_2	c_3	c_4	Error vs FVM	Tiempo CPU [s]
10	15	15	15	15	107830681.50	0.08
20	10	10	10	10	93190451.90	0.10
10	100	0.01	0.01	0.01	25767409.00	0.09
0.0009	100	0.0009	0.0009	0.0009	13572852.33	0.09
0.0009	10	0.0009	0.0009	0.0009	7407879.00	0.08
10	10	10	10	10	322196.66	0.09
85	85	85	85	85	2717.90	0.09
100	100	100	100	100	634.63	0.08
200	200	200	200	200	523.70	0.08
150	150	150	150	150	61.16	0.08

Tabla 6.2: Presión obtenida con RBF básico usando fullPivLU.

6.3.2. Cálculo de la presión con RBF aumentado

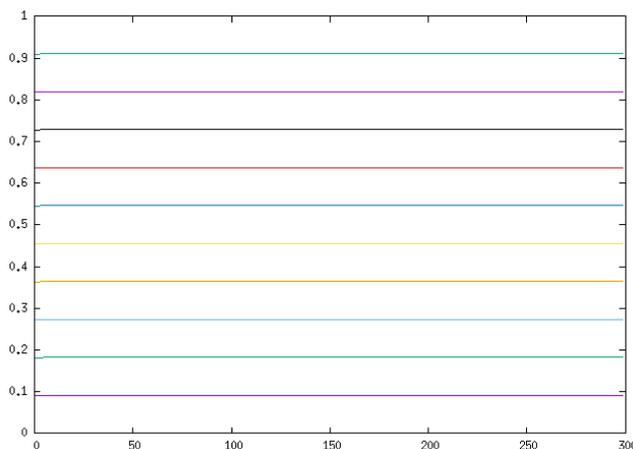
Finalmente, en la solución de la PDE de la presión (4.30), con la misma discretización del dominio y tiempo de simulación de la sección anterior, se usó el método RBF aumentado para generar su sistema lineal de ecuaciones correspondiente y se resolvió con el método *fullPivLU* de la biblioteca Eigen. En la tabla 6.3 se muestra, el error de la presión obtenida dependiendo de los valores que se le asignaron a c , tomando como referencia la que se obtuvo con FVM (figura 6.2), y el tiempo de CPU para realizar los cálculos.

c	c_1	c_2	c_3	c_4	Error vs FVM	Tiempo CPU [s]
100	0.0009	0.0009	0.0009	0.0009	186.66	0.1
10	0.0009	0.0009	0.0009	0.0009	0	0.1

Tabla 6.3: Presión obtenida con RBF aumentado usando fullPivLU.

6.3.3. Construcción de las líneas de corriente

Las líneas de corriente trazadas con el movimiento de las N_p partículas cuya posición inicial se establece dentro del campo vectorial de velocidad derivado de la presión del aceite, se muestran a continuación. En la figura 6.6, tienen una distribución uniforme. La posición inicial de las partículas es $x_i = 0$ y $y_{i+1} - y_i = K$ para $i = 1, 2, 3, \dots, N_p$ y K es constante. El tiempo de CPU para trazar una línea de corriente fue de 0.015 segundos.



(a) Distribución uniforme.

Figura 6.6: Trazo de las líneas de corriente.

6.3.4. Cálculo de la saturación con Forward Euler

Para la ecuación de saturación en cada línea de corriente (4.60) en el espacio de - coordenadas TOF, se discretizó el dominio (cada línea) esparciendo 100 puntos sobre él uniformemente, se usó el método RBF básico con base en Forward Euler con un parámetro de forma $c = 8000000$ y se hizo una simulación para 300 días.

La saturación en cada línea de corriente, presenta oscilaciones que crecen conforme avanza el tiempo de simulación (figura 6.7) y se obtiene un error de saturación $\|E^s\| = 0.2043$ que se calculó con la siguiente fórmula

$$\|E^s\| = \frac{1}{N} \sum_i^N |s_i^{RBF} - s_i^{Analitica}|. \quad (6.4)$$

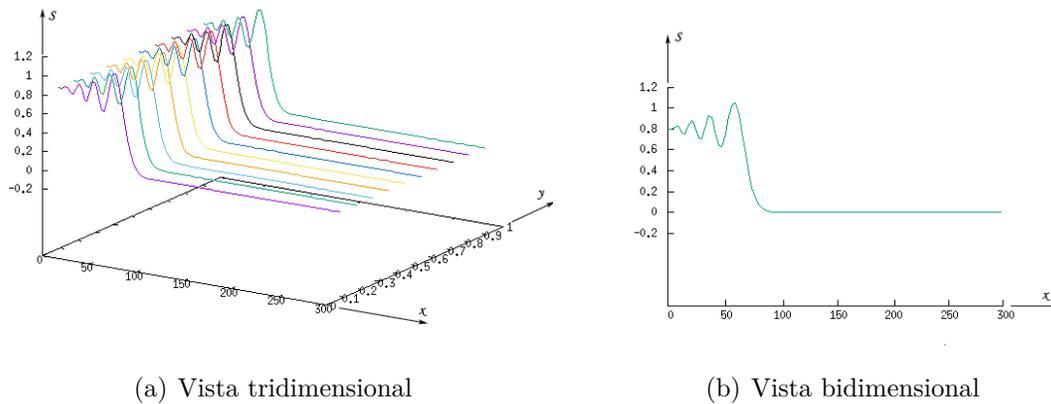


Figura 6.7: Saturación en líneas de corriente usando el método RBF básico con base en Forward Euler.

6.3.5. Cálculo de la saturación con Backward Euler

Para la ecuación de saturación en cada línea de corriente (4.60) en el espacio de coordenadas TOF, se discretizó el dominio esparciendo 100 puntos sobre él uniformemente, se usó el método RBF básico con base en Backward Euler con un parámetro de forma $c = 5005050$ y se hizo una simulación para 300 días.

La saturación obtenida en cada línea de corriente es estable (figura 6.8), es la que más se aproxima a la solución analítica con un error de saturación $\|E^s\| = 0.0075$, que se calculó con la fórmula 6.4.

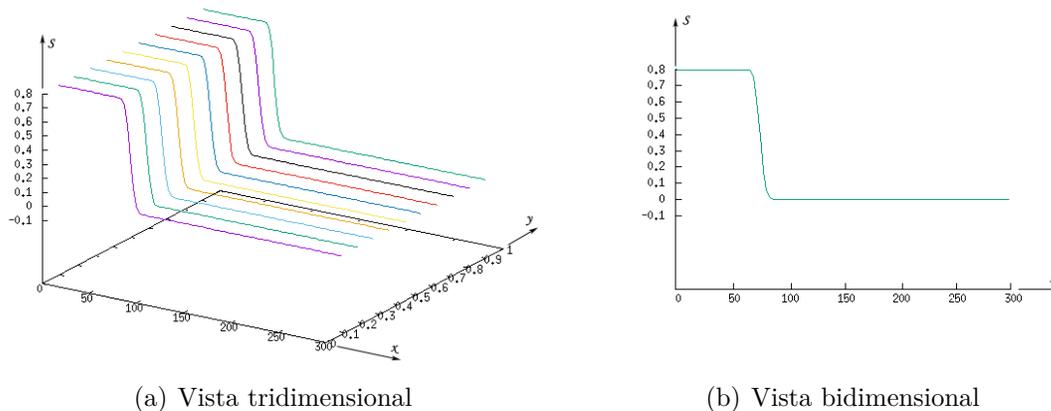
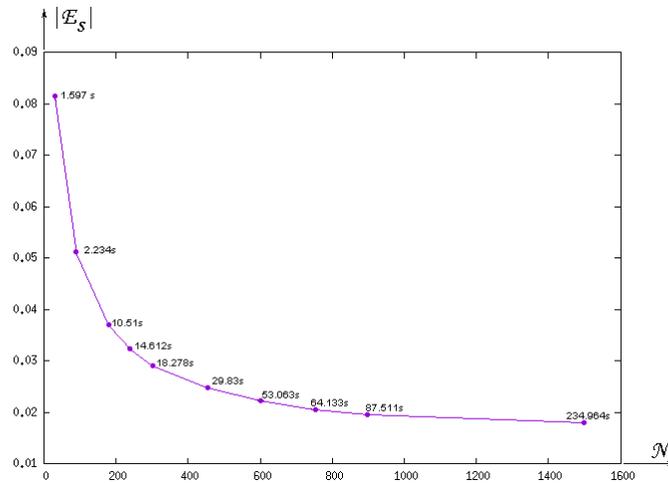


Figura 6.8: Saturación en líneas de corriente usando el método RBF básico con base en Backward Euler.

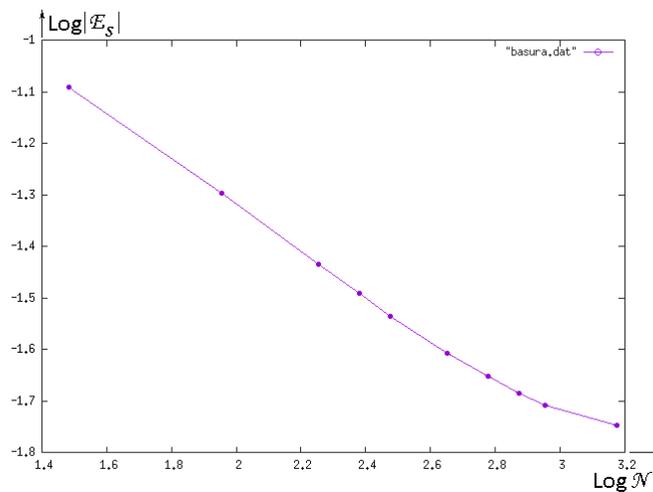
6.4. Análisis de resultados

6.4.1. Resultados con FVM

El número de volúmenes con que se discretizó el dominio de estudio para resolver con FVM (véase la sección 6.2), dependió del grado de precisión que se obtenía para la saturación, la cual se evalúa con la fórmula (6.2). Conforme se aumenta el número de puntos en el dominio del problema, la función resultante se aproxima más a la solución analítica mostrada en la figura 6.5, es decir, si $N \rightarrow \infty$, el error de la saturación $\|E^s\| \rightarrow 0$ (véase la figura 6.9), sin embargo, esto genera cada vez mayor consumo de tiempo y memoria de cómputo.



(a) Escala lineal



(b) Escala logarítmica

Figura 6.9: Error de la saturación dependiendo del número de puntos en el dominio.

Las soluciones aproximadas de ecuaciones diferenciales convergen a la solución analítica cuando se refina la malla. Así si S_N es la solución aproximada numérica de la saturación usando N puntos, y S es la solución analítica, entonces para cualquier punto x en el dominio, se tiene que $|S_N(x) - S(x)| \rightarrow 0$ cuando $N \rightarrow \infty$.

Dado que una solución numérica converge polinomialmente con orden p a la solución si existe una constante c tal que $|S_N(x) - S(x)| < cN^p$, podemos observar entonces en la gráfica a escala logarítmica (figura 6.9b), que al refinar la malla en el FVM, hay mayor precisión y se tiene una convergencia polinomial con orden $p = 1$ a la solución, teniendo una constante $c = 0.89$ que es la pendiente de la recta en $\log N \in [1.5, 2.6]$. Para el caso en que $\log N \geq 2.6$, la pendiente tiende a cero porque el error tiende a ser constante, es decir, en el rango de puntos mayor a 1000, se alcanza la convergencia prácticamente.

Para el equipo de cómputo en el que se programó la solución, se discretizó el dominio con 2500 puntos, lo que implica tener una matriz a resolver de $2500 \times 2500 = 6,250,000$ números de tipo double. La solución óptima obtenida con una simulación de 300 días, toma un tiempo de CPU de 234.96 segundos y un error de saturación $\|E^s\| = 0.0180$. Véase la figura 6.3.

6.4.2. Resultados con FVM y con la presión fuera del IMPES

Con base en lo mencionado en el apartado presión de la sección 6.2, se obtuvo la solución con la presión ya conocida (figura 6.2) y se consideró únicamente dentro del ciclo IMPES el cálculo de la saturación. Discretizando el dominio de igual forma que en esa sección, y haciendo el mismo tiempo de simulación. El resultado obtenido fue el mismo, con la excepción de que el tiempo de CPU en este caso fue de 35.25 segundos.

6.4.3. Resultados con RBF y con la presión fuera del IMPES

En este método, se discretizó el dominio con 500 puntos (100 puntos por cada línea de corriente), los cuales son suficientes para describir el comportamiento de la saturación. Con la finalidad de realizar una comparación con un fenómeno equivalente, se calculó la saturación también para un lapso de 300 días. El resultado que se obtuvo fue un tiempo de CPU de 3.30 segundos y un error de 0.0075.

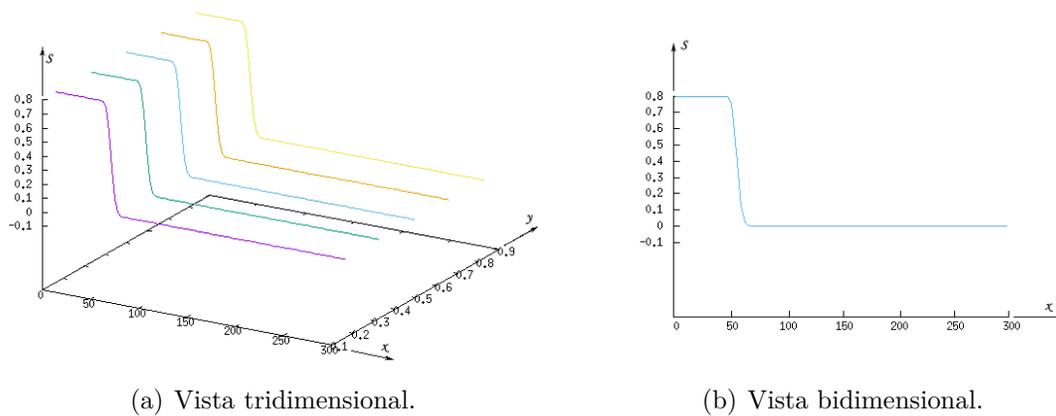


Figura 6.10: Función de la saturación obtenida con RBF.

6.4.4. Resultados con RBF, la presión fuera del IMPES y cómputo paralelo

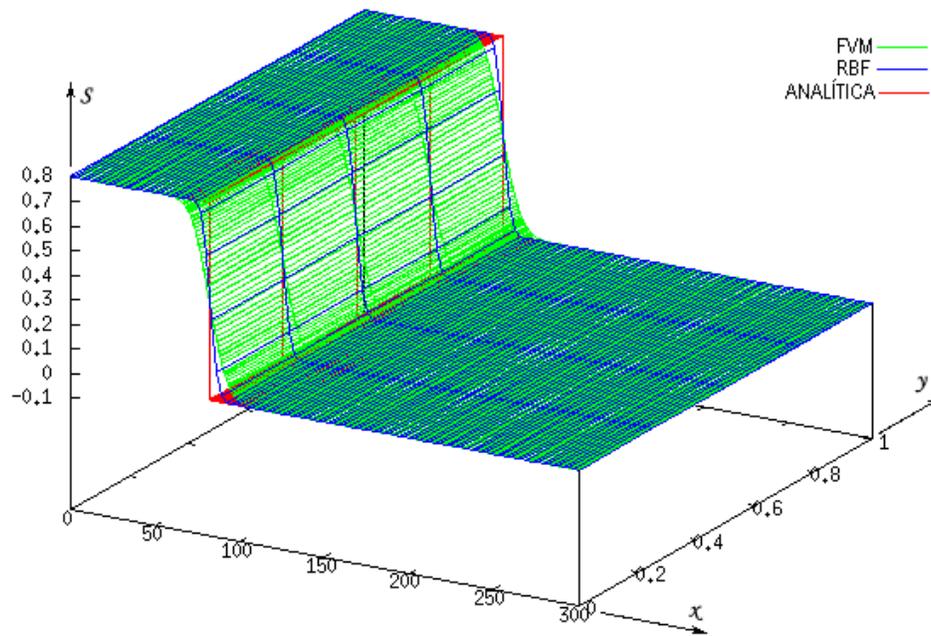
Se discretizó el dominio de igual forma que en la sección anterior. Se usó OpenMP para paralelizar el cálculo de la saturación de las líneas de corriente con 5 procesos. Se considerando un lapso de 300 días. La función obtenida de la saturación es el mismo pero con un tiempo real de cómputo de 1.59 segundos y un error de 0.0075. Véase la figura 6.10.

6.4.5. Resumen comparativo

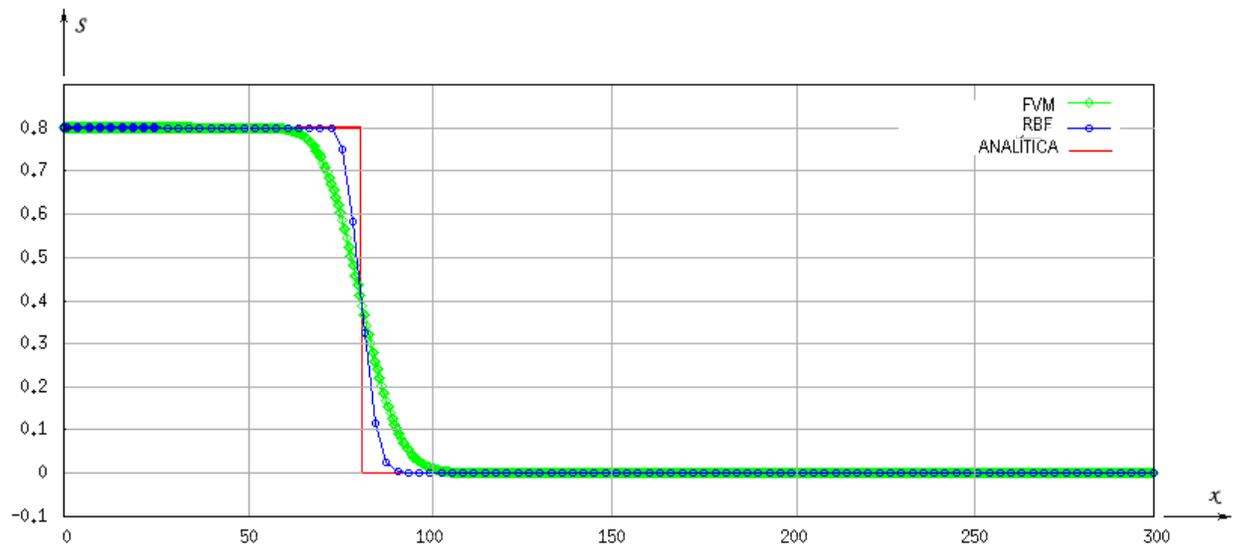
Para la simulación de 300 días que se ha hecho del desplazamiento de aceite por medio de agua, la tabla 6.4 muestra del método FVM y del RBF a manera de resumen, el número de puntos que se usaron en la discretización del dominio, el grado de precisión de la función saturación obtenida con respecto a la analítica y el tiempo de cómputo necesario para obtener la solución. En la figura 6.11, pueden visualizarse las gráficas de la saturación obtenidas en cada método comparadas con la analítica.

	N Puntos	$\ Error\ $	$tiempo$ [seg]
FVM IMPES completo	2500	0.018	234.96
FVM (<i>presión fuera de IMPES</i>)	2500	0.018	35.25
RBF (<i>presión fuera de IMPES</i>)	500	0.0075	5.67
RBF (<i>presión fuera de IMPES</i>) <i>en paralelo con 5 procesos</i>	500	0.0075	2.92

Tabla 6.4: Resumen comparativo para un tiempo de simulación de 300 días.



(a) Vista tridimensional.



(b) Vista bidimensional.

Figura 6.11: Función de la saturación obtenida con FVM, RBF y Analítica.

7

Conclusiones

7.1. Conclusiones

Resolver una PDE con RBF, requiere esencialmente de tres elementos: del amplio conocimiento de la solución esperada, del gran reto de encontrar la función de base radial junto con el valor de sus parámetros característicos que contribuirá a ello, y de un método iterativo eficiente para la solución del sistema de ecuaciones cuya matriz es densa y mal condicionada.

En este trabajo se combinaron los métodos SLS y RBF para la solución del flujo bifásico en medios porosos gobernado por las ecuaciones del modelo matemático Buckley-Leverett. Resolver con RBF la PDE de la presión, implicó conocer la solución ya obtenida por el trabajo del Dr. De la Cruz [58] con FVM, experimentar con la función de base radial multicuadrado y con diferentes valores del parámetro de forma c , donde el sistema de ecuaciones se resolvió con el método `fullPivHouseHolerQR` de la biblioteca `Eigen`. Los resultados obtenidos fueron funciones de la presión incongruentes con la esperada en todos los casos.

Posteriormente, se experimentó resolver el sistema de ecuaciones con el método `fullPivLu` de la biblioteca `Eigen`. Se obtuvo una función de la presión congruente con la esperada pero con cierto grado de imprecisión. Luego considerando la investigación del Dr. Kansa [59], se varió c de manera independiente para cada frontera del dominio y se obtuvieron funciones de la presión congruentes con la esperada a excepción de cuando su valor en los puntos del interior del dominio era menor que el de los puntos de las fronteras. Finalmente, se aplicó el mismo procedimiento pero generando el sistema de ecuaciones con el método RBF aumentado y se obtuvo una función para la presión casi exactamente igual a la esperada. Al variar el valor de c , en el interior del dominio y en las fronteras de tal forma que se aproximara más, se obtuvo una función exactamente igual a la esperada. Esto evitó la desventaja que el método presenta: mayor error en las fronteras del dominio. No fue necesario agregar puntos adyacentes.

El comportamiento de la presión que se observó en este fenómeno, es que conforme avanza el tiempo de simulación, no cambia. Esto permitió encontrar la función a la cual converge, tomarla como constante y excluirla del ciclo IMPES, obteniendo el mis-

mo resultado para la saturación que cuando no se excluye. Así el tiempo de cómputo tanto para FVM como para RBF es menor.

Para el cálculo de las PDEs de saturación a lo largo de las líneas de corriente, se aplicó primero RBF con base en Forward Euler, y se obtuvo una función de saturación con oscilaciones que incrementaban su amplitud conforme lo hacía el tiempo de simulación. Esto debido a que Forward Euler por su forma explícita es inestable. Luego se aplicó Backward Euler y se obtuvo una función estable para la saturación más próxima a la función escalón deseada. Su grado de precisión fue mucho mayor que la obtenida por FVM, la cual para hacerla más precisa, requirió de aumentar en gran medida el número de volúmenes y por ende el tiempo y memoria de cómputo.

Luego de encontrar la función de base radial junto con los valores de sus parámetros adecuados para la obtención de la solución de las PDEs, se necesitó menor número de puntos para discretizar el dominio que cuando se aplica FVM, se requirió menor tiempo y memoria de cómputo y se obtuvo mayor precisión.

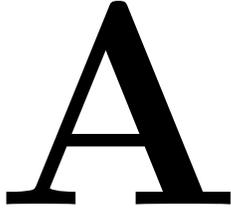
Para paralelizar el programa computacional, se observó que el mayor consumo de tiempo de cómputo, se llevaba a cabo en el cálculo de la solución de la PDE de saturación a lo largo de cada línea de corriente, y tomando en cuenta que cada una de ellas es independiente de las demás, con OpenMP se distribuyó el cálculo de cada una en diferentes procesos. Se obtuvo la misma precisión que el algoritmo serial, pero aún con menor tiempo de cómputo.

7.2. Direcciones futuras

Debido al espacio y tiempo delimitados de este trabajo, de los resultados, surgen direcciones futuras que lo enriquecerían y que lo harían aún más útil. A continuación, brevemente se escriben algunas de ellas:

- Encontrar el parámetro de forma c por medio de un algoritmo genético.
- Resolver el sistema de ecuaciones con algoritmos iterativos eficientes combinados con un preconditionador, como el GMRES que es un método de subespacios de Krylov. Ó con el método de descomposición de dominio el cual fue introducido por Schwarz en 1870.
- Resolver el problema para dominios irregulares.
- Resolver el problema con el método RBF de soporte compacto.
- Paralelizar no solo el cálculo de la saturación sobre cada línea de corriente, sino también la interpolación de la presión y la solución de los sistemas de ecuaciones.

Apéndices



Ecuación de presión

Para obtener la ecuación de presión, primero se divide la ecuación (4.8) por ρ_α , con lo que se obtiene

$$\frac{1}{\rho_\alpha} \left[\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \underline{u}_\alpha) - q_\alpha \right] = 0,$$

y sumando sobre todas las fases se tiene

$$\sum_\alpha \left\{ \frac{1}{\rho_\alpha} \left[\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \underline{u}_\alpha) - q_\alpha \right] \right\} = 0,$$

desarrollando las derivadas de la ecuación anterior se llega a lo siguiente

$$\sum_\alpha \left\{ \frac{1}{\rho_\alpha} \left[\rho_\alpha S_\alpha \frac{\partial \phi}{\partial t} + \phi S_\alpha \frac{\partial \rho_\alpha}{\partial t} + \rho_\alpha \phi \frac{\partial S_\alpha}{\partial t} + \rho_\alpha \nabla \cdot \underline{u}_\alpha + \underline{u}_\alpha \cdot \nabla \rho_\alpha - q_\alpha \right] \right\} = 0$$

y finalmente reorganizando y aplicando la relación (4.12) a la ecuación anterior, se puede escribir

$$\frac{\partial \phi}{\partial t} + \sum_\alpha \nabla \cdot \underline{u}_\alpha + \sum_\alpha \frac{1}{\rho_\alpha} \left[\phi S_\alpha \frac{\partial \rho_\alpha}{\partial t} + \underline{u}_\alpha \cdot \nabla \rho_\alpha \right] - \sum_\alpha \frac{q_\alpha}{\rho_\alpha} = 0. \quad (\text{A.1})$$

Se define ahora la velocidad total \underline{u} como

$$\sum_\alpha \underline{u}_\alpha, \quad (\text{A.2})$$

y aplicando la divergencia a esta relación se llega a

$$\nabla \cdot \underline{u} = \nabla \cdot \sum_\alpha \underline{u}_\alpha = \sum_\alpha \nabla \cdot \underline{u}_\alpha. \quad (\text{A.3})$$

Insertando (A.3) en (A.1) se obtiene la siguiente ecuación

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \underline{u} + \sum_\alpha \frac{1}{\rho_\alpha} \left[\phi S_\alpha \frac{\partial \rho_\alpha}{\partial t} + \underline{u}_\alpha \cdot \nabla \rho_\alpha \right] - \sum_\alpha \frac{q_\alpha}{\rho_\alpha} = 0. \quad (\text{A.4})$$

Ahora, de la definición de la ley de Darcy para flujo multifásico se tiene que

$$\underline{u} = \sum_\alpha \underline{u}_\alpha = \sum_\alpha \left[-\underline{k} \lambda_\alpha (\nabla p_\alpha - \rho_\alpha \underline{g}) \right]. \quad (\text{A.5})$$

Definiendo una función de flujo fraccional para la fase α como

$$f_\alpha = \frac{\lambda_\alpha}{\lambda} \implies \lambda_\alpha = f_\alpha \lambda \quad (\text{A.6})$$

donde $\lambda = \sum_\alpha \lambda_\alpha$ es la movilidad total y por tanto se cumple que $\sum_\alpha f_\alpha = 1$.
Sustituyendo (A.6) en (A.5) se obtiene

$$\underline{u} = -\lambda \underline{k} \left[\sum_\alpha f_\alpha \nabla p_\alpha - \sum_\alpha f_\alpha \rho_\alpha \underline{g} \right]. \quad (\text{A.7})$$

Con esta formulación de la velocidad total, la ecuación (A.4) se transforma en una ecuación para la presión que se escribe como sigue

$$\frac{\partial \phi}{\partial t} - \nabla \cdot \lambda \underline{k} \left[\sum_\alpha f_\alpha \nabla p_\alpha - \sum_\alpha f_\alpha \rho_\alpha \underline{g} \right] + \sum_\alpha \frac{1}{\rho_\alpha} \left[\phi S_\alpha \frac{\partial \rho_\alpha}{\partial t} + \underline{u}_\alpha \cdot \nabla \rho_\alpha \right] - \sum_\alpha \frac{q_\alpha}{\rho_\alpha} = 0. \quad (\text{A.8})$$

B

Inyección de agua

Se impone una razón de inyección constante de agua en la frontera $\partial\Omega_1$, que va en la dirección $(1,0)$. Dado que la normal a dicha frontera es $n = (-1, 0)$, entonces para imponer la condición de frontera se tiene que

$$-\underline{n} \cdot \underline{u} = g_p^{in} \quad (\text{B.1})$$

donde \underline{u} es la velocidad total de Darcy.

$$\underline{u} = \underline{u}_o + \underline{u}_w \quad (\text{B.2})$$

Sustituyendo la ecuación (4.9) en la ecuación (B.2) sin considerar la fuerza de gravedad \hat{g} , se obtiene

$$\underline{u} = -\underline{k} \frac{k_{ro}}{\mu_o} (\nabla P_o) - \underline{k} \frac{k_{rw}}{\mu_w} (\nabla P_w). \quad (\text{B.3})$$

Tampoco se está considerando la presión capilar, es decir $P_c = 0$, entonces

$$P_c = P_o - P_w = 0 \quad \implies \quad P_o = P_w. \quad (\text{B.4})$$

Sustituyendo (B.4) en (B.3) se obtiene

$$\underline{u} = -\underline{k} \frac{k_{ro}}{\mu_o} (\nabla P_o) - \underline{k} \frac{k_{rw}}{\mu_w} (\nabla P_o) \quad (\text{B.5})$$

$$\underline{u} = -\underline{k} \frac{k_{ro}}{\mu_o} (\nabla P_o) \left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} \right).$$

De acuerdo con la ecuación (4.10) se tiene que

$$\lambda_o = \frac{k_{ro}}{\mu_o} \quad y \quad \lambda_w = \frac{k_{rw}}{\mu_w}$$

las cuales al sustituirlas en la ecuación (B.5) se obtiene

$$\underline{u} = -\underline{k} (\nabla P_o) (\lambda_o + \lambda_w) \quad (\text{B.6})$$

Pero la movilidad total $\lambda = \lambda_o + \lambda_w$, entonces la ecuación (B.6) puede escribirse

$$\underline{u} = -\lambda \underline{k}(\nabla P_o). \quad (\text{B.7})$$

sustituyendo la ecuación (B.7) en (B.1) se obtiene

$$-\underline{n} \cdot \underline{u} = \underline{n} \cdot [\lambda \underline{k}(\nabla P_o)] = g_p^{in}.$$

$$(-1, 0) \cdot \left[\lambda \begin{pmatrix} k_{11} & 0 \\ 0 & k_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial P_o}{\partial x} \\ \frac{\partial P_o}{\partial y} \end{pmatrix} \right] = g_p^{in} \quad (\text{B.8})$$

$$(-1, 0) \cdot \left[\lambda \begin{pmatrix} k_{11} \frac{\partial P_o}{\partial x} \\ \frac{\partial P_o}{\partial y} k_{22} \end{pmatrix} \right] = g_p^{in}$$

$$-\lambda k_{11} \frac{\partial P_o}{\partial x} = g_p^{in}.$$

C

Interpolación de S

De acuerdo a las condiciones iniciales definidas en la tabla 3.1, en todos los puntos la presión inicial es de $1.0e+07$ pa y la saturación es cero.

Sin embargo, al empezar la saturación de agua en la frontera $\partial\Omega_1$ debido a la inyección constante que de ella se hace, la saturación S en todos los puntos de esa frontera es 0.8 y en los demás cero. En ese instante, la movilidad $\lambda(0.8) = 1000$.

La saturación S se conoce en todos los puntos y por tanto $\lambda(S)$ también con la ecuación (5.25).

Dado que la saturación S se conoce en cada uno de los puntos dispersos en el dominio, se usa el método RBF para encontrar una ecuación de S que los interpole. Esta ecuación nos permitirá a su vez, conocer su derivada en cada uno de ellos.

De acuerdo a la definición del interpolante básico RBF dada en la sección 5.2.2.1, expresamos la ecuación que interpola los datos dispersos de saturación de la siguiente forma

$$S(\underline{x}_i) = \sum_{j=1}^N w_{s_j} \varphi_j(\|\underline{x}_i - \underline{x}_j\|) \quad \text{para } i = 1, \dots, N, \quad (\text{C.1})$$

al expandir la sumatoria se puede expresar matricialmente como

$$\begin{pmatrix} \varphi_1(r_1) & \varphi_2(r_1) & \cdots & \varphi_N(r_1) \\ \varphi_1(r_2) & \varphi_2(r_2) & \cdots & \varphi_N(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(r_N) & \varphi_2(r_N) & \cdots & \varphi_N(r_N) \end{pmatrix} \begin{pmatrix} w_{s_1} \\ w_{s_2} \\ \vdots \\ w_{s_N} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{pmatrix}. \quad (\text{C.2})$$

Resolviendo el sistema lineal de ecuaciones se obtienen los valores de los coeficientes de expansión w_{s_j} .

Conocidos los valores w_{s_j} , es posible obtener la derivada de la saturación en cada uno de los puntos esparcidos en el dominio (véase el apéndice D)

$$\frac{\partial S(\underline{x}_i)}{\partial x} = \sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial x} \quad \text{y} \quad \frac{\partial S(\underline{x}_i)}{\partial y} = \sum_{j=1}^N w_{s_j} \frac{\partial \varphi_j(r_i)}{\partial y} \quad \text{para } i = 1, \dots, N \quad (\text{C.3})$$

D

Derivadas de RBF

Dada una función de base radial como la (5.10)

$$u(\underline{x}_i) = \sum_{j=1}^n w_j \varphi_j(r_i) \quad \text{para } i = 1, \dots, n$$

las derivadas de la función $u(\underline{x}_i)$ son calculadas por

$$\frac{\partial^k u(\underline{x}_i)}{\partial x_m \cdots \partial x_n} = \sum_{j=1}^n w_j \frac{\partial^k \varphi_j(r_i)}{\partial x_m \cdots \partial x_n}, \quad (\text{D.1})$$

$$\frac{\partial u(\underline{x}_i)}{\partial x} = \sum_{j=1}^n w_j \frac{\partial \varphi_j(r_i)}{\partial x}, \quad (\text{D.2})$$

$$\frac{\partial^2 u(\underline{x}_i)}{\partial x^2} = \sum_{j=1}^n w_j \frac{\partial^2 \varphi_j(r_i)}{\partial x^2}, \quad (\text{D.3})$$

donde $r_i = \|\underline{x}_i - \underline{x}_j\|$.

E

Cálculo analítico de $\frac{\partial F_w}{\partial S}$

De acuerdo a la definición de flujo fraccional de la sección 3.1.3, el del agua es

$$F_w = F_w(S) = \frac{\lambda_w}{\lambda} \quad (\text{E.1})$$

donde λ_w es la movilidad del agua y λ es la movilidad total. Las movilidades del agua y la total, están dadas por las ecuaciones (5.23) y (5.25). Sustituyéndolas en la ecuación (E.1), se obtiene

$$F_w(S) = \frac{S - S_{rw}}{S - S_{rw} + \frac{\mu_w}{\mu_o} (1 - S_{ro} - S)}, \quad (\text{E.2})$$

entonces

$$\frac{\partial F_w}{\partial S}(S) = \frac{\mu_w}{\mu_o} \frac{1 - S_{ro} - S_{rw}}{\left(S - S_{rw} + \frac{\mu_w}{\mu_o} (1 - S_{ro} - S) \right)^2} . \quad (\text{E.3})$$

Bibliografía

- [1] I. Herrera and G. F. Pinder. *Mathematical modeling in Science and Engineering: An Axiomatic Approach*. John Wiley & Sons, 2012.
- [2] Zhangxin Chen. *Reservoir Simulation, Mathematical Techniques in Oil Recovery*. Siam, 2007.
- [3] Charles Hirsch. *Numerical Computation of Internal and External Flows*. Butterworth-Heinemann, 2007.
- [4] D. H. Fenwick, M. R. Thiele and R. P. Batycky . *Streamline simulation for Modern Reservoir-Engineering Workflows*. Distinguished Author Series, SPE, 2010.
- [5] J. H. Abou-Kassem, M. Rafiq, S. M. Farouq. *Petroleum Reservoir Simulation. A Basic approach*. Gulf publishing company, 2006.
- [6] E.J. Kansa. Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics-i. surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19:127–145, 1990.
- [7] 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. *Comput. Math. Appl.*, 19:147–161, 1990.
- [8] Marthin D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003.
- [9] K. Stüben, M. Förster, M. R. Thiele and R. P. Batycky. Parallelization of a commercial streamline simulator and performance on practical models. *Society of Petroleum Engineers*, 2010.
- [10] David Lee Davidson. The rol of computational fluid dynamics in process industries. *The Bridge.*, 32:9–14, 2002.
- [11] I. Herrera and G. F. Pinder and M. B. Allen. *Numerical modeling in Science and Engineering*. John Wiley & Sons, USA, 1988.
- [12] R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2004.

- [13] Z. Chen, G. Huan and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media*. Siam, 2006.
- [14] S. E. Buckley and M. C. Leverett. Mechanism of fluid displacement in sands. *New York Meetig*, 1:107–116, 1941.
- [15] Walter Rose. Attaching new meanings of the equations of buckley and leverett. *Journal of petroleum science and engineering*, 1:223 – 228, 1988.
- [16] Z. Chen, G. Huan and Baoyan Li. An improved impes method for two-phase flow in porous media. *Department of mathematics*, 54:361–376, 2004.
- [17] Akhil Datta-Gupta and Michael J. King. *Streamline Simulation: Theory and practice*. SPE, 2007.
- [18] S. Yoon, A. H. Malallah, A. Datta-Gupta, D. W. Vasco and R. A. Behrens. *A multiscale approach to production-data integration using streamlin models*. SPE, 2001.
- [19] M. Muskat. The theory of potentiometric models. *Trans. AIME*, 179:216–221, 1948.
- [20] Higgins R. V. and Leighton A. J. A computer method to calculate two-pahe flow in any irregularly bounded porous medium. *J. Pet. Tech.*, 14:679–693, 1962.
- [21] Martin J. C. and Wegner R. E. Numerical solution of mutiphase, two-dimensional incompressible flow using stream-tube relationships. *Soc. Pet. Eng. J.*, 19:313–323, 1979.
- [22] Wang Ben, Lake L. W. and Pope G. A. Development and application of a stream-line micellar/polymer simulator. *SPE*, pages 521–532, 1981.
- [23] Thiele M. R., Blunt M. J. and Orr F. M. Modeling flow in heterogeneous media using streamtubes - miscible and immiscible displacement. *In Situ*, 19:299–339, 1995.
- [24] Hamdi A. Tchelepi Sébastien F. Matringe, Ruben Juanes. Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids. *Journal of Computational Physics*, 219:992–1012, 2006.
- [25] D. W. Pollock. Semianalytical computation of pathlines for finite-difference models. *Groundwater*, 6:743, 1988.
- [26] F. Bradvet. Frontline and frontsim. two full scale, two-phase, black oil reservoir simulators based on front tracking. *Surv. Math. Ind.*, 3:185, 1993.
- [27] F. Bradvet. Streamline computations for porous media flow including gravity. *Transport porous med.*, 25:63, 1996.

- [28] Richard L. Burden and J. Douglas Faires. *Análisis numérico*. Thomson Learning, 2002.
- [29] A. Rafiee, M. T. Manzari and M. Hosseini. An incompressible sph method for simulation of unsteady viscoelastic free-surface flows. *International Journal Non-Linear Mech.*, 42:1210–1223, 2007.
- [30] P. Villon, B. Nyroles and G. Touzot. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech.*, 10:301–318, 1992.
- [31] R. Brighenti. Application of the element-free galerkin meshless method to 3-d fracture mechanics problems. *Eng. Frac. Mec.*, 72:2808–2820, 2005.
- [32] Y.F. Zhang W.K. Liu, S. Jun. Reproducing kernel particle methods. *Int. J. Num. Methods in Fluids*, 20:2808–2820, 2005.
- [33] R L Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.*, 176:1905–1915, 1971.
- [34] R L Hardy. Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968/1988. *Computers Math. Applic.*, 10(8/9):163–208, 1990.
- [35] R. Franke. Scattered data interpolation: Test of some methods. *Math. Comput.*, 38:181–200, 1982.
- [36] W R Madych and S A Nelson. Multivariate interpolation and conditionally positive definite functions. *Approx. Theory Applic.*
- [37] A. I. Fedoseyev, M.J. Friedman and E. J. Kansa . Improved multiquadric method for elliptic partial differential equations via pde collocation on the boundary. *An international Journal Computers & Mathematics with applications*, 43:439–455, 2002.
- [38] B. J. C Baxter. Preconditioned conjugate gradients, radial basis functions, and toeplitz matrices. *International Journal Computers & Mathematics with Applications*, 43:305–318, 2002.
- [39] R. K. Beatson and G N Newsam. Fast evaluation of radial basis functions: Moment-based methods. *SIAM Journal of Scientific Computing*, 19(5):1428–1449, 1998.
- [40] R. K. Beatson, J. B. Cherrie and C. T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration. *Advances in Computational Mathematics*, 11:253–270, 1999.
- [41] R. K. Beatson, W. A. Light and S Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal of Scientific Computing*, 22(5):1717–1740, 2000.

- [42] E.J. Kansa and Y. C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *An International Journal Computers & Mathematics with Applications*, 39:123–137, 2000.
- [43] X. Zhou, Y. C. Hou and Jichun Li. Overlapping domain decomposition method by radial basis functions. *Applied Numerical Mathematics*, 44:241 – 255, 2003.
- [44] E.J.Kansa Leevan Ling. Preconditioning for radial basis functions with domain decomposition methods. *Math. Comput. Modeling.*, 40:1413–1427, 2004.
- [45] E.J.Kansa Leevan Ling. A least-squares preconditioner for radial basis functions collocation methods. *Advances in Computational Mathematics*, 23:31–54, 2005.
- [46] Luis Miguel de la Cruz Salas. MQ-RBF Meshless Method for solving CFD problems using an Object-Oriented Approach. *STFC-CSED Daresbury Laboratory*, 1:1–41, 2008.
- [47] Grady B. Wright. *Radial Basis Function Interpolation: Numerical and Analytical Developments*. PhD thesis, Univeristy of Colorado, 2003. Thesis directed by Prof. Bengt Fornberg.
- [48] A. Haar. Die minkowskische geometrie und die ann an stetige funktionen. *Math. Ann.*, 18:294–311, 1918.
- [49] Theory and applications of the multiquadric-biharmonic method: 20 years of discovery. *Comput. Math. Appl*, 19:163–208, 1990.
- [50] International Symposium on Surface Approximation. *Lecture notes on global basis functions methods for scattered*, 1983. Govagno, Italy, University of Milano.
- [51] C. A. Michelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Aprox.*, 2:11–22, 1986.
- [52] D. Yang. *C++ and Object-Oriented Numeric Computing*. Springer-Verlag, 2001.
- [53] D. Vandevoorde and N. M. Josuttis. *C++ Templates*. Addison?Wesley, 2003.
- [54] Michael J. Quinn. *Parallel programing in C with MPI and OpenMP*. Mc Graw Hill, 2004.
- [55] Library eigen. Technical report, 2015. <http://eigen.tuxfamily.org>.
- [56] Jan M. Rabaey. *Digital Integrated Circuits: a design perspective*. Prentice-Hall, 1996.
- [57] Jesús Savage Carmona, Gabriel Vázquez and Norma Elva Chávez Rodríguez. *Diseño de microprocesadores*. Facultad de Ingeniería, UNAM.

-
- [58] Luis Miguel de la Cruz Salas. Flujo en una y dos fases en medios porosos: modelos matemáticos, numéricos y computacionales. *Reportes Internos 2012-04, GMMC*, 1:1–41, 2012.
- [59] Grupo de Geofísica Matemática y Computacional del IGEF. *Radial Basis Functions: Achievements and challenges*, 2015. Ph. Edward Kansa, Convergent Solutions, USA.