



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**USO DE UN MÉTODO HÍBRIDO PARA TRATAMIENTO DE IMÁGENES
DE EVOLUCIÓN DE CÁNCER DE MAMA**

TESIS

**QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**

**PRESENTA:
ELPIDIO CRUZ CASTILLO**

**DIRECTOR DE TESIS:
Dra. Suemi Rodríguez Romo
Facultad de Estudios Superiores Cuautitlán, UNAM**

Ciudad Universitaria, CD. MX.

Enero, 2018



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

A la Universidad, por la formación que me ha dado.

A mi familia que me ha apoyado en los momentos más difíciles.

A la Doctora Suemi Rodríguez Romo, por haberme apoyado en todo este tiempo.

En verdad, gracias.

Reconocimientos

Agradezco al proyecto PAPIIT IT101817 por el apoyo para la realización de este trabajo y la facilidad de las instalaciones del Centro de Investigaciones Teóricas de la Facultad de Estudios Superiores Cuautitlán.

También quiero reconocer a CONACYT por el apoyo otorgado para la realización de este trabajo.

A mis compañeros y amigos, quienes me acompañaron siempre.

Resumen

En los años recientes se han establecido nuevas maneras de abordar los temas de investigación acerca del cáncer de mama, como es el caso de las pruebas para detectar células tumorales circulantes (CTC) que consiste en buscar mediante una prueba sanguínea a las pequeñas células cancerosas que se desprenden del seno y recorren el cuerpo a través del torrente sanguíneo, o la Gammagrafía mamaria (imagenología molecular del seno) en la que se inyecta en una vena un trazador ligeramente radiactivo que se une a las células del cáncer de seno, y luego se utiliza una cámara especial para detectarlas [1].

Las imágenes médicas, especialmente las mamografías son uno de los mejores métodos de estudio y diagnóstico por su facilidad de interpretación para los especialistas, aunque siguen siendo estudiadas para determinar la confiabilidad en la búsqueda de cánceres de seno. Generalmente a los médicos les es útil para observar áreas sospechosas encontradas en las mamografías convencionales, aunque siempre con el margen de error humano que implica la observación. La investigación actual está dirigida a proporcionar una herramienta computacional que potencialice las capacidades del ser humano para evaluar el posible avance de crecimiento de tumores en situaciones específicas, tales como en el caso de tejidos densos en mujeres más jóvenes, específicamente de cáncer de mama en donde se han realizado diferentes análisis y propuestas para poder desarrollar métodos de pronóstico de crecimiento basados en imágenes y en la manera en que las masas o tejidos densos pueden propagarse.

Esta tesis aborda la propuesta de un modelo híbrido que relaciona las propiedades de las imágenes, los tejidos densos y la propagación de fluidos en un medio para poder obtener contornos de áreas determinadas de la imagen y posteriormente simular una posible forma de propagación de los tejidos densos que pueden llevar a diagnosticar un cáncer.

Índice general

Índice de figuras	VIII
1. Introducción.	1
1.1. Presentación.	2
1.2. Motivación.	4
1.3. Objetivo.	5
1.4. Planteamiento del problema.	6
1.5. Metodología.	7
1.6. Contribuciones.	7
1.7. Estructura de la tesis.	8
1.8. Bibliografía capítulo 1.	9
2. Marco teórico	11
2.1. Introducción	11
2.2. Imágenes Médicas.	11
2.3. Segmentación y Contornos.	16
2.3.1. Modelos de bordes	19
2.3.2. Muestreo.	19
2.3.3. JPG	20
2.4. Autómatas Celulares de Gas.	21
2.5. Método de redes de Boltzmann (LBM).	22
2.5.1. La ecuación de transporte de Boltzmann	24
2.5.2. La ecuación de Boltzmann en redes.	26
2.6. LBM en contornos.	31
2.6.1. El modelo de redes de Boltzmann	32
2.7. Palabos en redes de Boltzmann.	34
2.7.1. Simulaciones de redes de Boltzmann.	34
2.8. Bibliografía capítulo 2.	39

3. Implementación de redes de Boltzmann para obtención de contornos en imágenes de cáncer de mama.	42
3.1. Introducción	42
3.1.1. Método de ajuste de nivel	42
3.1.2. Algoritmo para la obtención de contornos usando redes de Boltzmann.	43
3.1.3. Esténcil D2Q5.	45
3.1.4. Esténcil D2Q9.	47
3.2. Obtención de imágenes con contorno.	52
3.2.1. Máscara normalizada.	52
3.3. Bibliografía capítulo 3.	54
4. Análisis de Resultados	56
4.1. Resumen.	56
4.2. Resultados.	56
4.2.1. Acerca de la eficiencia en los bordes obtenidos.	58
4.2.2. Relación con la máscara obtenida.	59
4.3. Ecuaciones de Crecimiento.	61
4.4. Bibliografía capítulo 4.	64
5. Conclusiones	65
5.1. Conclusiones.	65
5.2. Trabajo futuro.	66
5.3. Bibliografía capítulo 5.	67
A. Apéndice: Un método rápido y robusto de conjuntos de nivel para la segmentación de imágenes utilizando el agrupamiento difuso y el método Lattice Boltzmann	68
A.1. Código	68
A.2. Bibliografía	73
B. Apéndice: Código D2Q9 en redes de Boltzmann	74
B.1. Características básicas y ecuaciones	74
B.2. Bibliografía	80
C. Apéndice: FEniCS para resolver ecuaciones diferenciales	81
C.1. Sistema de reacción-difusión-advección	81
C.2. Problema de Ecuaciones Diferenciales Parciales	81
C.3. Formulación variacional	83
C.4. Implementación en FEniCS	84
C.4.1. Código completo	87
C.5. Bibliografía	91

Índice de figuras

1.1. Modelo de crecimiento de tumores basados en probabilidad con distintas poblaciones iniciales [5].	3
2.1. Tomógrafo Philips Brilliance 64 CT.	12
2.2. Imagen de tomografía computarizada (Especializado May 21, 2015 Blog Médico, Radiología, Rayos X, Tomografía Axial Multicorte (TAC)).	13
2.3. Escala de Hounsfield para densidades del cuerpo humano.	14
2.4. Otra forma de representar la figura 2.3. Escala de Hounsfield para densidad de tejido en órganos.	15
2.5. Imágenes de resonancia magnética convencional.	16
2.6. Ejemplo. (a) Imagen con gotas sobre un fondo gris. (b) Imagen que muestra sólo los contornos de los objetos.	17
2.7. La sección transversal de un árbol. (a) Imagen original en gris. (b) Borde ideal de la imagen mejorada mostrando los anillos de crecimiento. c) La mejora del borde que se podría esperar utilizando un algoritmo real.	18
2.8. El efecto del muestreo en un borde escalonado. (a) Un borde de paso ideal. (b) vista tridimensional del escalón en el borde. (c) Borde de paso muestreado en el centro de un pixel, en lugar de en un margen. (d) El resultado, en tres dimensiones, tiene la apariencia de una escalera.	20
2.9. Vectores unitarios del modelo FHP.	22
2.10. Espacio μ con N partículas de masa m.	24
2.11. Esquema de velocidades c_α para la malla D2Q9 con valores 0,1 y $\sqrt{2}$. La velocidad c_0 se localiza en el centro.	31
2.12. Simulación D3Q19 de una malla de medio poroso.	37
2.13. Simulación D3Q19. Comparación de estrés y tensión en las partículas.	38
2.14. Simulación de una esfera calentada en un medio.	38

3.1. Obtención de contornos en imagen de rodilla [1].	44
3.2. Esquema del estencil para la malla D2Q5 con velocidades 0,1 y $\sqrt{2}$. La velocidad 0 se localiza en el centro.	46
3.3. Obtención de contornos en imagen de prueba.	46
3.4. Esquema del estencil para la malla D2Q5. Con coordenadas y pesos por columnas.	47
3.5. Variables de inicio.	48
3.6. Obtención de contornos en imagen de prueba.	48
3.7. Código D2Q5 modificado a D2Q9.	49
3.8. Código D2Q5 comparado con D2Q9.	50
3.9. Diagrama de flujo que describe el proceso de obtención de contornos en D2Q9 comparado con D2Q5.	51
3.10. Obtención de contornos. a) imagen original, b) región de interés y contorno de referencia, c) contorno obtenido, d) máscara obtenida.	53
4.1. Obtención de contornos. Fila 1: imagen de prueba. Fila 2: contorno con el método Chan-Vese. Fila 3: contorno con el método utilizado en este trabajo.	57
4.2. Valores de los pixeles antes de normalizarlos.	60
4.3. Valores de los pixeles después de normalizarlos.	61
B.1. Esquema del estencil para la malla D2Q9 con velocidades 0,1 y $\sqrt{2}$. La velocidad 0 se localiza en el centro.	75
C.1. Gráfica de concentraciones [A], [B] y [C] (desde arriba hacia abajo).	86
C.2. Gráfica de concentraciones [A], [B] y [C] en diferentes pasos de tiempo.	86

Capítulo 1

Introducción.

El método de redes de Boltzmann o "Lattice Boltzman Method" (LBM por sus siglas en inglés) es un método relativamente nuevo puesto que se ha utilizado desde el año 1985 en diferentes campos de la ciencia, principalmente para la simulación de fluidos con un alto grado de confiabilidad con respecto a los métodos numéricos comunes [2].

Es un método que se apoya en la Ley de Fick [3] y las ecuaciones de Fourier [4], y que se aplica para resolver problemas de transferencia de masa, momento y energía, simulando las propiedades de masa, temperatura y momento a diferentes escalas. En muchos casos son ecuaciones que son difíciles de resolver de forma analítica o que las aproximaciones no son lo suficientemente adecuadas a nuestras expectativas.

Se trata de ver en una macro escala cuál será el efecto de la interacción de pseudo partículas a nivel microscópico, de tal manera que, aunque tengamos una gran cantidad de pseudo partículas en un medio, analizaremos el efecto que tienen en conjunto todas ellas. Por tal motivo hay que tener muy en cuenta las características de velocidad, momento y fuerza con las que interactúan todas las partículas, ya que esto determina el efecto de transferencia de información (masa, calor, energía, etc) en cuestión dentro del medio que analizamos.

En este trabajo se realiza una asociación de las características de los fluidos y la intensidad de brillo de la imagen médica, tomando en cuenta que las variaciones de brillo se pueden expresar como flujos de información que se desplazan en un medio o rejilla y se pueden localizar como pseudo partículas con coordenadas en espacio y tiempo (tal es el caso de las imágenes), y además, se aplica este concepto al campo de las ciencias médicas, con lo cuál tenemos un amplio margen de trabajo y exploración de posibilidades.

Para poder trabajar las imágenes médicas como los medios en los que se desplaza nuestra información (los píxeles se consideran como la rejilla y las intensidades de brillo de los píxeles se consideran como un flujo de información para los fines del método de redes de Boltzmann) es necesario relacionar directamente la densidad del tejido de interés con su respectiva intensidad de brillo del píxel, por lo que abordaré la manera más adecuada de realizar esta asociación de los valores.

1.1. Presentación.

La idea de este trabajo surge en el año 2015, tras una exhaustiva revisión de posibilidades en las que se pueden aplicar las redes de Boltzmann, dicho esto, se desarrolló una propuesta inicial basada sólo en crecimiento de células cancerosas. Sin embargo es un estudio en el que encontró la característica de que el crecimiento parte de un conjunto de células sanas y que los tumores pueden crecer siguiendo una probabilidad que depende de cada individuo [5], tal como se muestra en la figura 1.1.

Donde P_s es la probabilidad de que las células cancerosas crezcan a un ritmo uniforme, por lo que, $P_s = 1$ aparenta ser una esfera. $P_s = 0.1$ representa que el crecimiento como mayor probabilidad de que crezcan células cancerosas sin ser uniforme en la periferia, y por último $P_s = 0.03$ nos indica que las células no crecen de manera controlada, por lo que para poblaciones de 150 000 células ya es una forma completamente irregular en su periferia. Por otro lado, la tabla nos indica también en color rojo que las células madre [6] comienzan a morir y proliferan las células necróticas en color gris (las células que ya se han convertido en un tumor).

Tomando en cuenta esta condición de poblaciones iniciales uniformes se decidió complementar la idea con la extracción de contornos de las mamografías. De manera que al partir de una mamografía, obtendría sólo el contorno del tejido sospechoso y usarlo como una condición inicial real del método y complementado con el uso de las redes de Boltzmann simular la propagación de células utilizando las reglas de crecimiento propuestas por Anderson et. al. [7] con las cuales se complementa la propuesta de modelo de crecimiento basado en una población inicial real y las características de consumo de nutrientes de células, grado de apoptosis (tasa de mortalidad de las células), consumo de oxígeno, entre otras.

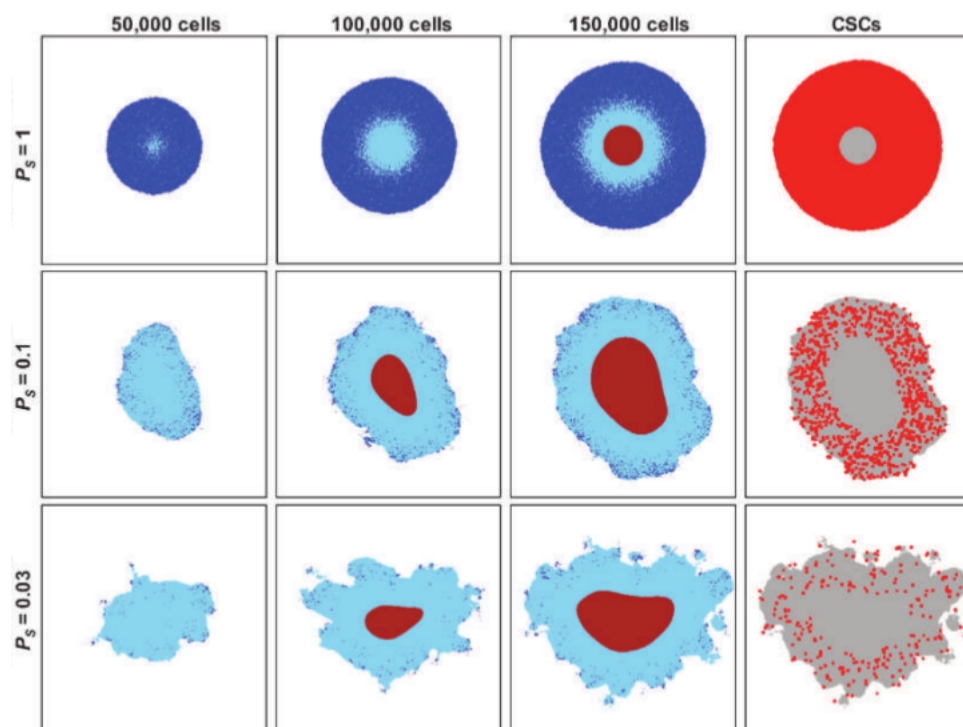


Figura 1.1: Modelo de crecimiento de tumores basados en probabilidad con distintas poblaciones iniciales [5].

1.2. Motivación.

El motivo por el cuál se llevó a cabo este trabajo es porque se considera que se deben reducir los tiempos de diagnóstico y además poder predecir qué tan rápido puede llegar a crecer un cáncer.

En México, el Instituto de Canceología y el Instituto de Física de la Universidad Nacional Autónoma de México publicaron en 2006 una media de tiempo basada en normas oficiales mexicanas, las cuales muestran estadísticas que indican que para que un cáncer pueda evolucionar de manera considerable, deben pasar cerca de 7 años, y se deben realizar mamografías de preferencia cada 6 meses para poder observar la evolución. Son datos que contrastan con estadísticas como por ejemplo en España, donde las mamografías son recomendadas cada dos años [8]. Tomando estos datos se puede decir que para un expediente médico de algún paciente se necesitan tomar imágenes en un intervalo de 2 por año. Transcurridos 7 años obtenemos para obtener 14 imágenes que muestren la evolución de un tejido y poder diagnosticar una calcificación o un cancer benigno o maligno, por lo que se reduce la oportunidad de poder atender a tiempo con el tratamiento adecuado. Por esta razón es una meta proponer una herramienta que ayude a simular el crecimiento y evolución de ese tejido sin tener que esperar mucho tiempo, aplicando las redes de Boltzmann como un método de aproximación a la solución de las ecuaciones que representan el fenómeno de crecimiento de las células del cuerpo humano. Esta propuesta en redes de Boltzmann se considera un problema computacional debido a la gran cantidad de ciclos e iteraciones necesarias para llegar al resultado, por lo que es un excelente punto de partida para utilizar las poderosas herramientas computacionales actuales combinadas con la implementación de métodos numéricos.

Esta y muchas herramientas más en conjunto pueden ayudar a mejorar la calidad de vida de las personas y a no reducir sus expectativas de vida.

Además de lo antes mencionado, la obtención de los parámetros necesarios para la simulación de crecimiento requieren como base la obtención de los contornos de interés, cuyas dimensiones y densidad nos brindarán el punto de partida para el modelo de crecimiento, por tal motivo, es una buena alternativa partir de la obtención de contornos por el mismo método. Esta manera de obtener los contornos de la imagen se basa en el método de conjuntos de nivel, resolviendo su ecuación por el método de redes de Boltzmann, de tal manera que así se puede introducir al lector en la idea general del proceso de simulación de crecimiento.

1.3. Objetivo.

Desde los años 90 se han hecho estimaciones acerca de la cantidad de personas que se diagnostican con algún tipo de cáncer, y especialmente de cáncer de mama, que es uno de los más comunes en la población tanto de hombres como de mujeres. En México el cáncer de mama es la segunda causa de muerte en mujeres. (Instituto Nacional de Estadística y Geografía e informática INEGI 2014 [9] y representa el 16 % de todos los cánceres en mujeres a nivel mundial según la Organización Mundial de la Salud.

El alto número de personas que lo padecen y los grandes tiempos diagnóstico han hecho que se necesiten herramientas eficientes para poder tener un diagnóstico oportuno y certero. Por esa razón se propone una herramienta computacional que ayude en primera instancia, y como objetivo principal de este trabajo, a la obtención de contornos en imágenes de cáncer de mama, utilizando redes de Boltzmann como una variante del método de ajustes de nivel en imágenes. Posteriormente, generar una nueva red de Boltzmann que tenga como condiciones iniciales dicho contorno obtenido en la primera parte, Con esto se obtendrá una simulación de crecimiento de anomalías en una zona específica del tejido. Esto puede ayudar al diagnóstico y a la determinación de la velocidad de crecimiento y así contribuir a la toma de decisiones en el tratamiento.

También se propone determinar un método de obtención de información de las imágenes médicas a partir del tratamiento digital de imágenes, utilizando bancos de imágenes públicas con el fin de hacer pruebas de confiabilidad del resultado obtenido, corroborar los resultados con imágenes de casos clínicos nuevos y formular conclusiones acerca de la eficiencia del método empleado.

Este trabajo tiene por objetivo general buscar la aplicación de un proceso de análisis digital de imágenes que permita resaltar regiones con posibles anomalías mediante técnicas híbridas de tratamiento de imágenes en las que, por un lado se toman en cuenta las condiciones biológicas de crecimiento de tumores a partir del brillo de los píxeles, y por otro lado se simula una propagación de dicho tejido con condiciones iniciales de frontera del dominio dadas por la imagen, todo esto ayudándose de los autómatas celulares que se basan en la mecánica estadística.

1.4. Planteamiento del problema.

El problema específico que se considera es obtener el contorno de un tejido considerado como una calcificación o un conjunto de células cancerosas a partir de una imagen de mamografía y extraer ese contorno de la imagen médica mediante el uso del método de Redes de Boltzmann. Posteriormente establecer ese contorno como una condición de concentraciones iniciales de las células y utilizar una nueva Red de Boltzmann acoplada a la inicial para modelar y simular cómo se difunden esas concentraciones a través de la rejilla en un intervalo de tiempo determinado.

El procedimiento está dividido en dos secciones:

Primera parte. La obtención de contornos y la generación de la máscara inicial para encontrar las concentraciones iniciales en base a la imagen.

En este trabajo se presenta la primera parte, con sus respectivas fases y resultados y en la sección final se plantea lo que se necesita para iniciar con la segunda parte del proceso general, por lo que nos centraremos en la obtención de contornos y los métodos utilizados.

Segunda parte. La generación de la nueva red de Boltzmann a partir de la máscara e iniciar la simulación de crecimiento basado en las ecuaciones que lo describen.

Se debe generar y extraer ese nuevo contorno y generar una imagen que represente la forma final que tendrá ese conjunto de células, por lo que se considera que es un problema de suma importancia para el área de las ciencias de la salud.

Debido a que este tipo de simulaciones se deben hacer por cada red de Boltzmann, se requiere mucho tiempo de cómputo y es necesario adaptarlas para ejecutar los algoritmos de manera serial en una computadora o en cómputo paralelo. Para ello se plantea la estrategia a seguir en los capítulos posteriores.

1.5. Metodología.

- Etapa inicial en donde se efectuará un estudio exhaustivo del estado del arte del procesamiento digital de señales, analizando las técnicas clásicas para el filtrado y restauración de imágenes, y todo lo referente a la teoría e implementación de las redes de Boltzmann [10] en particular aplicado a imágenes de tumores de cáncer de mama.
- Acotar el alcance del trabajo y enfocarse en la obtención de contornos mediante redes de Boltzmann.
- Analizar resultados de la obtención de contornos.
- Realizar la selección de parámetros que intervienen en el crecimiento de células cancerosas, fundamentado en el tratamiento digital de señales y realizar los respectivos algoritmos y su programación.
- Realizar los algoritmos y programas que simulen los fenómenos de reacción-difusión reales que ocurren en la evolución de un tumor.

La metodología consta también de marcar el avance por objetivos, y el objetivo inicial es realizar un algoritmo para la obtención de los contornos en las imágenes. Este trabajo requiere mucho tiempo de estudio debido a que es una alternativa poco explorada en ambos casos (imágenes y crecimiento) y se ha acotado el primer objetivo a corto plazo para poder alcanzarlo. Es necesario hacer una selección precisa de los parámetros de inicio y las ecuaciones de redes de Boltzmann, por lo que se considera que este trabajo tiene los elementos necesarios para ser estudiado en esta tesis de nivel de maestría.

El siguiente objetivo es generar un algoritmo computacional para simular crecimiento de tumores basados en su forma de crecimiento con variables físicas y químicas, por lo que se ha decidido trabajar a futuro en ello. Finalmente se trata de apoyar con una herramienta computacional al diagnóstico y tratamiento de los tumores de cáncer de mama.

1.6. Contribuciones.

La principal contribución de este trabajo es que se genera un modelo computacional que combina el tratamiento digital de imágenes con la segmentación de imágenes en redes de Boltzmann para asociar las propiedades de la imagen con

las concentraciones de células cancerosas en el cuerpo humano, posteriormente simular el probable crecimiento de un tumor de cáncer de mama y es altamente probable su implementación en cómputo paralelo debido a las características de no dependencia de cada nodo en las redes de Boltzmann. Esto quiere decir que, el fenómeno de colisión y propagación se puede analizar en cada nodo de manera local sin tener que depender del valor de concentraciones en un nodo con respecto a otro. También es probable su implementación en un futuro para diferentes tipos de cáncer, aunque es objeto de otro estudio más profundo.

1.7. Estructura de la tesis.

Este trabajo está dividido en 5 capítulos en los que se desarrolla el modelo de crecimiento y simulación de las Redes de Boltzmann a partir de imágenes médicas, la implementación de algoritmos computacionales, las características que se consideran en nuestras imágenes y el resultado de la aplicación del método.

En el capítulo 1 se establecen los objetivos, la motivación y la metodología utilizada para entender el problema y cuáles son los alcances a los que llega este trabajo.

En el capítulo 2 se encuentra una breve introducción al estado del arte de los métodos actuales para ayudar al diagnóstico de cáncer de mama, el tratamiento de las imágenes obtenidas de las mamografías y cuales son tejidos y densidades de tejidos que son relevantes para su posterior estudio. También se plantea la base teórica de las redes de Boltzmann y su relación con el tratamiento de imágenes. Se hace una descripción de lo que son los autómatas celulares y sus ventajas en el momento de realizar simulaciones de sistemas dinámicos.

En el capítulo 3 se explica detalladamente cómo aplicar las Redes de Boltzmann para la obtención de contornos en imágenes, basado en un algoritmo computacional propuesto por Souleymane Balla-Arabé, Xinbo Gao y Bin Wang [11] y la extracción del contorno obtenido como una máscara normalizada. De igual forma se describen las ecuaciones de crecimiento de células propuesto por Anderson et. al. [12], con las que se establecen las condiciones iniciales de la simulación y las condiciones de frontera del dominio apoyados en el método de solución de ecuaciones diferenciales parciales (EDP) en dos dimensiones para un sistema de ecuaciones diferenciales parciales no lineales acopladas.

En el capítulo 4 se discutirán los resultados obtenidos de la generación de contornos, extracción de los contornos y la manera en que se pueden hacer más eficientes los algoritmos de cómputo utilizados. También se discutirá acerca del modelo de crecimiento de células cancerosas y la implementación en un nuevo modelo que conjunte los contornos obtenidos para utilizarlos como base para una nueva red, a partir de los resultados obtenidos de las EDP.

En el capítulo 5 se concluye el resultado obtenido de todos los experimentos y propuestas realizadas en este trabajo y se pone de manifiesto el trabajo a futuro.

1.8. Bibliografía capítulo 1.

- [1] Morrow M, Burstein HJ, Harris JR. Chapter 79: Malignant Tumors of the Breast. In: DeVita VT, Lawrence TS, Rosenberg SA, eds. DeVita, Hellman, and Rosenberg's Cancer: Principles and Practice of Oncology. 10th ed. Philadelphia, Pa: Lippincott Williams & Wilkins; 2015.
- [2] Seminar. Lattice Boltzmann Method. Igor Mele, Iztok Tiselj. University of Ljubljana, March 2013.
- [3] Quasichemical Models of Multicomponent Nonlinear Diffusion. Alexander N. Gorban, Hrachya P. Sargsyan and Hafiz A. Wahab. Department of Mathematics, University of Leicester, Leicester, Republic of Armenia. 2011.
- [4] N. Gershenfeld. The nature of mathematical modeling. Cambridge University Press, Cambridge, 1999.
- [5] Cancer Stem Cell Tumor Model Reveals Invasive Morphology and Increased Phenotypical Heterogeneity. Andrea Sottoriva, Joost J.C. Verhoeff, Tijana Borovski, Shannon K. McWeeney, Lev Naumov, Jan Paul Medema, Peter M.A. Slood and Louis Vermeulen. Cancer Res January 1 2010 (70) (1) 46-56; DOI: 10.1158/0008-5472.CAN-09-3663
- [6] Reya T, Morrison SJ, Clarke MF, Weissman IL. Stem cells, cancer, and cancer stem cells. Nature 2001;414:105–11.
- [7] Anderson, Alexander. (2005). A hybrid mathematical model of solid tumour invasion: The importance of cell adhesion. Mathematical medicine and biology : a journal of the IMA. 22. 163-86. 10.1093/imammb/dqi005.

- [8] María Ester Brandan y Yolanda Villaseñor Navarro. Detección del Cáncer de Mama: Estado de la Mamografía en México. Instituto de Física, Universidad Nacional Autónoma de México. Instituto Nacional de Cancerología. (2006) : 147-162.
- [9] Instituto Nacional de Geografía y Estadística e Informática INEGI. 2017 <http://www.inegi.org.mx/>
- [10] Sauro Succi. The Lattice Boltzmann Equation. For Fluid Dynamics and Beyond. Oxford, 2001.
- [11] Souleymane Balla-Arabé, Xinbo Gao, and Bin Wang. A Fast and Robust Level Set Method for Image Segmentation Using Fuzzy Clustering and Lattice Boltzmann Method. Junio 2013.
- [12] A. R. A. Anderson, M. A. J. Chaplain, E. L. New Man, R. J. C. Steele and A. M. Thompson. Mathematical Modelling of Tumour Invasion and Metastasis. Department of Mathematics, University of Dundee, Scotland. August 1999.

2.1. Introducción

En este capítulo se plantean los distintos conceptos necesarios para poder relacionar las redes de Boltzmann con el tratamiento de imágenes, el concepto de imágenes médicas y la escala de medida en que se utilizan, además de la implementación de redes en la obtención de contornos de éstas imágenes.

2.2. Imágenes Médicas.

El mundo de la imagenología médica se ha visto beneficiado por dos principales acontecimientos. El primero de ellos, fue el descubrimiento de los rayos-X por el físico Wilhelm Konrad Röntgen en 1895, quien alcanzó a diferenciar y representar en una película fotográfica el grado de atenuación de estos rayos en los diferentes tejidos de una mano, visualizándose por primera vez la piel y el hueso. Este acontecimiento marcó el inicio de la imagenología médica. Posteriormente, en 1968 aconteció el descubrimiento de la tomografía computarizada (TC) por Godfrey Hounsfield [1], surgiendo la necesidad de almacenamiento y manipulación de las imágenes médicas para el diagnóstico basado en imágenes digitales. Este descubrimiento, proporcionó por primera vez la representación tridimensional de objetos a partir de la representación de imágenes rayos-X individuales, apiladas paralelamente en un volumen.

Desde entonces, se han desarrollado técnicas en la adquisición de imágenes de diferentes modalidades, las cuales son diferenciadas por la naturaleza de los principios físicos involucrados en el proceso de adquisición de las imágenes, ver figura 2.1. Las modalidades más comunes de imagenología médica son la tomografía computarizada (TC) que mide el coeficiente de atenuación de los rayos X, la resonancia magnética (RM) que mide las ondas de radio frecuencia y campos magnéticos ver figura 2.2. La imagenología por ultrasonidos (US) que representa

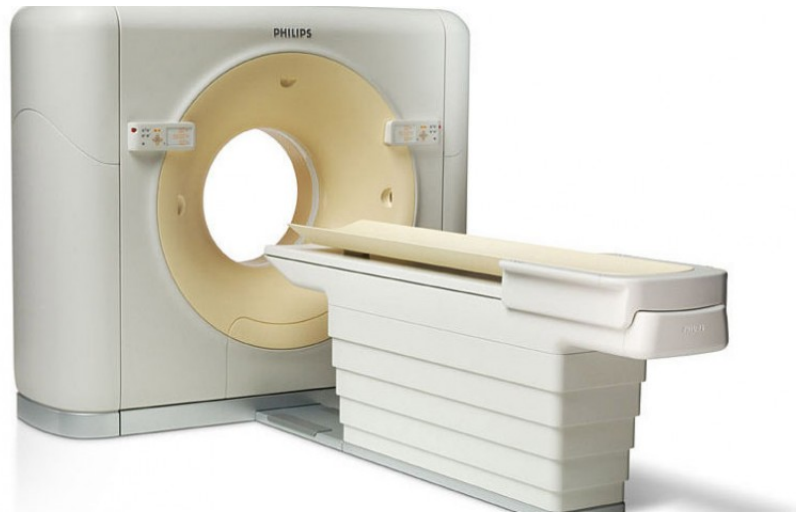


Figura 2.1: Tomógrafo Philips Brilliance 64 CT.

la atenuación de ondas acústicas, y la imagenología por medicina nuclear (MN), entre otras. En este trabajo se utilizaron las imágenes de TC y RM.

El alto grado de aceptación de las técnicas de imagenología médica y los diversos fabricantes de equipos médicos, originaron la aparición de diferentes especificaciones para el manejo de estas imágenes, por lo que integrar todas ellas en un sistema común para su almacenamiento y transferencia de manera flexible, era prácticamente imposible. Como consecuencia, en el año 1983, se inició la estandarización del proceso de manipulación de imágenes médicas digitales, desarrollándose el estándar Digital Imaging and Communications in Medicine DICOM [2], como iniciativa del Colegio Estadounidense de Radiología (ACR) y la Asociación Nacional de Fabricantes Eléctricos (NEMA). Este estándar describe el formato de archivos y la especificación de los datos primordiales de un paciente en la imagen, así como como el encabezado requerido para conocer las características de las imágenes obtenidas, describiendo un lenguaje común a distintos sistemas médicos. [3]

La aplicación de las rutinas planteadas en los casos de estudio analizados en imágenes médicas en formato DICOM, sin embargo, también se implementaron rutinas para leer imágenes en otros formatos como: JPG, RAW, BMP, VTK, los cuales son muy utilizados en el mundo de la visualización médica.

La tomografía computarizada (TC) es una técnica de diagnóstico utilizada en medicina, la cual al ser empleada en el cuerpo humano, obtiene cortes transver-

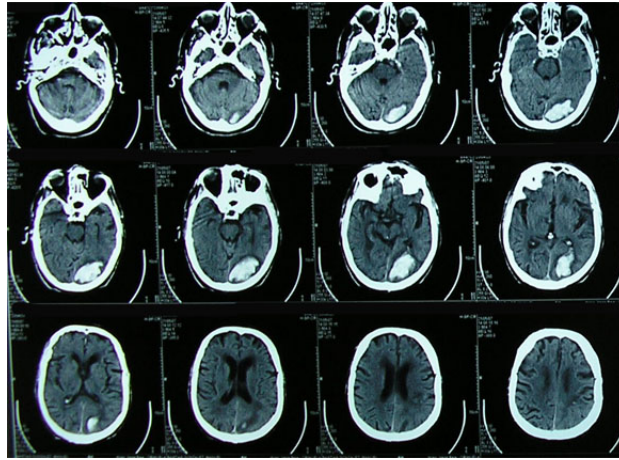


Figura 2.2: Imagen de tomografía computarizada (Especializado May 21, 2015 Blog Médico, Radiología, Rayos X, Tomografía Axial Multicorte (TAC)).

sales a lo largo de una región concreta del cuerpo (o del cuerpo entero). La TC es una técnica de imagenología de exploración de rayos X que produce imágenes detalladas de cortes del cuerpo. En lugar de obtener una imagen como la radiografía convencional, la TC obtiene múltiples imágenes al rotar alrededor del cuerpo. Una computadora combina todas estas imágenes en una imagen final que representa un corte del cuerpo como si fuera una rodaja, obteniéndose múltiples imágenes en rodajas o cortes (slices) de la parte del cuerpo que está siendo estudiada. Cada imagen o corte representa un perfil de intensidades medido por los detectores del tomógrafo, el cual es reconstruido empleando la transformada de Radon [4] que tiene sus bases en el teorema de Fourier [5].

En las imágenes de los cortes originales, a cada elemento pixel de la tomografía computarizada se le asigna un valor numérico que representa el valor de atenuación del haz radiológico absorbido por el tejido en ese punto, resultando la densidad de la tomografía directamente proporcional al coeficiente de absorción. El nivel de gris de cada pixel se expresa en unidades de Hounsfield (HU).

El coeficiente de atenuación de los rayos X expresado en Unidades Hounsfield para los diferentes tejidos vivos u otros materiales sometidos a un examen tomográfico de rayos X viene dado por la siguiente fórmula:

$$HU = (1000) \left[\frac{\mu_t - \mu_{agua}}{\mu_{agua}} \right] \quad (2.1)$$

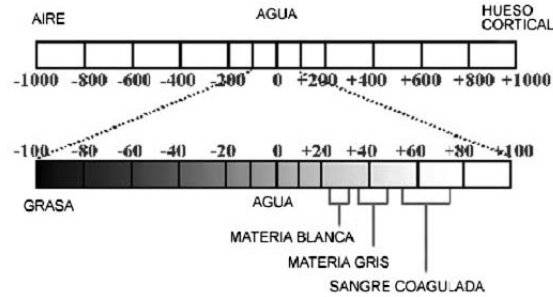


Figura 2.3: Escala de Hounsfield para densidades del cuerpo humano.

Donde μ_{agua} es el coeficiente de atenuación lineal del agua destilada y μ_t el coeficiente de atenuación lineal del tejido o material de interés.

Para la reconstrucción 3D a partir de la tomografía, se consideran los siguientes parámetros:

1. Resolución espacial. Compuesta por el número de cortes y el número de píxeles de cada corte. Este parámetro es representado por una matriz $3D$ de la imagen, cuyas medidas x, y dependen del tamaño en píxeles de cada corte y z depende del número de cortes obtenidos. Por ejemplo, en un conjunto de datos de TC con una matriz de imagen de 512 píxeles x 512 píxeles y 300 cortes, obtendremos un volumen de datos con la resolución 512 x 512 x 300.
2. La distancia entre voxeles (voxel spacing). Es calculada a partir de la distancia en x, y de los píxeles dentro de cada corte (en el plano) y la distancia z de los píxeles entre dos cortes, llamado grosor del corte (fuera del plano). Si x, y, z tienen mismo valor, se tendrían voxeles isotrópicos, caso contrario se denominan anisotrópicos.

En el momento de reconstruir cada corte, el valor de intensidad calculado para cada píxel representa la densidad del objeto escaneado. En las imágenes médicas, estos valores de intensidad son normalizados en unidades Hounsfield (HU). La normalización de los mapas del rango de datos para una representación numérica de 12 bits, se asume como la intensidad del agua en valor cero y el aire con valor -1000. Esta normalización se formaliza con la ecuación 2.1.

En esta primera figura se observa que los valores están dentro de un amplio rango de posibilidades, pero que están bien definidos para zonas específicas del

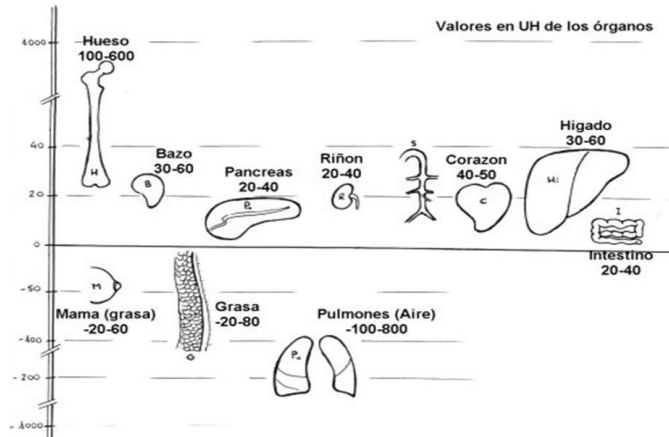


Figura 2.4: Otra forma de representar la figura 2.3. Escala de Hounsfield para densidad de tejido en órganos.

cuerpo. Se puede distinguir grasa, hueso o identificar aire.

En la figura 2.3 se puede encontrar el valor de escala que nos interesa, que es en tejidos grasos en mamas, de un valor aproximado de entre -20 y -60 HU. Con este valor podemos centrarnos ya en encontrar su correspondiente valor en la escala de grises de la imagen.

Resonancia magnética (RM).

El descubrimiento de la resonancia magnética (RM) se atribuye a dos grupos de investigadores independientes: (a) Felix Bloch de la Universidad de Stanford y colaboradores y (b) Edward Purcell de la Universidad de Harvard y colaboradores, quienes en 1945 decidieron investigar el comportamiento de la resonancia magnética nuclear en materia condensada.

Desde entonces, la resonancia magnética se ha convertido en una sofisticada técnica con aplicaciones en una amplia variedad de disciplinas que incluyen ahora la física, química, biología y medicina.

Un equipo de RM está constituido por un conjunto de aparatos emisores de electromagnetismo, antenas receptoras de radio frecuencias y computadoras, las cuales analizan datos para producir imágenes detalladas en 2D o 3D, con un gran nivel de precisión que permite detectar, o descartar, alteraciones en los órganos y los tejidos del cuerpo humano. La obtención de las imágenes de RM se pro-

duce sin la intervención de radiaciones ionizantes (rayos gama o X), sometiendo al paciente a un campo electromagnético con un imán que atrae a los protones que están contenidos en los átomos de hidrógeno que conforman los tejidos humanos, los cuales, al ser estimulados por ondas de radio frecuencia, salen de su alineamiento normal. Cuando el estímulo se suspende, los protones regresan a su posición original, liberando energía que se transforma en señales de radio para ser captadas por la computadora que las transforma en imágenes, las cuales describen la forma y funcionamiento de los órganos. Ver figura 2.5.

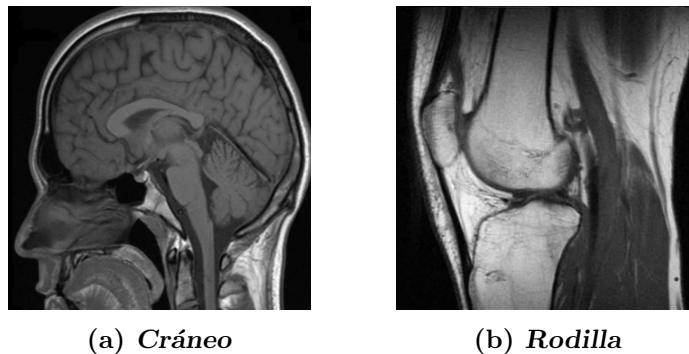


Figura 2.5: Imágenes de resonancia magnética convencional.

De manera muy similar a las imágenes de TC, en los cortes originales de RM, a cada elemento pixel se le asigna un valor numérico que representa el valor de atenuación de la onda de radiofrecuencia incidente en el tejido en ese punto.

Las imágenes de RM permiten visualizar de mejor manera los tejidos blandos, aunque también permiten ver el hueso. Sin embargo la TC es ideal para la visualización de estructuras óseas.

2.3. Segmentación y Contornos.

La detección de bordes es una de las operaciones más utilizadas en el análisis de imágenes, y hay probablemente más algoritmos en la literatura para mejorar y detectar bordes que cualquier otra cosa. La razón de esto es que los bordes forman el contorno de un objeto, en un sentido muy generalizado. Los objetos son temas de interés en sistemas de análisis de imágenes y de visión. Un borde es el límite entre un objeto y el fondo e indica el límite entre la superposición de objetos [6]. Esto significa que si los bordes de una imagen pueden ser identificados con

precisión, todos los objetos se pueden localizar, y las propiedades básicas, tales como el área, perímetro y forma pueden ser medidas.

Dado que el tratamiento de imágenes involucra identificación y clasificación de objetos en una imagen, la detección de bordes se convierte en una herramienta esencial.

En la figura 2.6, se ilustra un ejemplo directo de detección de bordes. Existen dos objetos superpuestos en la imagen original: la imagen de la izquierda contiene un color gris uniforme de fondo y la imagen de la derecha es la versión mejorada de la misma imagen de la izquierda, pero tiene líneas negras que delimitan los tres objetos. Hay que tener en cuenta que no hay manera de saber qué partes de la imagen son de fondo y cuales son objetos, sólo los límites entre las regiones se identifican. Sin embargo, dado que las figuras en la imagen son las regiones, se puede determinar que el objeto número 3 cubre una parte del objeto 2 y por lo tanto está en un primer plano.

La detección de bordes es parte de un proceso llamado segmentación (la identificación de regiones dentro de una imagen). Las regiones que pueden ser objetos de la figura 2.6 se han aislado y el procesamiento posterior determinará qué clase de objeto representa cada región. Mientras que en este ejemplo la detección de bordes es simplemente un paso en el proceso de segmentación, a veces es todo lo que se necesita, especialmente cuando los objetos en una imagen son líneas.

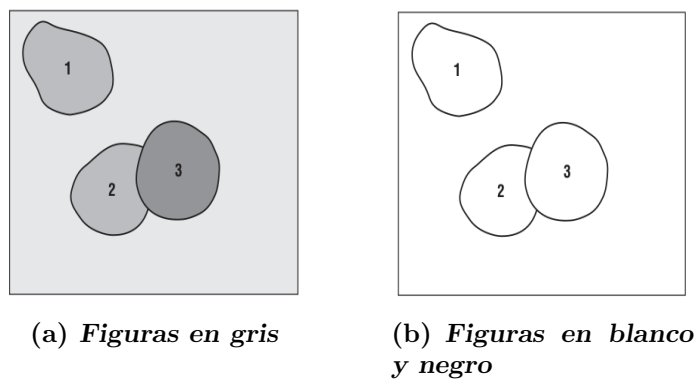


Figura 2.6: Ejemplo. (a) Imagen con gotas sobre un fondo gris. (b) Imagen que muestra sólo los contornos de los objetos.

Consideremos la imagen de la figura 2.7, que es una fotografía de la sección transversal de un árbol. Los anillos de crecimiento son los objetos de interés en

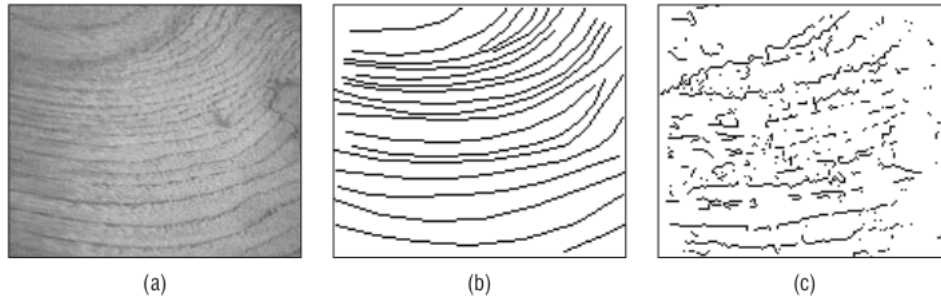


Figura 2.7: La sección transversal de un árbol. (a) Imagen original en gris. (b) Borde ideal de la imagen mejorada mostrando los anillos de crecimiento. c) La mejora del borde que se podría esperar utilizando un algoritmo real.

esta imagen. Cada anillo representa un año de la vida del árbol, y el número de anillos es por lo tanto el mismo que la edad del árbol. Encontrar los anillos usando un detector de bordes como se muestra en la Figura 2.7 es todo lo que se necesita para segmentar la imagen en primer plano (los objetos son los anillos) y fondo (todo lo demás).

Técnicamente, la detección de borde es el proceso de localización de los píxeles de borde, y la mejora de los bordes es el proceso de aumentar el contraste entre los bordes y el fondo para que los bordes se vuelvan más visibles. En la práctica, sin embargo, los términos se utilizan indistintamente, ya que la mayoría de los programas de detección de píxeles del borde usan un nivel o color de gris específico para que puedan ser fácilmente vistos. Además, el rastreo de bordes es el proceso de seguir los bordes, generalmente recoger los píxeles de borde en una lista. Esto se hace en una dirección consistente, en sentido horario o antihorario alrededor de los objetos.

La codificación de cadena es un ejemplo de un algoritmo específico para el rastreo de bordes. El resultado es una representación de los objetos que pueden usarse para calcular medidas de forma o de lo contrario identificar o clasificar el objeto.

Existen otras propuestas de detección de bordes incluyendo métodos tradicionales como el de Canny y el detector de bordes Shen-Castan, o ISEF. Ambos se basan sólidamente en consideraciones teóricas, y ambos reivindican un grado de optimalidad; es decir, ambos afirman ser lo mejor que se puede hacer bajo ciertas circunstancias especificadas.

Estas afirmaciones serán examinadas en la práctica en el capítulo 3 de esta

tesis.

2.3.1. Modelos de bordes

El borde escalonado de la Figura 2.8 es ideal porque es fácil de detectar en ausencia del ruido, cualquier cambio significativo en el nivel de gris indicará que hay un borde. Un borde escalonado nunca ocurre realmente en una imagen porque: a) los objetos rara vez tienen un contorno tan definido, b) una escena nunca es muestreada de manera que los bordes estén exactamente dentro del margen de un pixel, y c) debido al ruido. Es decir, al haber varios valores intermedios entre los valores límites de las capas en primer plano y los objetos de fondo.

2.3.2. Muestreo.

Las figuras 2.8a y 2.8b muestran un borde de paso ideal y el conjunto de pixeles implicados. Hay que tener en cuenta que el borde se produce en el extremo izquierdo de los pixeles del borde blanco. A medida que la cámara se desplaza hacia la izquierda en cantidades menores que el ancho de un pixel, el borde se mueve hacia la derecha. En la Figura 2.8c el borde se ha movido a la mitad de un pixel, y los pixeles a lo largo del borde ahora contienen parte de la imagen en negro y algunas partes en blanco. Esto se reflejará en el nivel de gris como un promedio ponderado:

$$v = \frac{v_w a_w + v_b a_b}{a_w - a_b} \quad (2.2)$$

donde v_w y v_b son los niveles de gris de las regiones blancas y negras, y a_w y a_b son las áreas de las partes blancas y negras de los pixeles del borde. Por ejemplo, si el nivel de blanco es 100 y el nivel de negro es 0, entonces el valor de un pixel en el borde para el cual el borde pasa a través del centro será 50. El resultado es un doble paso en lugar de un solo paso en el borde, como se muestra en la Figura 2.8d.

Si el efecto de un contorno borroso es extender el cambio de nivel de gris sobre un número de pixeles, entonces el único escalón se convierte en una escalera. La rampa es un modelo en el que el borde debía haberse visto originalmente para producir una escalera, y por lo tanto es una idealización, una interpolación de los datos encontrados.

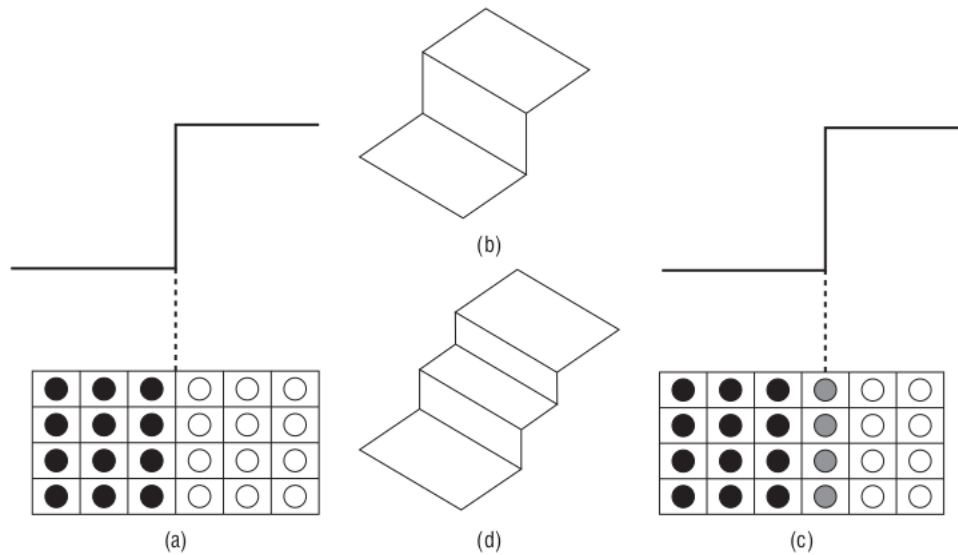


Figura 2.8: El efecto del muestreo en un borde escalonado. (a) Un borde de paso ideal. (b) vista tridimensional del escalón en el borde. (c) Borde de paso muestreado en el centro de un pixel, en lugar de en un margen. (d) El resultado, en tres dimensiones, tiene la apariencia de una escalera..

2.3.3. JPG

JPG o JPEG (para Joint Photographic Experts Group) [7], es el formato de archivo más común en formato digital fotografía. Soporta una profundidad de color de 24 bits (3 canales de color de 8 bits cada uno). Casi todos los dispositivos de adquisición son capaces de escribir imágenes utilizando este formato y es ampliamente apoyado en diferentes programas de visualización. Produce archivos pequeños y compresión con pérdidas. Cada vez que se guarda una imagen en este formato, la calidad se degrada porque se realiza una compresión con pérdidas. La compresión con pérdida elimina los detalles de la imagen de modo que pueda representarse con menos información. Utiliza análisis matemáticos complejos para hacer menos perceptible aquellos bits de información de la imagen para los que el ojo humano no puede distinguir la diferencia. Tenga en cuenta que puede ver una imagen JPG tantas veces como quiera sin alterar su calidad; la calidad se degrada sólo cuando se guarda. Además, la pérdida de calidad en la mayoría de los casos es muy pequeña y casi imperceptible por el ojo humano si se guarda en un ajuste de alta calidad. Sin embargo, guardar un archivo JPG varias veces puede llevar a una considerable pérdida de calidad de imagen. Por lo tanto, se recomienda que las imágenes digitales se guarden en otros formatos mientras se realiza el procesamiento sobre ellas, y sólo guardar a JPG como el último paso cuando no hay más modificaciones que hacer.

Tal es el caso de las imágenes utilizadas en este trabajo de tesis, se guardan en formato JPG porque al momento de la adquisición no necesitan ningún tratamiento en especial, por lo que pueden almacenarse de inmediato, además de que es el formato estándar para visualización de archivos en todos los sistemas operativos.

2.4. Autómatas Celulares de Gas.

Por definición un autómata celular (CA por sus siglas en inglés) es una entidad que ocupa una posición en una malla o un punto de una rejilla y que interactúa con sus vecinos idénticos. Generalmente examina su propio estado y el estado de sus vecinos en un determinado paso de tiempo, luego reinicia su propio estado para evaluar el siguiente paso de tiempo siguiendo una serie de reglas. Además de estas características se cuenta también con condiciones de frontera iniciales las cuales gobiernan el comportamiento del autómata y su evolución en el tiempo [8].

El método de redes de Boltzmann tiene sus orígenes en los autómatas celulares de gas, que son un modelo que cuenta con una malla discreta y un tiempo discreto, comenzando con los primeros modelos hexagonales en donde Frish, Hasslacher y Pomeau obtuvieron las ecuaciones de Navier-Stokes por primera vez de manera correcta. El modelo es conocido como FHP (Frish, Hasslacher y Pomeau) desde ese momento [9].

El modelo se construye a partir de triángulos equiláteros que forman un malla al tener puntos comunes. A partir del punto central se trazan líneas conocidas como “unidades de la red” (Lattice Unit. LU por sus siglas en inglés) y todas las partículas tienen una sola velocidad $v = LU/\delta t$, es decir, una velocidad dada por la distancia de cada partícula con respecto de la partícula central y el tiempo que tarda en desplazarse. En cada punto de la malla, pueden haber hasta 6 partículas que corresponden a las 6 posibles direcciones e_i del punto central $e_i = (\cos\frac{\pi a}{3}, \sin\frac{\pi a}{3})$ donde $a = 1, 2, \dots, 6$ y e_i es cada velocidad que parte del origen $(0,0)$ en coordenadas cartesianas $(\cos\frac{\pi a}{3}, \sin\frac{\pi a}{3})$.

Al ser un modelo discretizado en tiempo y espacio, sus velocidades también son discretas y corresponden a cada posible dirección en la que se desplazan las partículas de la malla [10].

En esencia, estos métodos son ampliamente utilizados en dinámica de fluidos, donde las partículas atómicas se mueven de acuerdo a las leyes de Newton. Sin embargo, estas herramientas realizan una descripción mesoscópica de un fluido, aún

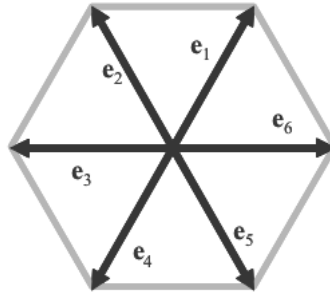


Figura 2.9: Vectores unitarios del modelo FHP.

cuando no necesariamente representan como partículas a los átomos individuales. Pueden representar conjuntos de partículas, campos cuantizados o transferencia de información o intensidad, como es el caso de este trabajo de tesis en el manejo de las imágenes.

2.5. Método de redes de Boltzmann (LBM).

El método de redes de Boltzmann es un modelo prometedor y altamente fácil de implementar, en términos de facilitar la solución numérica de sistemas de diferentes ecuaciones diferenciales hiperbólicas.

Los modelos basados en LBM son un poco diferentes a los de aproximaciones en diferencias finitas, elemento finito y volumen finito. Por un lado en una precisión microscópica, y por otro lado por una precisión macroscópica (Navier-Stokes eq) [11].

El punto clave es el siguiente: A pesar de que la dinámica microscópica depende en gran parte de la naturaleza del fluido considerado (gases y líquidos), se puede decir que la descripción macroscópica de un fenómeno es generalmente más insensible a las interacciones subyacentes inherentes en la escala microscópica a las interacciones a lo largo de las partículas, lo cual quiere decir que la probabilidad de que el fenómeno a micro escala se reproduzca en macro escala es muy alta. A su vez, un mismo sistema macroscópico es consistente con un conjunto de sistemas microscópicos. Siguiendo la aproximación estadística clásica de Boltzmann [12], este tratamiento se conoce como multiescala y ha trascendido al mundo de la física.

Por esta razón es posible diseñar una descripción microscópica idealizada, la cual, sin embargo, nos lleva a recobrar las ecuaciones macroscópicas. Además de esto, los gases reales o líquidos reales, se consideran un ensamble de micro mundos artificiales de pseudo partículas que “viven” en mallas con interacciones que conservan masa y momento. La micro dinámica de muchos micro mundos sería muy simple en una computadora para simularla eficazmente. Consideremos por ejemplo una malla cuadrada con cuatro celdas en cada lado, en los cuales una célula está asociada con otra a través de un nodo vecino. Esas celdas deben estar vacías u ocupadas por a lo mucho una partícula con masa unitaria, por lo que cada celda tiene solo dos posibles estados.

La velocidad y también el momento pueden ser asignados a cada partícula por el vector que conecta un nodo con su siguiente nodo vecino. La interacción microscópica es estrictamente local y por lo tanto masivamente paralelizable, debido a que envuelve sólo una partícula en un único nodo. Las partículas intercambian momentos o transfieren información mientras conservan masa y los momentos se suman en cada nodo (estas son las colisiones).

En pocas palabras, estamos asumiendo una relajación del sistema fuera de su estado de equilibrio hacia su equilibrio en un tiempo de relajación para uno o varios casos complejos. Los tiempos de relajación determinan la dinámica microscópica, macroscópica y los coeficientes de transporte al mismo tiempo. Una vez definido el modelo mesoscópico las constantes de tiempo de relajación son prácticamente parámetros controlados para ambos casos, microscópico y macroscópico.

El algoritmo para seleccionar las constantes de tiempo de relajación es llamado estrategia de búsqueda mesoscópica.

El método de redes de Boltzmann ha mostrado en los últimos años, ser una herramienta computacional confiable para realizar simulaciones de problemas de fluidos complejos. Con este método se han simulado problemas clásicos como el de Rayleigh-Bénard [13], flujo a través de obstáculos (con condiciones de frontera inmersas), propagación de ondas de sonido y turbulencia. Además de haber presentado excelentes resultados en otras áreas como la segmentación de imágenes [14], y la creación de simulaciones de crecimiento celular, entre otras.

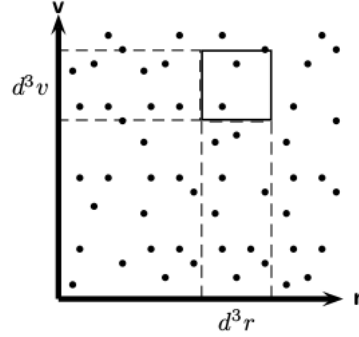


Figura 2.10: Espacio μ con N partículas de masa m .

2.5.1. La ecuación de transporte de Boltzmann

Si consideramos un gas mono atómico formado por N partículas cada una con masa m y confinadas en un volumen v , el estado del gas al tiempo t está determinado por la posición \mathbf{r} y velocidad \mathbf{v} de cada una de las partículas. Dicho gas lo podemos representar en un espacio de seis dimensiones, tres para la velocidad y las tres restantes para la posición, conocido como el espacio μ , que se muestra esquemáticamente en la figura 2.10. La ecuación de transporte de Boltzmann describe el cambio temporal y espacial de la función de distribución de velocidades f en el espacio μ que definimos como:

$$f(\mathbf{r}, \mathbf{v}, t) d^3r d^3v = \begin{cases} \text{distribución de probabilidad de partículas que se encuentran} \\ \text{en el tiempo } t \text{ en un elemento de volumen } d^3r \text{ alrededor de } \mathbf{r} \\ \text{con velocidades } d^3v \text{ alrededor de } \mathbf{v}. \end{cases} \quad (2.3)$$

En ausencia de colisiones, una partícula que se encuentra en (\mathbf{r}, \mathbf{v}) al tiempo t se encontrará en $(\mathbf{r} + \mathbf{v}\delta t, \mathbf{v} + (F/m)\delta t)$ al tiempo $t + \delta t$, siendo F una fuerza externa actuando sobre el gas. Entonces podemos escribir:

$$f(r, v, t) d^3r d^3v = f(r + v\delta t, v + (F/m)\delta t, t + \delta t) d^3r' d^3v' \quad (2.4)$$

Si δt es pequeño se puede demostrar que:

$$d^3r d^3v = d^3r' d^3v' \quad (2.5)$$

Por lo tanto podemos decir que:

$$f(\mathbf{r}, \mathbf{v}, t) = f(\mathbf{r} + \mathbf{v}\delta t, \mathbf{v} + (F/m)\delta t, t + \delta t) \quad (2.6)$$

Al tomar en cuenta las colisiones entre partículas, la diferencia entre las funciones de distribución ya no es cero, por lo que obtenemos:

$$f(\mathbf{r} + \mathbf{v}\delta t, \mathbf{v} + (F/m)\delta t, t + \delta t) - f(\mathbf{r}, \mathbf{v}, t) = \left(\frac{\partial f}{\partial t} \right)_{col} \delta t \quad (2.7)$$

donde el lado derecho en la ecuación anterior es el término de colisión. Expandiendo en series de Taylor el lado izquierdo de la ecuación a primer orden en δt obtenemos:

$$\left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} + \frac{F}{m} \cdot \nabla_{\mathbf{v}} \right) f(\mathbf{r}, \mathbf{v}, t) = \left(\frac{\partial f}{\partial t} \right)_{col} \quad (2.8)$$

donde:

$$\nabla_{\mathbf{r}} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right), \quad \nabla_{\mathbf{v}} = \left(\frac{\partial}{\partial u}, \frac{\partial}{\partial v}, \frac{\partial}{\partial w} \right), \quad (2.9)$$

x, y y z se refieren a los tres ejes espaciales y u, v y w las componentes de la velocidad para cada uno de los ejes respectivamente. La densidad de partículas n , la velocidad \mathbf{u} y la energía térmica ε son momentos de la función de distribución:

$$n(\mathbf{r}, t) = \int f(\mathbf{r}, \mathbf{v}, t) d^3v \quad (2.10)$$

$$\mathbf{u}(\mathbf{r}, t) = \frac{1}{n} \int \mathbf{v} f(\mathbf{r}, \mathbf{v}, t) d^3v \quad (2.11)$$

$$\varepsilon(\mathbf{r}, t) = \frac{1}{n} \int \frac{m}{2} (\mathbf{v} - \mathbf{u})^2 f(\mathbf{r}, \mathbf{v}, t) d^3v \quad (2.12)$$

respectivamente, donde $\rho = nm$ es la densidad de masa. Boltzmann supuso que el gas es diluido por lo que su comportamiento termodinámico debe corresponder a un gas ideal. Por ello, puede definirse una temperatura local a partir de:

$$\varepsilon(\mathbf{r}, t) = \frac{D}{2} nkT \quad (2.13)$$

donde D es la dimensión espacial, T la temperatura local y k la constante de Boltzmann.

Boltzmann encontró una forma explícita para el término de colisión que es:

$$\left(\frac{\partial f}{\partial t} \right)_{col} = \int \sigma(\Omega) d\Omega \int d^3v_2 |\mathbf{v}_2 - \mathbf{v}_1| (f'_1 f'_2 - f_1 f_2) \quad (2.14)$$

donde $\sigma(\Omega)$ es la sección diferencial de choque, Ω es el ángulo sólido de dispersión para la colisión entre dos partículas f_1 , f_2 con velocidades \mathbf{v}_1 y \mathbf{v}_2 que

tienen velocidades \mathbf{v}'_1 y \mathbf{v}'_2 después de la colisión, respectivamente, y f'_1 y f'_2 son las funciones de distribución después de la colisión.

Para llegar a esta ecuación, Boltzmann consideró un gas de partículas con colisiones en las cuales se conserva la masa, la cantidad de movimiento y la energía. Además usó la hipótesis de caos molecular la cual establece la independencia estadística de las colisiones binarias.

Se puede demostrar que una condición necesaria y suficiente para alcanzar el equilibrio termodinámico en el cual la función de distribución no dependa del tiempo es que:

$$f'_1 f'_2 = f_1 f_2 = 0 \quad (2.15)$$

De esta manera obtenemos la función de distribución de velocidades en equilibrio, conocida como la función de distribución de Maxwell–Boltzmann dada por:

$$f^{eq}(\mathbf{v}) = n \left(\frac{m}{2\pi kT} \right)^{D/2} \exp \left(-\frac{m(\mathbf{v} - \mathbf{u})^2}{2kT} \right) \quad (2.16)$$

2.5.2. La ecuación de Boltzmann en redes.

Si derivamos la ecuación de Boltzmann en redes a partir de la ecuación de transporte de Boltzmann con la aproximación de Bhatnagar, Groos y Krook (BGK) [15] y discretizando el tiempo y el espacio de velocidades obtenemos la ecuación de Boltzmann en redes para el espacio bidimensional.

La ecuación de transporte de Boltzmann es:

$$\left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} + \frac{F}{m} \cdot \nabla_{\mathbf{v}} \right) f(\mathbf{r}, \mathbf{v}, t) = \left(\frac{\partial f}{\partial t} \right)_{col} \quad (2.17)$$

donde $f(\mathbf{r}, \mathbf{v}, t)$ es la función de distribución y v es la velocidad microscópica. Tomamos la ecuación en ausencia de fuerzas de cuerpo externas y obtenemos:

$$f(\mathbf{r} + \mathbf{v}\delta t, \mathbf{v}, t + \delta t) - f(\mathbf{r}, \mathbf{v}, t) = \left(\frac{\partial f}{\partial t} \right)_{col} \quad (2.18)$$

y reemplazamos la función de colisión por un relajamiento local al equilibrio dado por la aproximación BGK, obtenemos:

$$\left(\frac{\partial f}{\partial t} \right)_{col} = -\frac{1}{\tau} [f(\mathbf{r}, \mathbf{v}, t) - f^{eq}(\mathbf{r}, \mathbf{v}, t)] \quad (2.19)$$

donde τ es el tiempo de relajación adimensional. Las ecuaciones (2.18) y (2.19) forman la ecuación de evolución de las funciones de distribución discretizada en el tiempo. Si consideramos que la velocidad del gas es cercana a cero, a la función de distribución de Maxwell–Boltzmann la podemos escribir en términos de potencias de \mathbf{u} como:

$$f^{eq} = \frac{\rho}{(2\pi kT)^{D/2}} \exp(-\mathbf{v}^2/2kT) \exp\left(\frac{(\mathbf{v} \cdot \mathbf{u})}{kT} - \frac{\mathbf{u}^2}{2kT}\right) \quad (2.20)$$

y realizando una expansión en series de Taylor alrededor de $\mathbf{u} = 0$

$$f^{eq} = 1 + \nabla f^{eq}(0, 0) \cdot \mathbf{u} + \frac{1}{2} \mathbf{u} \cdot \mathbf{H}(0, 0) (\mathbf{u})^2 + \dots \quad (2.21)$$

donde:

$$\nabla f^{eq}(0, 0) \cdot \mathbf{u} = \frac{(\mathbf{v} \cdot \mathbf{u})}{kT} \quad (2.22)$$

y (\mathbf{u}) es la matriz traspuesta de \mathbf{u} y \mathbf{H} es la matriz Hessiana dada por:

$$\mathbf{H}(0, 0) = \begin{pmatrix} -\frac{1}{kT} + \left(\frac{v_x}{kT}\right)^2 & \frac{v_x}{kT} \left(\frac{v_y}{kT}\right) \\ \frac{v_y}{kT} \left(\frac{v_x}{kT}\right) & -\frac{1}{kT} + \left(\frac{v_y}{kT}\right)^2 \end{pmatrix} \quad (2.23)$$

donde v_x y v_y son los componentes horizontal y vertical de \mathbf{v} . Evaluando los términos hasta orden $O(\mathbf{u}^2)$ obtenemos la aproximación para números de Mach bajos:

$$f^{eq} = \frac{\rho}{(2\pi kT)^{D/2}} \exp\left(\frac{-v^2}{2kT}\right) \left[1 + \frac{\mathbf{v} \cdot \mathbf{u}}{kT} + \frac{(\mathbf{v} \cdot \mathbf{u})^2}{2(kT)^2} - \frac{(\mathbf{u})^2}{2kT} \right] \quad (2.24)$$

Los momentos hidrodinámicos de la función de equilibrio los podemos calcular de la siguiente manera:

$$\begin{aligned} I &= \int \psi(\mathbf{v}) f^{eq} d^2v \\ &= \frac{\rho}{(2\pi kT)^{D/2}} \int \psi(\mathbf{v}) \exp\left(\frac{-v^2}{2kT}\right) \left[1 + \frac{\mathbf{v} \cdot \mathbf{u}}{kT} + \frac{(\mathbf{v} \cdot \mathbf{u})^2}{2(kT)^2} - \frac{(\mathbf{u})^2}{2kT} \right] d^2v \end{aligned} \quad (2.25)$$

donde $\psi(\mathbf{v})$ es un polinomio de \mathbf{v} . La integral anterior tiene la forma:

$$\int e^{-x^2} \psi(x) dx \quad (2.26)$$

por lo que puede ser calculada numéricamente usando una cuadratura de Gauss-Hermite [16]:

$$\int \psi(\mathbf{v}) f^{eq}(\mathbf{r}, \mathbf{v}, t) d\mathbf{v} = \sum_{\alpha} W_{\alpha} \psi(\mathbf{v}_{\alpha}) f^{eq}(\mathbf{r}, \mathbf{v}_{\alpha}, t) \quad (2.27)$$

donde W_{α} son los coeficientes de la cuadratura Gauss-Hermite y \mathbf{v}_{α} es el conjunto de velocidades discretas. Ahora la masa, la cantidad de movimiento y la energía las podemos escribir como:

$$\rho(\mathbf{r}, t) = m \sum_{\alpha} W_{\alpha} f^{eq}(\mathbf{r}, \mathbf{v}_{\alpha}, t) \quad (2.28)$$

$$\mathbf{u}(\mathbf{r}, t) = \frac{1}{\rho} \sum_{\alpha} W_{\alpha} \mathbf{v}_{\alpha} f^{eq}(\mathbf{r}, \mathbf{v}_{\alpha}, t) \quad (2.29)$$

$$\varepsilon = \frac{m}{2n} \sum_{\alpha} W_{\alpha} |\mathbf{v}_{\alpha} - \mathbf{u}|^2 f^{eq}(\mathbf{r}, \mathbf{v}_{\alpha}, t) \quad (2.30)$$

El polinomio $\psi(\mathbf{v})$ tiene la forma:

$$\psi_{m,n}(\mathbf{v}) = v_x^m v_y^n \quad (2.31)$$

Con todo esto nos queda la siguiente expresión:

$$\begin{aligned} I = \int \psi_{m,n}(\mathbf{v}) f^{eq} d^2v &= \frac{\rho}{(2\pi kT)} \left[\iint v_x^m v_y^n \exp(-\eta^2) \left(1 - \frac{u^2}{2kT} \right) dv_x dv_y \right] \\ &+ \frac{\rho}{(2\pi kT)} \left[\iint v_x^m v_y^n \exp(-\eta^2) \left(\frac{\mathbf{v} \cdot \mathbf{u}}{kT} \right) dv_x dv_y \right] \\ &+ \frac{\rho}{(2\pi kT)} \left[\iint v_x^m v_y^n \exp(-\eta^2) \left(\frac{(\mathbf{v} \cdot \mathbf{u})^2}{2(kT)^2} \right) dv_x dv_y \right] \end{aligned} \quad (2.32)$$

donde $\eta = v/\sqrt{2kT}$. Al evaluar cada uno de los términos anteriores obtenemos:

$$\begin{aligned}
 I &= \int \psi_{m,n}(\mathbf{v}) f^{eq} d^2v = \frac{\rho}{\pi} (\sqrt{2kT})^{m+n} \left[\left(1 - \frac{\mathbf{u}^2}{2kT}\right) I_m I_n \right] \\
 &\quad + \frac{\rho}{\pi} (\sqrt{2kT})^{m+n} \left[\frac{2(u_x I_{m+1} I_n + u_y I_m I_{n+1})}{\sqrt{2kT}} \right] \\
 &\quad + \frac{\rho}{\pi} (\sqrt{2kT})^{m+n} \left[\frac{u_x^2 I_{m+2} I_n + 2u_x u_y I_{m+1} I_{n+1} + u_y^2 I_m I_{n+2}}{kT} \right]
 \end{aligned} \tag{2.33}$$

donde:

$$I_m = \int_{-\infty}^{\infty} e^{-\eta^2} \eta^m d\eta \tag{2.34}$$

La ecuación anterior la podemos escribir como la sumatoria de términos de η con sus pesos respectivos:

$$I_m = \sum_{j=1}^3 \omega_j \eta_j^m \tag{2.35}$$

donde se considera la suma hasta tercer orden. Al evaluar las sumatorias de la ecuación obtenemos:

$$I = \frac{\rho}{\pi} \sum_{i,j=1}^3 \omega_j \omega_i \psi(\mathbf{v}_{i,j}) \left(1 + \frac{\mathbf{v}_{i,j} \cdot \mathbf{u}}{kT} + \frac{(\mathbf{v}_{i,j} \cdot \mathbf{u})^2}{2(kT)^2} - \frac{\mathbf{u}^2}{2kT} \right) \tag{2.36}$$

De la ecuación anterior podemos identificar a la función de equilibrio. Para el caso particular de $\psi(\mathbf{v}) = 1$ la ecuación queda como:

$$\begin{aligned}
 I &= \sum_{\alpha} W_{\alpha} f^{eq}(\mathbf{r}, \mathbf{v}, t) = \frac{\rho}{\pi} \sum_{i,j=1}^3 \omega_i \omega_j \left(1 + \frac{\mathbf{v}_{i,j} \cdot \mathbf{u}}{kT} + \frac{(\mathbf{v}_{i,j} \cdot \mathbf{u})^2}{2(kT)^2} - \frac{\mathbf{u}^2}{2kT} \right) \\
 &= \sum_{i,j=1}^3 f_{i,j}^{eq}
 \end{aligned} \tag{2.37}$$

donde:

$$f_{i,j}^{eq} = \frac{\omega_i \omega_j}{\pi} \rho \left(1 + \frac{\mathbf{v}_{i,j} \cdot \mathbf{u}}{kT} + \frac{(\mathbf{v}_{i,j} \cdot \mathbf{u})^2}{2(kT)^2} - \frac{\mathbf{u}^2}{2kT} \right) \quad (2.38)$$

Para determinar los valores de $\omega_i \omega_j$, empleamos los polinomios de Hermite calculando los valores de η_i . Los polinomios son de la forma:

$$\mathbf{H}(\eta) = (-1)^n e^{\eta^2} \frac{d^n}{d\eta^n} e^{-\eta^2} \quad (2.39)$$

para $n = 3$ tenemos:

$$\mathbf{H}_3(\eta) = 8\eta^3 - 12\eta \quad (2.40)$$

por lo que $\eta_1 = -\sqrt{3/2}$, $\eta_2 = 0$ y $\eta_3 = \sqrt{3/2}$.

Finalmente, para encontrar los pesos, utilizamos la formula de Hermite [15]:

$$\omega_i = \frac{2^{(n+1)n\sqrt{\pi}}}{[H'_n(\eta_i)]^2} \quad (2.41)$$

donde H'_n es la derivada del polinomio de Hermite de orden n . Para $n = 3$

$$H'_3(\eta) = 24\eta^2 - 12 \quad (2.42)$$

entonces $H'_3(\eta_1) = 24$, $H'_3(\eta_2) = -12$ y $H'_3(\eta_3) = 24$, por lo tanto:

$$\omega_1 = \frac{\sqrt{\pi}}{6}, \quad \omega_2 = \frac{2\sqrt{\pi}}{3} \quad y \quad \omega_3 = \frac{\sqrt{\pi}}{6} \quad (2.43)$$

Entonces tenemos deducida la manera en que encontraremos los pesos para cada una de las direcciones de las partículas, tal como se muestra a continuación:

$$\omega_\alpha = \frac{\omega_i \omega_j}{\pi} = \begin{cases} 4/9, & i = j = 2, & \alpha = 0 \\ 1/9, & i = 1, j = 2, \dots, & \alpha = 1, 2, 3, 4 \\ 1/36, & i = j = 1, \dots, & \alpha = 5, 6, 7, 8 \end{cases} \quad (2.44)$$

la función de distribución de equilibrio la podemos escribir como:

$$f_\alpha^{eq}(\mathbf{r}, t) = \omega_\alpha \rho \left(1 + \frac{3(\mathbf{c}_\alpha \cdot \mathbf{u})}{c^2} + \frac{9(\mathbf{c}_\alpha \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right) \quad (2.45)$$

donde:

$$\mathbf{c}_\alpha = \begin{cases} (0, 0), & \alpha = 0 \\ (\cos\theta_\alpha, \sin\theta_\alpha)c, & \theta_\alpha = (\alpha - 1)\frac{\pi}{2}, \quad \alpha = 1, 2, 3, 4 \\ \sqrt{2}(\cos\theta_\alpha, \sin\theta_\alpha)c & \theta_\alpha = (\alpha - 5)\frac{\pi}{2} + \frac{\pi}{4}, \quad \alpha = 5, 6, 7, 8 \end{cases} \quad (2.46)$$

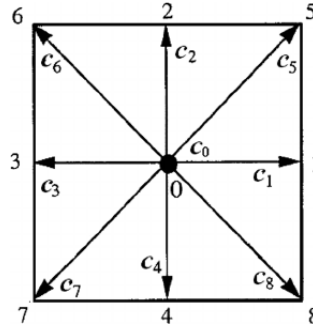


Figura 2.11: Esquema de velocidades c_α para la malla D2Q9 con valores 0,1 y $\sqrt{2}$. La velocidad c_0 se localiza en el centro.

y $c_s^2 = c^2/3$ es la velocidad del sonido y $c = \delta_x/\delta_t$, donde δ_x es la distancia entre nodo y nodo y usualmente es unitaria y δ_t es el intervalo de tiempo discretizado, también unitario. Entonces la ecuación de Boltzmann en redes [17] es:

$$f_\alpha = (\mathbf{r} + \delta t c_\alpha, t + \delta t) - f_\alpha(\mathbf{r}, t) = \frac{1}{\tau} (f_\alpha^{eq}(\mathbf{r}, t) - f_\alpha(\mathbf{r}, t)) \quad (2.47)$$

donde $\alpha = 0, \dots, 8$ para la malla D2Q9, que se muestra en la figura 2.11.

2.6. LBM en contornos.

Típicamente, la segmentación de imágenes se logra mediante la identificación de los puntos que pertenecen a cada uno de los objetos de interés o a través de la localización de aquellos que determinan su contorno [18]. Entre las distintas alternativas planteadas, existen algoritmos de segmentación que son capaces de dividir una imagen en regiones uniformes de color o textura, lo cual es una importante técnica que facilita la segmentación de regiones con propiedades de textura significativas [19].

La textura describe la existencia de alguna regularidad en una región continua de la imagen, la cual se puede manifestar como un patrón o forma repetitivos, aunque no está definido cuán regular debe ser [20]. Este proceso resulta de gran importancia en numerosas aplicaciones de procesamiento de imágenes, especialmente clasificación de objetos, síntesis de texturas para computación gráfica y visión computacional [21].

Entre las distintas técnicas usadas para segmentación se pueden mencionar los campos aleatorios de Markov, dimensión fractal, polígonos de Voronoi, filtros de Gabor y descomposición wavelet [22]. Los tres primeros extraen las características de textura por medio del cálculo de algunos parámetros de la imagen. Cabe señalar que al hablar de textura nos referimos a las diferentes formas y objetos que se distinguen visualmente en una imagen, y que podemos ver en primer o segundo plano. Mientras que los dos últimos primero transforman la imagen al dominio de frecuencia y luego lo dividen en diferentes sub-bandas. La distribución de energía en cada una de estas sub-bandas es utilizada para distinguir diferentes texturas.

El principal problema en la mayoría de los métodos se relaciona a la discriminación entre estructuras con diferente textura. Además, la similitud de texturas no puede ser exactamente definida ya que depende por lo general de cada aplicación particular.

El algoritmo de segmentación mediante redes de Boltzmann es capaz de detectar distintas texturas presentes en una imagen digitalizada analizando magnitudes y parámetros asociados al comportamiento de un fluido simulado.

2.6.1. El modelo de redes de Boltzmann

Los autómatas de redes de Boltzmann son aproximaciones relativamente nuevas que utilizan computación paralela para simular fenómenos de transporte de diferentes tipos, y son vistos como métodos prometedores para resolver una variedad de ecuaciones diferenciales parciales y modelar fenómenos físicos. La ventaja del método de redes de Boltzmann es la velocidad y facilidad del cálculo diferencial a partir de reglas aritméticas simples, presentándose como una alternativa interesante a los métodos numéricos tradicionales.

A diferencia de los métodos clásicos que aproximan con esquemas discretos las ecuaciones diferenciales parciales que representan comportamientos macroscópicos, las redes de Boltzmann construyen modelos cinéticos locales que incorporan los mecanismos esenciales de procesos que ocurren microscópica o mesoscópicamente, de manera que las propiedades macroscópicas promediadas obedezcan las ecuaciones de transporte deseadas.

Es esencialmente una representación discreta del espacio mediante una rejilla o malla regular de celdas cuyo estado está caracterizado por una población de “partículas”, esto quiere decir que un nodo en la malla no representa necesi-

riamente a una partícula del sistema en cuestión. El estado de las partículas se expresa con una función de densidad de probabilidad $f_i(\mathbf{r}, t)$ que representa la cantidad de partículas en la celda x en el tiempo t moviéndose con los vectores de velocidad c_α (tratada como una variable interna). La función de distribución cambia de estado según reglas que se ejecutan secuencialmente, simulando mecanismos de transporte y colisión de partículas.

El conjunto de velocidades c_α está definido por el tipo de malla, tal como se vio en la ecuación 2.44 y la figura 2.11. La vecindad de Moore en 2D es la más popular, la cual lleva a la resolución conocida como D2Q9, tal como vimos en la ecuación 2.44 (dos dimensiones y nueve velocidades) debido a la naturaleza de la vecindad, tal como se muestra:

$$\begin{aligned} c_0 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, & c_1 &= \begin{pmatrix} v \\ 0 \end{pmatrix}, & c_2 &= \begin{pmatrix} 0 \\ v \end{pmatrix}, \\ c_3 &= \begin{pmatrix} -v \\ 0 \end{pmatrix}, & c_4 &= \begin{pmatrix} 0 \\ -v \end{pmatrix}, & c_5 &= \begin{pmatrix} v \\ v \end{pmatrix}; \\ c_6 &= \begin{pmatrix} v \\ -v \end{pmatrix}, & c_7 &= \begin{pmatrix} -v \\ v \end{pmatrix}, & c_8 &= \begin{pmatrix} -v \\ -v \end{pmatrix}; \end{aligned}$$

donde $v = \Delta x / \Delta t$ es una velocidad característica dada por el cociente entre el tamaño de la red y el paso de tiempo asociado a cada cambio del estado del autómata.

Como en cualquier autómata celular, el estado de cada celda de una red de Boltzmann cambia siguiendo un esquema de reglas explícitas que sólo dependen del estado actual de la celda y sus vecinos. La ecuación 2.48 muestra el transporte desde el tiempo t al $t + 1$ para una celda en la posición x :

$$f(\mathbf{r} + \mathbf{v}\delta t, t + \delta t) = f(\mathbf{r}, t) \quad (2.48)$$

En el paso de colisión, se aplica el operador de colisión $\Omega(f) = df/dt$ [23] a la función de distribución de partículas en cada celda. Si las colisiones se dan entre las partículas habrá una diferencia entre el estado inicial y el estado final en el caso de que no exista una fuerza externa actuando en el sistema. Este operador es clave para el funcionamiento del modelo de Boltzmann, ya que simula las colisiones de partículas dentro de la celda. Para aplicar el operador de colisión a una celda dada, debe calcularse la función de distribución en equilibrio f_i^{eq} . En el caso de un fluido, la densidad ρ de una celda es la medida del número de partículas en la celda:

$$\rho = \sum_{i=0}^{q-1} f_i \quad (2.49)$$

La velocidad u de la celda es la velocidad promedio de las partículas en tal celda:

$$u = \frac{1}{\rho} \sum_{i=0}^{q-1} f_i e_i \quad (2.50)$$

Para el caso del modelo BGK, el operador de colisión es:

$$\frac{1}{\tau} (f_i^{eq}(\mathbf{r}, t) - f_i(\mathbf{r}, t)) \quad (2.51)$$

La densidad y la velocidad son usadas para calcular la función de distribución en equilibrio f_i^{eq} . Esta función representa la colisión entre partículas y se define de manera que a nivel macroscópico se recobre el comportamiento de lo que se está simulando (gas, líquido, etc.). En cada paso temporal, se calcula un nuevo conjunto de valores de f_i usando los valores actuales de f_i y los valores de equilibrio f_i^{eq} . La regla clásica de actualización de $f_i(\mathbf{r}, t)$ es la ecuación discreta de Boltzmann [24]:

$$f_i(\mathbf{r} + \delta t c_\alpha, t + \delta t) = f_i(\mathbf{r}, t) + \frac{1}{\tau} (f_i^{eq}(\mathbf{r}, t) - f_i(\mathbf{r}, t)) + S_i \quad (2.52)$$

donde S_i es un término fuente y τ es un tiempo característico de relajación usado para controlar la velocidad con la que la simulación alcanza el equilibrio. Tal coeficiente se corresponde con la viscosidad del fluido simulado, de modo que valores muy altos de τ corresponden a viscosidades muy bajas. La ecuación de Lattice Boltzmann generalmente se resuelve mediante un algoritmo de paso temporal explícito en el que se separa la fase de colisión y la de propagación.

Finalmente, en el paso de colisión se incluye en la función de equilibrio basada en la entropía máxima, donde $C = 1/3$.

$$f_i^{eq} = \frac{\rho C^{-\frac{\sqrt{|u_\alpha - v_\alpha|}}{c}}}{\left(\sum_{\alpha=0}^{q-1} C^{-\frac{\sqrt{|u_{alpha} - v_\alpha|}}{c}} \right)} \quad (2.53)$$

2.7. Palabos en redes de Boltzmann.

2.7.1. Simulaciones de redes de Boltzmann.

Palabos [25] es una plataforma para resolver problemas de dinámica de fluidos complejos (CFD por sus siglas en inglés. Complex Fluid Dynamics) basado en

el método de redes de Boltzmann. Es una plataforma muy amplia y flexible que permite acoplar nuestros problemas con condiciones particulares y su programación para realizar simulaciones de muy elevado grado de exactitud. Cuenta con muchos ejemplos y código abierto en Python [26] que está a nuestra disposición para un mejor control de la plataforma.

Se puede utilizar esta herramienta para construir programas en C++ y es en general muy eficiente aunque no siempre fácil de utilizar. Palabos es una herramienta que combina un ambiente similar al de Matlab y las herramientas propias de Palabos para realizar la ejecución de código [27].

Entre sus más distinguidas características se encuentran:

- Combina alto rendimiento e interactividad
- Ofrece una conveniente plataforma para desarrollar nuevos modelos

Un ejemplo muy claro de su uso es en la creación de geometrías de medios porosos, y el algoritmo básico de uso es el siguiente:

- Ejecución de simulaciones.
- Visualización de resultados.
- Validación del modelo matemático que gobierna el sistema.

A continuación se presenta un ejemplo escrito en Python. Para realizarlo es necesario descargar los paquetes de instalación de Palabos de la página oficial y consultar el ejemplo *examples/porous.py*. El resultado de la simulación se muestra en las figuras 2.12 y 2.13.

Código completo de ejemplo de medio poroso.

```
#!/usr/bin/python
from palabos import *
nx, ny, nz = 100, 50, 50
tau= 0.6

lattice = Block3D(nx,ny,nz, D3Q19, BGK(omega=1/tau))
pout.display( lattice[10,10,10].getPopulations() )
lattice[10,10,10].initializeAtEquilibrium(1.1, [0.,0.,0.] )
```

```

lattice.collideAndStream()

for i in range(1,100):
    lattice.collideAndStream()
    imShow(lattice[:, :, nz/2].velocityNorm())
    lattice[10:30, 10:30, 10:30].defineDynamics(BounceBack(1.))
    dir(lattice)
    help(lattice.defineDynamics)

for i in range(1,100):
    lattice.collideAndStream()
    imShow(lattice[:, :, nz/2].velocityNorm())
    lattice.defineDynamics(BGK(omega=1./tau))
    x,y,z = lattice.meshGrid()
    media = (x-10)**2 + (y-10)**2 + (z-10)**2 < 10**2
    pout.isoSurface(media, [1.])

nobst = 30
r= 8

for cx,cy,cz in zip(rand_int(nobst,nx), rand_int(nobst,ny), rand_int(nobst,nz)):
    pout.display('Sphere at position', cx, cy, cz)
    media += (x-cx)**2 + (y-cy)**2 + (z-cz)**2 < r**2
    pout.isoSurface(media>0, [1.])
    boundary.regularized().definePressureBC(lattice[0, :, :])
    lattice[0, :, :].setBoundaryDensity(1.02)
    boundary.regularized().definePressureBC(lattice[nx-1, :, :])
    lattice[nx-1, :, :].setBoundaryDensity(1.)
    lattice[media>0].defineDynamics(BounceBack(1.))

for i in range(1,200):
    lattice.collideAndStream()
    imShow(lattice[:, :, 20].velocityNorm())
    pout.display( lattice[media==0][nx/2, :, :].velocityComponent(0).average() )
    pout.imagesc( [ lattice.velocity().strainRate().symmetricTensorNorm()[:, :, 20],
    lattice.strainRateFromStress().symmetricTensorNorm()[:, :, 20] ] )

```

En cada nodo se tienen diferentes reglas de colisión.

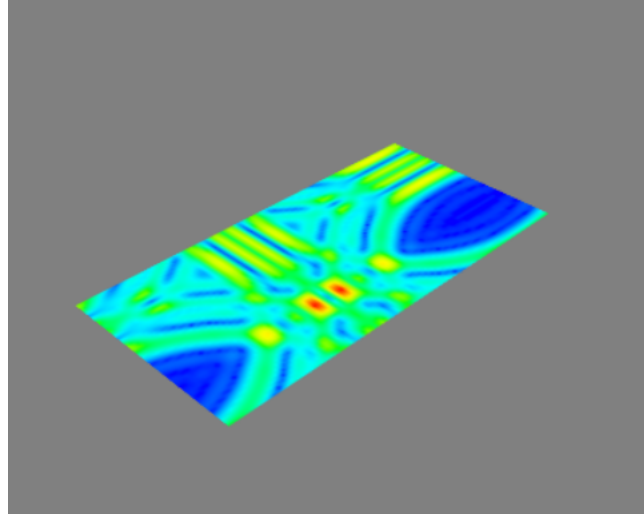


Figura 2.12: Simulación D3Q19 de una malla de medio poroso.

Se trata de una malla en tres dimensiones de $100 \times 50 \times 50$ con $\tau = 0.6$ según el modelo de BGK ($\Omega = \frac{1}{\tau}$) y 19 direcciones de propagación de las partículas de cada malla.

Se realizan 100 iteraciones en las que se evalúan la colisión y propagación de las partículas y se muestra la gráfica correspondiente:

Colisión local: es definida por reglas locales.

$$\Pi = \sum_{i=0}^{q-1} c_i c_i f_i \quad (2.54)$$

$$S = -\frac{1}{2} \frac{\omega}{\rho c_s^2} \Pi \quad (2.55)$$

La velocidad de los gradientes tiene un enfoque no local y se define como operaciones de rango limitado:

$$S = \frac{1}{2} \left(\nabla u + (\nabla u)^T \right) \quad (2.56)$$

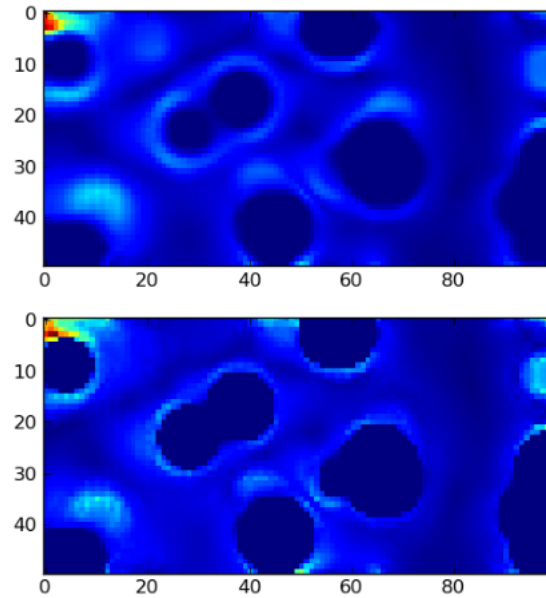


Figura 2.13: Simulación D3Q19. Comparación de estrés y tensión en las partículas.

Palabos
CFD, COMPLEX
PHYSICS

[Home](#)
[Gallery](#)
[Software](#)
[Benchmarks](#)
[Project](#)
[Commercial Support](#)

Gallery > Thermal flow > Heated Sphere in a Channel

Heated Sphere in a Channel

The unsteady transient flow around a sphere placed in a rectangular channel is presented. The Reynolds number is equal to 420. Thermal convection is also simulated, since the sphere has a constant temperature which is higher than that of the surrounding fluid. The Prandtl number is 1.2 This problem may seem simple, but it is quite difficult, since at this Reynolds number a complex flow pattern is formed downstream the sphere, as shows in the animations that follow.

First let's have a side-view which includes the sphere. Both velocity magnitude and temperature are displayed:

Velocity
JW Player

Temperature

QUICK LINKS

[Download](#)

[Documentation](#)

[Forum](#)

[LBM Wiki](#)

Figura 2.14: Simulación de una esfera calentada en un medio.

```
pout.display( lattice[media==0][nx/2, :, :] .  
velocityComponent(0).average() )  
pout.imagesc( [ lattice.velocity().strainRate().  
symmetricTensorNorm()[:, :, 20],  
lattice.strainRateFromStress().  
symmetricTensorNorm()[:, :, 20] ] )
```

Este y muchos ejemplos más de implementación de redes de Boltzmann pueden ser encontrados en el sitio oficial de Palabos y ofrecen una gran cantidad de métodos de solución de problemas de simulación de fluidos en distintos medios, basados en las ecuaciones de Navier-Stokes.

2.8. Bibliografía capítulo 2.

- [1] Godfrey N. Hounsfield. Premio Nobel de Medicina (1979). Seguridad Nuclear. III Trimestre 2004 (32). 2004. p. 34.
- [2] DICOM PS3.1 2017e - Introduction and Overview. p 12.
- [3] Franco Martínez, Edgardo Adrián. (2011). Análisis digital de imágenes tomográficas sin contraste para la búsqueda de tumores cerebrales (Tesis de Maestría). Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Unidad Zacatenco. Departamento de Computación. México, D.F.
- [4] Human identification based on temporal lifting using 5/3 wavelet filters and Radon transform. Randa Atta, Samir Shaheen, Mohammad Ghanbari. Elsevier 2017.
- [5] Fourier. Principles of Computerized Tomographic Imaging. Avinash C. Kak. School of Electrical Engineering. Purdue University, Malcolm Slaney. Originally: Schlumberger Palo Alto Research, 1988. <http://www.slaney.org/pct/pct-toc.html>
- [6] Algorithms for Image Processing and Computer Vision, Second Edition. J.R. Parker. Ed, Wiley. Indianapolis, 2011
- [7] Beginning Digital Image Processing Using Free Tools for Photographers. Sebastian Montabone. Apress 2010.
- [8] Introduction to Atomistic Simulations. Leonid Zhigilei. University of Virginia. 2000. <http://people.virginia.edu/lz2n/mse627/>

- [9] Statistical Thermodynamics Fundamentals and Applications. Normand M. Laurendeau. Purdue University, Cambridge University Press. 2005.
- [10] Seminar Lattice Boltzmann method, Igor Mele, Iztok Tiselj Ljubljana, March 2013.
- [11] Finite Volume Methods for Hyperbolic Problems. Randall J. Leveque. University of Washington. Cambridge University Press. 2004.
- [12] S. Succi. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford Science Publications, 2001.
- [13] Two-Dimensional Lattice Boltzmann for Reactive Rayleigh–Bénard and Bénard–Poiseuille Regimes. Suemi Rodríguez Romo, Oscar Ibañez Orozco. Facultad de Estudios Superiores Cuautitlán, Universidad Nacional Autónoma de México, México. Septiembre 2015.
- [14] Lattice Boltzmann Method of Active Contour for Image Segmentation. ZhiQiang Wang, ZhuangZhi Yan, George Chen. Shenzhen Institutes of Advanced Technology Chinese Academy of Sciences, Shenzhen, China School of Communication.
- [15] Bhatnagar, P.L., Gross, E.P., Krook, M.: A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.* 94(3), 511–525 (1954)
- [16] Z. Guo, C. Shu, Lattice Boltzmann Method and its Applications in Engineering, World Scientific, 2013.
- [17] Levitación de partículas en ondas de sonido usando el método de la ecuación de Boltzmann en redes. Guillermo Barrios Del Valle, Raúl Rechtman Schrenzel. Facultad de Ingeniería UNAM. Agosto 2007.
- [18] Yeh, C., Soetikno, B., Hu, S., Maslov, K. I., & Wang, L. V. (2014). Microvascular quantification based on contour-scanning photoacoustic microscopy. *Journal of Biomedical Optics*, 19(9), 096011. <http://doi.org/10.1117/1.JBO.19.9.096011>
- [19] Detección de texturas en imágenes digitales usando el modelo de Lattice Boltzmann. Virginia Cifuentes, Mariana del Fresno, Alejandro Clausse. Argentina, 2010. <http://www.pladema.net/>
- [20] Jawerth, B., Lin, P., Sinzinger, E.: Lattice Boltzmann models for anisotropic diffusion of images. *J. Math. Imaging Vis.* 11, 231–237 (1999)

- [21] Digital Image Processing. Third Edition. William K. Pratt. Los Altos, California. 2001.
- [22] Analysis of wavelet-domain Wiener Filters. Hyeokho Choi and Richard Baraniuk. Department of Electrical and Computer Engineering, Rice University, Houston, TX 77251-1892, USA
- [23] R. Benzi, S. Succi, M. Vergassola, The lattice Boltzmann equation: theory and applications, In Physics Reports, Volume 222, Issue 3, 1992, Pages 145-197, ISSN 0370-1573.
- [24] Xiaoyi He, Shiyi Chen, Gary D. Doolen, A Novel Thermal Model for the Lattice Boltzmann Method in Incompressible Limit, In Journal of Computational Physics, Volume 146, Issue 1, 1998, Pages 282-300.
- [25] Palabos <http://www.palabos.org/>
- [26] Tutorial: the open-source library Palabos in your daily work. Jonas Latt. EPFL, Lausanne, Switzerland. 2010.
- [27] <http://www.palabos.org/gallery/thermal-flow/50-sphere-in-a-channel-1>

Implementación de redes de Boltzmann para obtención de contornos en imágenes de cáncer de mama.

3.1. Introducción

Para llegar al objetivo de este trabajo, es necesario obtener en primera instancia los contornos de interés de las imágenes de prueba. Se trabaja con las imágenes como una matriz que contiene información de brillo en cada uno de sus píxeles (cuya equivalencia en términos de fluidos para poder aplicar las redes de Boltzmann corresponde a una magnitud). Contiene también un coeficiente de viscosidad que se relaciona con un coeficiente de relajación en toda la imagen, con el cuál se va a propagar la magnitud dentro de toda la matriz y el cuál determina en cuanto tiempo se alcanza un equilibrio local antes de analizar un punto diferente de la malla.

En este capítulo se lleva a cabo la implementación de las redes de Boltzmann mediante el método propuesto como “Fast and Robust Lattice Boltzmann Method” [1], en 5 direcciones diferentes, y su adaptación a 9 direcciones diferentes para mejorar la calidad de los contornos. Obteniendo así, los parámetros más adecuados para poder aplicar el método a todo el conjunto de imágenes de prueba y posteriormente, a imágenes de casos clínicos actuales.

3.1.1. Método de ajuste de nivel

Level Set Method o LSM [2] es una técnica numérica para trazar interfaces y contornos que se usa como una representación de contornos activos, la cual tiene la ventaja de manejar automáticamente los cambios topológicos de contorno trazado. Esto quiere decir que el método trabaja comparando valores entre píxeles

de una imagen y almacena la información para después trazar una trayectoria que une todos los puntos, formando así el contorno activo.

$$\phi_t + V_n |\nabla\phi| = 0 \quad (3.1)$$

donde V_n es una componente de velocidad en dirección normal, también se conoce como la velocidad normal. Por lo que la cantidad de movimiento en esta curvatura se caracteriza por $V_n = -bk$. Esta es la ecuación de level set básica [3].

Ecuación de level set con efecto de convección-difusión

$$\frac{\theta}{t} + \vec{V} \cdot \nabla\phi = b\Delta\phi \quad (3.2)$$

donde θ es el ángulo del gradiente en una curvatura del objeto de la imagen, es decir, es la curvatura del contorno. \vec{V} es la velocidad del gradiente, $\nabla\phi$ y $\Delta\phi$ son gradiente y el laplaciano, que interactúan con una viscosidad “virtual” $b\Delta\phi$ que tiene un valor bajo. Esta ecuación representa el fenómeno de convección-difusión, que incluye el efecto de un campo de velocidad externo y un coeficiente de difusión [4].

El método de redes de Boltzmann tiene ventajas sobre el método de conjuntos de nivel por ser altamente paralelizable y simple, debido a que las colisiones y el tiempo de relajación en los nodos no dependen entre sí.

Lo interesante de esta variación del método de redes de Boltzmann propuesta por Souleymane et. al., es que combina el método de Boltzmann para recuperar la ecuación de LSM asociada con las características de Navier-Stokes, y lo hace incorporando una fuerza externa para minimizar el problema de dividir una imagen de N píxeles en distintas clases (c) [5,6,7].

Con este método se obtiene un contorno muy bien definido, descrito en la figura 3.1, y que es funcional en las imágenes médicas que se usarán en este trabajo (ver apéndice A).

3.1.2. Algoritmo para la obtención de contornos usando redes de Boltzmann.

El proceso consiste en obtener una imagen o sección de una imagen médica que sea de interés para obtener su contorno. Posteriormente, se debe iniciar una

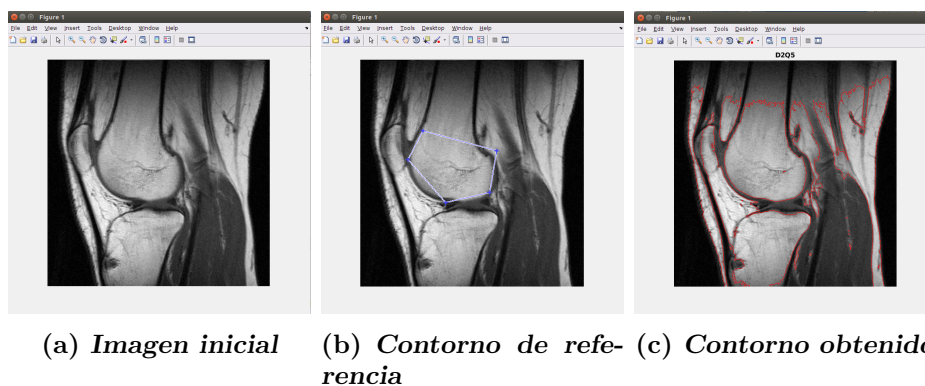


Figura 3.1: Obtención de contornos en imagen de rodilla [1].

función de equilibrio en cero para almacenar los resultados en cada paso de colisión. A continuación se obtienen las dimensiones en píxeles x y y de la imagen y se almacenan en dos vectores $U_1^p(x, y)$ y $U_2^p(x, y)$. Se establecen otros parámetros iniciales en la variable B. A partir de este momento se asignan valores iniciales para nuestros coeficientes de relajación, viscosidad de la malla y velocidad de propagación inicial, y se calcula el valor de una fuerza externa que inicia el proceso de colisión entre las magnitudes de brillo en los píxeles. En cada paso de colisión y relajación, se resuelve un problema de reacción-difusión en la malla, y, esto da paso a acumular el resultado de la ecuación en una función de equilibrio $f_i(\mathbf{r}, t)$ correspondiente a cada pixel de la malla para la dirección inicial. Finalmente se toma la función de equilibrio como nueva condición inicial y se repite el proceso para las 8 direcciones restantes obteniendo las distancias en cada punto y trazando un contorno en cada uno de ellos.

Este proceso de muestra en el Algoritmo 1.

Algorithm 1: Obtener contorno de imagen

Result: Contorno de imagen

- 1 inicio;
- 2 establecer la función de distribución f ;
- 3 calcular valores para $U_1^p(x, y)$ y $U_2^p(x, y)$;
- 4 calcular v_1 y v_2 ;
- 5 calcular los valores de B ;
- 6 **while** $\lambda \neq B$ **do**
- 7 | instrucciones;
- 8 | determinar parámetros de inicio ρ , τ , γ and v ;
- 9 | cálculo de fuerza externa;
- 10 | incluir la fuerza externa;
- 11 | resolver la ecuación de reacción-difusión;
- 12 | almacenar los valores de $f_i(\mathbf{r}, t)$ en cada punto de la malla, que
| generan las distancias en cada punto;
- 13 | trazar contorno;
- 14 | $\lambda = \lambda + 1$;
- 15 **end**

Dado que el contorno se obtuvo de una malla D2Q5 en el algoritmo (véase figura 3.2), se realizaron las primeras pruebas con imágenes de un banco de imágenes públicas conocido como Mammographic Image Analysis Society “MIAS” en donde podemos encontrar distintas imágenes previamente clasificadas en grupos de suma importancia, como lo son: a) Por tipo de cáncer. Maligno o Benigno; b) Por la forma de las acumulaciones de grasa. Espiculares, redondos, etc.; c) Por la densidad del tejido. Alta densidad, baja densidad, etc.

Utilizando imágenes de la clasificación de cáncer maligno de forma espicular (con forma de estrella) y con cáncer benigno, se obtuvo como resultado el contorno mostrado en la figura 3.3.

3.1.3. Esténcil D2Q5.

En la figura 3.4 se muestra parte del código implementado en Matlab. Se pueden ver asignados valores a los vectores ex y ey , los cuales por su valor de posición representan coordenadas de cada uno de los vecinos del esténcil, es decir, para c_0 tenemos las coordenadas $(0, 0)$ si las unimos de manera vertical, para c_1 tenemos

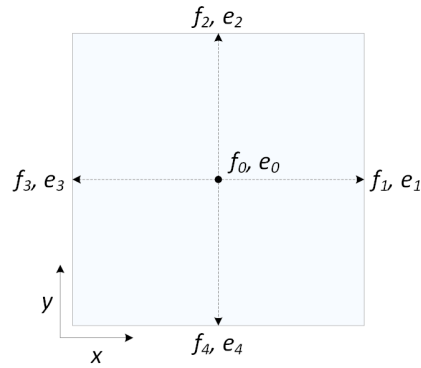
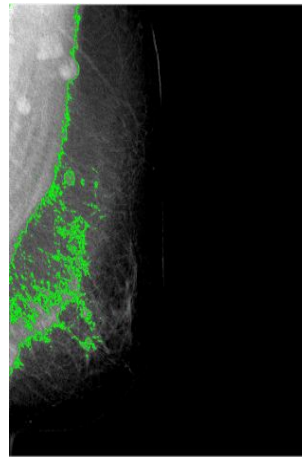


Figura 3.2: Esquema del estencil para la malla D2Q5 con velocidades 0,1 y $\sqrt{2}$. La velocidad 0 se localiza en el centro.



(a) *Imagen inicial*



(b) *Contorno obtenido*

Figura 3.3: Obtención de contornos en imagen de prueba.

```

Editor - /media/pidios/Beta/Documentos/Posgrado/Trabajo de imágenes médicas/respa.
FRLBM_Main.m x +
22
23 % D2Q5 LATTICE CONSTANTS
24 - ex = [0, 1, 0, -1, 0];
25 - ey = [0, 0, 1, 0, -1];
26 - A = [1/3, 1/6, 1/6, 1/6, 1/6];
27
28 % INITIAL DISTANCE FIELD
29 - J= imread ('Image_test.jpg');
30 - I=im2double(J);
31 - %I=imresize(I,0.8);
32 - I=rgb2gray(I);
33 - h = figure;
34 - imshow(J);
35 - hold on;
36 - binary_image = roipoly;
37 - phi = Initial_Mask(binary_image);
38 - contour(phi, [0 0], 'b')
39
40 % LOG TRANSFORMATION
41 - lx = size(I,1);
42 - ly = size(I,2);
43 - c = 1/log(2);
44 - Y = zeros(lx,ly);
45 - for x = 1:lx
46 -     for y = 1:ly
47 -         %I(x,y)=log(I(x,y).*255);
48 -         Y(x,y)= c.*log(1 + I(x,y));
49 -         % I=I(x,y).*255;
50 -     end
51 - end

```

Figura 3.4: Esquema del estencil para la malla D2Q5. Con coordenadas y pesos por columnas.

las coordenadas $(1, 0)$ y así sucesivamente. Además tenemos otro vector que nos indica el valor de los pesos de cada una de las posibles direcciones de la malla. Siendo para $e_0 = 1/3$, $e_1 = e_2 = e_3 = e_4 = 1/6$ [8].

Hay que trazar un contorno de referencia para poder inicializar la variable ϕ del código. y comenzamos con el algoritmo de obtención de contorno.

También se pueden observar los demás parámetros de inicio que necesitamos (figura 3.5): $\gamma = 30$ y $\tau = 68$ que son valores que determinan el tiempo de relajación; v = coeficiente de convección del sistema, $p = 2$ que representa a la viscosidad ρ en el código.

3.1.4. Estencil D2Q9.

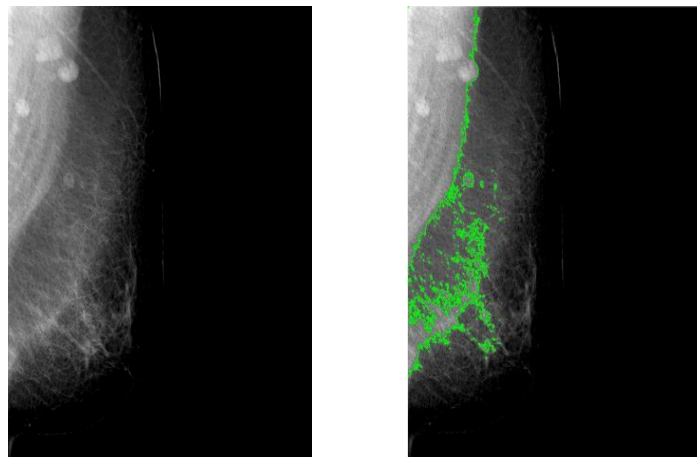
Como resultado de aplicar el algoritmo con las modificaciones a D2Q9 se obtuvieron las siguientes imágenes en la figura 3.6, con contornos mucho más definidos.

```

Editor - /media/pidios/Beta/Documentos/Posgrado/Trabajo de imágenes médicas/respa.
FRLBM_Main.m x +
70 - phi2 = phi - min(phi(:));
71 - for x = 1:lx
72 -     for y = 1:ly
73 -         for li = 1:5
74 -             Feq(li,x,y) = phi2(x,y).* A(li);
75 -         end
76 -     end
77 - end
78 - Fint = Feq;
79
80 - % SETTING PARAMETERS
81 - gamma = 30;
82 - tau = (9*gamma+2)/4;
83 - its_max = 1;
84 - v = 0.05; % Convection coefficient
85 - p = 2;
86 - pb = zeros(lx,ly);
87 - tic;
88 - for its = 1:its_max
89 -     % COMPUTATION OF THE FUZZY EXTERNAL FORCE
90 -     for x = 1:lx
91 -         for y = 1:ly
92 -             pb(x,y) = Y(x,y)-B(x,y)-v2;
93 -             U1d(x,y) = (Y(x,y)-B(x,y)-v1)./(Y(x,y)-B(x,y)-v2);
94 -             U2d(x,y) = (Y(x,y)-B(x,y)-v2)./(Y(x,y)-B(x,y)-v1);
95 -             U1(x,y) = 1./(1+((Y(x,y)-B(x,y)-v1)./(Y(x,y)-B(x,y)-v2)).^(
96 -             U2(x,y) = 1./(1+((Y(x,y)-B(x,y)-v2)./(Y(x,y)-B(x,y)-v1)).^(
97 -         end
98 -     end
99 -

```

Figura 3.5: Variables de inicio.



(a) Imagen inicial

(b) Contorno obtenido

Figura 3.6: Obtención de contornos en imagen de prueba.

```
Editor - /media/pidios/Beta/Documentos/Posgrado/Trabajo de imágenes médicas/re.
FRLBM_MainQ9_modificado.m  +
19 % E-mail: balla_arabe_souleymane@ieee.org
20
21 - clear all;
22
23 % direcciones D2Q9
24 - ex = [1, 0, -1, 0, 1, -1, -1, 1, 0];
25 - ey = [0, 1, 0, -1, 1, 1, -1, -1, 0];
26
27 - A = [4/9, 1/9, 1/9, 1/9, 1/9, 1/36, 1/36, 1/36, 1/36];
28
29 % INITIAL DISTANCE FIELD
30 - J= imread ('cc.jpg');
31
32 - imshow(J); %muestra la imagen
33 - rec=imcrop; %selecciona con el mouse la region a recortar
34
35 - f=im2double(rec);
36 - %I=imresize(I,0.8);
37 - I=rgb2gray(f);
38 - h = figure;
39 - imshow(rec);
40
41 - hold on;
42 - binary_image = roipoly;
43 - phi = Initial_Mask(binary_image);
44 - contour(phi, [0 0], 'b')
45
46
47 % LOG TRANSFORMATION
48 - lx = size(I,1);
```

Figura 3.7: Código D2Q5 modificado a D2Q9.

El código modificado y adaptado se muestra en la figura 3.7.

El algoritmo define los bordes que hay en toda la imagen, esto sucede debido a que el contorno inicial (delimitado por el polígono azul), sirve como mecanismo para encontrar un nivel de brillo promedio de los píxeles alcanzados por él, de tal forma que al encontrarlo comienza a fluir esa información a través de todos los píxeles de la imagen. A diferencia de un level set básico, este algoritmo no se limita a segmentar una región específica de la imagen, por lo que, sin importar el lugar donde se trace el polígono azul, el resultado siempre será en toda la imagen.

A simple vista no se puede apreciar un gran cambio, sin embargo, al realizar una cercamiento a ciertas partes de la imagen podemos parecer una mejor definición del contorno y algunas secciones que inclusive, ya no aparecen de un estencil a otro, tal como se muestra en la figura 3.8.

De esta manera se obtiene la imagen y el estencil adaptados para comenzar a utilizar las diferentes imágenes de prueba, y para trabajar con el algoritmo 2 que se propone en este trabajo de tesis; mismo que se describe en un diagrama

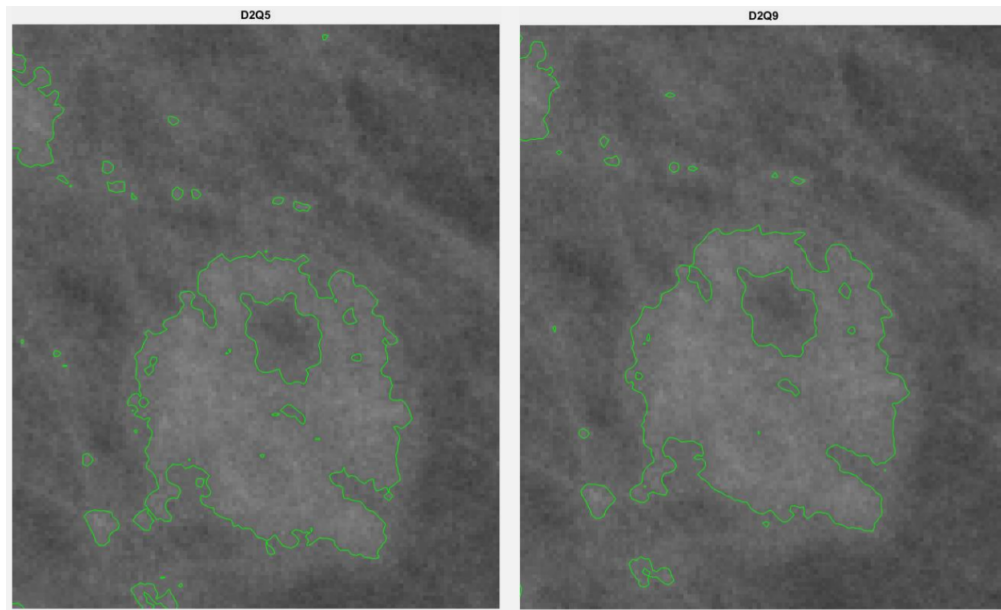


Figura 3.8: Código D2Q5 comparado con D2Q9.

de flujo simple en la figura 3.9.

Hay que mencionar que el procedimiento inicial fué generar el código dada la necesidad de trabajar con una vecindad de 8 individuos en el estencil D2Q9 y que a partir de este momento seguiremos el algoritmo 2 para obtener los contornos de ciertas partes específicas de la imagen.

Algorithm 2: Obtención de contorno de un área específica.

Result: Contorno de las áreas de interés

- 1 abrir el archivo de la imagen original;
 - 2 seleccionar el área de interés de la imagen;
 - 3 obtener un contorno de referencia para inicializar el Algoritmo 1;
 - 4 cálculo de la fuerza externa;
 - 5 resolución de la ecuación de reacción-difusión;
 - 6 acumulación de las funciones de equilibrio;
 - 7 realizar el paso de propagación (actualización);
 - 8 obtención del contorno.
-

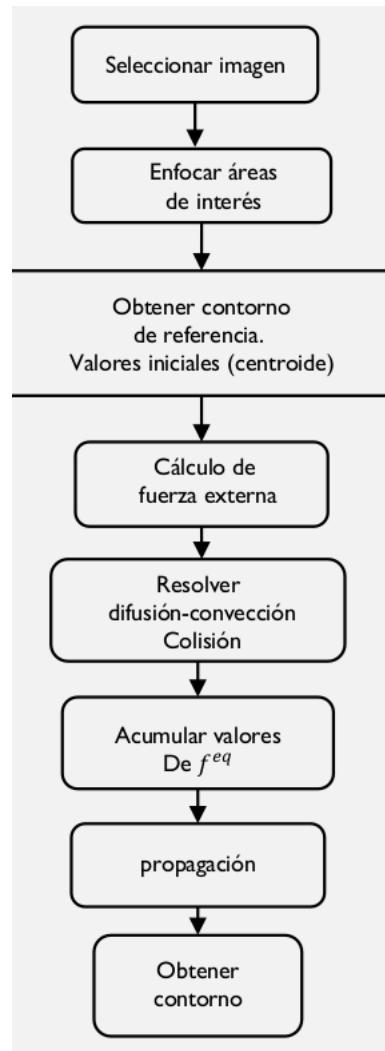


Figura 3.9: Diagrama de flujo que describe el proceso de obtención de contornos en D2Q9 comparado con D2Q5.

3.2. Obtención de imágenes con contorno.

En esta sección se utilizaron diferentes imágenes de la base de datos pública y algunas imágenes proporcionadas por instituciones de salud del país, y el resultado obtenido es el siguiente:

En la figura 3.10, de izquierda a derecha, en la primera columna tenemos las imágenes de la base de datos en formato *.jpg*. En la segunda columna tenemos recortadas sólo las regiones de interés en la imagen y un contorno de referencia trazado con polígonos azules para inicializar el programa. En la tercer columna tenemos trazados contornos obtenidos después de aplicar el algoritmo. En la cuarta columna se muestran los resultados de la máscara obtenida a partir de los contornos obtenidos para cada imagen.

Podemos observar que no sólo aparecen los contornos de los tejidos más densos, sino que también aparece el borde de todo el seno de la paciente con respecto al fondo negro de la imagen, esto debido a que hay una gran diferencia en los valores de fondo con respecto a los pixeles que representan la orilla del tejido. Este no es inconveniente debido a que la máscara en escala de grises que genera Matlab para trazar el contorno se queda almacenada en la variable `phi2` del código y podemos extraerla por separado.

Para llevar a cabo este procedimiento es importante aclarar que se utilizaron condiciones de frontera periódicas [9,10], en el dominio de la imagen (el tamaño de la imagen); esto quiere decir que para realizar una propagación adecuada de f_i se considera que tenemos un dominio en el que al llegar al pixel de cualquiera de los bordes, se continua analizando con el pixel de la fila o columna del otro extremo, simulando un efecto de medio cíclico en el que no hay límites.

Es como si la imagen se transformase en un toriodes. De esta manera se garantiza que no se está limitando la propagación de información de niveles de brillo en una sola sección de la imagen.

Aunque existen otros tipos de condiciones de frontera que se establecen brevemente en las siguientes secciones.

3.2.1. Máscara normalizada.

Debido a que la máscara obtenida consta de un dominio, que es el tamaño de la imagen recortada, se almacenan los distintos valores de escalas de gris que po-

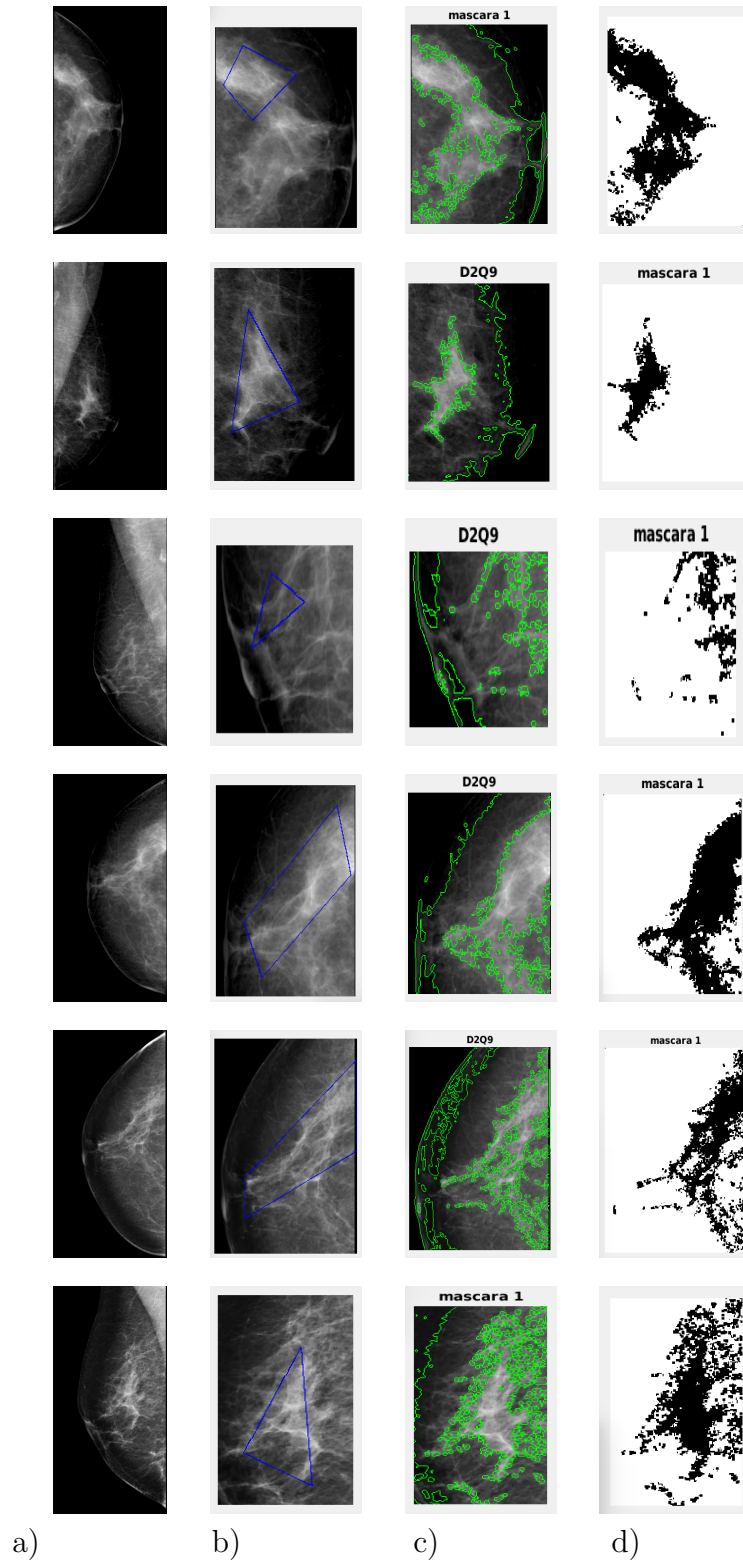


Figura 3.10: Obtención de contornos. a) imagen original, b) región de interés y contorno de referencia, c) contorno obtenido, d) máscara obtenida.

demos obtener con 2^8 posibles combinaciones. Esto nos da 256 diferentes niveles de gris (0 – 255) lo cuál dificultaría obtener el comportamiento de la distribución de probabilidad de toda la imagen (debido a la magnitud del valor de los pixeles). Por tal motivo se debe normalizar la imagen para que los valores queden entre 0 y 1 y se reduce el tiempo de cómputo para la siguiente etapa de este trabajo, que es iniciar una nueva red de Boltzmann con condiciones iniciales dadas por la máscara normalizada.

El tamaño de la máscara regularmente queda en tamaños pequeños de hasta 256 x 256 pixeles.

3.3. Bibliografía capítulo 3.

- [1] A Fast and Robust Level Set Method for Image Segmentation Using Fuzzy Clustering and Lattice Boltzmann Method. Souleymane Balla-Arabé, Xinbo Gao, and Bin Wang. 2013.
- [2] Y. Chen, Z. Yan, and Y. Chu, “Cellular automata based level set method for image segmentation,” in Proc. IEEE/ICME, Beijing, China, May 2007, pp. 23–27.
- [3] W. Chen and M. L. Giger, “A fuzzy c-means (FCM) based algorithm for intensity inhomogeneity correction and segmentation of MR images,” in Proc. IEEE Int. Symp. Biomed. Imaging: Nano Macro, 2004, vol. 2, pp. 1307–1310.
- [4] Level Set Methods and Dynamic Implicit Surfaces. Stanley Osher, Ronald Fedkiw. Applied Mathematical Sciences Volume 153. Springer-Verlag New York. 2003.
- [5] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag and T. Moriarty, “A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data,” in IEEE Transactions on Medical Imaging, vol. 21, no. 3, pp. 193–199, March 2002. doi: 10.1109/42.996338
- [6] A. Hagan and Y. Zhao, “Parallel 3-D image segmentation of large data set on a GPU cluster,” in Proc. ISVC, 2009, pp. 960–969.
- [7] W. Chen and M. L. Giger, “A fuzzy c-means (FCM) based algorithm for intensity inhomogeneity correction and segmentation of MR images,” 2004

2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821), 2004, pp. 1307-1310 Vol. 2. doi: 10.1109/ISBI.2004.1398786

- [8] Simulations of gas-surface reactions. S. A. Adelman and J. D. Doll, J. Chem. Phys. 61, 4242 (1974); J. Chem. Phys. 62, 2518 (1975); J. Chem. Phys. 64, 2375(1976); J. Chem. Phys. 63, 4908 (1975).
- [9] Simulations of many other phenomena, e.g. liquid phase reactions in Chem. Phys. Lett. 90, 215, (1982); J. Chem. Phys. 79, 6312, (1983).
- [10] University of Virginia, MSE 4270/6270: Introduction to Atomistic Simulations, Leonid Zhigilei. 2016.

Análisis de Resultados

4.1. Resumen.

En el capítulo anterior se implementó un algoritmo y se adaptó a las necesidades del problema planteado. Obtener un contorno dentro de una imagen médica que resalte un tejido de interés para el usuario. En este capítulo se va a analizar la calidad de los contornos obtenidos para las imágenes de prueba y se determinará que tan cercano es este contorno a la forma real del tejido denso, para saber que tan confiable es, y además cuales son las posibles alternativas que se tienen para obtener otros resultados. A partir de los contornos obtenidos se establecerán nuevas condiciones iniciales para generar una nueva red de Boltzmann que simule su posible evolución y crecimiento, ahora enfocándose en las condiciones biológicas del crecimiento de tumores.

Los resultados de contornos se evaluarán comparando el método de Chan-Vese con ajustes de nivel, contra el método propuesto y se tomará como referencia la unidad de medida de distancias de Hausdorff.

4.2. Resultados.

Como pudimos observar en las imágenes anteriores de contornos, en la figura 3.10. Es un procedimiento sencillo el poder obtener el contorno de un área en especial de nuestra imagen médica. A pesar de obtener contornos no sólo de los tejidos, sino también del borde externo de la mama, es posible obtener mediante la extracción de una máscara sólo el contorno del tejido en cuestión.

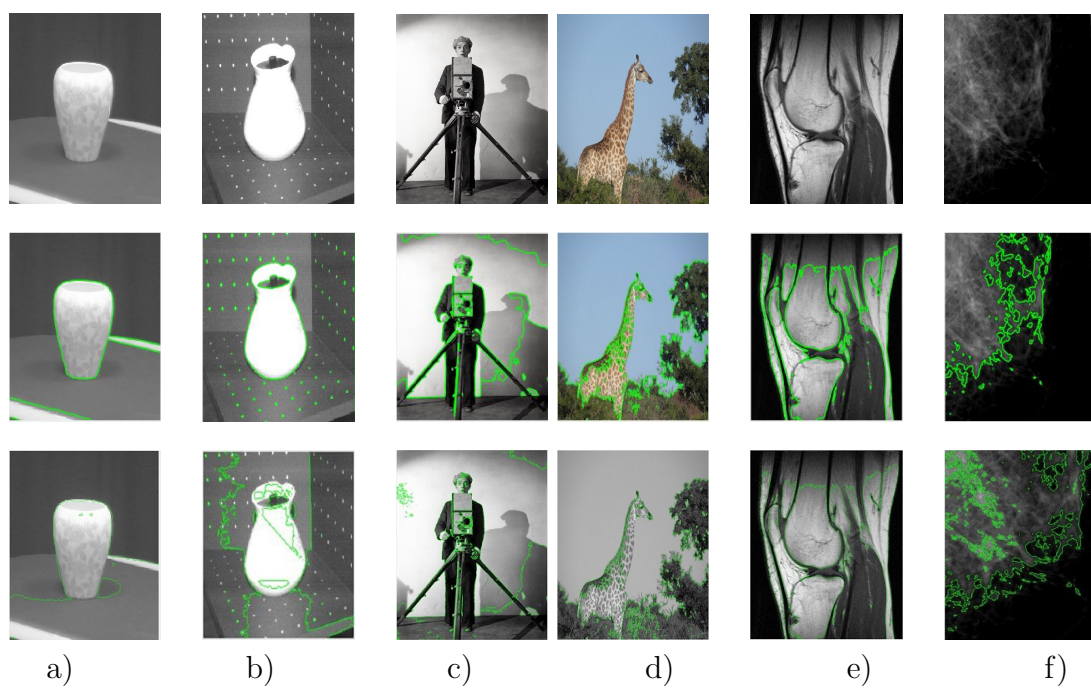


Figura 4.1: Obtención de contornos. Fila 1: imagen de prueba. Fila 2: contorno con el método Chan-Vese. Fila 3: contorno con el método utilizado en este trabajo.

	Distancias Hausdorff
a) Jarrón	114.2979
b) Florero	168.3568
c) Camarógrafo	176.1164
d) Jirafa	273.9069
e) Rodilla	95.8958
f) Mama	226.3957

Tabla 4.1: Distancias obtenidas con la métrica de Hausdorff

Con los resultados obtenidos se completa la primera parte de este trabajo, que consiste en obtener regiones delimitadas por un método confiable y que esté adaptado a generar los nuevos dominios de trabajo para la implementación de la siguiente parte del modelo propuesto, que es obtener ahora las condiciones iniciales de la nueva red de Boltzmann y simular a través de éstas, una posible forma de propagación del contorno. Sin embargo, esto será en función de las características biológicas de crecimiento de un cáncer.

4.2.1. Acerca de la eficiencia en los bordes obtenidos.

En la figura 4.1 se muestra el resultado de aplicar dos métodos distintos sobre la misma imagen. Es claro que al ser una validación con dos métodos que trazan bordes, es necesario obtener una medida de referencia cuantitativa y cualitativa de la calidad de los resultados, por lo cual, se aplicó una función en Matlab para poder realizar la comparación de las imágenes y se obtuvieron los siguientes resultados:

En la tabla 4.1 se puede observar que las distancias obtenidas corresponden visualmente con los contornos en la figura 4.1. Las distancias obtenidas son muy elevadas debido a que cada método traza una gran cantidad de bordes pequeños, que son los que generan estas diferencias muy grandes. Esto refleja que el método implementado en este trabajo no se limita a obtener el contorno en una región de la imagen de prueba (como en el caso de los level set básicos), sino que delimita los contornos sobre toda la imagen. Esto es especialmente importante para el caso de las imágenes de cáncer de mama que se usaron como prueba. El resultado es elemental para poder decir que hay un flujo de información sobre toda la imagen y que es muy importante esta característica para poder realizar ajustes sobre el trabajo futuro.

Los bordes obtenidos por el método propuesto pueden variar en consecuencia a dos parámetros principalmente, que son el tipo de estencil utilizado (en este caso pasamos de un estencil D2Q5 a D2Q9), y al coeficiente de difusión (ρ) en el algoritmo 1 .

Por tal motivo y mediante experimentación se obtuvieron los valores más adecuados en la simulación para encontrar los contornos más cercanos visualmente a la frontera entre el tejido denso y el tejido normal en las imágenes de mamografías, realizando una variación en el tiempo de relajación que depende de γ , y la densidad del medio ρ sin afectar el tiempo de cómputo necesario para la simulación, por lo que el resultado obtenido se considera óptimo en términos de tiempo del cómputo necesario para generar el contorno.

En general, se han propuesto una gran variedad de métodos para resolver problemas de segmentación de imágenes. El más conocido y aplicado es el de contornos activos. Básicamente entra dentro de tres categorías: basado en bordes, basado en regiones, y los modelos de contornos activos híbridos. Utilizan principalmente la información de los bordes como un gradiente en la imagen para poder detener la evolución del contorno, tal es el caso de los contornos activos geodésicos que se desarrollan a partir del modelo de Snakes. Sin embargo, son muy sensibles al ruido y también difícilmente detectan bordes suaves. Tal es el caso del resultado del método empleado para la comparación en la imagen 4.1.

Además, la segmentación depende fuertemente del lugar de la curva inicial. Estos métodos de segmentación comunes tienen limitaciones al segmentar imágenes con grandes variaciones en la intensidad de los píxeles, de hecho, esta variación de intensidad es un fenómeno común en las imágenes médicas, por lo que los resultados obtenidos de aplicar estos modelos queda a discusión del observador [1]. En el método empleado en este trabajo, se obtiene el mismo resultado de la segmentación independiente de la región del contorno inicial, y además, se obtienen bordes en toda la imagen debido a que, en el método de redes de Boltzmann se utiliza una sola interacción para obtener los contornos activos, que, comparado con el método de Chan-Vese que se implementó con 500 iteraciones, delimita la parte central de la mamografía.

4.2.2. Relación con la máscara obtenida.

En todos los casos de aplicación del algoritmo computacional obtuvimos máscaras de la misma dimensión que nuestra imagen de prueba y que cuentan con valores de 0 a 256 en escala de gris (figura 4.1), tal que, después de normalizar los valores no cambia nuestra máscara y se prepara la imagen para un mejor manejo

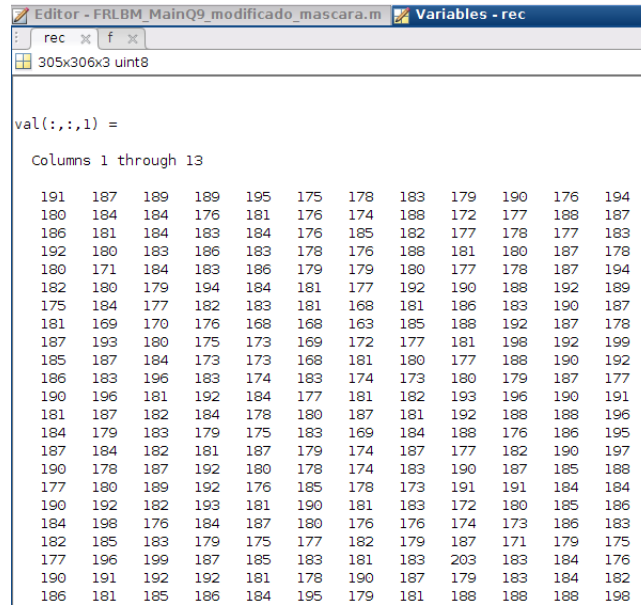


Figura 4.2: Valores de los pixeles antes de normalizarlos.

en la siguiente red de Boltzmann (ver figura 4.2).

En datos experimentales podemos encontrar que las densidades para calcificación de tejidos de cáncer de mama se encuentra en el rango de las -20 y -60 HU y se encuentran el rango de valores 50 y 180 en escala de gris, con lo cual se establece que para estos límites se encuentran las unidades HU.

La relación entre las densidades se obtiene de la siguiente manera:

Si la densidad para los tejidos de cáncer de mama se encuentra en el rango de las -20 y -60 HU y las imágenes en escala de gris sólo muestran valores entre 0 y 255, podemos correlacionar estos valores en una escala lineal debido a que el formato de imagen .jpg utilizado ya está linealizado con respecto al formato de imagen DICOM.

Posteriormente, obtenemos el valor medio de las unidades Hounsfield características del tejido de la mama y tomando en cuenta el valor absoluto obtenemos:

$$V_{med} = \left| \frac{-20 - 60HU}{2} \right| \quad (4.1)$$

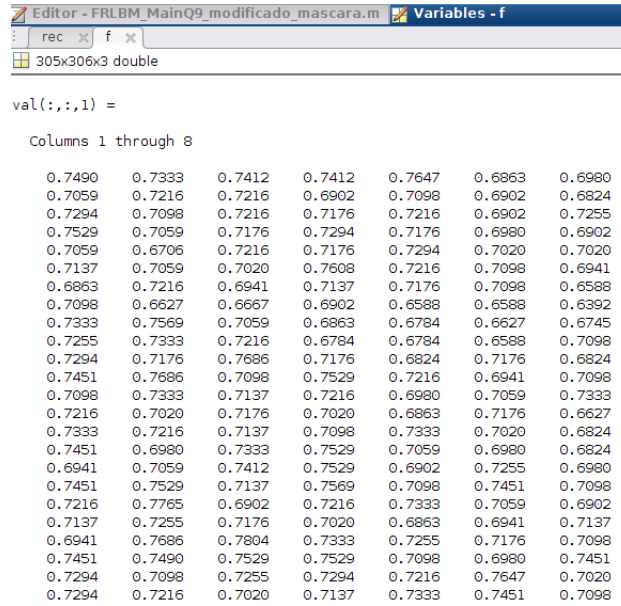


Figura 4.3: Valores de los pixeles después de normalizarlos.

Hasta este punto del trabajo se hizo el planteamiento de resultados, debido al alcance establecido por los tiempos de implementación del método, lo cuál quiere decir que a partir de este punto se concluye con la primera parte.

En la siguiente sección se establecerán los criterios para el trabajo que se realizará como segunda parte del proyecto, y en la que ya se tienen avances significativos para establecer los criterios de la segunda red de Boltzmann aplicada al crecimiento de los tumores.

4.3. Ecuaciones de Crecimiento.

En el trabajo realizado por Anderson et. al. [2,3] podemos observar que durante años se han utilizado algunos valores supuestos hasta el día de hoy para representar la concentración y comportamiento del crecimiento de tumores, partiendo de un modelo de crecimiento conocido como Modelo de invasión [4].

El modelo presentado se clasifica como “híbrido”, ya que es un modelo determinístico en continuo basado en un sistema de ecuaciones de reacción-difusión que combina un modelo discreto de los autómatas celulares basado en una cami-

nata al azar que controla la migración celular y la interacción de células.

En este modelo tenemos los siguientes parámetros:

- Densidad de las células del tumor, denotado por la letra n .
- Una matriz de enzimas degradativas (MDE), denotado por la letra m .
- Mezcla de macromoléculas (MM), denotado por la letra f .
- Concentración de oxígeno, denotado por la letra c .

Donde cada una de estas variables es función del tiempo y espacio.

El modelo consiste en una mezcla de MM como colágeno, fibronectina, laminina y otras. La mayoría de las MM son importantes para la adhesión de células, esparcimiento y motilidad se fijan o se unen al tejido circundante.

Se asume que las células tumorales pueden moverse por simple difusión (motilidad aleatoria), esto da como resultado un gradiente de estas moléculas de adhesión celular unidas. Por lo tanto, se puede constituir una barrera para el movimiento celular normal que también proporciona un sustrato al que las células pueden adherirse y sobre las cuales pueden moverse. La mayoría de los tipos de células de mamíferos requieren que algunos de estos elementos estén presentes para el crecimiento y la supervivencia.

Por definición, la haptotaxis es la respuesta migratoria dirigida de las células a gradientes de químicos (es decir, productos químicos no difusibles). Aunque todavía no se ha demostrado explícitamente que el haptotaxis se produce en una situación in vivo. Sin embargo, dada la estructura del tejido humano, no es irrazonable asumir que la haptotaxis es un componente principal del movimiento dirigido en la invasión de células tumorales. De hecho, ha sido un esfuerzo muy reciente para caracterizar dicho movimiento dirigido. Por lo tanto este movimiento dirigido de las células tumorales es conocido como haptotaxis, es decir, una respuesta a gradientes de MM unidas.

$$J_{hпто} = \chi n \nabla f, \text{ donde } \chi > 0 \text{ es el coeficiente haptotáctico (constante).}$$

Para describir la motilidad aleatoria de las células tumorales, se asume un flujo de la forma $J_{rand} = -D_n \nabla n$, donde D_n es el coeficiente de motilidad aleatoria constante.

Por lo tanto, la ecuación de conservación para la densidad de células tumorales n está por lo tanto dada por:

$$\frac{\partial n}{\partial t} + \nabla \cdot (\mathbf{J}_{rand} + \mathbf{J}_{hapto}) = 0 \quad (4.2)$$

Por lo que la ecuación que gobierna el comportamiento de las células de un tumor es:

$$\frac{\partial n}{\partial t} = D_n \nabla^2 n - \chi \nabla \cdot (n \nabla f) \quad (4.3)$$

También se asume que las matrices de enzimas se degradan en macro moléculas, por lo que este proceso se modela de la siguiente manera:

$$\frac{\partial f}{\partial t} = -\delta m f \quad (4.4)$$

donde δ es una constante positiva.

Una matriz de encimas degradadas consiste en un cúmulo de células de cáncer, que se difunde a través del tejido y bajo un cierto nivel de decaimiento, que podemos representar con la siguiente ecuación:

$$\frac{\partial m}{\partial t} = D_m \nabla^2 m + g(n, m) - h(n, m, f) \quad (4.5)$$

Donde d_m es una constante positiva, g es el coeficiente de difusión en el medio y es función de las células del tumor pertenecientes a la matriz de enzimas, h es el decaimiento. $g = \mu n$ (producción de enzimas por el tumor), y $h = \lambda m$ (decaimiento natural). Obtenemos la siguiente ecuación:

$$\frac{\partial m}{\partial t} = D_m \nabla^2 m + \mu n - \lambda m \quad (4.6)$$

Y como sabemos, las células de los tumores necesitan también de un nivel de consumo de oxígeno para poder sobrevivir. Puede difundirse este oxígeno mediante angiogénesis [3,4] por lo que la ecuación que gobierna este fenómeno es:

$$\frac{\partial c}{\partial t} = D_c \nabla^2 c + \beta f - \gamma n - \alpha c \quad (4.7)$$

Donde D_c , β , γ y α son constantes positivas dadas mediante experimentos.

Por lo que nos queda un sistema de ecuaciones diferenciales parciales de la siguiente manera:

$$\begin{aligned}\frac{\partial n}{\partial t} &= D_n \nabla^2 n - \chi \nabla \cdot (n \nabla f) \\ \frac{\partial f}{\partial t} &= -\delta m f \\ \frac{\partial m}{\partial t} &= D_m \nabla^2 m + \mu n - \lambda m \\ \frac{\partial c}{\partial t} &= D_c \nabla^2 c + \beta f - \gamma n - \alpha c\end{aligned}\tag{4.8}$$

Este sistema de ecuaciones está acoplado y representa como pueden crecer los tumores tomando en cuenta las 4 características más relevantes de un tumor, además de poder ser aplicado a los tumores de cáncer de mama. La manera más adecuada es realizar la simulación y obtener el resultado de las concentraciones utilizando el método de elemento finito mediante la herramienta de FEniCS, tal como se muestra en el apéndice D de esta tesis.

Posteriormente se realizará la simulación en redes de Boltzmann para comparar los resultados de ambos procedimientos y obtener las métricas respectivas para determinar la eficiencia del método propuesto en esta tesis.

4.4. Bibliografía capítulo 4.

- [1] Fangfang Dong, Zengsi Chen, Jinwei Wang, A new level set method for inhomogeneous image segmentation, *Image and Vision Computing*, Volume 31, Issue 10, 2013, Pages 809-822, ISSN 0262-8856, <https://doi.org/10.1016/j.imavis.2013.08.001>
- [2] A. R. Anderson, M. A. J. Chaplain, E. L. Newman, R. J. C. Steele and A. M. Thompson. *Mathematical Modelling of Tumour Invasion and Metastasis* University of Dundee. *Journal of Theoretical Medicine*. Vol 2, pp. 129-151.
- [3] Alexander R. Anderson. A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion. *Division of Mathematics, University of Dundee, UK. Mathematical Medicine and Biology* (2005) 22, 163–186.
- [4] Alexander R. A. Anderson and Vito Quaranta. Integrative mathematical oncology. *Nature reviews cancer*, volume 8 march 2008.

5.1. Conclusiones.

El resultado obtenido de este trabajo deja en claro distintivos puntos en los que principalmente destacan:

- La calidad o cercanía del contorno obtenido con respecto a la forma del tejido explorado puede variar dependiendo del parámetro de viscosidad del dominio y contorno de referencia. Sin embargo, aunque es preferible usar $\rho = 2$, es más acertado el contorno si se incrementa a 2.5, debido a que este parámetro nos indica en qué momento debe dejar existir una propagación.
- La máscara indica cuales serán las condiciones iniciales de la nueva red de Boltzmann aplicada a las ecuaciones de crecimiento de células, mostrando la relación existente entre las unidades de Hounsfield y la densidad de tejido.
- El objetivo del método no es segmentar una región o forma específica de la imagen. Se trata de tomar una imagen médica, seleccionar una región de interés en ella y buscar contornos en toda esa región. Este proceso es indispensable para definir un contorno inicial confiable para generar una nueva red de Boltzmann que complete la segunda parte del trabajo.

La viscosidad en una imagen representa la probabilidad de que se propaga el brillo de un pixel a otro en función de la distancia entre vecinos en la que se está trabajando. Para imágenes es común encontrar que la viscosidad o densidad es el flujo macroscópico en el dominio, que depende de la distancia máxima del valor de cada pixel.

Se obtuvo mediante esta experimentación la parte inicial de una propuesta de simular un crecimiento, que aunque, es una etapa temprana del trabajo en general, sin duda no había sido abordada de esta manera, y por lo tanto considero que es relevante en el ámbito de la investigación de esta enfermedad y por

supuesto en la implementación computacional, visto como un problema de una alta necesidad de cómputo para obtener resultados en un tiempo razonable.

5.2. Trabajo futuro.

Se implementará un nuevo programa que simule el crecimiento de las concentraciones de nutrientes, consumo de oxígeno en células cancerosas, consumo de oxígeno en células sanas y haptotaxis. Esto hará que el contorno crezca y al final del crecimiento simulado se obtendrá un nuevo contorno que se va a comparar con historiales clínicos para verificar que tan cercanos son y que tan probable es poder simular este efecto de crecimiento mediante este método de redes. De esta manera se obtendrá el resultado de una simulación que entrega la posible nueva forma del tumor anticipando a que transcurra un largo periodo de tiempo y se podrá comparar contra el resultado obtenido mediante elemento finito a través de la herramienta FEniCS como método de validación adicional.

Debido a que se debe usar una red por cada concentración que se desea encontrar y aplicarla a cada dirección de la malla D2Q9, los cálculos se convierten en un problema que requiere una solución en términos de cómputo masivo. Se requiere paralelizar el algoritmo computacional, lo cual demanda un alto nivel de recursos cómputo para su implementación, sin embargo, queda también en claro que la alternativa más adecuada para resolver este problema es implementar los nuevos algoritmos en pequeñas secciones de cómputo paralelo para reducir el tiempo de los cálculos, y poder utilizar un equipo de cómputo que satisfaga esta necesidad y esté al alcance de un centro de investigación, probablemente la implementación sea realizada en CUDA [1]. Esto ayudará a concretar la simulación de un proceso de propagación final asociado a cada máscara obtenida en la propuesta.

También se establecerán las mejores condiciones de frontera (en este caso es un sistema cerrado) para que los pasos de propagación nos entreguen el resultado de la simulación en un dominio determinado.

5.3. Bibliografía capítulo 5.

- [1] A new approach to the lattice Boltzmann method for graphics processing units. Christian Obrecht, Frédéric Kuznik, Bernard Tourancheau, Jean-Jacques Roux. *Computers and Mathematics with Applications*. 2011. pp. 3628–3638.

Apéndice: Un método rápido y robusto
de conjuntos de nivel para la
segmentación de imágenes utilizando el
agrupamiento difuso y el método Lattice
Boltzmann

A.1. Código

Esta sección es para identificar el algoritmo propuesto por los autores Souleymane Balla-Arabé, Xinbo Gao, y Bin Wang, y fué implementado en Matlab por Souleymane Balla-Arabé para una malla D2Q5 [1].

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Level Set Segmentation based on Lattice Boltzmann Method %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
% This Matlab code is an serial implementation of the paper:  
% S. Balla-Arabé, X. Gao, and B. Wang, A fast and robust  
% level set method for image segmentation using  
%fuzzy clustering and lattice boltzmann method.  
% IEEE Trans. Syst., Man, Cybern. Part B, Cybern., vol. 43, no. 3,  
% pp. 910 920, 2013.  
%  
% The definition of some symbols are as follow:  
%     J: Image to be segmented  
%     its_max: number of iterations  
%     phi2: updated level set function after level set evolution  
%
```

```
% Author: Souleymane Balla-Arabé, all rights reserved
% School of Electronic Engineering
% Xidian University, Xi'an, China
% E-mail: balla_arabe_souleymane@ieee.org

clear all;

% D2Q5 LATTICE CONSTANTS
ex = [0, 1, 0, -1, 0];
ey = [0, 0, 1, 0, -1];
A = [1/3, 1/6, 1/6, 1/6, 1/6];

% INITIAL DISTANCE FIELD
J= imread ('Image_test.jpg');
I=im2double(J);
%I=imresize(I,0.8);
I=rgb2gray(I);
h = figure;
imshow(J);
hold on;
binary_image = roipoly;
phi = Initial_Mask(binary_image);
contour(phi, [0 0], 'b')

% LOG TRANSFORMATION
lx = size(I,1);
ly = size(I,2);
c = 1/log(2);
Y = zeros(lx,ly);
for x = 1:lx
    for y = 1:ly
        %I(x,y)=log(I(x,y).*255);
        Y(x,y)= c.*log(1 + I(x,y));
        % I=I(x,y).*255;
    end
end

% INITIALIZATION OF THE LBM DISTRIBUTION FUNCTION

Feq = zeros(5,lx,ly);
FEF1 = zeros(lx,ly);
```

```

FEF = zeros(5,lx,ly);
U1 = zeros(lx,ly);
U2 = zeros(lx,ly);
U1d = zeros(lx,ly);
U2d = zeros(lx,ly);
B = zeros(lx,ly);
v1n = zeros(lx,ly);
v1d = zeros(lx,ly);
v2n = zeros(lx,ly);
v2d = zeros(lx,ly);
v1 = 0;
v2 = 1;
Fout = zeros(5,lx,ly);
phi2 = phi - min(phi(:));
for x = 1:lx
    for y = 1:ly
        for li = 1:5
            Feq(li,x,y) = phi2(x,y).* A(li);
        end
    end
end
Fint = Feq;

% SETTING PARAMETERS
gamma = 30;
tau = (9*gamma+2)/4;
its_max = 1;
V = 0.05; % Convection coefficient
p = 2;
pb = zeros(lx,ly);
tic;
for its = 1:its_max
    % COMPUTATION OF THE FUZZY EXTERNAL FORCE
    for x = 1:lx
        for y = 1:ly
            pb(x,y) = Y(x,y)-B(x,y)-v2;
            U1d(x,y) = (Y(x,y)-B(x,y)-v1)./(Y(x,y)-B(x,y)-v2);
            U2d(x,y) = (Y(x,y)-B(x,y)-v2)./(Y(x,y)-B(x,y)-v1);
            U1(x,y) = 1./(1+((Y(x,y)-B(x,y)-v1)./(Y(x,y)-B(x,y)-v2)).^(2/(p-1))));
            U2(x,y) = 1./(1+((Y(x,y)-B(x,y)-v2)./(Y(x,y)-B(x,y)-v1)).^(2/(p-1))));
        end
    end
end

```

```

end

for x = 1:lx
    for y = 1:ly
        v1n(x,y) = (U1(x,y).^p).*(Y(x,y)-B(x,y));
        v1d(x,y) = U1(x,y).^p;
        v2n(x,y) = (U2(x,y).^p).*(Y(x,y)-B(x,y));
        v2d(x,y) = U2(x,y).^p;
    end
end

v1 = sum(v1n(:))/sum(v1d(:));
v2 = sum(v2n(:))/sum(v2d(:));
for x = 1:lx
    for y = 1:ly
        B(x,y) = Y(x,y)-(((U1(x,y).^p)*v1 +
            (U2(x,y).^p)*v2)/(U1(x,y).^p + U2(x,y).^p));
    end
end

for x = 1:lx
    for y = 1:ly
        FEF1(x,y) = (U1(x,y).^p)*(Y(x,y)-B(x,y)-v1).^2 -
            (U2(x,y).^p)*(Y(x,y)-B(x,y)-v2).^2;
    end
end

for x = 1:lx
    for y = 1:ly
        for li = 1:5
            FEF(li,x,y) = FEF1(x,y);
        end
    end
end

% PERFORMING STREAMING-COLLISIONS
for x = 1:lx
    for y = 1:ly
        for li = 1:5
            Fef(li,x,y) = phi2(x,y).* A(li);
        end
    end
end

```

```

for x = 2:lx-1
    for y = 2:ly-1
        for li = 1:5
            Fout(li,x,y)= Fint(li,x,y)-1/tau .*
                (Fint(li,x,y)-Feq(li,x,y))+(2*tau-1)/(2*tau)
                .* FEF(li,x,y).*1000000000000;
            %Fout(li,x,y)= Fint(li,x,y)+(2*tau-1)/(2*tau) .*
                FEF(li,x,y).*1000000000000;
        end
    end
end

for x = 2:lx-1
    for y = 2:ly-1
        for li = 1:5
            Fint(li,x+ex(li),y+ey(li)) = Fout(li,x,y);
        end
    end
end

% UPDATING OF THE DISTANCE VALUE
phi2 = zeros(lx,ly);
for x = 1:lx
    for y = 1:ly
        for li = 1:5
            phi2(x,y) = phi2(x,y) + Fint(li,x,y);
        end
    end
end

figure(h);imshow(J); title('D2Q5');
hold on;
%figure;
contour(phi2,[-min(phi(:)) -min(phi(:))], 'r');

end

tEdII = toc;

```

A.2. Bibliografía

- [1] S. Balla-Arabé, X. Gao, and B. Wang, A fast and robust level set method for image segmentation using fuzzy clustering and lattice boltzmann method. *IEEE Trans. Syst., Man, Cybern. Part B, Cybern.*, vol. 43, no. 3, pp. 910 920, 2013.

Apéndice: Código D2Q9 en redes de Boltzmann

B.1. Características básicas y ecuaciones

Los modelos de redes de Boltzmann simplifican enormemente el diseño conceptual original de Boltzmann, reduciendo el número de posibles posiciones espaciales de las partículas y los momentos microscópicos de un continuo a tan sólo un puñado y similarmente discretizando el tiempo en pasos distintos. [1]

Las posiciones de las partículas son confinadas a los nodos de la red. Las variaciones en los momentos podrían deberse a las direcciones de velocidad y magnitudes y variaciones en la masa de las partículas se reducen a 8 direcciones, 3 magnitudes y una única partícula de masa.

En la siguiente figura se muestra el gráfico cartesiano de cada nodo y las velocidades e_a donde $a = 0, 1, \dots, 8$ es un índice de dirección y $e_0 = 0$ denota las partículas en reposo. Este modelo se conoce como D2Q9 ya que es en 2 dimensiones y contiene 9 velocidades. Este esquema de clasificación LBM fue propuesto por Qian et al. (1992) [2] y es un estencil generalizado.

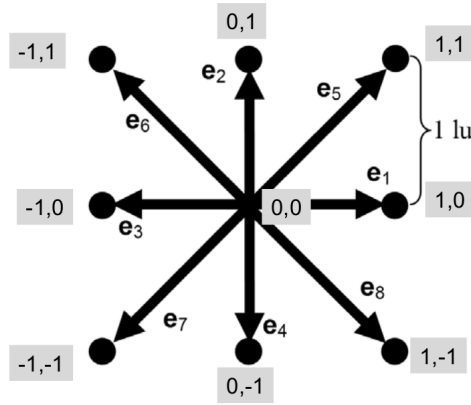


Figura B.1: Esquema del estencil para la malla D2Q9 con velocidades 0,1 y $\sqrt{2}$. La velocidad 0 se localiza en el centro.

Debido a que la masa de la partícula es uniforme (1 unidad de masa desde el enfoque más simple), estas velocidades microscópicas y los momentos son siempre equivalentes. La unidad es la medida fundamental de la longitud en los modelos de Boltzmann y la medida de los pasos de tiempo es la unidad de tiempo.

```

%*****Poiseuille flow using LBM*****%
%Based on procedures as explained in 'Lattice Gas Cellular Automata and Lattice
%Models'by Wolf Gladrow
%code may have errors as it is my first experience with LBM.Readers are
%suggested to check the code for errors.For feedback
%vinuvargheseijk@gmail.com
OMEGA=0.2;%Relaxation factor
TAU=1/OMEGA;
XMAX=20;%Mesh size in x-direction
YMAX=20;%Mesh size in y-direction
FORCING=(1.024)/YMAX^3;%Forcing term in Poiseuille equation
KVISC = ((1/OMEGA)-0.5)/3;%lattice viscosity
UX=(0.5*FORCING*YMAX*YMAX)/KVISC;%Velocity in x-direction
cx=[1 0 -1 0 1 -1 -1 1 0];%components of lattice velocities in x-direction
cy=[0 1 0 -1 1 1 -1 -1 0];%components of lattice velocities in y-direction
jx=zeros(XMAX,YMAX);%creating an array for storing momentum in x-direction
jy=zeros(XMAX,YMAX);%creating an array for storing momentum in y-direction
rho=ones(XMAX,YMAX);%creating an array for storing densities
f=zeros(XMAX,YMAX);%creating an array for storing distributions
fprop=zeros(XMAX,YMAX);%creating an array for storing propagating distributions
x=1:XMAX;
    
```

```

y=1:YMAX;
MAXIT=input('Number of iterations');
%Initializataion of distributions
    u=jx./rho;
    v=jy./rho;
    f(x,y,1) = (rho./9)*(1+3.*u+4.5.*u.*u-1.5*(u.^2+v.^2));
    f(x,y,2) = (rho./9)*(1+3.*v+4.5.*v.*v-1.5*(u.^2+v.^2));
    f(x,y,3) = (rho./9)*(1-3.*u+4.5.*u.*u-1.5*(u.^2+v.^2));
    f(x,y,4) = (rho./9)*(1-3.*v+4.5.*v.*v-1.5*(u.^2+v.^2));

    f(x,y,5) = (rho./36)*(1+3.*(u+v)+4.5*(u+v).^2-1.5*(u.^2+v.^2));
    f(x,y,6) = (rho./36)*(1+3.*(-u+v)+4.5*(-u+v).^2-1.5*(u.^2+v.^2));
    f(x,y,7) = (rho./36)*(1-3.*(u+v)+4.5*(u+v).^2-1.5*(u.^2+v.^2));
    f(x,y,8) = (rho./36)*(1+3.*(u-v)+4.5*(u-v).^2-1.5*(u.^2+v.^2));

    f(x,y,9) = (4/9)*rho.*(1-1.5*(u.^2+v.^2));
%Initialization completed
%Assign first fprop to equilibrium distributions.
    fprop=f;
    feq=zeros(20,20,9);
for iter=1:MAXIT
    u=jx./rho;
    v=jy./rho;

    feq(x,2:YMAX-1,1) = (rho(x,2:YMAX-1)./9).*
        (1+3.*u(x,2:YMAX-1)+4.5.*u(x,2:YMAX-1).*u(x,2:YMAX-1)-1.5.*
        (u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
    feq(x,2:YMAX-1,2) = (rho(x,2:YMAX-1)./9).*
        (1+3.*v(x,2:YMAX-1)+4.5.*v(x,2:YMAX-1).*v(x,2:YMAX-1)-1.5.*
        (u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
    feq(x,2:YMAX-1,3) = (rho(x,2:YMAX-1)./9).*
        (1-3.*u(x,2:YMAX-1)+4.5.*u(x,2:YMAX-1).*u(x,2:YMAX-1)-1.5.*
        (u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
    feq(x,2:YMAX-1,4) = (rho(x,2:YMAX-1)./9).*
        (1-3.*v(x,2:YMAX-1)+4.5.*v(x,2:YMAX-1).*v(x,2:YMAX-1)-1.5.*
        (u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));

    feq(x,2:YMAX-1,5) = (rho(x,2:YMAX-1)./36).*
        (1+3.*(u(x,2:YMAX-1)+v(x,2:YMAX-1))+4.5*(u(x,2:YMAX-1)
        +v(x,2:YMAX-1)).^2-1.5*(u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
    feq(x,2:YMAX-1,6) = (rho(x,2:YMAX-1)./36).*

```



```
(1+3.*(-u(x,2:YMAX-1)+v(x,2:YMAX-1))+4.5*(-u(x,2:YMAX-1)
+v(x,2:YMAX-1)).^2-1.5*(u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
feq(x,2:YMAX-1,7) = (rho(x,2:YMAX-1)./36).*
(1-3.*(u(x,2:YMAX-1)+v(x,2:YMAX-1))+4.5*(u(x,2:YMAX-1)
+v(x,2:YMAX-1)).^2-1.5*(u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
feq(x,2:YMAX-1,8) = (rho(x,2:YMAX-1)./36).*
(1+3.*(u(x,2:YMAX-1)-v(x,2:YMAX-1))+4.5*(u(x,2:YMAX-1)
-v(x,2:YMAX-1)).^2-1.5*(u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
feq(x,2:YMAX-1,9) = ((4/9)*rho(x,2:YMAX-1)).*
(1-1.5*(u(x,2:YMAX-1).^2+v(x,2:YMAX-1).^2));
```

%propagating distributions fprop by applying kinetic equation.

```
fprop = (1.0-OMEGA).* f + OMEGA.* feq;
force =FORCING/6;
fprop(x,2:YMAX-1,1) = fprop(x,2:YMAX-1,1) + force;
fprop(x,2:YMAX-1,3) = fprop(x,2:YMAX-1,3) - force;
fprop(x,2:YMAX-1,5) = fprop(x,2:YMAX-1,5) + force;
fprop(x,2:YMAX-1,6) = fprop(x,2:YMAX-1,6) - force;
fprop(x,2:YMAX-1,7) = fprop(x,2:YMAX-1,7) - force;
fprop(x,2:YMAX-1,8) = fprop(x,2:YMAX-1,8) + force;
```

%Copying the boundary values.

```
fprop(x,1,:) = f(x,1,:);
fprop(x,YMAX,:) = f(x,YMAX,:);
```

```
%      C6  C2  C5      ^  y
%      \  |  /      |
%      C3-C0-C1      |
%      /  |  \      |
%      C7  C4  C8      -----> x
p=2:XMAX;
q=2:YMAX;
r=1:XMAX-1;
s=1:YMAX-1;
```

%Propagating C1

```
f(p,y,1) = fprop(p-1,y,1);
```

%Propagating C2

$$f(x,q,2) = fprop(x,q-1,2);$$

%Propagating C3

$$f(r,y,3) = fprop(r+1,y,3);$$

%Propagating C4

$$f(x,s,4) = fprop(x,s+1,4);$$

%Propagating C5

$$f(p,q,5) = fprop(p-1,q-1,5);$$

%Propagating C6

$$f(r,p,6) = fprop(r+1,p-1,6);$$

%Propagating C7

$$f(r,s,7) = fprop(r+1,s+1,7);$$

%Propagating C8

$$f(p,s,8) = fprop(p-1,s+1,8);$$

%Propagating C9

$$f(x,y,9) = fprop(x,y,9);$$

%Complete Bounce Back Boundary Conditions

%1.Implementing Periodic BC.

$$f(x(1),y,1)=fprop(XMAX,y,1);$$

$$f(XMAX,y,3)=fprop(x(1),y,3);$$

$$f(x(1),2:YMAX,5)=fprop(XMAX,1:YMAX-1,5);$$

$$f(XMAX,2:YMAX,6)=fprop(x(1),1:YMAX-1,6);$$

```

f(XMAX,1:YMAX-1,7)=fprop(x(1),2:YMAX,7);
f(x(1),1:YMAX-1,8)=fprop(XMAX,2:YMAX,8);

    %2.Bounce Back Begins.
temp=f(1:XMAX,1,2);
f(1:XMAX,1,2)=f(1:XMAX,1,4);
f(1:XMAX,1,4)=temp;

temp=f(1:XMAX,YMAX,4);
f(1:XMAX,YMAX,4)=f(1:XMAX,YMAX,2);
f(1:XMAX,YMAX,2)=temp;

temp=f(1:XMAX,1,5);
f(1:XMAX,1,5)=f(1:XMAX,1,7);
f(1:XMAX,1,7)=temp;

temp=f(1:XMAX,YMAX,7);
f(1:XMAX,YMAX,7)=f(1:XMAX,YMAX,5);
f(1:XMAX,YMAX,5)=temp;

temp=f(1:XMAX,1,6);
f(1:XMAX,1,6)=f(1:XMAX,1,8);
f(1:XMAX,1,8)=temp;

temp=f(1:XMAX,YMAX,8);
f(1:XMAX,YMAX,8)=f(1:XMAX,YMAX,6);
f(1:XMAX,YMAX,6)=temp;

rho(x,y) = f(x,y,1)+f(x,y,2)+f(x,y,3)+f(x,y,4)+f(x,y,5)
+f(x,y,6)+f(x,y,7)+f(x,y,8)+f(x,y,9);
jx(x,y)=f(x,y,1)-f(x,y,3)+f(x,y,5)-f(x,y,6)-f(x,y,7)+f(x,y,8);
%Distributions multiplied by lattice velocities in x-directions
jy(x,y)=f(x,y,2)-f(x,y,4)+f(x,y,5)+f(x,y,6)-f(x,y,7)-f(x,y,8);
%Distributions multiplied by lattice velocities in y-directions
uprofile(1:YMAX)=sum(jx(1:XMAX,1:YMAX)./rho(1:XMAX,1:YMAX))/XMAX;
%Velocity profile in x-direction
plot(uprofile)
pause(.1)
end

```

B.2. Bibliografía

- [1] Mark E. Davis. Numerical Methods and modeling for chemical engineers. California Institute of Technology. May 14, 2001.
- [2] Cellular automata and lattice Boltzmann methods: a new approach to computational fluid dynamics and particle transport. Bastien Chopard, Alexandre Masselot. University of Geneva, Computer Science Department. March 1999.

Apéndice: FEniCS para resolver ecuaciones diferenciales

C.1. Sistema de reacción-difusión-advección

En ocasiones encontramos problemas con una característica en común: involucran modelos que se expresan como un sistema de ecuaciones diferenciales parciales (EDP) que describen el comportamiento de dicho sistema analizando una o varias variables físicas. Tal es el caso de las ecuaciones de Navier-Stokes. Un método para resolver estos sistemas de ecuaciones es resolver una ecuación a la vez y alimentar la siguiente ecuación con la solución de la anterior.

Una ventaja que posee FEniCS es la facilidad con que uno puede definir problemas variacionales con los que cuentan algunos sistemas de ecuaciones. A continuación se muestra un ejemplo de como implementar estas ecuaciones en python y obtener la solución del sistema.

C.2. Problema de Ecuaciones Diferenciales Parciales

Se presenta el siguiente sistema de reacción-difusión-advección:

$$\frac{\partial u_1}{\partial t} + w \cdot \nabla u_1 - \nabla \cdot (\epsilon \nabla u_1) = f_1 - K u_1 u_2 \quad (\text{C.1})$$

$$\frac{\partial u_2}{\partial t} + w \cdot \nabla u_2 - \nabla \cdot (\epsilon \nabla u_2) = f_2 - K u_1 u_2 \quad (\text{C.2})$$

$$\frac{\partial u_3}{\partial t} + w \cdot \nabla u_3 - \nabla \cdot (\epsilon \nabla u_3) = f_3 - K u_1 u_2 - K u_3 \quad (\text{C.3})$$

Este sistema modela la reacción química entre dos especies A y B en un dominio Ω :



Suponemos que la reacción es de primer orden, es decir, que el nivel de la reacción es proporcional a las concentraciones $[A]$ y $[B]$ de las dos especies A y B:

$$\frac{d}{dt}[C] = K[A][B]$$

También suponemos que la especie formada C decae con un rango proporcional a la concentración $[C]$. En el sistema de ecuaciones presentado, utilizamos las variables u_1 , u_2 y u_3 para denotar las concentraciones de las 3 especies:

$$u_1 = [A], \quad u_2 = [B], \quad u_3 = [C].$$

Las reacciones químicas toman parte en cada punto del dominio Ω . Además, suponemos que las especies A, B y C se difunden a través del dominio con una difusividad ϵ (los términos $-\nabla \cdot (\epsilon \nabla u_i)$) y advección con una velocidad w (los términos $w \cdot \nabla u_i$).

Para hacer las cosas visual y físicamente interesantes, dejaremos que la reacción tenga lugar en el campo de velocidad dado por la solución de las ecuaciones de fluidos incompresibles de Navier-Stokes alrededor de un cilindro. En resumen, se dará solución al sistema de ecuaciones de la siguiente manera:

$$\rho \left(\frac{\partial w}{\partial t} + w \cdot \nabla w \right) = \nabla \cdot \sigma(w, p) + f \tag{C.4}$$

$$\nabla \cdot w = 0 \tag{C.5}$$

$$\frac{\partial u_1}{\partial t} + w \cdot \nabla u_1 - \nabla \cdot (\epsilon \nabla u_1) = f_1 - K u_1 u_2 \tag{C.6}$$

$$\frac{\partial u_2}{\partial t} + w \cdot \nabla u_2 - \nabla \cdot (\epsilon \nabla u_2) = f_2 - K u_1 u_2 \tag{C.7}$$

$$\frac{\partial u_3}{\partial t} + w \cdot \nabla u_3 - \nabla \cdot (\epsilon \nabla u_3) = f_3 - K u_1 u_2 - K u_3 \tag{C.8}$$

Asumimos que $u_1 = u_2 = u_3 = 0$ en $t = 0$ y que colocamos las especies dentro de un sistema especificando los términos de "no cero" f_1 y f_2 cerca de las esquinas de la entrada, y tomamos $f_3 = 0$. El resultado será que A y B están conectados por advección y difusión en todo el medio, y cuando se mezclan forman la especie C.

C.3. Formulación variacional

Obtendremos la formulación variacional de nuestro sistema multiplicando cada ecuación por una función de prueba, integrando los términos de segundo orden $-\nabla \cdot (\epsilon \nabla u_1)$ por partes, y sumando las ecuaciones. Al trabajar con FEniCS es conveniente pensar en el sistema de ecuaciones diferenciales parciales como un vector de ecuaciones. Las funciones de prueba se acumulan en un vector también, y la formulación variacional es el producto interno del vector de ecuaciones y el vector de la función de prueba.

También es necesario introducir una discretización en tiempo. Se usará el método de Backward Euler [1] para resolver la ecuación de calor y aproximar las derivadas en tiempo $(u_i^{n+1} - u_i^n) / \Delta t$.

Usaremos v_1 , v_2 y v_3 como las funciones de prueba o los componentes del vector de prueba: El producto interno resulta en:

$$\begin{aligned}
 & \int_{\Omega} (\Delta t^{-1} (u_1^{n+1} - u_1^n) v_1 + w \cdot \nabla u_1^{n+1} v_1 + \epsilon \nabla u_1^{n+1} \cdot \nabla v_1) dx & (C.9) \\
 & + \int_{\Omega} (\Delta t^{-1} (u_2^{n+1} - u_2^n) v_2 + w \cdot \nabla u_2^{n+1} v_2 + \epsilon \nabla u_2^{n+1} \cdot \nabla v_2) dx \\
 & + \int_{\Omega} (\Delta t^{-1} (u_3^{n+1} - u_3^n) v_3 + w \cdot \nabla u_3^{n+1} v_3 + \epsilon \nabla u_3^{n+1} \cdot \nabla v_3) dx \\
 & - \int_{\Omega} (f_1 v_1 + f_2 v_2 + f_3 v_3) dx \\
 & - \int_{\Omega} (-K u_1^{n+1} u_2^{n+1} v_1 - K u_1^{n+1} u_2^{n+1} v_2 + K u_1^{n+1} u_2^{n+1} v_3 - K u_3^{n+1} v_3) dx = 0
 \end{aligned}$$

Para este problema es natural pensar que tenemos condiciones de frontera de tipo Neumann a lo largo de toda la frontera u_1 , u_2 y u_3 ; eso significa que $\partial u_i / \partial n = 0$ para $i = 1, 2, 3$, por lo tanto, los términos en la frontera desaparecen al integrar por partes.

C.4. Implementación en FEniCS

El primer paso es leer el archivo con la rejilla que se sugiere en la página de los ejemplos [2]:

```
mesh = Mesh('navier_stokes_cylinder/cylinder.xml.gz')
```

La rejilla se guarda en un archivo de FEniCS de formato XML nativo. Después, se necesita definir el espacio de funciones de elemento finito. Para este problema se definirán diferentes espacios.

El primer espacio creado es para el campo de velocidad w de la simulación de Navier-Stokes. Se llamará W y se definirá como:

```
W = VectorFunctionSpace(mesh, 'P', 2)
```

Es importante que ese espacio sea exactamente del mismo tamaño que el espacio usado para el campo de velocidad de la solución de Navier-Stokes. Para leer los valores del campo de velocidad se usa la instrucción `TimeSeries`:

```
timeseries_w = TimeSeries('navier_stokes_cylinder/velocity_series')
```

Esto inicializa el objeto `timeseries_w` el cual llamaremos después en el ciclo de pasos de tiempo (time-stepping loop) para devolver los valores del campo de velocidad del archivo `velocity_series.h5` en un formato HDF5.

Para las 3 concentraciones u_1 , u_2 y u_3 tenemos que crear un espacio que mezcle las 3 funciones que representan nuestro sistema completo (u_1, u_2, u_3) como una única entidad. Para hacerlo, se define un espacio “MixedElement” como el producto de 3 elementos finitos simples y se usa el elemento mezclado para definir el espacio de funciones:

```
P1 = FiniteElement('P', triangle, 1)
element = MixedElement([P1, P1, P1])
V = FunctionSpace(mesh, element)
```

Una vez que este espacio ha sido creado, hay que definir la función de prueba y la función del elemento finito. Las funciones mezcladas pueden ser creadas reemplazando las funciones de prueba por la variable funciones de prueba:

```
v_1, v_2, v_3 = TestFunctions(V)
```


Dado que el problema es no lineal, necesitamos trabajar con funciones en lugar de funciones de prueba para las incógnitas. Esto puede hacerse utilizando las funciones de construcción en FEniCS. Sin embargo, como tendremos que acceder a la función para todo el sistema en sí, primero tenemos que crear esa función y luego acceder a sus componentes:

```
u = Function(V)
u_1, u_2, u_3 = split(u)
```

Esas funciones se usarán para representar las incógnitas u_1 , u_2 y u_3 al nuevo nivel $n + 1$. Los valores correspondientes al nivel de tiempo anterior n son denotados por u_{n1} , u_{n2} y u_{n3} en este programa de ejemplo. Cuando sean conocidas todas las funciones de prueba que teníamos definidas, podemos expresar el problema variacional no lineal como:

```
F=((u_1 - u_n1) / k)*v_1*dx + dot(w, grad(u_1))*v_1*dx \
+ eps*dot(grad(u_1), grad(v_1))*dx + K*u_1*u_2*v_1*dx \
+ ((u_2 - u_n2) / k)*v_2*dx + dot(w, grad(u_2))*v_2*dx \
+ eps*dot(grad(u_2), grad(v_2))*dx + K*u_1*u_2*v_2*dx \
+ ((u_3 - u_n3) / k)*v_3*dx + dot(w, grad(u_3))*v_3*dx \
+ eps*dot(grad(u_3), grad(v_3))*dx - K*u_1*u_2*v_3*dx + K*u_3*v_3*dx \
- f_1*v_1*dx - f_2*v_2*dx - f_3*v_3*dx
```

Los pasos de tiempo simplemente consisten en resolver este problema variacional en cada paso de tiempo mediante una llamada a la función de solución:

```
t=0
for n in range(num_steps):
    t+=dt
    timeseries_w.retrieve(w.vector(), t)
    solve(F == 0, u)
    u_n.assign(u)
```

En cada paso de tiempo primero se lee el valor actual del campo de velocidad de la serie de tiempos que teníamos previamente almacenada, después se resuelve el sistema no lineal y se asignan los valores calculados al lado izquierdo de la rejilla para el siguiente paso de tiempo. Cuando se reciben valores de la serie de tiempo, los valores serán por default interpolados (linealmente) para el tiempo dado t , si el tiempo no corresponde exactamente con la muestra en la serie.

La solución final del tiempo se muestra en la siguiente figura. Se puede ver claramente la advección (unión) de las especies A y B, y la formación de la especie C a lo largo del canal donde A y B se unen.

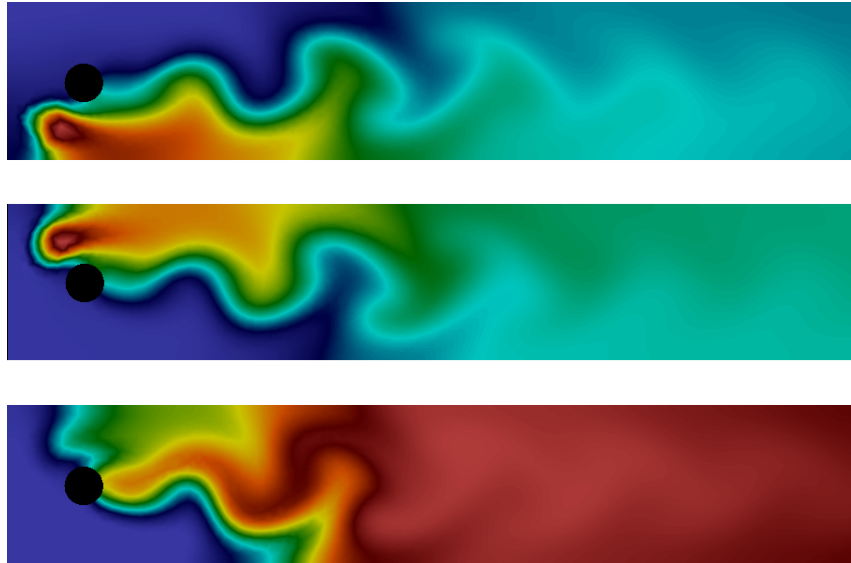


Figura C.1: Gráfica de concentraciones $[A]$, $[B]$ y $[C]$ (desde arriba hacia abajo).

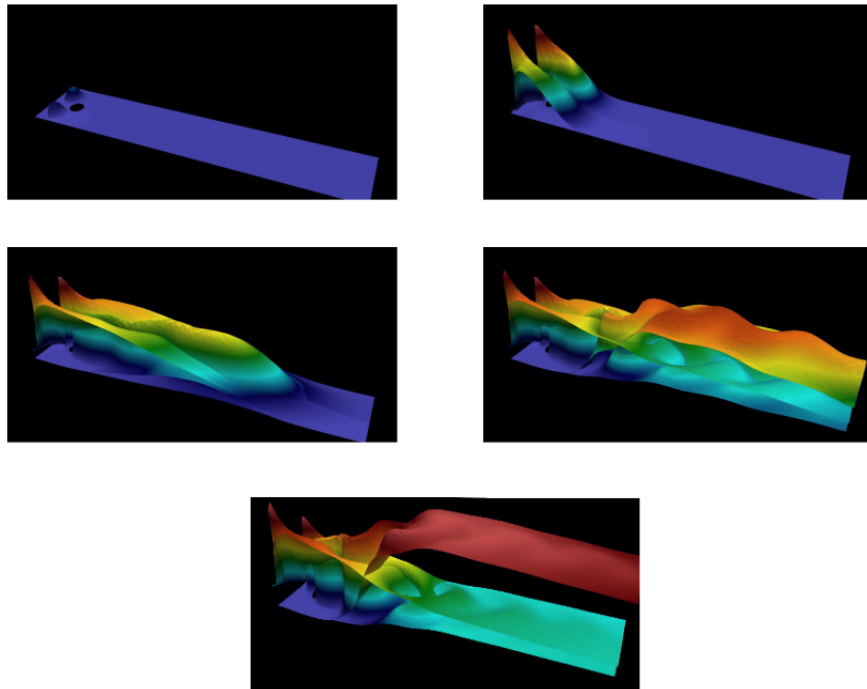


Figura C.2: Gráfica de concentraciones $[A]$, $[B]$ y $[C]$ en diferentes pasos de tiempo.

C.4.1. Código completo

```
from fenics import *
T = 5.0 #final time
num_steps = 500 #number of time steps
dt = T / num_steps #time step size
eps = 0.01 #diffusion coefficient
K = 10.0 #reaction rate

# Read mesh from file
mesh = Mesh('navier_stokes_cylinder/cylinder.xml.gz')

# Define function space for velocity
W = VectorFunctionSpace(mesh, 'P', 2)

# Define function space for system of concentrations
P1 = FiniteElement('P', triangle, 1)
element = MixedElement([P1, P1, P1])
V = FunctionSpace(mesh, element)

# Define test functions
v_1, v_2, v_3 = TestFunctions(V)

# Define functions for velocity and concentrations
w = Function(W)
u = Function(V)
u_n = Function(V)

# Split system functions to access components
u_1, u_2, u_3 = split(u)
u_n1, u_n2, u_n3 = split(u_n)

# Define source terms
f_1 = Expression('pow(x[0]-0.1,2)+pow(x[1]-0.1,2)<0.05*0.05 ? 0.1 : 0',
degree=1)

f_2 = Expression('pow(x[0]-0.1,2)+pow(x[1]-0.3,2)<0.05*0.05 ? 0.1 : 0',
degree=1)

f_3 = Constant(0)
```

```
# Define expressions used in variational forms
k = Constant(dt)
K = Constant(K)
eps = Constant(eps)

# Define variational problem
F = ((u_1 - u_n1) / k)*v_1*dx + dot(w, grad(u_1))*v_1*dx \
+ eps*dot(grad(u_1), grad(v_1))*dx + K*u_1*u_2*v_1*dx \
+ ((u_2 - u_n2) / k)*v_2*dx + dot(w, grad(u_2))*v_2*dx \
+ eps*dot(grad(u_2), grad(v_2))*dx + K*u_1*u_2*v_2*dx \
+ ((u_3 - u_n3) / k)*v_3*dx + dot(w, grad(u_3))*v_3*dx \
+ eps*dot(grad(u_3), grad(v_3))*dx - K*u_1*u_2*v_3*dx + K*u_3*v_3*dx \
- f_1*v_1*dx - f_2*v_2*dx - f_3*v_3*dx

# Create time series for reading velocity data
timeseries_w = TimeSeries('navier_stokes_cylinder/velocity_series')

# Create VTK files for visualization output
vtkfile_u_1 = File('reaction_system/u_1.pvd')
vtkfile_u_2 = File('reaction_system/u_2.pvd')
vtkfile_u_3 = File('reaction_system/u_3.pvd')

# Create progress bar
progress = Progress('Time-stepping')
set_log_level(PROGRESS)

# Time-stepping
t = 0
for n in range(num_steps):

# Update current time
t += dt

# Read velocity from file
timeseries_w.retrieve(w.vector(), t)

# Solve variational problem for time step
solve(F == 0, u)

# Save solution to file (VTK)
_u_1, _u_2, _u_3 = u.split()
```

```
vtkfile_u_1 << (_u_1, t)
vtkfile_u_2 << (_u_2, t)
vtkfile_u_3 << (_u_3, t)

# Update previous solution
u_n.assign(u)

# Update progress bar
progress.update(t / T)

# Hold plot
interactive()
```

Este programa es un ejemplo que se puede encontrar en el archivo *ft09_reaction_system.py*

Finalmente, comentamos tres técnicas importantes que son muy útiles al trabajar con sistemas de PDEs:

1. Establecer las condiciones iniciales,
2. Condiciones de frontera,
3. Extracción de componentes del sistema para graficar o post procesar.

Establecimiento de condiciones iniciales para sistemas mixtos.

En este ejemplo, no es necesario preocuparse por establecer una condición inicial, ya que comenzamos con $u_1 = u_2 = u_3 = 0$. Esto ocurre automáticamente en el código cuando establecemos $u_n = Función(V)$. Esto crea una función para todo el sistema y los grados de libertad se ponen a cero. Si se quiere establecer las condiciones iniciales para los componentes del sistema de manera separada, la solución más fácil es definir las condiciones iniciales como un valor vectorial expresión y luego proyectar (o interpolar) éste a la función representativa del sistema completo. Por ejemplo:

```
u_0 = Expression(('sin(x[0])', 'cos(x[0]*x[1])', 'exp(x[1])'), degree=1)
u_n = project(u_0, V)
```

Esto define a u_1 , u_2 y u_3 como proyecciones de $\sin x$, $\cos(xy)$, y $\exp(y)$, respectivamente.

Establecimiento de condiciones de frontera para sistemas mixtos.

En este ejemplo, tampoco fué necesario preocuparse por establecer condiciones de frontera ya que utilizamos una condición natural de Neumann. Si queremos establecer las condiciones de Dirichlet (componentes individuales del sistema), esto puede hacerse con la clase *DirichletBC*, pero debemos especificar para qué subsistema establecemos la condición de frontera. Por ejemplo, para especificar que u_2 debe ser igual a xy en el límite definido por la frontera, tenemos qué:

```
u_D = Expression('x[0]*x[1]', degree=1)
bc = DirichletBC(V.sub(1), u_D, boundary)
```

El objeto *bc* es una lista de muchos objetos que contiene diferentes condiciones de frontera. A continuación, se pasa a la función de la solución como de costumbre. Se debe tener en cuenta que la numeración comienza en 0 en FEniCS por lo que el subespacio correspondiente a u_2 es *V.sub(1)*.

Acceso a los componentes de sistemas mixtos

Acceso a componentes de sistemas mixtos. Si u es una función definida en un espacio de función mixta en FEniCS, hay varias maneras en que u pueda ser dividida en componentes. Ya hemos visto un ejemplo:

```
u_1, u_2, u_3 = split(u)
```

Esto extrae los componentes de u como símbolos que pueden ser utilizados en un problema variacional. Este estado es, de hecho, equivalente a:

```
u_1 = u[0]
u_2 = u[1]
u_3 = u[2]
```

Se debe tener en cuenta que $u[0]$ no es realmente un objeto *Function*, sino una expresión simbólica como el $\text{grad}(u)$ en FEniCS es una expresión simbólica y no una función que representa el gradiente. Esto significa que u_1 , u_2 , u_3 pueden ser utilizados en un problema variacional, pero no pueden ser usados para graficar o post procesar.

Para acceder a los componentes de u para graficar y guardar la solución en un archivo, necesitamos usar una variante de la función por separado:

```
u_1_, u_2_, u_3_ = u.split()
```

Esto devuelve tres subfunciones como objetos reales con acceso a los datos subyacentes almacenados en u , lo que hace que la gráfica y el almacenamiento en el archivo sea posible. Alternativamente, tenemos qué:

```
u_1_, u_2_, u_3_ = u.split(deepcopy=True)
```

que creará $u_1_$, $u_2_$ y $u_3_$ como objetos función independientes, cada una de las cuales contiene una copia de los datos de subfunción extraídos de u . Esto es útil en muchas situaciones, pero no es necesario para graficar y guardar soluciones en un archivo.

C.5. Bibliografía

- [1] Mark E. Davis. Numerical Methods and modeling for chemical engineers. California Institute of Technology. May 14, 2001.
- [2] The FEniCS tutorial <https://fenicsproject.org/tutorial>.