

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

MODELOS FORMALES PARA LA SIMULACIÓN DE LA EPIDERMIS HUMANA

CARLOS ALFONSO CASTAÑEDA MARROQUIN

**Tesis para optar por el título de Doctor
en Ingeniería Informática y de Telecomunicación**

Director: Manuel Alfonseca Moreno

Director: Alfonso Ortega de la Puente



ENERO DE 2014

Resumen

La epidermis humana es un ejemplo de sistema complejo. Está compuesto por multitud de copias de diferentes tipos de células. El comportamiento del sistema completo emerge de las conductas individuales de sus células. Se han desarrollado muchos modelos que describen la conducta individual de las células. En muchas ocasiones, el conocimiento que se tiene de la aportación de la conducta de cada célula a la conducta del tejido completo es de alto grado de abstracción y la poseen los expertos de este dominio. Estas conductas individuales son bien conocidas. Son muchas las razones por las que puede resultar de interés la simulación de este órgano. Por todo esto, estamos ante las características que habitualmente hacen provechoso un enfoque basado en modelos de cómputo bioinspirados. Los autómatas celulares serían uno de los sistemas que, por su definición, podrían considerarse ideales para abordar la simulación de la epidermis. Sin embargo, una característica fundamental tiene que ser incorporada a cualquier simulador de la epidermis: mantener una configuración de mínima energía en cada instante. Los modelos matemáticos más prometedores en la simulación de tejidos y que, por tanto, incorporan esta característica, son la familia de modelos que parte del modelo de Ising, sigue por el modelo de Potts y el modelo extendido a células de Potts y termina con el modelo CPM-GGH. Una de las principales limitaciones para la aplicación de este modelo para problemas reales relacionados con la epidermis es el rendimiento. Una vía tradicional para subsanar esta limitación es el acceso a recursos masivamente paralelos mediante versiones paralelas, concurrentes y distribuidas de los algoritmos de simulación.

La presente tesis doctoral utiliza una implementación del modelo CPM-GGH para definir un modelo básico de epidermis que simula con éxito el proceso de homeostasis y regeneración de pérdida de capas celulares en rasguños.

Este modelo permitirá, en líneas futuras abordar la simulación de fenómenos más complejos.

También se han abordado dos posibles aproximaciones a la ejecución mediante hardware paralelo de versiones de los algoritmos que simulan los modelos básicos que subyacen al CPM-GGH. Estas aproximaciones permitirán en el futuro proporcionar versiones paralelas y más eficientes que permitan abordar la simulación de fragmentos grandes de epidermis.

Abstract

Human skin is an example of a complex system. It is made of several copies of different cell types. The behavior of the complete system emerges from the individual behaviors of its cells. Many models have been developed that describe the behavior of individual cells. The knowledge about the contribution of each cell to the behavior of the complete tissue is usually quite complicated and only known by the experts on the field. The behavior of individual cells is, however, quite well known.

The simulation of skin tissue is interesting for many reasons. The system to be simulated is very appropriate for the use of bioinspired computational models such as cellular automata, which can be considered ideal for the simulation of the epidermis because they can assure at every instant the maintenance of a minimum energy situation. The most promising family of mathematical models in tissue simulation are those based on the Ising model, the Potts model and the CPM-GGH model.

Performance is one of the main limitations posed by this family of models. A possible way to solve it is the access to massively parallel resources by means of parallel, concurrent and distributed versions of the simulation algorithms.

This doctoral thesis implements the CPM-GGH model and successfully represents the processes of homeostasis and regeneration after the loss of cell

layers in small wounds and scratches. This model will make it possible to tackle the simulation of more complex phenomena in the future.

Two different ways to implement it by means of parallel hardware have been envisaged, which in the future should make it possible to address the simulation of bigger fragments of epidermis.

A mis padres Gloria y Alfonso y mis hermanos, Claudia y Alejandro y para ti
Fernando, que muchas montañas aun nos aguardan por escalar.

Índice General

Capitulo 1 Introducción	1
1.1 Motivación.....	5
1.2 Organización del Documento	7
Capitulo 2 Modelos de simulación	8
2.1 Modelo y Simulación Asistida por Ordenador.....	8
2.2 Modelos de Simulación Celular Basados en Rejillas	12
2.2.1 Autómatas Celulares	13
2.3 Modelos Celulares Basados en Mecánica Estadística	18
2.3.1 Resumen de Mecánica Estadística	18
2.3.2 Modelo de Ising	26
2.3.3 Modelo de Potts.....	34
2.3.4 Modelo CPM-GGH para simulación celular	36
2.3.5 Selección del Modelo CPM-GGH como Entorno para Las Simulaciones.....	42
Capitulo 3 Mecanismos para la Aceleración de los Modelos Basados en CPM- GGH	45
3.1 Computación Paralela	47
3.1.1 Clústeres de Ordenadores.....	49
3.1.2 Programación Paralela	52
3.1.3 Java en Entornos Paralelos	57
3.1.4 Sistemas paralelos Auto Ajustables	60

3.1.5 Balanceo de Carga y Distribución de Cargas de Trabajo	61
3.2 Modelos para la Aceleración de la Dinámica de Metrópolis	63
3.2.1 Modelos Q2R, Swendsen–Wang y Wolff.....	63
3.2.2 Descomposición por Dominios.....	66
3.2.3 Algoritmo del Tablero de Ajedrez.....	67
3.2.4 Algoritmos “Rejection-Free”	68
3.2.5 Algoritmo de “N-FOLD WAY”	69
Capitulo 4 La Piel	73
4.1 La Célula	74
4.1.1 Membrana Celular	75
4.1.2 Orgánulos Intracelulares	76
4.1.3 Citoesqueleto.....	77
4.1.4 Adhesión Celular	78
4.1.5 Reordenamiento Celular “Cell Sorting”	79
4.1.6 Ciclo Celular	80
4.2 Estructura de la Piel	82
4.2.1 La Epidermis.....	83
4.2.2 Dermis.....	87
4.2.3 Hipodermis.....	89
4.3 Homeostasis de la Epidermis	90
4.3.1 Unidades De Proliferación Epidérmica	91
4.3.2 Moléculas Celulares Epidérmicas	93
4.4 Heridas en la Epidermis	95
Capitulo 5 Aceleración del Modelo de Ising y del Modelo de Potts a Través de Implementaciones Paralelas	98
5.1 Una Implementación Concurrente Usando Java E Hilos Para El Algoritmo De Metrópolis - Monte Carlo En El Modelo De Ising.....	99
5.1.1 Autómatas Celulares, Modelo de Ising y sus Variantes	102

5.1.2 Modelo de Ising	103
5.1.3 Implementación Paralela del Modelo de Ising	106
5.1.4 Métodos e Implementación	108
5.1.5 Resultados	111
5.2 Paralelización del Método de Metrópolis – Monte Carlo Para el Modelo De Potts Usando una Partición Dinámica de la Rejilla.....	114
5.2.1 Definiciones y Antecedentes.....	117
5.2.2 Modelo Matemático	119
5.2.3 Modelo de Potts.....	123
5.2.4 Modelo de División Dinámica de Cargas de Trabajo de la Rejilla en un Hnoc.....	125
5.2.5 Implementación del Algoritmo y Balanceo de Carga en un Clúster Heterogéneo	125
5.2.6 Resultados.....	129
Capitulo 6 Simulación de la epidermis Humana Usando GGH-CPM.....	132
6.1 Heridas en la Epidermis	135
6.2 Descripción Informal de un Modelo para el Cierre de Heridas Epidérmicas	137
6.3 Materiales y Métodos	140
6.3.1 COMPUCCELL 3D	140
6.3.2 Modelo CPM-GGH Para Simulación Epidérmica.....	142
6.4 Resultados	148
6.5 Discusión	151
6.6 Observaciones Finales.....	153
Capitulo 7 Conclusiones y trabajo futuro.....	154
7.1 Conclusiones	154
7.2 Trabajo Futuro.....	158
Referencias.....	161

Índice de Figuras

Figura 2.1 Proceso de simulación y modelado	11
Figura 2.2: Vecindad de Von Neumann y Vecindad de Moore en $d=2$	16
Figura 2.3. Esquema de evolución de un LGCA.....	18
Figura 2.4. Esquema de un conjunto de partículas en un sistema	21
Figura 2.5. Distribución espacial de los espines en el modelo de Ising bidimensional	28
Figura 2.6. Simulaciones del modelo de Ising a diferentes temperaturas. ...	30
Figura 2.7. Diferentes configuraciones en un modelo de Ising con 4 partículas	31
Figura 2.8. Rejilla bidimensional para el modelo de Potts.	34
Figura 2.9. Representación de la rejilla y células en el modelo CPM-GGH....	38
Figura 2.10 Simulación de patrones de formación de células usando diferentes valores de J	41
Figura 2.11 Representación esquemática de diferentes modelos de simulación celular – adaptado de [44]	43
Figura 3.1. Diferentes modelos de procesamiento	49
Figura 3.2. Esquema de un clúster	51
Figura 3.3. Enlaces o <i>Bonds</i> en los modelos de actualización tipo clúster para el modelo de Ising o Potts.	65
Figura 3.4. Rejilla en forma de tablero de ajedrez para algoritmo de Ising/Potts	68
Figura 4.1. Célula Animal	75
Figura 4.2. Ciclo Celular.....	80
Figura 4.3. Estructura de la piel	83

Figura 5.1. Configuraciones a diferentes temperaturas usando el modelo de Ising	113
Figura 5.2. Ejemplo de un clúster heterogéneo con 4 nodos. Cada servidor tiene su propia cola para recibir las solicitudes entrantes de los restos de los nodos de la granja.....	119
Figura 5.3. Ejemplo para el sistema <i>Gs</i> Los elementos del grafo son: Nodos: $P = \{A,B,C,D\}$, enlaces $L = \{ca, ab, ad, bd, dc\}$, pesos de procesamiento = $\{2, 5, 4, 1\}$ latencia en la comunicación $\delta = \{1, 0, 7, 1, 2\}$ y vecindario (A) = $\{B,D\}$, $path(C,D) = \{C, A,B,D\} \cup \{C, A,D\}$ y $pathmin(C,D) = \{C,A,B,D\}$	121
Figura 5.4. Grafos <i>Gma</i> obtenidos para cada nodo para el grafo de la Figura 5.3.	122
Figura 5.5. HNoC formado por 5 nodos, 4 maestro/esclavo y 1 nodo <i>supermaster</i> . El supernodo contiene el grafo completo con toda la comunicación posible entre cada nodo y el valor estadístico de las esperas y la carga efectiva de cada procesador.....	127
Figura 5.6. A partir de una distribución de datos arbitraria de una cuadrícula de 3×3 (cuadrícula con 3 procesadores).....	128
Figura 5.7. Simulación de burbujas usando el modelo de Potts.....	129
Figura 6.1. Ilustración 1. Mitosis Asimétrica / Simétrica.....	138
Figura 6.2. Condiciones iniciales de la simulación epidérmica	144
Figura 6.3. La migración celular en la curación de heridas	147
Figura 6.4. Instantáneas de una simulación en diferentes etapas de Monte Carlo.....	149
Figura 6.5. Una comparación entre los datos reales (datos <i>in vivo</i>) y los resultados de diferentes ejecuciones de nuestra simulación. Las cuatro fases sucesivas en el proceso real son correctamente capturadas por nuestro modelo.....	151

Índice de Tablas

Tabla 3.1 Diferentes Clases Modelo N-Fold Way	70
Tabla 5.1. El tiempo de ejecución, en segundos, de la versión de tres del mismo algoritmo, 5000MCSs. El algoritmo de agrupación utilizado en esta ejecución fue la DFS. Temperatura = 2. La mejora alcanzada con este ejemplo era casi el 112%.....	130
Tabla 5.2. El tiempo de ejecución, en segundos, de la versión de tres del mismo algoritmo, 5000MCSs. El algoritmo de agrupación utilizado en esta ejecución fue la DFS. Temperatura= 15	131
Tabla 6.1. Valores de unión para J	145

No intentes averiguar a dónde te lleva cada uno de tus pasos.

Sólo quien ve a distancia acierta el camino.

Dag Hammarskjöld

Agradecimientos

Llegar a este momento, a la culminación de esta investigación no ha sido del todo sencillo. Desde que decidí embarcarme en esta aventura y cambiar de residencia de Colombia a España para cumplir mis sueños, ha habido varios momentos difíciles en los cuales muchos de ellos hicieron mermar mis ganas de continuar, pero que sin embargo, han sido compensados con creces cuando observas que las cosas a la cual te has dedicado varios años empiezan a dar fruto, dándote como única recompensa, el estar satisfecho contigo mismo. Muchos de estos sucesos han determinado mi personalidad y marcado el camino que quiero recorrer a partir de hoy.

Como reza el refrán castellano, “es de bien nacido ser agradecido” he intentado no olvidar a nadie quien de alguna u otra forma han colaborado para que esta tesis pudiera llegar a ser realizada y culminada. Espero no tener tan mala memoria y disculpadme si a alguien no menciono explícitamente.

En primer lugar, quiero agradecer a mi familia, mis padres Alfonso y Gloria, mis hermanos Claudia, Alejandro y Fernando, que a pesar de la distancia siempre me han arropado con su amor y apoyo, sin ellos terminar este trabajo habría sido imposible. También a mi familia extensa, mis tíos y primos.

Ofrezco mi más sincero agradecimiento a mis tutores a Manuel Alfonseca y Alfonso Ortega de la Puente que me han apoyado durante toda mi tesis con su paciencia y conocimiento. Haber trabajado con ellos ha sido un placer y privilegio.

Me gustaría agradecer a Maciej Swat y James Glazier en la universidad de Indiana Bloomington – USA, por el tiempo compartido allí durante mis es-

tancias y a los innumerables correos intercambiados con mis dudas en el campo de la modelización usando el modelo CPM-GGH. También quiero agradecer a Alberto Gandarillas investigador de la Fundación Marqués de Valdecilla y a Pedro Tarazona del Departamento de Física Teórica de la Materia Condensada de la Universidad Autónoma de Madrid por compartir sus experiencias y sus conocimientos invaluable.

A Carmen Navarrete, con quien pude hacer muchos de los experimentos de esta tesis y compartir muchas horas en discusiones fructíferas a cerca de la utilización de la paralelización para la aceleración del modelo CPM-GGH.

Muchas gracias a todos los miembros del departamento que han hecho de éste, un lugar especial durante todos estos años. Deseo expresar mi gratitud a Juana Calle quien en todo momento estuvo pendiente de mí como si fuese su hijo adoptivo. A la antigua gente que perteneció al laboratorio B-207, Pablo, Pedro, Manuel Freire, Leila y Abraham. A las personas del Grupo GHIA Miguel, Álvaro y Marina. A todos y todas en el servicio de Biblioteca de la escuela politécnica.

A todos los amigos que he cosechado en España, a los cuales cada vez que les contaba mi tesis me dieron ánimo: a Julián e Irene y su hermosa hija. A Jorge Enrique, a Gustavo, a Juan Gabriel, a mi gran amigo Rodrigo y a su esposa Sandra junto con esa niña tan hermosa, Violeta. A Ricardo Lozano y Martha. A Laura, por ayudarme con las ilustraciones incluidas en esta tesis, ¡la verdad no sé cómo lo haces! A Erika María y la familia Rodríguez Pinzón. A Rocío y mis compañeros de trabajo en Afina. A mi amigo Andrés H. A los hermanos Miguel y Javier Blanquer con quien he aprendido a subir montañas y a conocer los mejores paisajes que tiene esta tierra.

A Carlos Martin, con el cual hemos vivido una amistad inquebrantable desde que nos conocimos en Chile allí por el año 1999 y quien fue la persona que me tendió su mano para poder venir a España.

A Myriam, mi gran amiga y hermana, solo tú y yo sabemos que es estar tan lejos y emprender una aventura como esta.

A mi familia adoptiva en España, Efraín e Isabel, y sus hijos Bibiana, Lorena y Efraín.

A Daniel, Pablo, Christian y José, los últimos en unirse en esta aventura.

Y en especial a ti, Soraya, por tu paciencia, ánimos y mimos. Lo hemos conseguido.

Capítulo 1 Introducción

Las herramientas informáticas son cada vez más imprescindibles en otras disciplinas científicas. En particular, los simuladores posibilitan enormemente la investigación, sobre todo en aquellos entornos en los que resultaría demasiado costoso, peligroso o incluso poco ético.

La biología celular proporciona una gran cantidad de datos sobre los mecanismos biológicos que determinan los comportamientos globales de un tejido. Para poder comprender estos mecanismos de forma predictiva se requiere la integración de la información que suministran tales modelos y la complementación, usando modelos computacionales para su contrastación. Esta integración puede darse a diferentes niveles, usando escalas tanto microscópicas como macroscópicas.

La migración y movimiento celular juegan un papel fundamental en una variedad de procesos. Durante el proceso de embriogénesis, la organización de los diferentes tipos de células determinará el adecuado desarrollo de los diferentes órganos y tejidos del organismo. En los organismos adultos, el movimiento celular no se presenta de igual manera con la misma escala que en los organismos en desarrollo. Sin embargo, se pueden observar procesos concretos tales como el inmunológico (donde los leucocitos migran activamente de los vasos sanguíneos a los tejidos infectados) [1, 2] y en la cicatrización de heridas (donde la migración contribuye a la reparación de la membrana basal y los tejidos conectivos). En condiciones anormales, la migración celular está relacionada con enfermedades inflamatorias crónicas tales como arteriosclerosis o la invasión de células cancerosas.

En el campo de la modelización y de la simulación, ha surgido en los últimos años el concepto de Individual Cell Model o Single Cell-Based Model [3] en el cual, usando técnicas concretas, es posible predecir el comportamiento colectivo de un grupo de células a través de las interacciones individuales de cada una de ellas. Los modelos basados en la célula o Cell-Based Model son modelos teóricos de un sistema biológico en los que las células se representan explícitamente como unidades únicas.

Se suelen utilizar para modelar sistemas en los que se requiere recoger el comportamiento complejo y la geometría de las células que determinan el comportamiento global del sistema.

En biología, el término modelo puede diferenciarse del concepto que se usa en las matemáticas. En este campo, el término hace referencia a la representación abstracta de los fenómenos biológicos. En tal sentido, un modelo biológico es una descripción cualitativa de un conjunto de hipótesis sobre los mecanismos y relaciones que tratan de explicar un fenómeno. Un modelo matemático es una formalización cuantificable de un modelo biológico, en el que las reglas que regulan el comportamiento biológico se declaran explícitamente de una forma cuantitativa. Junto a estos, un modelo computacional es una implementación de un modelo matemático en forma de algoritmos que utilizan tales métodos junto con un conjunto de parámetros.

La construcción de modelos multi-escala (que incluyen características microscópicas y macroscópicas) es compleja. En primer lugar se deben combinar los diferentes contextos biológicos de manera coherente y verosímil, es decir, debe existir una conexión biológica entre los mecanismos que explican los diferentes procesos y se deben colocar en un contexto matemático. Implementar el modelo computacional requiere una atención similar usando la misma coherencia empleada entre el modelo y el fenómeno biológico modelado.

Con estas herramientas los científicos pueden plasmar en un modelo en funcionamiento su conocimiento actual sobre un sistema, contrastar posibles hipótesis en el entorno virtual y planificar mejor los experimentos que realmente llevarán a cabo, descartando aquellas hipótesis que resulten poco prometedoras.

En los últimos años, el aumento de la potencia de los ordenadores ha permitido la simulación eficiente de modelos computacionales alternativos al de von Neumann, que fueron descartados anteriormente por la ausencia de arquitecturas físicas que los implementaran y por su ineficiente simulación en los anteriores computadores convencionales.

Por otro lado, la proximidad al techo de la potencia de la tecnología basada en la miniaturización de los circuitos integrados impulsa el estudio de arquitecturas de computadores alternativas. Debido a éstas, entre otras razones, asistimos en la actualidad al resurgimiento del interés por nuevos modelos computacionales que, a menudo, nacen de la observación e imitación de sistemas complejos reales cuya potencia computacional se suele garantizar porque en un entorno natural solucionan eficientemente problemas de gran complejidad.

Tal es el caso de paradigmas como los autómatas celulares, la computación inspirada en el ADN, en las membranas o en otros fenómenos bioquímicos. Todos estos sistemas son tan potentes como los ordenadores actuales (computacionalmente completos). Sin embargo, la implementación de este tipo de simulaciones tiene un inconveniente en el límite de velocidad de procesamiento que poseen los ordenadores, el cual determina el tiempo que tarda en alcanzar la solución de un problema computable.

Una de las alternativas existentes para mejorar esta situación es el uso de clúster de ordenadores, es decir, la utilización de múltiples ordenadores que realizan una tarea como si fuesen un solo gran ordenador. Un cluster se puede ver como una clase de ordenador extendido, construido con base en un conjunto de componentes hardware estándar interconectado mediante una red privada de gran velocidad que permite la distribución de tareas mediante algún tipo de estrategia para ser ejecutada en paralelo. Generalmente un clúster se compone de un grupo de dos o más estaciones de trabajo y es especialmente útil cuando se dedica a ejecutar tareas que precisan alta capacidad de cálculo. Existen sistemas operativos de libre distribución, como por ejemplo Linux o FreeBSD, que permiten la administración y gestión de los clústeres.

Desde la aparición de los ordenadores, la necesidad de potencia de cálculo que requieren las aplicaciones ha ido en continuo crecimiento, presentándose en los últimos veinte años un aumento significativo en las prestaciones de las CPU. Este fenómeno ha planteado enormes desafíos tanto en el diseño de hardware como en el de software, centrándose en el avance en el desarrollo de arquitecturas, algoritmos y compiladores. Sin embargo, el poder potencial de las plataformas actuales no puede ser explotado plenamente sin dificultad.

Una de las razones principales que limitan la eficacia de una aplicación es el límite que impone la memoria y las comunicaciones. Mientras en los últimos años la velocidad de las CPU ha crecido de manera exponencial, los soportes de memoria y la transmisión de datos a través de un medio han causado menor impresión.

Este fenómeno ha incrementado la brecha en el rendimiento, siendo la principal causa de la ineficiencia de las aplicaciones secuenciales. Varios enfoques arquitectónicos se han adoptado para mitigar el impacto de esta brecha.

Una solución natural para aumentar el número de operaciones ejecutadas por unidad de tiempo es el uso de arquitecturas paralelas. El paralelismo se ha estudiado durante décadas y se han explorado diferentes soluciones [4]. Existen muchos compiladores y lenguajes que han ayudado a los programadores a extraer el paralelismo de forma automática a partir de sus aplicaciones, pero no siempre proporcionan soluciones óptimas. De hecho, el mejor enfoque es exponer de forma explícita el paralelismo a nivel de diseño de algoritmos.

Un obstáculo importante para el logro de un alto rendimiento en arquitecturas paralelas está representado por el coste de la comunicación, dado que la relevancia de este factor aumenta con el tamaño del problema a resolver. Por consiguiente, la reducción de las necesidades de comunicación en los algoritmos es de suma importancia. El coste de comunicación depende de la distancia entre los procesadores implicados, teniendo en cuenta el ancho de banda y la latencia.

Por lo tanto, es evidente que el paralelismo, por una parte, y la gestión de la memoria y de la interconexión en la otra juegan un papel clave en el rendimiento de las aplicaciones.

La tesis se enmarca en el campo de la biología computacional, que es un campo interdisciplinar que aplica las técnicas de la ciencia de la computación, matemáticas y estadística, intentando interpretar problemas biológicos y abarcando los campos de la bioinformática, el biomodelado computacional y la biología de sistemas entre otras [5, 6].

1.1 Motivación

El tejido epitelial es un componente esencial en los animales pues determina su arquitectura corporal. Se encuentra revistiendo todas las cavidades internas del cuerpo y la piel.

Las capas epiteliales también tienen un papel importante en la secreción, absorción selectiva y transporte intracelular. Los carcinomas representan alrededor del 90% de los cánceres humanos, originándose en los tejidos epiteliales [7].

Los epitelios se renuevan constantemente durante toda la vida del organismo, y con ello la existencia de mecanismos que aseguren la conservación y uniformidad del tejido, mostrando un proceso de muerte celular o de división para formar nuevas células. Este ciclo determina los procesos homeostáticos o de equilibrio del tejido manteniendo su robustez estructural y funcional de los tejidos vivos y de los órganos. Cuando se presenta una alteración de la homeostasis, la preservación de tejido se encuentra en riesgo; por ejemplo, cuando se desarrollan tumores, la homeostasis epitelial está dañada y alterada.

La epidermis humana se puede considerar un ejemplo de sistema biológico complejo. A partir de ello, esta investigación busca expresar las interacciones que existen en las células de la epidermis para formar un tejido, usando para ello un tipo de especial de autómatas celulares: el Cellular Potts Model [8, 9]. Uno de los intereses de esta investigación es comprender cómo surge una estructura provista por las células en el desarrollo y conservación del tejido

epidérmico. ¿Es la organización igual cuando el tejido esta en equilibrio o cuando debe ser remodelado? Por ejemplo, cuando dos porciones del tejido se encuentran separadas, se chocan y se funden en el proceso de cierre de heridas.

El interés principal es integrar elementos de la teoría computacional junto con las observaciones biológicas para recoger las conductas de las células del tejido epidérmico para desarrollar un modelo computacional donde, a partir de fáciles relaciones expresadas en reglas que se vinculan en el Hamiltoniano, surjan comportamientos complejos que permitan generar hipótesis sobre el proceso de equilibrio entre el nacimiento, diferenciación y descamación que se presenta en la epidermis.

Asimismo, el grupo de investigación en el cual se desarrolló esta tesis ha estado interesado en proveer modelos y herramientas para el diseño y programación de estas nuevas arquitecturas con el objetivo de solucionar problemas concretos y proporcionar nuevas herramientas informáticas útiles en otras áreas de investigación. Por esto, durante el proceso de la investigación se entró en contacto con el "Departamento de biología molecular y terapia génica" del CIEMAT, dirigido por el Dr. D. José Luis Jorcano y con D. Antonio Bernar, del CSIC, ambos de reconocido prestigio, en el estudio de la piel humana y de sus patologías; por último con Alberto Gandarillas perteneciente a la Fundación Marqués de Valdecilla, quien ha validado el modelo y ha sido coautor de una de las publicaciones de la tesis.

El objetivo de esta tesis se resume en los siguientes puntos:

- Mostrar la expresividad de los modelos discretos, tales como los autómatas celulares y el modelo CPM-GGH, de acuerdo a lo ya expuesto en los párrafos anteriores.
- Instalación y configuración de un clúster de ordenadores que permita realizar las simulaciones a lo largo de la investigación.
- Proponer mecanismos de aceleración de los modelos empleados, usando una aproximación que modifique la manera de evaluar la rejilla o acudiendo a una descomposición espacial sobre un entorno multi-máquina.

- Desarrollar de un modelo macroscópico de la epidermis donde se incluyan el máximo número de elementos biológicos tomados de la literatura, que permita expresar el comportamiento macroscópico de la epidermis
- Ofrecer una herramienta de simulación que permita a experimentadores contrastar de manera más sencilla la hipótesis sobre sus modelos y que ofrezca tiempos razonables de simulación.

1.2 Organización del Documento

La tesis se divide en siete capítulos, distribuidos en dos partes.

El primer capítulo es esta introducción.

La primera parte, que comprende los capítulos 2, 3 y 4, describe el estado del arte en los campos necesarios para el mejor entendimiento de la tesis: una revisión de los modelos de simulación basados en la informática teórica y la mecánica estadística (capítulo 2); una introducción a las técnicas de paralelización (capítulo 3); y un resumen de la biología de la epidermis (capítulo 4).

La segunda parte del documento comprende los capítulos 5 y 6 y desarrolla los dos objetivos de la tesis: la paralelización del modelo CPM-GGH (aceleración del modelo de Ising y del modelo de Potts a través de implementaciones paralelas) usando dos aproximaciones diferentes (capítulo 5); y la simulación de la homeostasis y del cierre de heridas en la epidermis humana, mediante un modelo CPM-GGH que se ejecuta sobre el sistema CompuCell 3D (capítulo 6).

Finamente se especifican las conclusiones de este estudio y algunos trabajos futuros que pueden derivarse de él.

*-Minino de Cheshire, ¿podrías decirme, por favor, qué camino debo seguir para salir de aquí?
-Esto depende en gran parte del sitio al que quieras llegar - dijo el Gato.*

Lewis Carroll, Alicia en el País de las Maravillas



Capitulo 2 Modelos de simulación

En este capítulo se introducen los conceptos básicos de simulación y modelado, desde la óptica de la informática teórica y la mecánica estadística. Se describen brevemente las características más relevantes de los modelos que ofrecen estas dos áreas de conocimiento y que han sido revisados en el proceso de confección de esta investigación, detallando con más atención aquellos elementos que son fundamentales para el entendimiento de los siguientes capítulos en esta tesis.

2.1 Modelo y Simulación Asistida por Ordenador

Los conceptos de modelado y simulación están estrechamente relacionados uno con el otro, encontrándose diferentes definiciones de acuerdo con el área de su aplicación [10].

Un modelo es una representación de un evento o de un objeto real o artificial en la que, usando diferentes metodologías, se realiza una abstracción o descripción formal. En otros términos la modelización permite a un observador comprender desde una visión controlada el comportamiento del todo y sus partes. Una mejor definición para entender el concepto de modelo la sugiere Minsky M. Matter [11]: "para un observador B, un objeto A* es un modelo de un objeto A, si el observador B puede usar A* para responder preguntas que le interesan acerca de A".

La simulación es la aplicación de un modelo con el objetivo de encontrar estrategias que ayuden a resolver un problema o responder preguntas acerca de un sistema [12]. Una definición acertada sobre simulación es hecha por Cellier: "Simulación es un experimento realizado sobre un modelo" [13].

Normalmente no se incluye en la simulación todo el sistema, sino aquellos fenómenos o eventos que pueden ser relevantes para el analista. El proceso de simplificación o abstracción del modelo permite [14]:

- **El tratamiento del sistema abstraído.** El modelo es, por lo general, computacionalmente menos complejo que el modelo base. Este hecho es aplicable mientras exista un balance entre la complejidad y la relevancia de los elementos que se incorporan en el modelo resultante.
- **Un mejor entendimiento del objeto de estudio.** Una abstracción permite una reducción de los componentes y procesos, facilitando su mejor manipulación y control.
- **El desarrollo de bibliotecas de modelos que representan el mismo proceso.** Este repositorio representa un camino secuencial en el proceso de modelado: una aproximación jerárquica tiende a reducir la complejidad y facilita el uso de distintos niveles de abstracción. La modularidad ayuda a reducir la complejidad y facilita las modificaciones.

La importancia de la modelización y la simulación no tiene cabida sin la ayuda de los medios y métodos computacionales, cada vez más potentes. Es entonces cuando la simulación asistida por ordenador puede ser de utilidad, en contextos donde el proceso de experimentación implica altos costes y esperas de tiempo para la observación de cambios o resultados, la experimentación es imposible de repetir en un medio real, o el realizarlo podría llegar a ir en contra de la seguridad o legalidad.

La creación de un modelo y posterior uso dentro de una simulación consta de un conjunto de entidades jerarquizadas [15] que se modelan por separado.

- **El sistema real.** Es el sujeto de quien se habla y se desea construir un modelo que permite responder preguntas sobre él.
- **El marco experimental.** Es el mecanismo que el investigador posee para observar el sistema real como si fuese un *prisma* ayudándole a interpretar sus observaciones y su abstracción de la realidad.
- **El modelo base.** Es un modelo general con el que es posible razonar acerca del comportamiento del sistema real. Al igual que en el caso del sistema real, el modelo base podría ser tan complejo y propiciaría la dificultad de su entendimiento y manipulación y con ello se hace necesario recurrir a una abstracción.
- **Modelo simplificado.** Es aquel que ha sido obtenido a partir del modelo base y que sólo es válido dentro un marco experimental.
- **Modelo computacional.** Es la representación simbólica en código fuente de un modelo simplificado, es decir, la implementación del modelo en una máquina que será capaz de procesarlo.

A continuación se muestra en la Figura 2.1 un esquema de las entidades en el proceso de simulación y modelado:

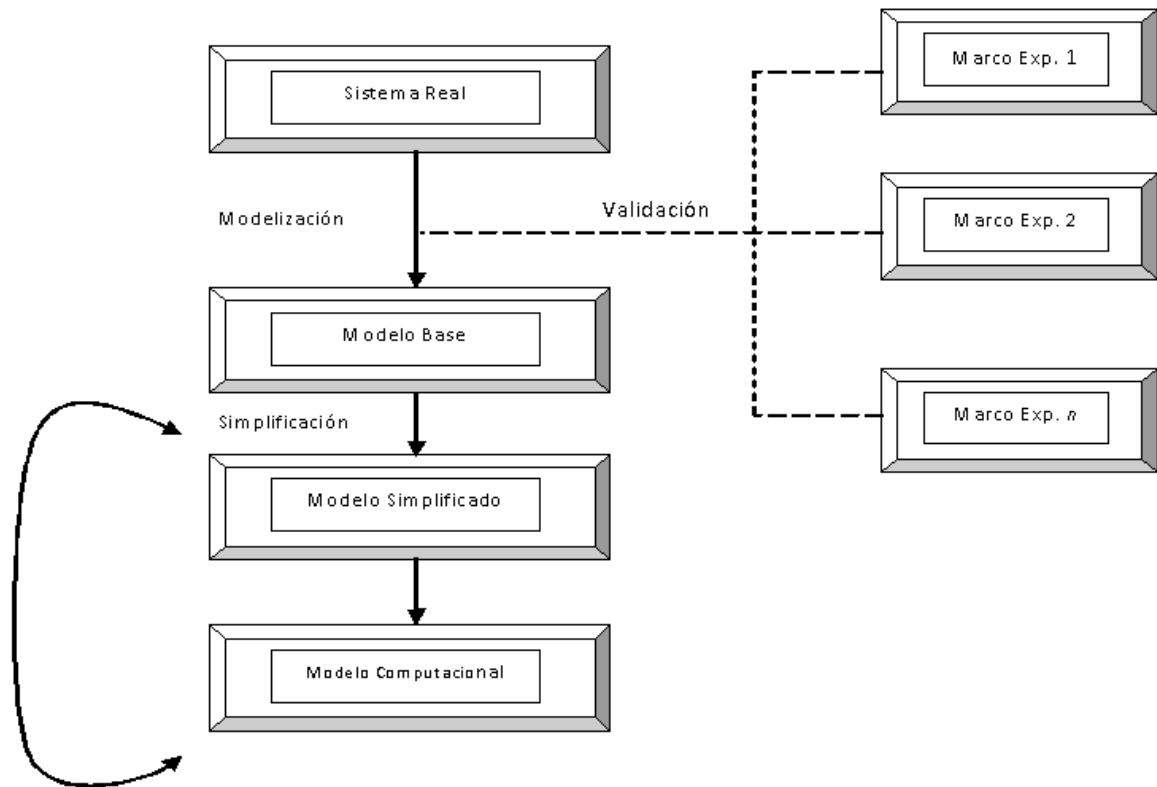


Figura 2.1 Proceso de simulación y modelado

El proceso de modelización es un proceso iterativo. Por un lado, la realidad física proporciona una información experimental y por otro lado existe el proceso gradual de abstracción que, mediante la elaboración de hipótesis o supuestos, llega a la construcción de un modelo reflejo de esa realidad física.

La simulación es ajustable a diferentes áreas de conocimiento, donde se pueden encontrar ejemplos de su aplicación: el pronóstico del estado del tiempo, el comportamiento del precio de una acción o el desarrollo de la reacción de un fármaco en una población, entre otros.

Existen dos enfoques claramente diferenciados de simulación [16]:

- Por un lado, los métodos clásicos donde se agrupan aquellos basados en ecuaciones diferenciales y métodos de Monte Carlo.

- Por otro, los inspirados en computación natural, los cuales intentan explicar los fenómenos a través de la computación inspirada en la naturaleza en términos de procesamiento de información.

Se encuentran tres áreas bien diferenciadas [17]:

- Computación inspirada en la naturaleza.
- Modelado, simulación y emulación de sistemas naturales a través de medios computacionales.
- Computación con materiales naturales.

La variable tiempo t o representación temporal determina cómo se dará la transición de estados del modelo entre un momento y el siguiente. Éstos pueden ser continuos o discretos y la elección de uno u otro dependerá del propio proceso de modelado del sistema.

2.2 Modelos de Simulación Celular Basados en Rejillas

Existen diferentes aproximaciones aplicadas a la modelización de fenómenos celulares donde, independientemente de cuál sea el enfoque, la finalidad es que a través de ellos se pueda entender el porqué de ciertos procesos, como la migración celular, el desarrollo y crecimiento de los tejidos, la proliferación anormal de un conjunto de células y la embriogénesis, entre otros.

De acuerdo con la literatura, los modelos desarrollados pueden estar clasificados en dos grandes categorías: un enfoque continuo, donde las células y variables del modelo consideradas se plasman como puntos en un espacio cartesiano y el tiempo cambia continuamente; y otro enfoque discreto en el que las células se representan como unidades en un espacio virtual o rejilla donde las interacciones se suceden en un tiempo a modo de saltos.

Cada uno de ellos tiene sus ventajas y desventajas en función del fenómeno indagado y de la escala de longitud sobre la que se quiere investigar el fenómeno.

Los modelos continuos emplean básicamente las ecuaciones diferenciales parciales (**PDE**) para describir la variación temporal y espacial de las células bajo la influencia, por ejemplo, de la concentración de una sustancia específica o un campo de fuerza que ejercen algún tipo de efecto sobre la población de células. Las células se modelan como una población global, donde se recoge los comportamientos de cambio, diferenciación o movimiento celular, en respuesta a los elementos que afectan a dicha población. Los modelos continuos son adecuados para representar el crecimiento a escala del tejido, donde las propiedades locales de las células individuales se pueden promediar y suponer cambiantes suave y gradualmente en el tiempo.

Por otro lado, los modelos basados en la célula (*cell-based-models*), predicen el comportamiento colectivo de las células a partir de sus comportamientos individuales definidos a través de las reglas que gobiernan las relaciones de unas con otras. Los resultados de una aproximación de este estilo permiten observar los patrones y conductas que siguen las células y sus relaciones a través del tiempo. En estos modelos las propiedades de las células se definen a nivel local e independiente, dejando observar el destino de cada célula dentro de la simulación. Frente a los modelos continuos, los modelos basados en células individuales permiten investigar los comportamientos resultantes de pequeñas poblaciones de células.

A continuación se revisarán brevemente varios modelos que se enmarcan en los grupos descritos previamente y se justificará el porqué de la selección de un modelo en particular, que será usado en posteriores capítulos.

2.2.1 Autómatas Celulares

La historia de los autómatas celulares (**CA**) gira en tres hechos o períodos de mayor interés. El primero se da a finales de los años cuarenta cuando John von Neumann [18], extiende la teoría de la lógica de autómatas a una teoría probabilística basada en el trabajo de Claude Shannon sobre la teoría de la comunicación. El segundo tiene lugar cuando Martin Gardner [19] populariza *el juego de la Vida* de John Conway. El tercero se produce cuando Stephen

Wolfram [20] realiza su clasificación de los autómatas en cuatro categorías generales.

Los CA son una abstracción matemática de sistemas físicos en los que el espacio, el tiempo y las variables que describen el estado del sistema son discretos.

Un CA d -dimensional consta de una rejilla uniforme regular, habitualmente de extensión infinita, poseyendo en cada una de sus celdas un autómata finito, es decir, una variable descrita que cambia de estado en función de la entrada que recibe del exterior. El estado del autómata celular queda totalmente definido mediante el valor de los autómatas finitos de cada celda. El CA evoluciona a intervalos discretos de tiempo, donde el valor de cada autómata finito en un instante determinado depende de los valores de sus vecinos en el instante anterior. El vecindario de una posición está habitualmente formado por el autómata de la posición considerada y los autómatas adyacentes. El estado de todas las posiciones es actualizado simultáneamente de forma que el comportamiento del autómata celular está definido en función de reglas de transición locales.

Un CA clásico consta de los siguientes elementos:

- Un conjunto finito de símbolos de entrada.
- Un conjunto finito de estados.
- Una rejilla d -dimensional, donde cada célula contiene un autómata finito determinista.
- Una función de transición determinista, que define el próximo estado del autómata.
- Un estado inicial del autómata.
- Un conjunto finito de estados objetivo.
- El estado actual.

La definición formal de un autómata celular se presenta a continuación:

Un CA n -dimensional determinista es la séxtupla

$$(G, G_o, N, Q, f, T) \tag{2-1}$$

Donde,

- G es una rejilla n-dimensional de autómatas.
- Q es el conjunto de estados: un conjunto finito y no vacío de posibles estados de cada autómata de la rejilla.
- G_o es el estado inicial del autómata celular n-dimensional determinista y es la configuración inicial (t=0) de G y Q y $G_o : G \rightarrow Q$, una función inyectiva que asigna un estado inicial a cada autómata en la rejilla.
- N es una vecindad d-dimensional que determina el conjunto de los vecinos de cada autómata de G .
- f es la función de transición, que asigna a cada autómata de la rejilla junto a sus vecinos el nuevo estado del autómata:

$$f : Q \times Q^n \rightarrow Q \tag{2-2}$$

- $T \subseteq Q$ es el conjunto de estados finales.

El conjunto de autómatas finitos en CA se distribuye sobre una estructura de topología regular sin limitación en su dimensión y con la posibilidad de no ser finita. El autómata puede describirse con contorno periódico o cerrado.

En el caso bi-dimensional periódico la rejilla tiene forma toroidal, donde los vecinos del borde superior son las celdas del borde inferior. Lo mismo sucede en relación al borde izquierdo con el derecho. En el caso cerrado el borde tiene valor cero.

Para una rejilla bi-dimensional, las vecindades en un CA comúnmente usadas son la de Von Neumann y la de Moore. La vecindad de von Neumann (N_{VN}) tiene en cuenta los vecinos que satisfacen la condición $\|\vec{y}\|_1 \leq 1$, usando la norma Manhattan:

$$\|(y_1, y_2, \dots, y_d)\|_1 = |y_1| + |y_2| + |y_3| + \dots + |y_d| \quad (2-3)$$

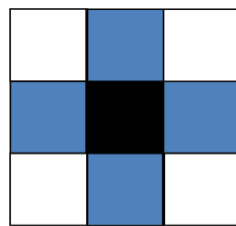
Esto significa que el autómata en la localización \bar{x} tiene $2d + 1$ vecinos. Es decir $N_M = (5, ((0,0), (0,1), (1,0), (0,-1), (-1,0)))$

La vecindad de Moore (N_M) contiene todos los vectores $(y_1, y_2, \dots, y_d) \vec{y} \in Z^d$ tal que $\|\vec{y}\|_\infty \leq 1$, de modo que:

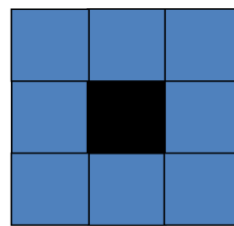
$$\|(y_1, y_2, \dots, y_d)\|_\infty = \max \{ |y_1| + |y_2| + |y_3| + \dots + |y_d| \} \quad (2-4)$$

Es la norma del máximo. Cada autómata tiene 3^d vecinos, es decir $N_M = (9, ((0,0), (-1,1), (0,1), (1,1), (1,0), (1,-1), (0,-1), (-1,-1), (-1,0)))$

La Figura 2.2 muestra las dos vecindades para el caso de $d=2$



Vecindad de Von
Neumann



Vecindad de Moore

Figura 2.2: Vecindad de Von Neumann y Vecindad de Moore en $d=2$.

La función de transición define el próximo estado en cada autómata finito, dada una entrada actual (el conjunto de estados del vecindario) y su propio estado.

$f : Q \times Q^k \rightarrow Q$ es la función de transición que asigna el próximo estado a cada autómata en la red dependiendo de su estado actual y del estado de los k vecinos.

Dependiendo de la función de transición los autómatas celulares se denominan deterministas, no deterministas y autómatas celulares probabilísticos.

2.2.1.1 Modelos LGCA

Los modelos “Autómatas Celulares de Gases de Red” o “Lattice Gas Cellular Automata” (LGCA) son CA con un conjunto particular de estados y dinámica. Son propuestos inicialmente para explicar el comportamiento de los fluidos. Fue formulado a en 1973 por Hardy, de Pazzis y Pomeau [21] (modelo HPP) y extendido en 1986, por Frisch, Hasslacher y Pomeau [22] (modelo FHP), donde se aplicaron diferentes cambios entre ellos la forma de la rejilla (hexagonal). Este modelo presenta un comportamiento macroscópico de acuerdo a las ecuaciones de Navier—Stokes (conjunto de PDE no lineales que describen el movimiento de un fluido).

La idea fundamental de los LGCA consiste en describir el comportamiento de un conjunto de partículas o autómatas que se desplazan sobre una rejilla regular con un conjunto particular de estados y dinámica representada de manera discreta.

Adicionalmente a la descripción formal de un CA, las nuevas características de un LGCA [23] son las siguientes:

- Las partículas sólo pueden estar en los nodos o vértices de la rejilla.
- La evolución del modelo se hace a través de la definición del concepto de velocidades o canales.
- Las reglas que rigen la relación entre las partículas han de cumplir las leyes de la conservación.
- El modelo debe cumplir el principio de exclusión, que impide la presencia simultánea de dos o más partículas en el mismo nodo con la misma velocidad.

Los canales o velocidades en un LGCA especifican la dirección y magnitud del movimiento de una partícula. Las reglas de transición de un LGCA se realizan en dos fases:

Colisión: En esta primera fase las partículas interactúan de acuerdo a unas reglas definidas, donde se actualiza la velocidad o canales de las partículas presentes en el nodo. Las reglas de colisión se desarrollan de tal forma que

cumplan las leyes de conservación, es decir algunas partículas pueden aparecer o desaparecer y deben no violar el principio de exclusión.

Propagación: En esta fase, las partículas se mueven sincrónicamente según su velocidad hasta alcanzar otro nodo en la rejilla. La fase de propagación es determinista y las partículas deben conservar su masa y momento.

En la **Figura 2.3** se muestra un ejemplo sencillo de un LGCA usando una rejilla hexagonal donde se muestran las dos fases de transición:

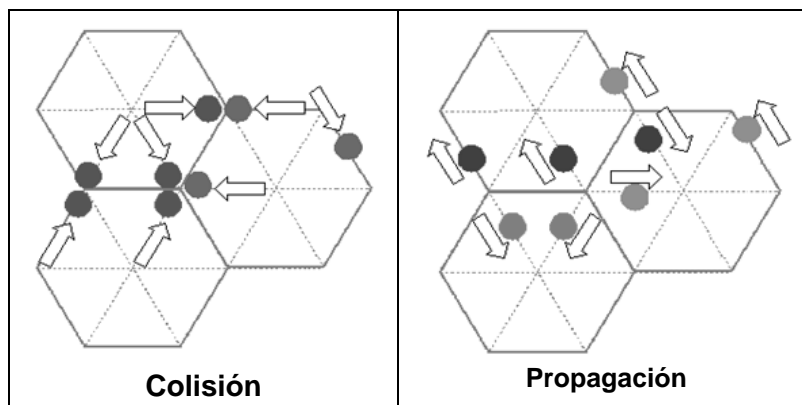


Figura 2.3. Esquema de evolución de un LGCA

2.3 Modelos Celulares Basados en Mecánica Estadística

El modelo que se usa en esta tesis tiene su origen en la mecánica estadística. A priori puede parecer que no existe conexión entre un fenómeno mitótico y la disciplina que intenta explicar la interrelación de millones de partículas atómicas. Por ello pretendo que esta sección sea más extensa y detallada, con el fin de ayudar al lector a entender la conexión entre los modelos de la física teórica con los modelos celulares, desarrollados hasta hace poco más de tres décadas por diferentes autores.

2.3.1 Resumen de Mecánica Estadística

La mecánica estadística es un área de investigación de la física que se apoya en la teoría matemática de probabilidades. Esta última fue fundada como un

intento de explicar la termodinámica de las interacciones entre los átomos y las moléculas. Su origen se basa en los trabajos desarrollados por Ludwig Boltzmann y Josiah Willard Gibbs [24].

La mecánica estadística se ocupa básicamente de los sistemas con N (muchas) componentes. Cada componente tiene una (o varias) variables, que en el modelo pueden tomar valores continuos (ej. las posiciones x , y , z de una partícula) o discretos (ej. ocupación = 0 o 1 de una celda o el $spin = +1$ o -1 en el modelo de Ising).

Su capacidad de predicción se ha explotado a lo largo del siglo pasado en muchos campos de la física. Uno de los logros más notables de la mecánica estadística ha sido la creación de paradigmas para una amplia gama de sistemas, donde a partir de modelos sencillos es posible describir realidades complejas.

Algunos conceptos que ofrece la mecánica estadística y que se usan en la tesis son los de energía (*hamiltoniano*), microestado (o simplemente estado), configuración, ensamble o conjunto, función de partición y distribución de Boltzmann. El lector podrá encontrar una definición más rigurosa en [25].

Cuando se habla de la energía total se hace referencia a la energía cinética y potencial del sistema. Es decir, se tiene en cuenta el movimiento y no movimiento que puede tener una partícula dentro del sistema. Cada partícula aporta su energía al total del sistema. Por conveniencia esta energía se denota con la $H(\sigma_i)$ o *hamiltoniano*, es decir, $H(\sigma_i)$ es la representación de las interacciones entre los componentes de un sistema, o microestado σ_i que determinan cómo el sistema va evolucionando en el tiempo; es decir, cómo $\sigma_i(t)$ para $t=t_0$ determina $\sigma_i(t)$ para cualquier $t>t_0$ y cómo a partir de unas posiciones y velocidades iniciales las partículas van a ir moviéndose. Las leyes de la física son la descripción matemática del modo en el que $H(\sigma_i)$ determina la derivada $\frac{d\sigma_i(t)}{dt}$ del estado del sistema en el tiempo.

¿Cómo se pueden aplicar entonces las leyes físicas? Depende de la complejidad del sistema que se intenta analizar. Se pueden definir cuatro niveles, para los que cada vez se sabe hacer menos predicciones, siendo éstos:

- 1) **Sistemas regulares.** Son sistemas sencillos donde se pueden resolver (analítica o numéricamente) las ecuaciones de movimiento y se puede predecir el estado $\sigma_i(t)$ para cualquier tiempo, a partir del estado inicial $\sigma_i(t_0)$. Dos ejemplos ilustrativos de sistemas regulares son el oscilador armónico o el planeta alrededor del Sol.

- 2) **Sistemas dinámicos caóticos.** Se desconoce cómo resolverlos analíticamente y al resolverlos de forma numérica resulta existir un límite para la predicción práctica. Es decir, sabiendo $\sigma_i(t_0)$ se puede predecir bien (con bastante exactitud) $\sigma_i(t)$ para tiempos próximos, sin embargo, la predicción se hace (exponencialmente) imprecisa conforme crece t , así que hay un *horizonte de predictibilidad*. Ésto ocurre para sistemas relativamente sencillos, pero se hace inevitable en sistemas complicados con muchos componentes. Por ejemplo, desde *el problema de tres cuerpos* (el Sol y dos planetas), que puede tener un horizonte de predictibilidad de millones de años, hasta modelos de predicción meteorológica, con un horizonte de unos pocos días.

- 3) **Sistemas macroscópicos en equilibrio.** Tienen un número tan grande de componentes que el horizonte de predictibilidad es casi nulo. Existe un ejemplo muy descriptivo en las bolas del bombo de la lotería, pues se es incapaz de calcular las posiciones de las bolas unos pocos segundos después de iniciar el movimiento.

Para estos sistemas se utiliza la Mecánica Estadística: se renuncia a conocer $\sigma_i(t)$ (el microestado del sistema) y, en su lugar, basta con saber la probabilidad $P(\sigma_i)$ de encontrar un microestado σ_i concreto del sistema bajo ciertas condiciones. Esta descripción es útil para predecir las *magnitudes macroscópicas*, que suman (o promedian) sobre propiedades de todos los componentes.

- 4) **Sistemas macroscópicos fuera del equilibrio.** Tienen un número muy grande de componentes. Son caóticos, pero no están aislados o interesa su evolución a tiempos suficientemente cortos. Por tanto, el estado inicial $\sigma_i(t_0)$ "no se llega a olvidar del todo" para las magnitudes macroscópicas, que dependerán del tiempo, pero a la vez serán indistinguibles para un enorme número de microestados. Un ejemplo es una corriente de agua, que forma remolinos, con velocidad distinta en distintos sitios y tiempos, aunque la corriente que se mide sea la suma de billones de moléculas.

2.3.1.1 Mecánica Estadística de Equilibrio: Colectivos Estadísticos

En la física, un ensemble o colectividad es la idealización de un sistema de partículas constituido por copias del sistema de moléculas teniendo en cuenta su posición y cantidad de movimiento o *momentum*, que en su totalidad representa todos los posibles microestados alcanzables por el sistema. Para un mejor entendimiento se puede observar en la siguiente imagen (Figura 2.4), donde es posible considerar un conjunto de partículas fijas en un sistema de volumen fijo.

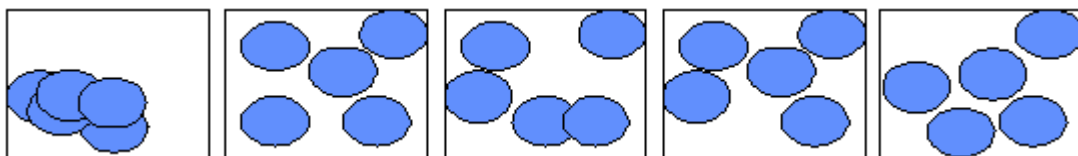


Figura 2.4. Esquema de un conjunto de partículas en un sistema

Se conocen las reglas que permiten calcular $P(\sigma_i)$ en casos de sistemas en equilibrio, bajo distintas condiciones o conjuntos, colectivos o ensambles estadísticos, que se caracterizan por unas cuantas (pocas) variables macroscópicas (A, B, \dots). Cada una de ellas es una función del microestado σ_i :

- a) **Colectivo microcanónico.** Representa un sistema aislado que evoluciona en todo el tiempo que sea necesario para alcanzar el *equilibrio estadístico*. [25]
- b) **Colectivo canónico.** Representa un sistema que no está aislado, pues está en *contacto térmico* con un baño, es decir, puede intercambiar energía con otro sistema mucho mayor que él. En ese caso cualquier microestado σ_i del sistema es posible, porque cualquier valor de la energía del sistema es posible, pero no todos son igual de probables. La probabilidad $P(\sigma_i)$ es proporcional a $e^{-\beta H(\sigma_i)}$ o denominado, factor de Boltzmann donde $\beta = \frac{1}{(k_b T)}$, k_b es la constante de Boltzman y T es la temperatura del sistema.
- c) **Colectivo macrocanónico.** Representa un sistema que no está aislado sino en *contacto térmico* con un baño, y además su número de componentes N es variable, pues puede tomar o ceder partículas de un *almacén*. Además de la temperatura, que da la energía típica que puede recibir del baño cada componente, hay un parámetro nuevo, el potencial químico μ , que representa el coste (en energía) de tomar una partícula del *almacén*. La probabilidad $P(\sigma_i)$ es proporcional al factor de Boltzmann.

2.3.1.2 Simulaciones Numéricas

Para los sistemas descritos anteriormente y con el objetivo de conocer las variables de estado o valores medios de los colectivos, tales como energía libre, energía interna o la entropía, se debe calcular una función particular para cada colectivo conocida como la función de partición: $\Gamma(E,N,V)$ para el colectivo microcanónico, $Z(T,N,V)$ para el colectivo canónico o $\Xi(T,\mu,V)$ para el colectivo macrocanónico. Según el colectivo que se use se deben obtener las pro-

babilidades $P(\sigma_i)$ para todos los microestados y a partir de ellas el cálculo de cualquier valor medio o valor esperado $\langle Q \rangle$.

El problema práctico consiste en que, aunque esas funciones de partición no son más que sumas de términos, se deben calcular uno a uno el número de términos en la suma es tan grande que no se pueden calcular exactamente. Los métodos de simulación permiten aproximar el resultado de los promedios estadísticos $\langle Q \rangle$ sin tener que sumar las funciones de partición.

Hay dos tipos distintos de simulaciones numéricas: Dinámica Molecular y Método de Monte Carlo.

2.3.1.2.1 Dinámica Molecular

Este mecanismo de simulación numérica resuelve las ecuaciones microscópicas del movimiento, obteniendo las trayectorias $\sigma(t)$ a partir de un estado inicial $\sigma(0) = \sigma(t_0)$ y según el *hamiltoniano* $H(\sigma_i)$. Esto garantiza que el sistema visite los estados con la probabilidad correcta. En un gas se seguirían las trayectorias de cada partícula, incluyendo los choques entre ellas, durante un tiempo relativamente corto (nanosegundos) pero suficiente para que las magnitudes de interés se promedien bien, es decir, no recuerden su estado inicial.

2.3.1.2.2 Métodos De Monte Carlo

De manera general se entiende como métodos de Monte Carlo aquellos procedimientos de simulación que utilizan números aleatorios. Estos métodos se pueden aplicar a problemas de tipo determinístico (solución exacta) o problemas estadísticos (no determinísticos). La exactitud de estos métodos es asegurada por la Ley de los Grandes Números, que garantiza la convergencia del método cuando el número de simulaciones tiende al infinito. A pesar de que los métodos de Monte Carlo se usan para resolver problemas muy diversos en áreas como matemáticas, físicas, economía, y en general problemas con gran número de grados de libertad, los métodos de Monte Carlo ayudan a resolver problemas usando las siguientes estrategias:

- Uso de muestras aleatorias.
- Aproximación de expresiones matemáticas complejas.
- Repetición un gran número de veces.

Los métodos de Monte Carlo tienen su origen en el Laboratorio Nacional de Los Álamos, y concretamente en los trabajos desarrollados por Stanislaw Ulam junto con John von Neumann. Éstos fueron posteriormente extendidos por Nicholas Metrópolis en su publicación [26]. Fue él mismo quien bautizó este tipo de métodos con el término "*Monte Carlo*".

Considerando los problemas de la mecánica estadística, los métodos de Monte Carlo permiten muestrear aleatoriamente las fluctuaciones térmicas del sistema de interés. Pasan de un estado a otro siguiendo un proceso estocástico de tipo cadena de Markov, en un espacio de configuraciones posibles, con el fin de estimar las propiedades de interés del modelo. La tarea principal de la mecánica estadística es calcular los promedios térmicos de interacción en un sistema con muchas partículas.

2.3.1.2.3 Cadenas de Markov

El concepto de cadenas de Markov es el eje central de los métodos de Monte Carlo. Para entender mejor este concepto se puede tener en cuenta el experimento del lanzamiento de una moneda al aire. Éste consta de una serie de procesos independientes, pues cada intento es independiente de los resultados anteriores. En caso contrario, la moneda podría estar trucada.

Un proceso de Markov difiere del experimento de la moneda porque cada prueba depende del resultado anterior, es decir, existe dependencia únicamente del estado anterior.

Formalmente [25], un proceso estocástico en un tiempo discreto $t_1, t_2, t_3, \dots, t_n$ para un sistema compuesto por un conjunto finito de estados $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n$, se define X_t como el estado del sistema σ , en el tiempo t . La probabilidad condicional que $X_{t_n} = \sigma_{i_n}$ es:

$$P(X_{t_n} = \sigma_{i_n} | X_{t_{n-1}} = \sigma_{i_{n-1}} | X_{t_{n-2}} = \sigma_{i_{n-2}} | X_{t_{n-3}} = \sigma_{i_{n-3}}, \dots, X_{t_1} = \sigma_{i_1}) \quad (2-5)$$

Sabiendo que, en el tiempo inmediatamente anterior, el estado del sistema $X_{t_{n-1}}$ ha transitado por el estado $\sigma_{i_{n-1}}$, y así para todos los estados en el sistema. Este proceso se denomina "proceso de Markov" si la probabilidad condicional definida anteriormente es un hecho independiente para todos los estados, pero no para el inmediatamente anterior. Es decir:

$$P(X_{t_n} = \sigma_{i_n} | X_{t_{n-1}} = \sigma_{i_{n-1}}) \quad (2-6)$$

La secuencia de estados $\{X_t\}$ se conoce como "cadena de Markov". La anterior probabilidad condicional puede ser interpretada como la probabilidad de transición de mover del estado i al estado j :

$$w_{i,j} = W(\sigma_i \rightarrow \sigma_j) = P(X_{t_n} = \sigma_{i_n} | X_{t_{n-1}} = \sigma_{i_{n-1}}) \quad (2-7)$$

Bajo las siguientes condiciones:

$$w_{i,j} \geq 0 \quad (2-8)$$

$$\sum_j w_{i,j} = 1 \quad (2-9)$$

La selección estadística de los microestados de un sistema debe ser la de los más característicos del sistema, pues ellos influyen en el momento de calcular las magnitudes de interés y determina el tiempo en que converge el sistema a un estado estable o de termalización. Es decir, generar estados al azar no es la mejor manera de encontrar los microestados porque se seleccionarán estados poco probables.

Ahora bien, relacionando los métodos de Monte Carlo con las cadenas de Markov, el problema se reduce entonces a encontrar microestados que sigan un proceso donde los nuevos estados dependan tan sólo del último estado y que se distribuyan usando una función de probabilidad concreta. La distribución que habitualmente es usada es la de Boltzmann. Para conseguir que la

cadena de Markov generada tienda a representar efectivamente esta distribución, existen dos condiciones: la condición de ergodicidad o accesibilidad y la condición del balance detallado o micro-reversibilidad.

- **Balance detallado.** Para asegurar que las simulaciones hechas a través de los métodos de Monte Carlo alcancen un estado de equilibrio se debe dar la condición del balance detallado, la cual permite imponer que exista una probabilidad donde se pueda transitar desde un estado cualquiera del sistema y salir de él. Es decir, intentar ir de un estado σ a otro estado σ' debe ser igual de probable que intentar ir del estado σ' al σ . Matemáticamente se puede expresar esta idea de la siguiente manera:

$$p_{\sigma}P(\sigma \rightarrow \sigma') = p_{\sigma'}P(\sigma' \rightarrow \sigma) \quad (2-10)$$

Donde p_{σ} es la probabilidad de que el sistema esté en un estado σ en un momento concreto:

$$\begin{bmatrix} P(1 \rightarrow 1) & P(2 \rightarrow 1) & \dots & P(M \rightarrow 1) \\ P(1 \rightarrow 2) & P(2 \rightarrow 2) & \dots & P(M \rightarrow 2) \\ \vdots & \vdots & \vdots & \vdots \\ P(1 \rightarrow M) & P(2 \rightarrow M) & \dots & P(M \rightarrow M) \end{bmatrix} \quad (2-11)$$

y $P(\sigma' \rightarrow \sigma)$ es la matriz de probabilidades de transición para $\sigma' \rightarrow \sigma$.

- **Ergodicidad.** La condición de ergodicidad en un proceso de Markov establece que debe alcanzarse cualquier estado σ_i del sistema a partir de otro estado cualquiera durante la evolución del sistema. Es decir, la condición exige que en el proceso de simulación se debe ser capaz de visitar todos los estados del sistema.

2.3.2 Modelo de Ising

El ferromagnetismo es la disposición magnética en todos los momentos magnéticos de una muestra de un material, en una misma dirección y sentido. Esta propiedad se puede encontrar en metales tales como el hierro y el níquel. Propiamente dicho, el ferromagnetismo es la presencia de magnetización espontánea incluso cuando no hay campo magnético externo. Esto es

ocasionado por los momentos magnéticos (o *spines*) de los átomos que se alinean en la misma dirección debido a la interacción entre los mismos, dando lugar a que la muestra del material se imante.

La alineación magnética se produce únicamente por debajo de la temperatura crítica o temperatura *de Curie*, a partir de la cual un cuerpo ferromagnético pierde su magnetismo. Por encima de dicha temperatura los *spines* están orientados de forma aleatoria, de tal manera que no hay un campo magnético neto.

Cualquier sustancia ferromagnética pierde dicho carácter a temperaturas altas, cuando la agitación térmica se hace lo suficientemente intensa como para alterar aleatoriamente la orientación de los momentos magnéticos. Entonces, la sustancia pierde la imanación permanente que pudiera tener y se convierte en paramagnética. A este fenómeno se le denomina transición de fase ferromagnético - paramagnético.

En 1920 Wilhelm Lenz [27] formuló un modelo para explicar el comportamiento de los *spines* de los átomos en materiales con características ferromagnéticas. En 1921, como profesor en la Universidad de Hamburgo, dirige a uno de sus primeros estudiantes de doctorado, Ernst Ising [28], el cual retoma su modelo y lo desarrolla en su tesis doctoral. Sin embargo, sus conclusiones fueron inexactas, pues erróneamente concluye que el modelo no exhibiría transiciones de fases en dos y tres dimensiones. El modelo se conoce desde entonces como el "modelo de Ising".

El modelo de Ising es un modelo sencillo que explica el fenómeno ferromagnético y permite su solución analítica en una o dos dimensiones.

Se parte de una rejilla regular que imita una red cristalina del hierro o níquel, en cuyas posiciones se colocan los momentos magnéticos asociados S_i ($i = 1, 2, \dots, N$) con valores $+1$ o -1 (arriba o abajo) de la siguiente forma:

1	1	-1	1	1	1
1	1	1	-1	1	1
1	1	1	-1	-1	-1
1	1	1	1	1	1
1	1	-1	-1	-1	-1
1	1	1	1	1	-1
1	1	1	1	-1	-1
1	1	1	-1	-1	-1

Figura 2.5. Distribución espacial de los espines en el modelo de Ising bidimensional

La energía asociada al sistema se define con la siguiente función de energía:

$$H(\sigma_i) = -\frac{1}{2} \sum_{i,j} J_{i,j} s_i s_j - B \sum_i s_i \quad (2-12)$$

Donde J_{ij} denota la fuerza de interacción entre los *spines* S_i y S_j , la i, j muestra que se recorren todas las posiciones adyacentes de la rejilla (vecinos más próximos), y B es el campo magnético externo. En ausencia de cualquier campo externo B , el *hamiltoniano* es la suma de las interacciones $J_{i,j} s_i s_j$ entre todos los pares de *spines* i, j con los vecinos más próximos ($|i-j|=1$). Habitualmente se considera que la interacción entre *spines* es isotrópica, es decir, $J_{i,j} = J > 0 \forall i, j$ y que solamente existe interacción entre sus vecinos más próximos, pudiendo reescribir (2-12) así:

$$H(\sigma_i) = -\frac{J}{2} \sum_{\langle i,j \rangle} s_i s_j \quad (2-13)$$

El modelo de Ising, y los que de él parten, habitualmente son tratados como ensambles canónicos. La distribución de probabilidad de ocurrencia de un microestado es definida por Gibbs [29], y es conocida comúnmente como la distribución de Boltzman:

$$P(\sigma_i) = \frac{e^{-\beta H(\sigma_i)}}{Z} \quad (2-14)$$

Donde β es:

$$\beta = \frac{1}{(k_b T)} \quad (2-15)$$

Y Z corresponde a la función de partición o constante de normalización. Z tiene su origen en la palabra alemana "*Zustandssumme*" que significa "suma sobre todos los estados".

$$Z = \sum_{\sigma} e^{-\beta H(\sigma_i)} \quad (2-16)$$

La constante de Boltzman k_b permite "traducir" unidades de temperatura en energía. La cantidad $k_b T$ da la energía típica que esta "a disposición" de cada componente del sistema.

De forma general, el valor esperado para una cantidad Q en equilibrio térmico se define como:

$$\langle Q \rangle = \sum_{\sigma} p(\sigma)q(\sigma) = \frac{\sum_{\sigma} e^{-\beta H(\sigma_i)} q(\sigma_i)}{Z} \quad (2-17)$$

Con la que se podrá calcular el promedio o valores esperados para cualquier magnitud macroscópica $\langle Q \rangle$, que dependerá de la temperatura y del número de componentes.

Si sólo se tuviera en cuenta la energía y se tratara de minimizarla, entonces la fase del sistema sería una fase perfectamente ordenada y, por tanto, ferromagnética. Sin embargo, la temperatura provoca un efecto aleatorio donde los *spines* pueden cambiar su valor al azar.

Este efecto es mayor, cuanto más alta es la temperatura y por ello, a temperaturas altas producen fases desordenadas. Dependiendo de cuál sea el efecto dominante entre energía y temperatura, aparecerán fases ferromagnéticas o no. A continuación se muestran tres diferentes configuraciones de una simulación del modelo de Ising en una rejilla cuadrada de 100x100 posiciones, con diferentes temperaturas: $T > T_c$, $T = T_c$, $T < T_c$.

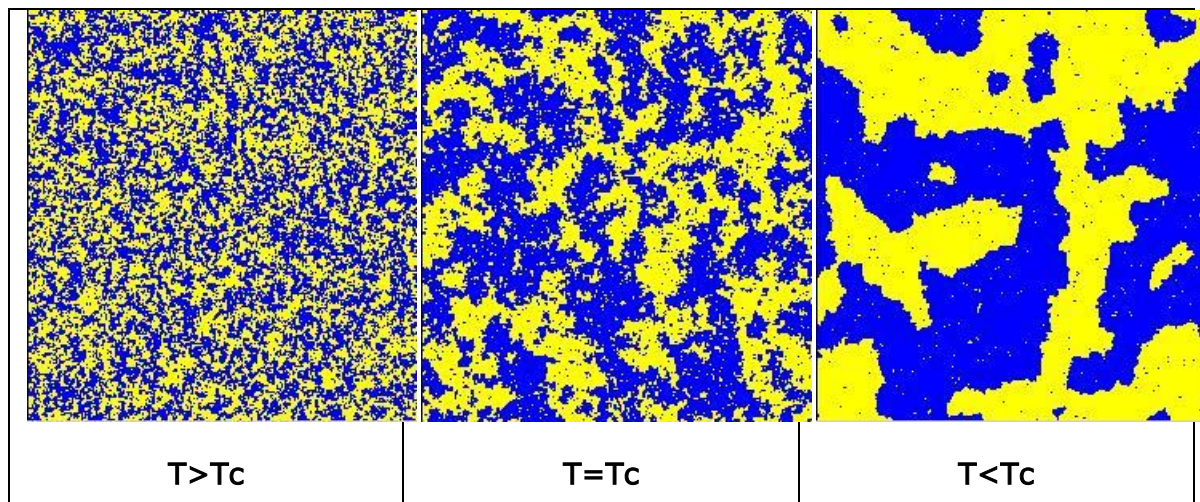


Figura 2.6. Simulaciones del modelo de Ising a diferentes temperaturas.

A continuación, se mostrará un sencillo ejemplo en el que se expondrá la dificultad que existe para calcular de manera directa los valores esperados en el modelo. Suponiendo que se tiene una rejilla de $n_x n_y = 2 \times 2$ donde existen 4 *spines* y cada uno de ellos puede tener 2 valores diferentes (1 o -1 arriba o abajo), para un total de $2^N = 2^4 = 16$ microestados diferentes, enumerados como se muestran la figura (Figura 2.7) siguiente:

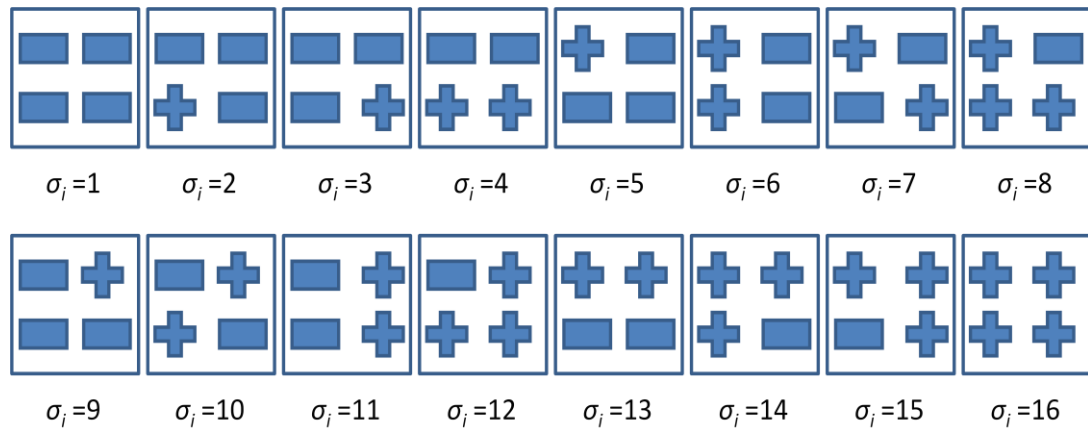


Figura 2.7. Diferentes configuraciones en un modelo de Ising con 4 partículas

La energía de los 16 microestados se resume a continuación:

Microestado	Energía del Micro Estado
i_1, i_{16}	$-8J$
i_7, i_{10}	$8J$
Los demás	0

La energía total del sistema es $E=0$.

La magnetización de cada microestado se define como es:

$$M(\sigma_i) = \sum_i s_i \tag{2-18}$$

De acuerdo con la mecánica estadística, la distribución de configuraciones para un conjunto de *spines* depende tan sólo del *hamiltoniano* y de la temperatura. La probabilidad de ocurrencia de un microestado puede expresarse usando las ecuaciones (2-14) y (2-16). En el ejemplo la probabilidad de ocurrencia del micro estado i_{16} es:

$$P(i_{16}) = \frac{e^{-8\beta J}}{z} \tag{2-19}$$

Donde la función de partición es:

$$z = 2e^{8\beta J} + 12 + 2e^{-8\beta J} \quad (2-20)$$

Usando (2-17), la magnetización esperada del sistema se define como M , donde:

$$\langle M \rangle = \sum_{\sigma} M(\sigma_i) \frac{e^{-\beta H(\sigma_i)}}{z} = \frac{2(4e^{8\beta J}) + 8(2)}{z} = \frac{8e^{8\beta J} + 16}{2e^{8\beta J} + 12 + 2e^{-8\beta J}} \quad (2-21)$$

Para sistemas de una sola dimensión, el sistema no produce transición de fase. En espacios de dimensión 1, los efectos del desorden provocado por la temperatura son siempre dominantes. Sin embargo, en un espacio de dos dimensiones y superior, sí que existe transición de fase. El cálculo de la solución exacta del modelo para dos dimensiones fue hecho en 1944 por Lars Onsager [30, 31].

En casos muy sencillos, la función de partición y los valores asociados al modelo (la energía libre, energía interna, entropía, etc.) son obtenibles a través de una solución analítica, como se ha mostrado en el ejemplo. Sin embargo, esto habitualmente no es común, pues con sólo considerar una rejilla de 8x8, en el modelo de Ising tenemos 2^{64} elementos en el denominador de la ecuación (3). Por tal razón, hay que acudir a otros métodos para evaluarlos.

Existen varios métodos que pueden generar cadenas de Markov en estados de un modelo de Ising [32]. Uno de los más sencillos y habituales es el algoritmo de Metrópolis [26], que basa su formulación en el cumplimiento de las condiciones de ergodicidad y balance detallado.

La probabilidad de ocurrencia del n -ésimo estado en un conjunto canónico está dada por la ecuación (1). Como se ha visto, el cálculo de Z es algo complejo. Para evitar esta complicación se pueden usar cadenas de Markov de los estados del sistema, es decir, generando un estado n directamente a partir

del estado predecesor m . Como resultado, la diferencia de energías entre los dos estados se define como:

$$\Delta H = H_n - H_m \quad (2-22)$$

La manera en que se implementa el algoritmo de Metrópolis se describe a continuación:

1. Generar una configuración inicial de *spines* para la rejilla.
2. Escoger al azar un *spin* de la rejilla i .
3. Realizar el cambio de valor del *spin* (multiplicar por -1).
4. Calcular el incremento de energía ΔH de la posición elegida $\Delta H = H_n - H_m$.
5. Aceptar este cambio, es decir, realmente modificamos el valor del *spin* con la siguiente función de probabilidad:

$$P(\sigma \rightarrow \sigma') = \begin{cases} e^{-\frac{\Delta H}{kT}}, & \Delta H > 0 \\ 1, & \Delta H < 0 \end{cases} \quad (2-23)$$

6. Regresar al paso número 2.

El modelo de Ising aporta dos ideas que retoma el modelo CPM-GGH [33]:

1. La energía (J) de los *spines* (fuerza de cohesión) en la rejilla pueden representar la energía de contacto entre dominios.
2. La temperatura (T) o fluctuación determina cómo será una configuración de *spines*.

Sin embargo, el modelo de Ising no puede llegar a ser un modelo que representa a las células biológicas debido a que:

- a. No reproduce la dinámica observada en los procesos biológicos.
- b. Existen varios dominios que comparten el mismo *spin*, mientras que en los procesos biológicos cada dominio necesita ser único para cada célula.

2.3.3 Modelo de Potts

En 1943, J. Ashikin y D. Teller [34] estudiaron una variante del modelo de Ising donde se consideraban cuatro valores de *spin* diferentes, que se conoce como "*Modelo de Ashikin-Teller*". Posteriormente, en 1952 [35], R.B Potts realiza un nuevo planteamiento donde retoma el modelo de Ising y lo extiende esta vez a q -componentes en espacios bidimensionales. En el año 1974, tres décadas después, C. Domb, [36], alumno de Potts en Oxford, retoma el trabajo de su profesor y lo generaliza desde ese momento como "*Modelo de Potts*".

En los años ochenta, Anderson, Grest, Sahni y Srolovitz [37] orientaron sus investigaciones a la aplicación del modelo de Potts para simular el comportamiento de crecimiento de granos en materiales metálicos expuestos a diferentes tipos de temperatura.

El modelo de Potts o Large- q Potts consiste, al igual que su predecesor, en una rejilla (habitualmente bidimensional cuadrada) con un total de N posiciones o *spines*, cada una de las cuales puede tomar un valor σ de un conjunto finito con Q de distintos valores $\{1,2,\dots,Q\}$. Se utiliza la notación σ para referirse al *spin* de la posición i,j . Se considera que un dominio es un conjunto conexo con el mismo valor de *spin*.

La rejilla podría tener la siguiente forma:

1	1	1	1	2	2	4	4	4	4	4
1	1	1	2	2	2	2	4	4	4	4
1	1	2	2	2	2	2	4	4	4	4
1	1	2	2	2	2	2	4	4	4	4
3	3	3	2	2	5	5	5	4	4	4
3	3	3	3	3	5	5	5	5	4	4
3	3	3	3	5	5	5	5	5	7	7
3	3	3	3	5	5	5	7	7	7	7
3	3	6	6	6	6	6	7	7	7	7
3	3	6	6	6	6	6	7	7	7	7

Figura 2.8. Rejilla bidimensional para el modelo de Potts.

Según esta disposición geométrica se considera la energía de cada posible configuración a partir de la energía necesaria para mantener juntos los valores de *spines* de celdas adyacentes. Este valor es 0 para celdas adyacentes dentro del mismo dominio.

El *hamiltoniano* es:

$$H = J \sum_{(i,j)(i',j')} [1 - \delta_{\sigma(i,j),\sigma(i',j')}] \quad (2-24)$$

Donde δ es la función *delta de Kronecker*, de modo que: $\delta(x, y) = 0$ si $x = y$ y 1 si $x \neq y$.

Al igual que el modelo de Ising, el cálculo analítico del modelo se vuelve intratable en la medida en que el tamaño de la rejilla aumenta. Por ello, es habitual que el modelo se simule usando el método de Metrópolis - Monte Carlo.

El sistema se desarrolla cambiando de σ a σ' a través del algoritmo de Metrópolis, escogiendo un elemento de la red aleatoriamente a una temperatura T dada. El cambio se produce según la siguiente distribución de probabilidad de Boltzman:

$$P(\sigma \rightarrow \sigma') = \begin{cases} e^{\frac{-\Delta H}{kT}}, \Delta H > 0 \\ 1, \Delta H \leq 0 \end{cases} \quad (2-25)$$

k es la constante de Boltzmann y ΔH es el incremento debido al cambio del *hamiltoniano*. El efecto global consiste en que los dominios se muevan mientras hay un ajuste de las fronteras. Existe una temperatura crítica T_c en la que los dominios tienden a separarse. Por debajo de esta temperatura los

spines tienden a unirse en dominios compactos, y a una temperatura $T=0$ los dominios experimentan una relajación.

El modelo de Potts contiene dos ideas importantes para simulaciones de células biológicas:

1. Los dominios poseen *spines* (índices) diferentes.
2. Los dominios tienen una energía J asociada, que representa una repulsión o atracción entre los dominios.

A pesar de ello, el modelo de Potts todavía tiene algunas deficiencias y no es capaz de expresar el comportamiento biológico de las células, ya que:

1. Muchos dominios pueden compartir el mismo índice. En las simulaciones biológicas es necesario que cada dominio represente un objeto único.
2. En el modelo de Potts se define una energía única de contacto J para todos los índices.
3. El modelo de Potts no incorpora un mecanismo para controlar el tamaño del dominio ni su forma.

2.3.4 Modelo CPM–GGH para simulación celular

El modelo celular de Potts (CPM), conocido actualmente como modelo Glazier-Grannier-Hogeweg (GGH) [33] (y que usaremos ampliamente en el presente trabajo), ha sido desarrollado por James Glazier y Francois Granier en los años noventa [8, 9], como extensión del modelo de Potts, a partir de la similitud de las formas que muestran las simulaciones de burbujas y el *crecimiento de granos* (*grain growth*) de los materiales con los que se ven por ejemplo, en el re-agrupamiento de células en un proceso de embriogénesis.

Glazier y Granier modificaron el modelo de Potts asemejando los dominios que se presentan en este modelo a células biológicas, e incluyeron restricciones de área para estas células. Asimismo, acuñaron el concepto de energía de contacto entre tipos de células (adhesión) para permitir a las células atraerse o repelerse según sea caso.

El modelo CPM-GGH se basa principalmente en la hipótesis del trabajo de Malcolm S. Steinberg (*Differential Adhesion Hypothesis* (DAH)) [38, 39], en la que, haciendo analogía a cómo se organiza dos diferentes tipos de líquidos en un recipiente, las células tienden a organizarse de manera similar. Por ejemplo, partiendo de un grupo aleatorio de células, éstas tienden a organizarse en conjuntos en los que aquellas células más cohesivas crean una estructura uniforme rodeada por aquellas células menos cohesivas. Este fenómeno se conoce como "*cell sorting*" o reorganización celular [40].

La observación de Steinberg propone que la configuración final de cualquier configuración inicial arbitraria de células minimiza la energía total del sistema, asemejándose al comportamiento de los fluidos viscosos.

Las diferentes fuerzas de adhesión que existen en un conjunto de células son los principales mecanismos que promueven la evolución de los sistemas biológicos. Sin embargo, la DAH no resuelve la dinámica de cómo se mueven las células, sólo ayuda a determinar cuál será la configuración más favorable de entre todas las posibles.

Del mismo modo que en los anteriores modelos descritos basados en mecánica estadística, en el modelo CPM o GGH los elementos que se simulan están dispuestos en una rejilla donde se desarrollan la mayoría de sus interacciones. La rejilla puede definirse en un espacio de dos o tres dimensiones (una dimensión menor a 2 o superior a 3 no tiene sentido en un modelo celular). En el modelo GGH, la rejilla es análoga al lugar donde las células cohabitan, por ejemplo: un tejido, un cultivo celular, etc. La rejilla se puede definir como periódica o fija en cada dirección según el objetivo de la simulación. La extensión volumétrica de la rejilla es igual al total del número de posiciones, que al mismo tiempo representa la unidad básica de duración del sistema.

El objeto principal en el modelo GGH es la "*célula extendida*" [41], que cobra sentido de acuerdo al interés del modelo que se desea implementar. La célula extendida puede asumir o representar diferentes entes biológicos como: una célula biológica, un grupo de células en un tejido, una bacteria, el medio que está alrededor de una o varias células, etc.

Cada célula extendida es un conjunto discreto de subcomponentes, el cual está formada por puntos o píxeles contiguos que comparten el mismo índice σ (Figura 2.9). El modelo GGH rechaza el uso del término *spin* y lo renombra como "índice" para evitar la confusión con los anteriores modelos y la carencia de significado en un contexto biológico.

Un caso particular se produce al hacer referencia al *medio*, en este caso el índice podrá tomar únicamente el valor de 0 ($\sigma = 0$). Los enlaces entre diferentes índices representan regiones de fronteras de contacto entre las membranas de dos células.

1	1	2	2	2	2
1	1	2	2	2	2
1	1	2	2	2	2
1	1	2	2	2	2
1	1	2	3	2	2
3	1	2	3	3	3
3	3	3	3	3	3
3	3	3	3	3	3

Figura 2.9. Representación de la rejilla y células en el modelo CPM-GGH

Como se ha visto anteriormente, el *hamiltoniano* define una función de energía de interacción entre las partículas. En un contexto más biológico, el *hamiltoniano* no determina la energía real de las células biológicas o de un tejido, sino que es una forma de especificar los factores que gobiernan y determinan los cambios e interacciones de las células en el tiempo. Utilizando esta función, junto con un sistema de ecuaciones diferenciales y la forma y condiciones de contorno de la rejilla, se pueden expresar diferentes tipos de modelos celulares [41].

En un entorno biológico, las células poseen diferentes atributos tales como volumen o superficie, e interactúan unas con otras a través de una condición de adhesión o repulsión.

Estas ideas intuitivas se recogen en la siguiente ecuación:

$$H = E_{\text{adhesión}} + E_{\text{surface}} + E_{\text{volumen}} \quad (2-26)$$

Donde de forma extendida el *hamiltoniano* se define a continuación:

$$H = \sum_{\substack{i,j \\ \text{Vecinos}}} J(\tau(\sigma_i), \tau(\sigma_j))(1 - \delta(\sigma_i, \sigma_j)) + \sum_{\sigma} [\lambda_{\text{vol}}(\sigma)(v(\sigma) - V_t(\sigma))^2] + \sum_{\sigma} [\lambda_{\text{surface}}(\sigma)(s(\sigma) - S_t(\sigma))^2] \quad (2-27)$$

La primera parte de la ecuación incorpora al modelo el mecanismo para que dos células que comparten una misma región experimenten un proceso de repulsión o adhesión. Este concepto se basa en la hipótesis DAH, comentada anteriormente. Los vectores \vec{i} y \vec{j} en el sumando evalúan la interacción de las diferentes posiciones dentro de la rejilla. $J(\tau(\sigma_i), \tau(\sigma_j))$ representa la energía de contacto por unidad de área para un par de células de cierto tipo $\tau(\sigma_i)$, donde J es un escalar $\in \mathbb{R}$. Como se ha mencionado previamente, el símbolo δ es la función *delta de Kronecker*, donde $\delta(x, y) = 0$ si $x = y$ y 1 si $x \neq y$. La relación $J(\tau(\sigma_i), \tau(\sigma_j))$ es una matriz simétrica de índices J entre tipos de células. Cuando se dan valores altos de J , se produce una relación de repulsión entre las células. Por el contrario, con valores bajos de J , muestran una mayor adhesión.

El segundo término es una restricción que penaliza las desviaciones en los atributos de las *células extendidas*, es decir, que este tipo de sumandos en el *hamiltoniano* realizan una acción de control, aumentando en la medida en que los atributos se desvían de su valor objetivo. En el caso concreto del segundo elemento de la ecuación (2-27), éste evalúa las desviaciones que puede sufrir el área de la célula respecto a su valor objetivo ($V_t(\sigma)$), donde un incremento de la energía efectiva penaliza la desviación. El parámetro λ_{vol} asume el rol de mecanismo de elasticidad de la célula, y también determina la ponderación o importancia relativa de la energía asociada a la restricción de volumen al modelo. Los valores altos de este parámetro muestran una reducción en las fluctuaciones del volumen de la célula con respecto a su volumen objetivo. Usando el mismo argumento se desarrolla el tercer ítem de la

ecuación (2-27), en el que se implementa la restricción de superficie (3d) o perímetro (2d) donde λ_{surface} hace referencia a la elasticidad de la membrana.

De forma estricta, el modelo CPM-GGH se concibe originalmente como un modelo discreto, y las actualizaciones de la rejilla se basan en un conjunto de reglas probabilísticas. En este sentido el modelo CPM-GGH puede ser considerado como un autómata celular (CA). A pesar de heredar muchos elementos de la mecánica estadística ofrecidos por sus antecesores, existen muchas diferencias que separan al CPM de los otros modelos, mientras que su actualización tiende a parecerse más a un autómata celular reversible. Esto podría ser motivo de discusión.

La evolución del modelo CPM-GGH a través del tiempo es un método estocástico repetitivo de la minimización de la energía libre que se expresa a través del hamiltoniano. El resultado de usar esta dinámica sobre los diferentes objetos dentro de la rejilla permite expresar diferentes formaciones de patrones y desplazamientos de las células dentro de la rejilla, imitando la movilidad celular.

El modelo CPM usa una versión modificada del algoritmo de Metrópolis-Monte Carlo similar al que utilizan los modelos de mecánica estadística ya descritos en las secciones anteriores. El algoritmo consiste en seleccionar, de manera aleatoria, una posición dentro de la rejilla para intentar modificar su índice. El modelo selecciona al azar una posición objetiva \bar{i} , y de la misma manera selecciona uno de sus vecinos más próximos, determinado a través de la distancia euclidiana (posición destino \bar{i}'). Si al momento de comparar las posiciones, éstas pertenecen a diferentes células, el algoritmo cambia el valor de la posición "temporalmente", igualando la posición-origen con la posición-destino ($\sigma_{\bar{i}} = \sigma_{\bar{i}'}$) con probabilidad $P(\sigma_{\bar{i}} \rightarrow \sigma_{\bar{i}'})$ y, usando la función de probabilidad de Boltzmann, evalúa si conserva o no el cambio de acuerdo al siguiente criterio:

$$P(\sigma_{\bar{i}} \rightarrow \sigma_{\bar{i}'}) = \begin{cases} 1 & : \Delta H \leq 0 \\ e^{-\frac{\Delta H}{T_m}} & : \Delta H > 0 \end{cases} \quad (2-28)$$

Donde ΔH es el cambio de la energía efectiva si el cambio ocurre y $T_m \in \mathbb{R}^+$.

Aquí el lector podrá encontrar ambigüedad en los términos usados, pues se conservan los mismos heredados de la mecánica estadística. La temperatura, por ejemplo, es uno de éstos, ya que trasladándolo al ámbito biológico se entiende como la fluctuación o movilidad de la membrana celular.

En una rejilla de N posiciones se define como un paso de Monte Carlo a las N evaluaciones de intento de cambio en los índices; también se define como "*tasa de aceptación*" al promedio de los cambios aceptados dividido por los intentos de cambio de los índices. Como regla general, la tasa de aceptación debe ser superior a 1%. Si la tasa de aceptación es pequeña, el método de Monte Carlo es ineficiente. La tasa de aceptación aumenta en la medida en que aumenta el valor de T .

A continuación, se muestran diferentes simulaciones usando el modelo CPM-GGH con diferentes valores de las energías de contacto, hallando como resultado diferentes patrones de formación de grupos células.

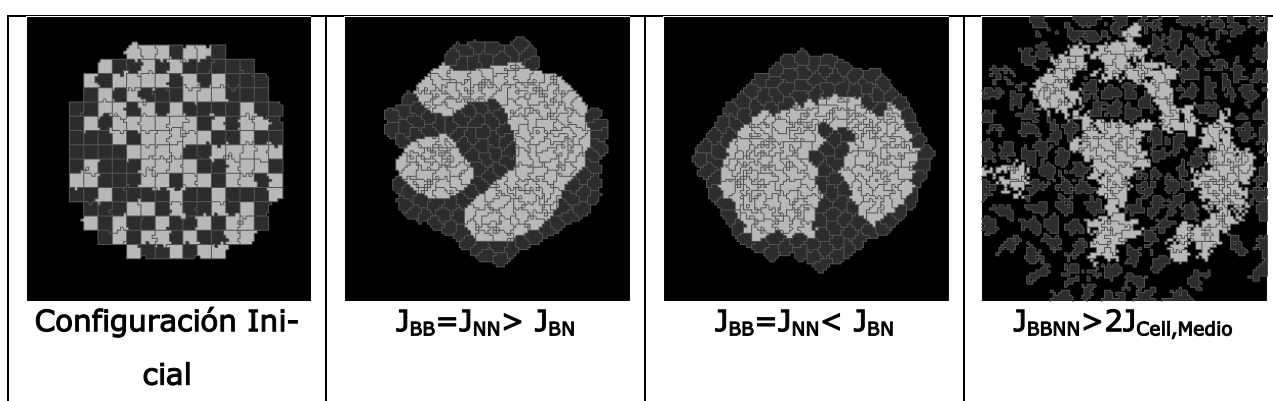


Figura 2.10 Simulación de patrones de formación de células usando diferentes valores de J

El movimiento de las células es primordial para el desarrollo de los diferentes fenómenos biológicos, tanto en situaciones normales (ej. el desarrollo embrionario) como patológicas (ej. el cáncer). Emplear el método de Metrópolis tiene su razón de ser, la cual no obedece solamente al hecho de ser el método usado tradicionalmente en los modelos de mecánica estadística, donde el CPM-GGH tiene su origen. El método de Metrópolis permite desarrollar una de las características más atractivas del modelo CPM-GGH, y es que con él, el

CPM-GGH es capaz de representar el comportamiento continuo de la migración de las células a través de un medio, donde las células usan extensiones parciales y retracciones de la membrana celular para moverse (filopodios). El método de Metrópolis, al actualizar los diferentes puntos de la rejilla, uno a la vez, es capaz de emular lo que sucede en la realidad: cómo las células se mueven poco a poco, en lugar de mostrar saltos desordenados en el espacio-tiempo. Si se usan otros métodos alternos al de Metrópolis, como el algoritmo de Kawasaki [42], donde su objetivo es buscar configuraciones para un estado en equilibrio dentro la mecánica estadística, no se obtendrán los mismos resultados o se tendrán pero no con un sentido biológico, en el que lo buscado es emular la cinética de las células.

Aunque el modelo CPM-GGH no incorpora explícitamente la descripción de una fuerza que actúa sobre la membrana celular, el *hamiltoniano* puede tratarse como tal [42]. La determinación de la probabilidad de prolongación y retracción de los pseudópodos se interpreta como una fuerza que se aplica localmente a la membrana celular, si los movimientos obedecen a una relación fuerza-velocidad de un movimiento oscilatorio sobre-amortiguado, es decir, la velocidad es proporcional a la fuerza aplicada sin inercia.

$$\vec{V} \sim \vec{F} = -\vec{\nabla} H \quad (2-29)$$

Esto es una suposición prudente para la matriz extracelular, que es altamente viscosa en las células de mamíferos.

2.3.5 Selección del Modelo CPM-GGH como Entorno para Las Simulaciones.

El CPM-GGH es tan solo uno de los diferentes modelos que hay para simular procesos celulares. Durante la revisión del estado del arte se han podido comparar diferentes aproximaciones que pueden ajustarse a los objetivos que se definieron en las hipótesis de partida de esta tesis. Las principales características que han determinado esta decisión se enumeran a continuación:

- El modelo que es capaz de expresar comportamientos complejos usando simples interacciones a través de modificar o adicionar nuevos componentes en su Hamiltoniano.
- Los modelos basados en autómatas celulares por definición son discretos tanto en sus componentes como su evolución. El CPM-GGH cumple estas condiciones, pero es posible adicionar elementos continuos como PDE, para aumentar su expresividad y desarrollar modelos híbridos [43].
- Muchos de los modelos que permiten describir sistemas de células biológicas, lo hacen desde el punto de vista de considerar las células como puntos o esferas, haciendo caso omiso que la forma de las células pueden ser un factor determinante y esencial en el desarrollo del modelo. En el CPM-GGH, la energía de contacto entre las diferentes células y la interacción con el medio determina la forma que desarrollará una célula.

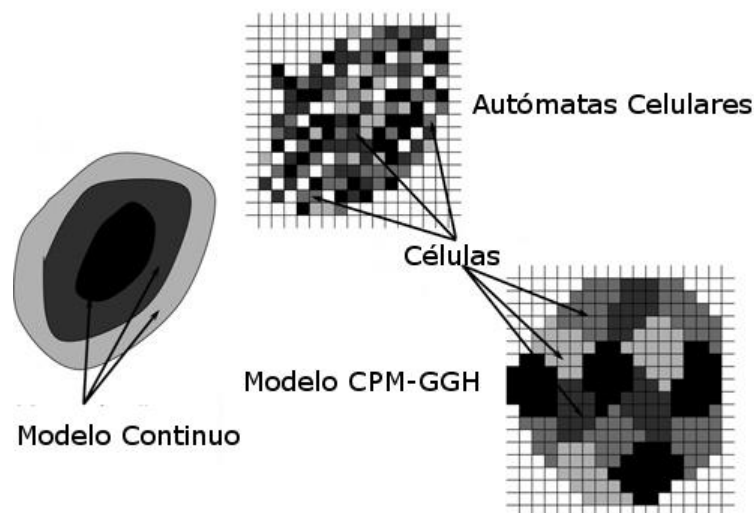


Figura 2.11 Representación esquemática de diferentes modelos de simulación celular – adaptado de [44]

- Otra ventaja de la CPM-GGH es la descripción clara de los conceptos de adhesión y de atracción celular [42], pues el modelo es capaz de distinguir la distancia en que se encuentran un par de células y con ello como será su interacción de atracción o repulsión.

- Como se ha descrito en apartados anteriores, uno de los factores más importantes que recoge el modelo GGH-CPM, es el uso de la hipótesis de adhesión diferencial (DAH) que es determinante para explicar la formación y conservación de muchos de los tejidos, determinada a la distribución y organización que presentan los diferentes tipos de células dentro de un grupo.

-Todos para uno y uno para todos
Alejandro Dumas , Los Tres Mosqueteros



Capitulo 3 Mecanismos para la Aceleración de los Modelos Basados en CPM-GGH

Los desarrollos e implementaciones paralelas del modelo GGH-CPM constituyen un área de investigación bastante activa [45]. Como se mostrará en este capítulo, una de las limitaciones del uso de esta aproximación es la actualización de la rejilla, pues el algoritmo que se utiliza (Metrópolis – Monte Carlo) evalúa una a una las posiciones de la rejilla para minimizar la energía del sistema. La manera en que se desarrolla la actualización de la rejilla determina la principal dificultad en el CPM-GGH, ya que para paralelizarlo hay que tener en cuenta que la evaluación de la minimización de la energía se hace de manera local, es decir, evaluando sus vecinos más próximos dentro de un proceso estocástico.

Esta manera de abordar el problema lleva a pensar en la necesidad de usar diferentes técnicas que permitan incrementar la velocidad con la que se realizan los cálculos. Se han propuesto diferentes opciones para mejorar la eficiencia de cálculo de la dinámica del modelo. No hay que olvidar que los objetivos de la simulación de los modelos ferromagnéticos difieren del interés del CPM-GGH, es decir, que mientras unos buscan obtener en un espacio-tiempo un número suficiente de configuraciones para encontrar ciertos valores del sistema, en el mo-

delo CPM-GGH el interés radica en poder observar la dinámica de la evolución biológica de los elementos que se incluyen en la rejilla.

Por ejemplo, para una simulación sencilla de 10000 a 20000 células, un ordenador común puede ser suficiente para afrontar un modelo y permite que un investigador pueda interactuar fácilmente con él, generando hipótesis o cambios para encontrar nuevos resultados.

Por consiguiente, la pregunta consiste en porqué esforzarse en paralelizar el modelo, si la potencia de cálculo y la velocidad de los ordenadores de escritorio son suficientes para ser usadas como herramientas de simulación. Hay varias justificaciones para esta pregunta, pero existen al menos tres razones que pueden apoyar el argumento:

- En primer lugar, un cálculo rápido y sencillo muestra que si se desea simular con gran detalle una porción de tejido epidérmico de algunos centímetros, ésta puede convertirse en una tarea compleja. 1 mm³ contiene aproximadamente 100.000 células [46] y cada objeto dentro de la rejilla requiere al menos 150 bytes para describir los diferentes atributos de cada célula, y con ello el proceso de evolución de la interacción entre éstas son computacionalmente costosas. Por ello las simulaciones que se realizan generalmente son pequeñas, de 200 x 200 hasta 1000 x 1000 sitios.
- En segundo lugar, la mayoría de los resultados de las simulaciones usando un ordenador convencional se limitan a usar objetos en 2D, lo cual puede ser una restricción para desarrollar hipótesis más ricas y realistas.
- Finalmente, un modelo que contenga más elementos y mayor detalle requerirá mayor potencia para que los resultados se obtengan en tiempos lineales.

Existen diferentes maneras de abordar la solución a estos problemas. Habitualmente, antes de contar con la capacidad de cómputo que ofrecen los procesadores actuales, la orientación principal de los investigadores fue la de plantear nuevas formas de actualizar la rejilla sin transgredir las normas físicas del mode-

lo. Estos modelos basan su hipótesis en el hecho de que el método de Metrópolis desarrolla lentitudes cuando se usan valores de T inferiores a la temperatura crítica, pues la tasa de aceptación del cambio de índice es muy baja, llevando a que el proceso ocasione miles de evaluaciones sin realizar ningún cambio en la rejilla.

Este capítulo de “estado del arte” se divide en dos partes: una primera donde se introduce al lector en apartados referentes a la computación paralela, y una segunda en la que se realiza una exposición de los modelos que se han revisado y que permiten evaluar de manera diferente los cambios de la rejilla.

3.1 Computación Paralela

La ley de Moore, propuesta por Gordon Moore en 1965, expone que aproximadamente cada dos años se duplicará el número de transistores en un circuito integrado [47]. Esta propuesta ha sido un ambicioso intento de prever la evolución tecnológica en el campo de la computación. Unos años más tarde la ley fue ajustada reduciendo el plazo a 18 meses. Esta ley sigue estando vigente hoy en día, casi 45 años después desde que se formuló. Aunque se debe aceptar la clara limitación que impone la miniaturización de los circuitos que determinará el fin en algún momento de la vigencia de esta ley, obligando a la búsqueda de alternativas para la aceleración del computo de los algoritmos [48].

Las posibilidades que ofrecen las nuevas infraestructuras de computación estimulan a las comunidades industriales y científicas a producir continuamente nuevos retos cada vez más complejos. El principal problema de tener éxito asignando tareas dentro de la computación en varias máquinas consiste en que debe tener un menor coste que si se realizase en una sola máquina, y esto no es siempre una tarea fácil.

En el ámbito de esta materia existe ambigüedad sobre qué es exactamente computación paralela, pues se asocia el mismo término a diferentes conceptos muy similares que se definen a continuación:

- **Paralelismo.** Hace referencia al cómputo o solución de un problema lo más rápidamente posible usando la estrategia de (“divide y vencerás”) a través del uso de diferentes arquitecturas hardware o software.

A nivel hardware, el paralelismo se puede implementar en procesadores con múltiples unidades de cómputo, múltiples procesadores que comparten una misma memoria, y de procesadores distribuido en diferentes máquinas en una red.

A nivel software, el paralelismo trata en general del uso de múltiples hilos de ejecución o (*threads*) que se procesan simultáneamente.

- **Concurrencia.** Es la simultaneidad de procesamiento, usando múltiples hilos (*threads*) de ejecución. La concurrencia se asocia más a la arquitectura ofrecida por el *software* en lugar de la arquitectura del *hardware*. El código puede ser escrito para usar la concurrencia con el fin de explotar el paralelismo ofrecido por el *hardware*. A menudo la concurrencia es necesaria para reaccionar a diferentes eventos asíncronos; por ejemplo, el caso de un hilo que funciona de forma independiente para realizar un tipo de cálculo concreto, mientras otro hilo espera un evento del clic del ratón del usuario.
- **Computación distribuida.** Es el procesamiento que se realiza en un conjunto de nodos conectados entre sí, cooperando entre ellos para desarrollar una misma tarea. Mientras en el modelo de procesamiento paralelo con frecuencia (aunque no siempre) se asume que puede existir una memoria compartida para compartir información, en los sistemas distribuidos el mecanismo para compartir información se basa fundamentalmente en el paso de mensajes.

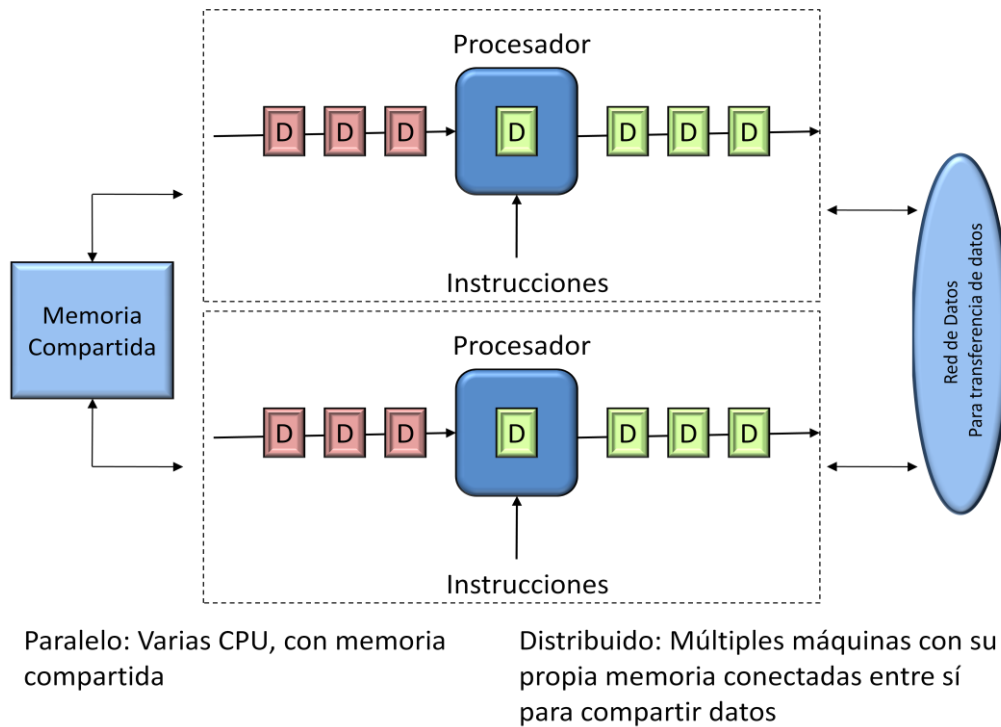


Figura 3.1. Diferentes modelos de procesamiento

Como se puede observar, a menudo este conjunto de términos se utiliza indistintamente haciéndose referencia uno a otro o solapándose entre sí. Lo más correcto es entender que la paralelización es un tipo de computación distribuida [49].

Históricamente, la computación en paralelo ha sido considerada como "el límite de la computación" [50] y se ha utilizado para resolver problemas difíciles en diferentes áreas de la ciencia y la ingeniería como:

- Predicción del estado del tiempo.
- Problemas relacionados con la Física nuclear de partículas.
- Biotecnología, genética o química.
- Matemáticas.
- Y en muchos otros campos de la ciencia.

3.1.1 Clústeres de Ordenadores

Junto con el desarrollo de los procesadores y el avance en la tecnología aplicada a las comunicaciones entre ordenadores, las cuales han permitido la transmisión de datos a grandes velocidades (hasta de 10gb/s), es posible unir varios ordenadores a una red y ponerlos a trabajar en pro de la solución de un problema. Ésta es una de las formas más económica y efectiva para hacer frente a los problemas más grandes que limitan el uso de un único equipo. Dicho enfoque de trabajo se conoce como clúster de estaciones de trabajo (CoWs) o de ordenadores personales (CoPs).

El cómputo científico ha cambiado mucho en las últimas décadas: del uso de procesadores vectoriales (procesadores capaces de ejecutar en un solo ciclo de reloj un vector de instrucciones) a las estaciones de trabajo personales, donde la tendencia actual es agrupar un gran número de unidades de procesamiento. En la actualidad cada superordenador se construye en torno al principio de la agregación que va desde las decenas a cientos de miles de CPUs, cada uno con potencia de cálculo relativamente pequeña, donde las aplicaciones se ejecutan en paralelo en todas estas unidades de procesamiento a fin de lograr un único objetivo.

La mayoría de máquinas altamente especializadas, tales como los superordenadores modernos, utiliza interconexiones patentadas, CPU homogéneas y sus propios dispositivos de entrada/salida, entre otros.

Un clúster es un conjunto de máquinas independientes, cada uno con su propia CPU y memoria conectados entre sí por una red privada rápida, comúnmente confiable. Actualmente, los clústeres proporcionan un alto rendimiento, disponibilidad, fiabilidad y escalabilidad. Las máquinas independientes que forman el clúster (comúnmente conocidos como nodos) son normalmente homogéneas, presentando idénticas características de *software* y *hardware*, aunque también existen muchos clústeres que presentan cierto grado de heterogeneidad, en ellos existen diferentes tipos de nodos que por lo general se encuentran dedicados a diferentes tareas específicas.

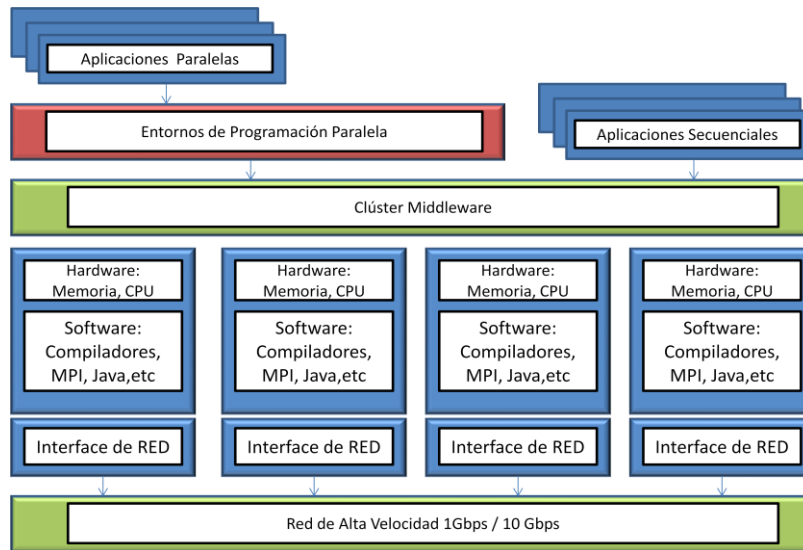


Figura 3.2. Esquema de un clúster

El control de un clúster habitualmente lo realiza un equipo concreto denominado "maestro", donde se controla todo el *hardware* y *software* de las máquinas. Su función básica es proporcionar los mecanismos necesarios para que todos los nodos trabajen juntos de manera coordinada.

El clúster puede usar un *middleware* [51] para mostrarse a los usuarios como una única máquina, haciendo que el total de nodos cooperen y actúen como una unidad monolítica o "*Single System Image (SSI)*" [51, 52], vigilando la interacción del usuario con el sistema, gestionando aspectos como el equilibrio de carga de los procesos, la tolerancia ante posibles fallos, etc.

El *middleware* por lo general trabaja conjuntamente con el sistema operativo controlando las diferentes operaciones y *tareas* a nivel de sistema, funcionando como una capa intermedia entre el usuario y los nodos del clúster, proporcionando:

- **Un fácil mantenimiento y optimización del clúster.** Las herramientas que conforman el *middleware*, ofrecen diferentes servicios: tareas de verificación o puntos de control para los trabajos lanzados, equilibrio de carga, tolerancia a fallos, programación de tareas, etc.

- **Una única interfaz de acceso al entorno.** El *middleware* ofrece a los usuarios del sistema un único punto de acceso, proporcionando la misma respuesta que cuando se realizan operaciones en un solo equipo. Esto ayuda al usuario a eliminar la complejidad que significa lanzar un trabajo dentro de un clúster, lo que es importante teniendo en cuenta que los clústeres pueden incorporar cientos o incluso miles de nodos.
- **Escalabilidad.** El *middleware* está preparado para detectar automáticamente nuevos módulos u otros recursos incorporados al clúster, siendo de inmediato proporcionados y aprovechados por todo él.

En la actualidad, existen varios single system image SSI o sistemas operativos capaces de ejecutarse desde una imagen, que pueden ser usados dependiendo de la misión con que se construya el clúster: libres, propietarios, basados en diferentes sistemas operativos o que ya están en desuso o no son soportados. Entre los más frecuentes se encuentran: MOSIX[53] , OpenMosix [54], Beowulf[55], SSI-OSCAR [56], OpenSSI [54], Kerrighed [57] entre otros.

Aunque la computación para clústers proporciona una alternativa relativamente barata a los superordenadores, a veces existe la idea errónea de que cualquier código de *software* se ejecutará más rápido en un clúster, ya que se supone que debe proporcionar un mayor rendimiento. Sin embargo, es importante recordar que, para lograr esto, el *software* debe someterse a una adaptación a este entorno distribuido.

3.1.2 Programación Paralela

Los programas que se desarrollan para que se ejecuten en un entorno paralelo son más difíciles de diseñar que los secuenciales, debido a que la concurrencia introduce nuevos desafíos y problemas al programar. La comunicación y la sincronización entre las diferentes partes suelen ser alguno de los más grandes obstáculos para conseguir un buen rendimiento de los programas paralelos [52].

Habitualmente los programas informáticos se han escrito para ser ejecutados de manera secuencial. Para resolver un problema, se construye un algoritmo y se implementa como una secuencia de instrucciones. Estas instrucciones se ejecutan en una CPU, donde sólo se puede ejecutar una instrucción a la vez; después de que ésta es ejecutada se ejecutará la siguiente.

Por otra parte, la programación paralela utiliza diferentes piezas de código que se ejecutan simultáneamente mediante la división del problema en partes independientes, de modo que cada elemento de procesamiento puede ejecutar su parte del algoritmo a la vez que las otras. Los elementos de procesamiento pueden ser diversos: como un único ordenador con varios procesadores, varios ordenadores en red, *hardware* especializado, o cualquier combinación de los anteriores.

Para un programa dado es posible conocer el aumento de velocidad al distribuir-la en un entorno paralelo (p.e. al tener disponibles más de un nodo o núcleo). El máximo incremento de velocidad de un programa como resultado de ser paralelizado, está gobernado por la ley de Amdahl [58], que se utiliza para determinar el máximo aumento teórico de velocidad S_n de un programa, expresándose de la siguiente manera:

$$S_n = \frac{1}{(1 - P) + \frac{P}{N}}$$

Donde P es el porcentaje de código que se puede ejecutar en paralelo, N el número unidades de procesamiento y (1-P) es la proporción del programa que se ejecuta en serie.

La ley establece que la pequeña porción de código que no se puede paralelizar limitará la aceleración global de la paralelización. Si ninguna parte del código puede ser paralelizada ($P = 0$) entonces la aceleración es ≈ 1 . Si todo el código es susceptible de ser paralelizado, la velocidad es máxima (N).

En el desarrollo de los sistemas multiprocesador y desde el punto de vista de la arquitectura de *hardware*, se han propuesto una variedad de modelos. Una de las principales diferencias entre ellos reside en la memoria principal, es decir, si los procesadores comparten la misma memoria para almacenar datos. De acuerdo con esta clasificación, hay dos familias principales de arquitecturas paralelas:

a) Memoria compartida

En este modelo todos los procesadores tienen acceso a la memoria como espacio de direcciones global. Por lo tanto, los cambios en una ubicación de memoria hechos por un procesador serán visibles para todos los demás.

La ventaja de las arquitecturas de memoria compartida es que la comunicación entre procesadores es muy rápida, pero presentan una deficiencia en su *bus* de datos principal, pues puede causar un problema de cuello de botella. Los modelos de programación de memoria compartida centran su interés en el acceso asíncrono al espacio de direcciones de memoria.

Las máquinas paralelas de memoria compartida se pueden clasificar como:

- UMA (*Uniform Memory Access*): Procesadores idénticos con igual acceso y tiempos a la memoria.
- NUMA (*Non-Uniform Memory Access*): Alternativa donde los procesadores incorporan módulos de memoria, donde además de acceder a su memoria local cada procesador puede acceder a otras memorias a través de un canal. El acceso inter-procesador puede verse limitado, ya que no todos los procesadores tardan el mismo tiempo de acceso para acceder a toda la memoria. El acceso a la memoria a través del *bus* es lento.

b) Memoria Distribuida

En los sistemas que se enmarcan en este grupo cada procesador tiene su propia memoria, no visible o accesible por los demás. La comunicación y sin-

cronización entre procesadores se lleva a cabo a través del paso de mensajes o *tokens*. En esta arquitectura no existe un *bus* de datos compartido. De acuerdo a la taxonomía propuesta por Flynn [59] las máquinas con memoria distribuida pueden ser clasificadas como:

- *SISD (Single Instruction, Single Data)*: La CPU procesa únicamente una instrucción por cada ciclo de reloj, y únicamente un dato puede ser usado o procesado en cada uno de dichos ciclos.
- *SIMD (Single Instruction, Multiple Data)*: Cada CPU procesa independientemente la misma instrucción en cada ciclo de reloj, y cada CPU puede a su vez procesar diferentes datos.
- *MISD (Multiple Instruction, Single Data)*: Cada CPU ejecuta una instrucción distinta en cada ciclo de reloj, y cada CPU procesa el mismo dato.
- *MIMD (Multiple Instruction, Multiple Data)*: Cada CPU procesa instrucciones distintas con datos diferentes.

El estándar de programación para la ejecución en máquinas paralelas o multi-núcleo empleando memoria compartida es el *thread*, donde un proceso puede tener múltiples ejecuciones concurrentes. Existen dos implementaciones principales [60] de *Threads*:

- *OpenMP*: Son directivas de compilación o *pragmas* que modifican el código en serie y lo habilitan para que se ejecute en varios procesadores. OpenMP es portátil y multi-plataforma, disponible para C / C ++ y Fortran.
- *POSIX Threads (o pthreads)*: Son librerías que habilitan a varios lenguajes la posibilidad de activar hilos de código para que realicen tareas en paralelo. Se requiere del programador mayor atención para obtener mejores resultados.

Desde el punto de vista de programación en memoria distribuida y paralela se pueden enumerar los siguientes enfoques:

a) Paso de mensajes

En este paradigma una porción o programa independiente es cargado en cada procesador utilizando su propia memoria (típicamente en una máquina MIMD). Las unidades de cómputo pueden residir en la misma máquina o en máquinas distribuidas en una red. Cada programa puede estar escrito en un lenguaje comúnmente C, C++ o Fortran, y el intercambio de datos se controla mediante llamadas y envío de mensajes. La Interfaz estándar para esta implementación es MPI [61].

El programador tiene el completo control sobre la distribución de los datos, la comunicación entre los procesos y las unidades de cómputo a quien asigna tareas, y así también le es otorgada la responsabilidad de organizar los procesos para que operen de forma colectiva y orquestada.

Claramente las comunicaciones inter-maquina introducen una sobrecarga adicional en la ejecución del programa, pero las hace parte esencial de la programación usando este paradigma. A menudo se obvia esta ralentización que imponen las comunicaciones.

Aunque este enfoque es muy flexible, el que lo desarrolle debe contemplar las posibles desventajas, como por ejemplo [61]:

- Cada procesador puede estar realizando una tarea diferente y también ejecutando diferentes porciones de código, haciendo que el entendimiento y depuración de una aplicación sea complejo.
- El programador debe asegurarse de que todos los procesadores estén ocupados tanto como sea posible para no subutilizar este recurso.
- Es importante asegurarse de que la aplicación está bien diseñada para no encontrarse con 'puntos muertos' y bloquear la ejecución.

b) Datos Paralelos (Data Parallelism)

La idea central en este enfoque es que los datos, y en concreto los de tipo vectorial o matricial (*arrays*), se distribuyen a través de las unidades de cómputo pudiendo tratarse en paralelo[62] .

En este enfoque, un único hilo de control de ejecución es el responsable de la distribución de los datos dentro del entorno paralelo. En la mayoría de los casos, la distribución real de los datos y la comunicación entre los procesadores es llevada a cabo por el compilador. Desde el punto de vista del programador, tan solo se verá un único espacio de memoria con detalles a bajo nivel. La ejecución del programa de manera distribuida no está sincronizada en todo momento, tan sólo se realiza en casos concretos de sentencias tales como: *Loop*, *do* o *for*. Aunque cada procesador puede ejecutar la misma porción de código, no hay garantía de que cada procesador esté ejecutando la misma instrucción en el mismo instante de tiempo. Los lenguajes de programación comunes que adoptan esta manera de programación, usando librerías particulares, son Fortran o C.

c) Nuevos enfoque y modelos híbridos

Existen modelos que son la combinación de varios de los anteriores usando el paso de mensajes (MPI) junto con el modelo de *threads* (OpenMP), siendo un ejemplo el uso de MPI con la tecnología *hardware* GPU. Asimismo, en la actualidad existen nuevas ramas de gran interés en el mundo de la computación paralela, como el mapReduce y su implementación Hadoop [63] CUDA (*Compute Unified Device Architecture*) [64] y todo lo referente al high performance computing - HPC sobre *Cloud Computing* [65].

3.1.3 Java en Entornos Paralelos

Durante el desarrollo de esta investigación, además de realizar una evaluación sobre los temas relacionados con clústeres y paralelización, fue igualmente importante la revisión de herramientas de desarrollo centradas en aquellas basadas en Java (entre otras por la independencia de plataforma que Java ofrece), posibilitando la ejecución en diferentes sistemas operativos, máquinas etc. A conti-

nuación se muestran las herramientas revisadas, algunas testeadas y otras tan solo fueron revisadas a nivel teórico:

a) Datos Paralelos (Data Parallel)

Como se ha mencionado en apartados anteriores, el paradigma de datos paralelo es el de distribuir diferentes conjuntos de datos en diferentes unidades de cómputo y operar sobre ellos. Un ejemplo usando Java junto con este paradigma de programación es HPJava [66], que es una extensión del Java estándar implementando programación de datos paralelos añadiendo el concepto de matrices multidimensionales o *multiarrays*. Estos *arrays* distribuidos, junto con una sintaxis particular, permiten a los datos distribuirse a través de los procesos que se ejecutan en paralelo.

Otro desarrollo más actual que se encuentra en la literatura, es el enfoque propuesto por el proyecto Ateji PX, donde se extiende el lenguaje java con una nueva sintaxis similar a openmp en C/C++ aunque con mayores características y dejando de lado las limitaciones del uso de threads [67].

b) Paso de Mensajes

Éste es probablemente el paradigma más popular para la programación paralela. Los programadores de Java pueden utilizar algunos mecanismos de paso de mensajes de alto nivel como *Message Passing Interface* (MPI), *Remote Method Invocation* (RMI)[68] o *Common Object Request Broker Architecture* (CORBA) [69].

Existen dos implementaciones de MPI sobre Java, ambas son librerías que dotan al lenguaje de características MPI. Éstas son mpiJava [70] y MPJ [71]. La comunicación entre procesos se habilita mediante la introducción de nuevas clases actuando de forma similar a las clases utilizadas en otros lenguajes que usan MPI. Usando mpiJava o MPJ, el programador necesita supervisar cómo se hace la comunicación de datos inter-máquina, siendo esto una tarea complicada y propensa a errores como ocurre en otros lenguajes.

Java RMI es un modelo cliente/servidor basado en la invocación de procedimientos remotos, es decir, se habilita Java para utilizar métodos de un objeto que se

encuentra en otra máquina virtual de Java (JVM). Una aplicación diseñada usando este enfoque habilita a un servidor RMI para ofrecer objetos públicos de tal manera que clientes remotos invoquen estos métodos, distribuyendo carga de trabajo.

CORBA es una arquitectura abierta y no propietaria para interconectar diferentes aplicaciones distribuidas en una red. Con prestaciones similares a las de RMI, CORBA permite la interacción entre programas y aplicaciones Java con terceros fabricantes que sean compatibles con este estándar. Al igual que RMI, CORBA también es un modelo difícil de programación.

c) Memoria Compartida

El paradigma de memoria compartida supone la existencia de un único contenedor de datos, donde el programador no necesita especificar directivas de código adicionales para las operaciones con éste.

En los clústeres, al no poseer memoria física compartida, los nodos deben comunicarse a través de una interfaz de red. El interés de la investigación en el apoyo a la programación de memoria compartida utilizando Java en clústeres se ha incrementado. Una razón importante es el *multithreading* de Java frente a otros lenguajes como C que necesitan bibliotecas externas para apoyar *multithreading*. Java la incluye en su definición.

A pesar de su importancia, la programación en un entorno de clúster continúa siendo una tarea difícil debido a su naturaleza de memoria distribuida. El programador generalmente está acostumbrado al paradigma de memoria compartida de un sistema único, y se enfrenta a una empinada curva de aprendizaje al tener que gestionar múltiples espacios de memoria disjuntos. Si el *software* de *clustering* pudiese aligerar este problema, proporcionaría un entorno de programación muy útil para un gran número de desarrolladores de aplicaciones.

El *Distributed Java Virtual Machine* (DJVM) es un conjunto de instancias de JVM extendidas e interconectadas que abarcan varios nodos del clúster para permitir la ejecución paralela de una aplicación multiproceso Java, como si se estuviese ejecutando en una única pero poderosa máquina. A diferencia de RMI que está

incluido en el estándar de Java, los enfoques paralelos son extensiones o desarrollos nuevos que no pertenecen a dicho estándar.

Haciendo relación al concepto de sistema operativo basado en una imagen (SSI) revisado en apartados anteriores, DJVM es un enfoque transparente, destinado a ser un soporte de ejecución para Java, que se ejecuta en el clúster. Esta idea surgió a principios del año 2000. Existen diferentes tipos de prototipos:

- Java/DSM [72]
- JavaParty [73]
- cJVM [74]
- Jackal [75]
- Hyperion [76]
- JSJM[77]
- Kaffemik [78]
- J/Orchestra [79]
- JESSICA2[80]
- JESSICA3 [81]
- JESSICA4 [82]
- dJVM[83]
- JavaSplit, [84]
- Terracotta Clustered JVM [85]
- RuggedJ [86]
- Ateji [67]

Sin embargo, la mayoría de estos sistemas de JVM se encuentran todavía en fase de investigación, excepto Terracotta JVM o Ateji.

3.1.4 Sistemas paralelos Auto Ajustables

En los anteriores apartados se han mencionado las técnicas y métodos que se usan para desarrollar una aplicación paralela. Sin embargo el desarrollo de una aplicación de este tipo es un proceso complejo, donde el programador que op-

timiza el software lo realiza de forma manual, teniendo en cuenta muchos variables, donde al descuidar alguna, repercutirá en el rendimiento propio de la aplicación. El programador es entonces responsable de buscar la mejor combinación de parámetros, la consecuente transformación del código y de ajustar los parámetros del entorno de ejecución.

Hoy en día, los sistemas que pueden componer un entorno paralelo son habitualmente heterogéneos, por ello la optimización del código se convierte en una tarea compleja e interminable. El estado ideal sería, que los códigos que hasta ahora funcionan de manera adecuada en entornos secuenciales, sigan siendo válidos sobre los nuevos sistemas informáticos paralelos.

El interés de esta área de investigación, es el desarrollo de mecanismos donde los propios sistemas se encarguen del proceso de optimización software usando mecanismos de auto ajuste, evitando en general la intervención humana.

Las técnicas de auto-ajuste se pueden agrupar en las siguientes clases:

- Librerías específicas de auto ajuste para tratamiento de problemas de algebra lineal y proceso de señales como ATLAS, FFTW, OSKI y SPIRAL [87].
- Herramientas de análisis automático de código para mejoramiento en el tiempo de compilación [87];
- Sistemas auto-ajustables que buscan un conjunto de parámetros a nivel de aplicación que determinan el rendimiento de una aplicación. [88, 89].

3.1.5 Balanceo de Carga y Distribución de Cargas de Trabajo

Un problema relacionado con la computación paralela viene derivado de la disponibilidad de las unidades de proceso, por lo que es otro elemento a tener en cuenta en el diseño de una aplicación distribuida. Una correcta evaluación de esta disponibilidad facilita un preciso reparto de cargas de trabajo entre los nodos mientras se atienden las colas de procesos, junto con la constante verificación de la alta disponibilidad tanto en las comunicaciones como de los nodos. Esto se conoce como la calidad de servicio en el funcionamiento de un clúster [90].

La evaluación del conjunto de los diferentes elementos que definen esta calidad y disponibilidad de un clúster es un problema NP [91], constituido tanto por la velocidad de transmisión en la red donde están los nodos, como por el número de nodos y sus variables a tener en cuenta. Para la optimización de asignación dinámica de recursos y procesos en un clúster de ordenadores, han sido propuestos varios algoritmos relacionados con éste problema [90] .

Resource Management Scheduler RMS es un conjunto de algoritmos que ayudan a predecir el rendimiento y consumo de recursos de diferentes configuraciones de un clúster. El problema reside en predecir la capacidad de cómputo de un nodo antes de que éste reciba alguna carga de trabajo; la dificultad del problema aumentará si se tiene en cuenta las diferentes topologías de comunicación que puede tener un clúster.

La tarea básica del RMS es aceptar peticiones de recursos hechas por las aplicaciones y asignarles recursos de aquellos disponibles de un conjunto dado. Usando RMS en un clúster, todos los ordenadores pueden ser igualmente maestro o esclavo (o en otros términos, receptores y generadores de tareas), existiendo un nodo especial que asume el rol de orquestador o *super-master* donde se ejecuta el RMS.

El RMS recurre a una función matemática para predecir cómo será el funcionamiento del clúster. Esta función define cómo se distribuirá la carga de trabajo entre los diferentes componentes del clúster a cada instante. Las técnicas para pronosticar el rendimiento del sistema se basan en técnicas de teorías de colas o en técnicas históricas que recopilan los tiempos de respuesta, disponibilidad y retrasos en la comunicación entre los nodos, asignando un peso a estas variables, cada una con ventajas e inconvenientes, requiriendo tiempo adicional de CPU para hacer esta predicción. Ambas técnicas se pueden combinar para aprovechar sus ventajas.

3.2 Modelos para la Aceleración de la Dinámica de Metrópolis

Como se ha mencionado anteriormente, por lo general el modelo de Ising, Potts y GGH-CPM son simulados a través del método de Montecarlo usando la dinámica de Metrópolis. Sin embargo, este método experimenta deficiencias de rendimiento debidas a la ralentización del algoritmo cuando se acerca a la temperatura crítica.

En esta investigación, el interés de usar modelos basados en el modelo de Potts es obtener configuraciones que desarrollan dominios (o células). En ese caso, el coste computacional de una dinámica en la cual existe tan sólo un cambio o *flip* es muy alto. Esto se debe a que la probabilidad de cambio de los índices dentro del dominio (que son la mayoría) se vuelve muy pequeña, y más a temperaturas bajas.

Los índices con mayor probabilidad de cambio son aquellos que se encuentran en los límites del dominio, en donde principalmente tiene lugar la dinámica de relajación. Pero el número de índices en el límite de los dominios es mucho menor que los índices dentro del dominio. De modo que, en la práctica, la mayor parte de la simulación estará “perdiendo el tiempo” intentando modificar índices que no van a ser cambiados.

3.2.1 Modelos Q2R, Swendsen–Wang y Wolff

En la literatura existen diferentes esfuerzos para reducir este problema. Por un lado Swendsen [92] y Wolff [93] atacan la ralentización del algoritmo, usando también modelos estocásticos, pero realizando una aproximación no local para el cálculo de las configuraciones.

Estos algoritmos consideran regiones que se interconectan entre sí, pues comparten el mismo índice. A aquellos grupos que contienen índices del mismo valor

se les denominan clústeres y se unen a través del concepto de enlace o *bonds*. La actualización se realiza de manera más veloz, pues el algoritmo actualiza en un sólo momento todos aquellos índices que pertenecen a un clúster determinado.

El algoritmo Swendsen-Wang se resume a continuación:

1. Se inicializa la rejilla de $n \times n$ lugares con una configuración arbitraria.
2. Se examina cada par de índices contiguos en el sistema: si dos índices son paralelos (comparten el mismo índice), un enlace o *bond* se introduce entre ellos con probabilidad $p = 1 - e^{-k}$ donde $k = \frac{J}{k_B T}$. En caso contrario, no se realiza ninguna operación. Si $p < 1$, un número aleatorio r es generado tal que $0 < r < 1$. Si $r < p$ un enlace o *bond* es introducido entre un índice i y j .
3. Una vez que se han creado todos los grupos de clústeres en la rejilla, se elige un clúster arbitrariamente.
4. Se genera otro número aleatorio R , tal que $1 \leq R \leq q$, donde q es número de índice.
5. A todos los índices del clúster elegidos les es asignado el nuevo valor $\sigma_i = R$ calculado en el punto 4.
6. Otro clúster se selecciona al azar y se vuelve al punto 4.
7. Cuando todos los clústeres se han evaluado, se eliminan los enlaces y se regresa al punto 2. El proceso se repite tantas veces como se quiera hasta obtener el número de configuraciones deseado que permitan estimar los valores críticos en el modelo.

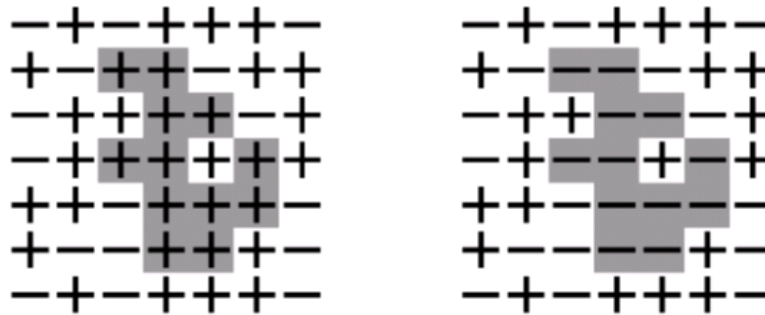


Figura 3.3. Enlaces o *Bonds* en los modelos de actualización tipo clúster para el modelo de Ising o Potts.

Por otro lado, Creutz [94] y Vichniac [95] proponen una simulación mediante autómatas celulares (esencialmente deterministas) para evitar la generación de números aleatorios, que son fuente de ralentización.

El autómata celular desarrollado por Vichniac fue denominado Q2R y se caracteriza por el hecho de que la rejilla se encuentra dividida en dos, como en una especie de tablero de ajedrez, en el cual cada sub-rejilla es actualizada alternativamente. La actualización sólo se da si no hay variación de energía al cambiar el valor del índice.

El autómata de Creutz desarrolla la idea en un "demonio" en una rejilla. El demonio es un "repositorio de energía", donde su tarea es desplazarse a través de la rejilla y visitar los índices facilitando su cambio y manteniendo constante la energía total del sistema. Para ello, absorbe (si puede) la energía cuando el cambio de un índice la produce y la aporta (si la tiene) en el caso de que el cambio la necesite. La temperatura del sistema se calcula a partir de la energía cinética media del mismo. El método sólo requiere números aleatorios al inicio de la simulación para determinar la configuración inicial de los *spines*.

Una segunda versión del autómata celular de Creutz [96] modifica su versión inicial haciendo un híbrido entre su modelo inicial y el modelo de Vichniac. Utiliza la forma de actualización de tablero de ajedrez de Vichniac y los demonios para mantener la energía del sistema. Estos demonios son de dos tipos: fijos uno en

cada índice o , en menor número, moviéndose al azar por la rejilla. Si se omiten los demonios, este algoritmo se reduce al autómata celular de Q2R como caso especial.

3.2.2 Descomposición por Dominios

El DDM (métodos de descomposición por dominios) es una rama de la matemática relacionada con el análisis numérico que se extiende y aplica a la informática, donde su objetivo es resolver problemas mediante la división de un problema o dominio en partes más pequeñas o sub dominios. Desde el punto de vista de la teoría de grafos y aplicando éste a la distribución y balanceo de trabajos en un clúster, se puede expresar con el grafo balanceado, donde sus vértices representan los procesos computacionales y sus líneas representan la comunicación entre procesos dependientes [97].

Uno de los primeros pasos en el diseño de un programa paralelo es conocer la manera de romper el problema en porciones más pequeñas de trabajo (comúnmente llamados dominios) que se pueden distribuir en múltiples unidades de proceso. Esta acción recibe el nombre de "descomposición" o "partición".

Hay dos formas de particiones para distribuir procesos en unidades de computación: la descomposición de dominio y la descomposición funcional.

- **Descomposición de dominio o paralelismo de datos:** En este enfoque los datos son divididos en partes aproximadamente del mismo tamaño y son asignadas a diferentes procesadores. Cada procesador trabaja tan sólo con el paquete de información que le es asignado, y en ciertos momentos los procesos se sincronizan para intercambiar sus resultados y recibir más datos para continuar con el proceso. Este enfoque permite mantener un flujo único de control.

- **Descomposición funcional o paralelismo de tareas:** Este enfoque busca descomponer las operaciones que se realizan sobre los datos, donde un problema complejo se divide en un gran número de partes más pequeñas y a su vez asignadas a los procesadores disponibles, de tal manera que la tarea global será resuelta una vez todas las sub-tareas hayan concluido. Esta aproximación implica un cuidadoso diseño para que se produzca una máxima paralelización, pues de lo contrario puede darse el caso de cuellos de botella o abrazo mortal.

3.2.3 Algoritmo del Tablero de Ajedrez

El enfoque estándar para paralelizar la familia de modelos de Ising a nivel de procesamiento informático sigue la estrategia del tablero de ajedrez, que es una manera de descomposición de dominios, propuesta por [25] para la evaluación del cambio de índice.

Esta forma es necesaria por la propia naturaleza del método de los algoritmos de Ising/Potts, pues parecería a priori sencillo pensar que la distribución de porciones de cálculo podría distribuirse en tantas unidades de procesamiento como se quisiesen. Sin embargo, esta opción no es posible, ya que por ejemplo dos o más procesadores que ejecutaran simultáneamente la evaluación de cambio de índice podrían encontrarse evaluando índices que sean adyacentes dentro de la rejilla. Si esto ocurriese, entonces se estaría evaluando el cambio utilizando información incorrecta acerca de los índices y de los vecinos. Eso estaría violando el principio del "balance detallado" que exige que dos (o más) sitios no se puedan cambiar al mismo tiempo, si estos interactúan entre sí a través de sus respectivos *hamiltonianos*.

Una alternativa es dividir la rejilla en porciones o subdominios menores (en cuatro partes en 2D y ocho en 3D), marcándolas con diferentes colores, usando una pequeña zona de superposición o zona fantasma para cada procesador. En dicha zona se garantiza que cada procesador tendrá la información necesaria de los vecinos próximos para poder hacer la evaluación del cambio de índice. La eva-

luación puede darse de manera independiente sin necesidad de información de las otras sub-rejillas.

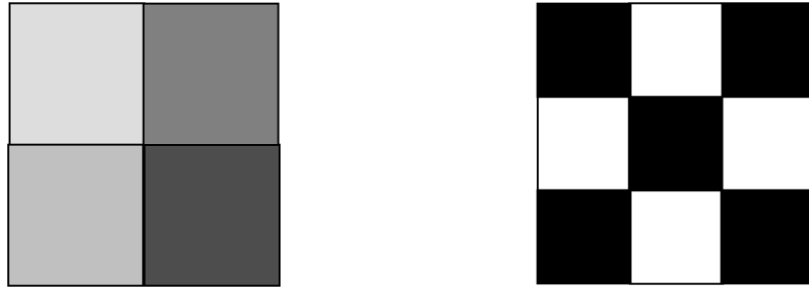


Figura 3.4. Rejilla en forma de tablero de ajedrez para algoritmo de Ising/Potts

Una vez dividida la rejilla principal, se desarrolla en cada procesador una dinámica del algoritmo de Metrópolis-Montecarlo, actuando de manera alterna. El procesamiento se inicia en aquellas rejillas de colores iguales, mientras permanecen a la espera aquellas donde los colores son diferentes. Después de cada paso de Montecarlo, las sub-rejillas se sincronizan y se inicia la evaluación en los procesadores que se encontraban a la espera en aquellas sub-rejillas de color diferente.

El enfoque de "tablero de ajedrez" permite evaluar índices a gran velocidad dentro de la rejilla frente al algoritmo secuencial visto en el capítulo anterior. Chen et al. [98] han seguido este enfoque para la paralelización del modelo CPM-GGH, donde se muestra la relación entre las sub-rejillas, alternancia entre sub-rejillas y eficiencia de la implementación en un clúster. Para obtener otros ejemplos de este método se puede consultar [99, 100]. Aunque hasta ahora se han propuesto varias aproximaciones, nadie ha planteado una versión este método totalmente optimizado para CPM-GGH.

3.2.4 Algoritmos "Rejection-Free"

Se presenta como alternativa a los métodos ya mencionados otro grupo de métodos de aceleración, encontrados en la literatura y denominados "*Rejection-Free*" [101]. En el caso del modelo Metrópolis-Montecarlo, no siempre tienen

éxito la evaluación o los ensayos de cambio de índice. En cambio, en los algoritmos de este tipo las evaluaciones se realizan de otra manera y casi todas tienen éxito. Así, mientras que en el Modelo de Metrópolis se pasa mucho más tiempo calculando posibles cambios sin éxito, los modelos "*Rejection-free*" aportan ganancias que pueden ser miles o incluso millones de veces más rápidas.

Algunos algoritmos que se enmarcan en este grupo son el N-fold way y el Montecarlo Cinético, siendo especialmente atractivos [32], pues facilitan la computación paralela y por tal razón fueron revisados en el estado del arte de esta tesis.

3.2.5 Algoritmo de "N-FOLD WAY"

Una manera de evitar el problema de eficiencia y poder así alcanzar escalas de tiempos mejores hace necesario el uso de métodos alternativos.

Los algoritmos que se han revisado hasta ahora son métodos que evolucionan paso a paso siguiendo el tic-tac de un reloj imaginario. Un enfoque diferente diseñado por Bortz, Kalos y Lebowitz en 1975 [102] y revisado por Novotny en 1995 [103] introduce un método alternativo conocido como "*n-fold way method*", donde n hace referencia al número de posibles cambios energéticos diferentes para un paso de Montecarlo. El tiempo que transcurre en la evolución del modelo también se calcula de manera distinta.

Este método fue aplicado al modelo de Ising pero puede ser usado para otros sistemas basados en rejilla y con las mismas reglas que éste. En la literatura de este método a veces se conoce como "Kinetic Monte Carlo" [32].

El algoritmo se inicia basándose en la siguiente observación: los *spines* pueden pertenecer o clasificarse dentro de un conjunto y así durante toda la evolución de la simulación. Estos conjuntos son un número n de "clases" de *spines* o índices, discriminados de acuerdo a su probabilidad de cambio. El modelo reconoce que existen pequeños micro-entornos y con ellos un limitado número de diferen-

tes probabilidades de realizar el cambio. Así, los *spines* que conforman el sistema se agrupan con valores energéticos idénticos.

Para el caso del modelo de Ising, en una rejilla cuadrada donde los *spines* asumen $\sigma = +1$ o $\sigma = -1$ existen 10 clases o micro-entornos. Una vez clasificados los *spines* de la rejilla se pasa a calcular el ΔH de cada una de las clases. La probabilidad total de cualquier clase l puede ser cambiada en un paso, definiéndose como:

$$p_l = n_l e^{\frac{-\Delta H_l}{k_b T}} \quad (3-1)$$

Donde n_l es el número de *spines* que pertenecen a la clase l . La probabilidad total de que ocurra cualquier cambio en un paso dado se define como:

$$Q_m = \sum_{l \leq M} p_l \quad (3-2)$$

Y Q_n es la probabilidad total para todas las N Clases.

En el modelo de Ising usando *n-fold way*, las 10 clases o grupos de *spines* se enumeran como se muestra en la siguiente tabla:

Clase	Spin	Número de <i>spines</i> Alineados hacia Arriba
1	ARRIBA	4
2	ARRIBA	3
3	ARRIBA	2
4	ARRIBA	1
5	ARRIBA	0
6	ABAJO	4
7	ABAJO	3
8	ABAJO	2
9	ABAJO	1
10	ABAJO	0

Tabla 3.1 Diferentes Clases Modelo N-Fold Way

Mientras que en el algoritmo de Metrópolis se elige un *spin* a la vez entre todos los lugares de la rejilla con igual probabilidad, en cambio en el *n-fold way* se elige un *spin* entre los sitios de la rejilla usando una probabilidad ponderada, de tal manera que la probabilidad de elegir un *spin* particular es proporcional a su probabilidad de cambio. Una vez que el *spin* es elegido, el cambio se realiza inmediatamente.

El procedimiento se resume a continuación:

Una vez clasificada la rejilla en clases, se genera un primer número aleatorio sobre el intervalo $[0, 1]$ que permita escoger la clase en la cual el próximo *spin* será cambiado. Seleccionada la clase, un segundo número aleatorio se genera para elegir un *spin* entre los diferentes *spines* que pertenecen a la clase seleccionada anteriormente. Por último, un tercer número aleatorio se utiliza para determinar la cantidad de tiempo que ha transcurrido antes de que este evento haya tenido lugar.

Ahora bien, en el método de Metrópolis existe una unidad de tiempo llamada paso de Montecarlo definido por lo general como $n_x n$ intentos de cambio, donde n es el número total de *spines* en la rejilla. Pero n iteraciones de la *n-fold way* no son equivalentes y por lo tanto no constituyen un paso de Monte Carlo. Por ello es necesario ajustar el paso temporal que involucra cada cambio.

Por esto en la *n-fold way* ocurre un cambio de *spin* ineludiblemente en cada iteración, y el tiempo que el suceso de ese evento hubiese demandado en un algoritmo de Monte Carlo estándar se calcula a partir de la probabilidad global de cambio del *spin* asociada a la configuración instantánea del sistema. Este cálculo se realiza después de cada cambio de índice y el resultado se va acumulando al tiempo transcurrido, definiéndose así:

$$\Delta t = -\frac{N \ln r}{Q_n} \quad (3-3)$$

Estos incrementos temporales son en general un número real donde la variable Δt se incrementa en fracciones.

Cada vez que un *spin* es intercambiado éste cambia de clase. Entonces el *spin* debe ser eliminado de la clase a la cual pertenecía, añadiéndolo a la nueva lista correspondiente. De este modo la totalidad de sus vecinos próximos también cambiará de clase y se deberá entonces actualizar toda la lista de *spines* y clases. La clave del algoritmo *N-fold* es el control y la actualización de las listas durante toda la simulación

- Lo esencial es invisible a los ojos.
-Lo esencial es invisible a los ojos -repitió el principito, a fin de acordarse

Antoine de Saint-Exupéry. – El Principito



Capitulo 4 La Piel

Una célula es la unidad funcional de todos los organismos vivos. Las células se dividen en dos grandes grupos: por un lado las procariotas (sistemas celulares más sencillos) que contienen organismos unicelulares como bacterias o archaeas. En un segundo grupo están las células de tipo eucariotas, que conforman la mayoría de los seres vivos. Estos últimos son extremadamente complejos y altamente estructurados, similares en el nivel básico a las diferentes especies.

Cada célula está compuesta de varios elementos, principalmente de la membrana celular, citoplasma y núcleo. La membrana celular está compuesta en su mayoría por lípidos y proteínas, y delimita a la célula de su entorno. El citoplasma contiene orgánulos que sintetizan proteínas y proporcionan la energía necesaria para que la célula pueda funcionar. El núcleo contiene el mapa genético de la célula en forma de ADN, el cual se traspassada de una generación a otra.

En la naturaleza, las células se dividen, diferencian y especializan en el proceso de morfogénesis para dar forma a un individuo maduro y viable. En un organismo adulto, se suceden procesos similares que ayudan a mantener el equilibrio de la renovación celular y a procesos como el de la curación de heridas. Estas similitudes, presentes en la mayoría de los animales, permiten extrapolar y esperar una comprensión general de los principios comunes en el funcionamiento de la célula, a pesar de que existan mecanismos propios codificados en los programas genéticos de cada una de las especies.

La organización de varias células permite expresar patrones espaciales y una estructura particular que tendrá cada tejido. Aunque los genes son importantes para la formación de dichos patrones, la genética no dice nada sobre qué mecanismos debe seguir cada célula para que se origine un patrón concreto.

En el programa genético ya se encuentra definido como van a ser las células, pero no especifica con exactitud dónde y cómo. Cada célula debe moverse dentro del organismo.

Los genes determinan la expresión de moléculas específicas en las células que conducen a ciertas propiedades físicas, como la viscosidad del citoplasma o de la adhesión de la membrana celular. Las diferentes expresiones de estas moléculas controlan la naturaleza de cada propiedad celular en particular. Junto con esto, las fuerzas físicas y procesos dinámicos como la gravedad, la adhesión y la tensión superficial determinarán cómo estas propiedades van a interactuar y conducir la formación de patrones determinando el movimiento y la forma del tejido.

A partir de cómo se desarrollan estas agrupaciones "naturales" de las células, se expresan y dan lugar a muchas de las propiedades de los tejidos. Los cambios en las propiedades de estas agrupaciones, tales como cambios en la adhesión durante la embriogénesis o en el cierre de heridas, pueden ocasionar patologías o procesos de desorden tales como el cáncer [104, 105].

En este capítulo se presentan algunos antecedentes biológicos a nivel celular, de tejido epidérmico y de procesos biológicos necesarios para entender el trabajo presentado en la tesis.

4.1 La Célula

La célula es la unidad fundamental de toda la materia viva. Existen dos tipos de células, las eucariotas y las procariotas. Estas últimas se diferencian de las primeras principalmente por poseer núcleo. La célula es una entidad separada de otras células y del medio por una pared o membrana, que contiene en su interior diversas estructuras, algunas de las cuales son comunes a todas las células. Las células vegetales difieren de las células animales en poseer una pared celular de celulosa que le da rigidez estructural, cloroplastos (orgánulos capaces de sintetizar azúcares a partir de dióxido de carbono, agua y luz solar en el proceso de fotosíntesis) y una vacuola llena de líquido que ocupa casi todo el interior de la célula vegetal.

A continuación, se hace una breve descripción de los principales componentes de la célula animal (Figura 4.1). Para una descripción extensa se invita al lector a consultar [106].

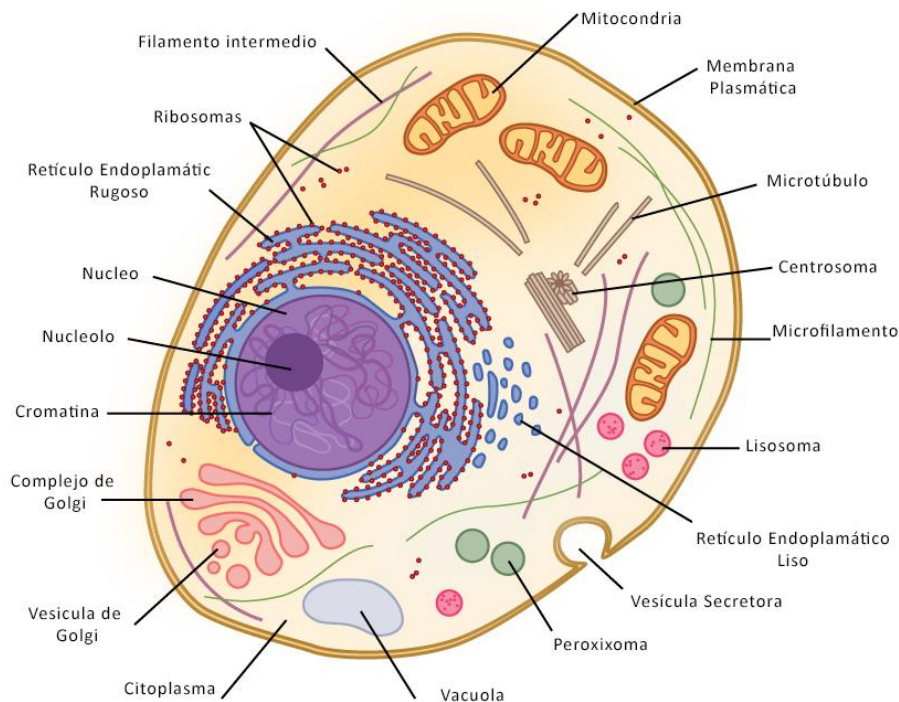


Figura 4.1. Célula Animal

4.1.1 Membrana Celular

La membrana celular es una estructura muy especializada que desarrolla tareas varias en la célula. Recubre y delimita a la célula manteniendo un balance entre el citoplasma y el entorno extracelular. Posee un grosor aproximado de entre 4 y 7,5 nm y está formada principalmente por fosfolípidos, colesterol, glúcidos y proteínas.

La membrana está compuesta por dos láminas que sirven de contenedor para el citosol o hialoplasma (parte soluble del citoplasma) y los distintos compartimentos internos de la célula. La fluidez o viscosidad de esta bicapa lipídica depende de su composición y de la temperatura. La estructura de la membrana celular está descrita habitualmente bajo al modelo de mosaico fluido[107] . El término fluido

es debido a que moléculas que allí se encuentran pueden moverse libremente a través de las bicapas, exponiendo diferentes tipos de desplazamientos como de rotación alrededor de su eje, de difusión lateral, de *flip-flop* (movimiento entre capas) y de flexión.

El rasgo principal de esta barrera es su permeabilidad selectiva que actúa como filtro de materiales y evita el escape de los metabolitos o iones.

Posee la capacidad de romperse y fusionarse de nuevo sin perder su organización, de tal manera que es una estructura flexible y elástica que se adapta a las necesidades de la célula. Está en permanente renovación, en la que se adicionan y eliminan moléculas permitiendo su adaptación a las necesidades fisiológicas de la célula.

Entre otras funciones importantes de esta barrera, tiene las de comunicación e interacción con otras células y el reconocimiento de señales extracelulares.

4.1.2 Orgánulos Intracelulares

Las células eucariotas contienen membranas intracelulares u orgánulos necesarios para el mantenimiento de la actividad vital de las células, constituyendo más de la mitad del volumen total de esta. Los principales tipos de orgánulos que están presentes en todas las células eucariotas son el retículo endoplásmico, el aparato de Golgi, el núcleo, las mitocondrias y varios tipos de vesículas.

- **El retículo endoplasmático.** Orgánulo distribuido por todo el citoplasma constituyendo la mayor parte de este, representando entre el 50 y 70% de las membranas internas. Existen dos tipos de retículos: el retículo endoplasmático liso y el retículo endoplasmático rugoso. Forman un conjunto de sáculos aplanados, sáculos globosos y túbulos sinuosos que se extienden por todo el citoplasma y comunican con la membrana nuclear externa. Su principal función es la síntesis y el transporte de la mayoría de los lípidos y la participación en procesos de detoxificación de la célula.

- **El aparato de Golgi.** Representa entre 5 y 10% del total de las endomembranas. Es un grupo de sáculos apilados, unido a la membrana celular, implicado en la transformación y clasificación de macromoléculas recibidas del retículo endoplasmático, para ser secretadas o para ser enviadas a otros orgánulos.
- **El núcleo.** Orgánulo propio de los organismos eucariontes y lugar donde se almacena el ADN. Está conformado por una membrana o envoltura nuclear (la cromatina) y por un nucléolo. La función del núcleo es mantener la integridad de esos genes y controlar las actividades celulares regulando la expresión génica.

Aunque en el interior del núcleo su contenido no es uniforme, existe una cierta cantidad de cuerpos subnucleares compuestos por tipos exclusivos de proteínas, moléculas de ARN y segmentos particulares de los cromosomas.

4.1.3 Citoesqueleto

El citoesqueleto es un componente presente en todas las células eucariotas, formando una estructura que controla la ubicación espacial de proteínas y orgánulos, proporcionando vías de comunicación entre éstos. La composición química se basa en un entramado de fibras de proteína que permite desarrollar microfilamentos, filamentos intermedios y microtúbulos descritos a continuación:

- **Microtúbulos.** Cilindros largos, relativamente rígidos y huecos compuestos de tubulina. Los microtúbulos tienen diferentes funciones, entre ellas establecer la disposición espacial de determinados orgánulos, formar un sistema de caminos mediante el cual se pueden transportar vesículas o macromoléculas entre compartimentos celulares. Cada microtúbulo es una estructura altamente dinámica que cambia por adición y pérdida de tubulina durante los ciclos de polimerización - depolimerización.

- **Filamentos intermedios.** Componentes formados por agrupaciones de proteínas fibrosas con un diámetro de alrededor de 10 nm. Son polímeros de una familia heterogénea de polipéptidos fibrosos. Estos filamentos resisten al estiramiento y desempeñan un papel estructural o de soporte en la célula.
- **Microfilamentos.** Filamentos mayoritariamente de actina que constituyen el córtex, el cual bajo la membrana plasmática da forma y elasticidad a la célula. Facilitan la emisión de prolongaciones citoplasmáticas o pseudópodos.

4.1.4 Adhesión Celular

De todas las interacciones que tienen las células en un organismo multicelular, una de las más relevantes son los procesos que determinan que las células se unan y mantengan juntas. En la mayoría de los organismos pluricelulares, las células se organizan en grupos denominados tejidos, que a su vez se asocian formando grandes unidades funcionales denominadas órganos.

Los mecanismos de cohesión que rigen la arquitectura del cuerpo, su forma, su resistencia y la disposición de los diferentes tipos de células están gobernados por los materiales extracelulares que éstas segregan. La adhesión juega un papel clave en la determinación del movimiento celular y es crucial en el crecimiento de los embriones en desarrollo [108], la gastrulación, la formación de órganos, y también en la adultez en procesos como, por ejemplo, la cicatrización de heridas. Los defectos en este proceso acarrearán una enorme variedad de patologías (la metástasis en el cáncer).

La adhesión facilita la unión de células a la matriz extracelular o a otras células, desarrollando diferentes tipos de uniones: de oclusión o estancas, de anclaje y de comunicación. Estas se hacen mediante moléculas de adhesión como las integrinas, cadherinas, selectinas o las inmunoglobulinas, sirviendo como anclaje y de mecanismo de fuerzas de compresión o de estiramientos para las células.

4.1.5 Reordenamiento Celular “Cell Sorting”

La comprensión de los factores que influyen y que permiten la organización tisular durante el desarrollo es uno de los objetivos principales de la biología del desarrollo. El reordenamiento celular es el proceso por el cual grupos de células desorganizadas establecen un orden estructurado. El reordenamiento sucede cuando dos o más células de diferentes tipos pasan a formar dominios homogéneos desde una configuración.

En consecuencia, este reordenamiento de células ha sido objeto de una gran cantidad de especulación teórica y de una variedad de hipótesis propuestas para determinar cuáles son las propiedades que deben tener celulares, y que puedan causar dicha organización y asimismo permitan predecir si un tipo de célula concreto va a clasificarse internamente o externamente envolviendo a otra y posibilitando el desarrollo de un tejido [39, 105].

Steinberg demostró que el comportamiento de los grupos de células en tejidos embrionarios aislados no se ordenan tan sólo con procesos en donde las células se mueven de acuerdo a diferentes niveles de concentración de alguna sustancia (quimiotaxis), sino que para que se produjese existían más elementos, y que la organización era más semejante al comportamiento que tienen diferentes tipos de líquidos en un recipiente. Steinberg, a partir de ello, formula la hipótesis de adhesión diferencial (DAH)[40, 109] y con ello explica el modo en el que las propiedades de la célula pueden operar para determinar cómo se alinean las células. El DAH propone:

- Las células de un mismo tipo tienen las mismas características de resistencia y adhesión frente a células de diferente tipo.
- Las células desarrollan movimiento para poder conformar un dominio o agregado de células.
- La organización final de un conjunto de células es producto derivado de la configuración que expone la menor energía libre interfacial.

Estas propuestas posibilitan el andamiaje para la formación o reconstrucción de un órgano. Los tejidos de muchos organismos presentan este comportamiento,

lo cual demuestra que es un proceso generalizable e independiente de genéticas específicas de las células.

4.1.6 Ciclo Celular

La única manera de que exista una nueva célula es a través de una ya existente. Todos los organismos vivos, desde una bacteria a un mamífero multicelular, son producto de un ciclo continuo y repetido de rondas de crecimiento y división celular.

Una célula se reproduce mediante la realización de una secuencia ordenada de eventos en los que se duplica su contenido y luego se divide en dos. Este ciclo de duplicación y división es conocido como el ciclo celular, y es el mecanismo esencial por el cual todos los seres vivos pueden reproducirse y mantenerse.

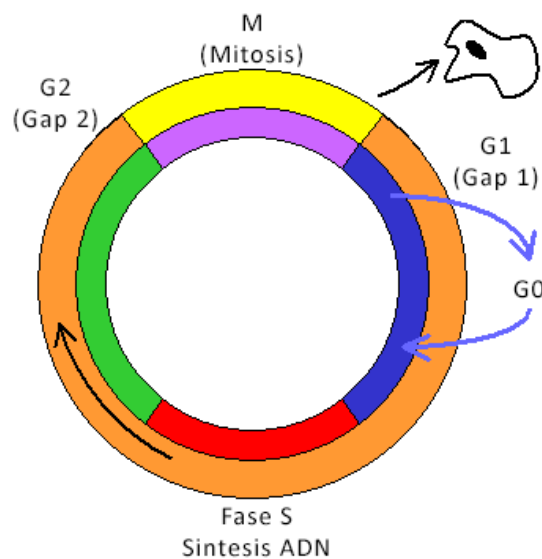


Figura 4.2. Ciclo Celular

Independiente de los procesos que se realicen en el interior de una célula al dividirse, la tarea principal es la transmisión de la información genética a la siguiente generación; razón por la cual, las futuras dos "hijas" deben ser genéticamente idénticas. El ADN de cada cromosoma se debe replicar fielmente para producir dos copias completas y, una vez listo, los cromosomas replicados deben ser distribuidos a las dos células "hijas", de tal manera que cada una recibe una copia de todo el genoma.

El ciclo celular de una célula eucariota se divide en cuatro fases diferentes tal como lo muestra la Figura 4.2:

- Fase G₁ (Gap 1): primera parte del ciclo, en la que la célula experimenta crecimiento y se inicia la síntesis de proteínas y de ARN.
- Fase S (Síntesis): se produce la replicación o síntesis del ADN, dando como resultado que los cromosomas se dupliquen, formándose dos cromátidas iguales.
- Fase G₂ (Gap 2): en esta fase de crecimiento se continúa la síntesis de proteínas y del ARN.
- Fase M (mitosis): se sucede la división celular, donde la célula progenitora se divide en dos células hijas idénticas.

La fase M o mitosis se divide en cuatro sub-etapas:

- Profase: se produce la condensación del material genético, formándose los cromosomas junto con el desarrollo bipolar del huso mitótico. La actividad metabólica se reduce.
- Metafase: los cromosomas condensados comienzan a moverse continuamente, hasta que migran a la zona media de la célula.
- Anafase: los cromosomas duplicados se separan en dos partes idénticas y se mueven hacia los polos opuestos.
- Telofase: los nuevos cromosomas se agrupan en los polos opuestos, la actividad metabólica se restaura y la envoltura nuclear se reconstruye.

El ciclo celular termina con la citocinesis en donde la célula se divide en dos, cada una con las mismas copias de material genético. La duración del ciclo celular total varía entre especies y entre diferentes tipo de tejidos.

Durante el ciclo, la célula tiene mecanismos de vigilancia que controlan los cambios que se están sucediendo, y aseguran que los procesos cruciales como el de replicación y división se realicen con alta fidelidad. Con los puntos de chequeo se asegura que se produzcan ciertos acontecimientos para que el ciclo continúe o se aborte. Los puntos de control también supervisan las condiciones del entorno de la célula, teniendo en cuenta las señales de otras células y estimulando en el caso de que haya mayor división celular cuando se necesitan más número de células.

Los puntos de control no sólo detectan daños en el ADN, sino que también forman parte de la red de múltiples mecanismos de reparación que detienen la progresión del ciclo celular para que se pueda aplicar algún correctivo, iniciando también procesos implicados en la reparación. Si es irrecuperable, se inicia la apoptosis o muerte celular programada, que previene que la célula defectuosa dañe a todo el organismo cuando el daño es demasiado grave.

4.2 Estructura de la Piel

La piel es un órgano complejo que protege al organismo del entorno y al mismo tiempo permite la interacción con el medio ambiente. Es una barrera casi impenetrable contra los diferentes elementos externos a los que el organismo puede verse expuesto. La auto-renovación de la piel es provista por varios mecanismos que mantienen la estructura del tejido y el equilibrio constante entre la renovación y la muerte celular.

Entre las funciones de la epidermis pueden mencionarse [110]:

- Barrera protectora contra las lesiones mecánicas, térmicas, físicas y de sustancias peligrosas gracias a las células epiteliales escamosas estratificadas y queratinizadas.
- Medio para evitar la pérdida de humedad gracias a la barrera formada por aminoácidos, esteroides y fosfolípidos.
- Agente reductor de los efectos nocivos de la radiación ultravioleta.
- Órgano sensorial, receptor y transmisor de señales a otros organismos, puesto que en la piel se encuentran múltiples receptores sensitivos somáticos que detectan estímulos, originando sensaciones generales o somáticas (presión, tacto, temperatura, dolor y vibración).
- Regulador de la temperatura, asegurando que el organismo mantenga un adecuado funcionamiento de la perfusión dérmica, aumentando o disminuyendo la vasoconstricción de los vasos sanguíneos.
- Excretor mediante la regulación del volumen y del contenido químico del sudor.

- Inmunidad: a través de células especializadas que fijan y destruyen microorganismos patógenos.

Estas distintas funciones anteriores son permitidas por uno o varios de los segmentos en la que está constituida (Figura 4.3). La piel está formada por un tejido epitelial (la epidermis) y uno conjuntivo especializado (la dermis). Debajo de estas dos, se encuentra el tejido subcutáneo o hipodermis, otra variedad de tejido conjuntivo especializado en el que predominan los adipocitos.

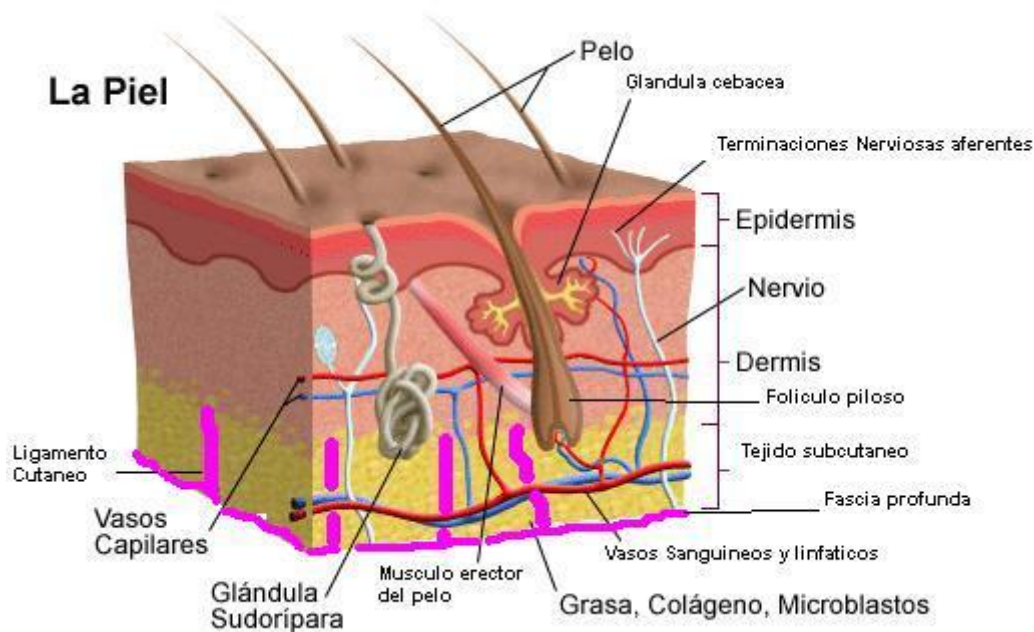


Figura 4.3. Estructura de la piel

4.2.1 La Epidermis

Es la parte más externa y visible de la piel, siendo un tejido epitelial, plano, estratificado y queratinizado, que se forma a partir del ectodermo embrionario (superficie externa del embrión). Es una estructura que se renueva continuamente junto con diferentes estructuras o apéndices que también se encuentran en uñas, glándulas sudoríparas o unidades pilosebáceas. La mayoría de las células que se encuentran en la epidermis son queratinocitos, constituyendo un 80% del tejido epidérmico. Los queratinocitos, en su viaje desde las capas más pro-

fundas a las más superiores, se someten a un proceso de diferenciación, de cambios estructurales y de morfología [111].

La epidermis se divide en varias capas de acuerdo a la diferenciación de los queratinocitos y en función de su ubicación dentro del tejido.

4.2.1.1 Tipos de Células en la Epidermis

Junto con los queratinocitos, se pueden hallar otras células en la epidermis: melanocitos, células de Langerhans y células de Merkel, que se encuentran esparcidas entre los queratinocitos.

Queratinocitos. Tipo de célula con mayor número de apariciones dentro del tejido, encontrándose el 99% de las células en esta zona. Estas células derivan del ectodermo embrionario.

Melanocitos. Células de citoplasma claro y núcleo pequeño. Se encuentran conformando el estrato basal junto con los queratinocitos, con una relación aproximada de un melanocito por cada diez queratinocitos [112]. Los melanocitos producen melanina, pigmento que absorbe la radiación ultravioleta y protege a los queratinocitos, evitando que ésta llegue al núcleo y produzca daños irreversibles en el ADN y proporcionando fotoprotección y termoprotección a las células vecinas.

Células de Langerhans. Células dendríticas que carecen de melanina, derivándose de células macrófagas u otras células mesenquimales. Están presentes en todas las capas de la epidermis, pero en especial en la capa espinosa, que se presenta más adelante. También están presentes en la dermis superior alrededor de los vasos sanguíneos pequeños. Su función no está muy clara, puesto que pueden tener un papel en el control de la queratinización o participar en el control y mantenimiento de las células de la epidermis. Pero se cree [113] que su función primordial se relaciona con el sistema inmunológico/antígeno cutáneo colaborando con las células T para desarrollar reacciones inmunológicas, constituyendo la primera barrera defensiva del organismo

Células de Merkel. Asociadas con terminaciones neuronales, contienen catecolaminas. No se conoce con claridad su función, pudiendo ser receptores sensoriales, transductores o mecanorreceptores. Parece que estas células desempeñan funciones sensoriales al estar asociadas a terminaciones nerviosas, presentes al otro lado de la membrana basal, siendo receptores del sistema nervioso. Sin embargo, todavía no se ha podido demostrar que cedan neurotransmisores.

Por tanto, la epidermis consiste en una serie de células ordenadas en estratos, con distintas características según sea su nivel de diferenciación. Es decir, en función del grado de diferenciación celular, dentro de la epidermis se pueden distinguir varios estratos.

4.2.1.2 Capa Basal

También conocida como estrato basal, esta capa es la responsable de la regeneración constante de las otras capas de la epidermis, producto de la repetida división mitótica. De entre todas las capas que conforman la epidermis, ésta es en la única donde las células pueden dividirse. Se encuentra en el nivel inferior de la epidermis limitando con la dermis. Normalmente comprende una sola fila de queratinocitos. La proliferación de células en la capa basal depende de la membrana basal, que es rica en proteínas y factores de crecimiento [111].

Las células que constituyen este estrato son indiferenciables ya que carecen de los marcadores bioquímicos característicos de las células diferenciadas de las capas superiores y su proliferación está regulada por diversos factores. Entre los reguladores se destacan el factor de crecimiento epidérmico (EGF), los factores de crecimiento tumoral, progesterona, epinefrina y vitaminas A y D.

Tras su proliferación, las células del estrato basal ascienden para situarse en una posición suprabasal: pierden su capacidad proliferativa e inician el complejo proceso de diferenciación para dar origen al estrato espinoso.

La epidermis nueva es formada constantemente por el proceso de división de las células madre de esta capa, exponiendo dos estrategias distintas de división ce-

lular para mantener el equilibrio entre las células madre y su progenie diferenciada (división celular asimétrica y división celular simétrica).

Las células de la lámina basal están unidas a la membrana basal por medio de hemidesmosomas que permiten el perfecto anclaje [114]. Cuando la unión entre las células de la lámina basal y las células de la membrana basal es débil o no es apropiada, se producen enfermedades cutáneas crónicas.

4.2.1.3 Estrato Espinoso

Esta capa debe su nombre a la morfología que poseen las células, otorgada por la existencia de numerosas estructuras proteicas que tienen las células llamadas desmosomas, las cuales son diversas moléculas de adhesión tomando la forma de espinas. Las células espinosas son poliédricas con un núcleo redondeado, ricas en ARN, que facilitan la síntesis de queratina (proteína).

A medida que las células se diferencian y se mueven hacia arriba a través de la epidermis, se van transformando, tornándose más planas y desarrollando progresivamente orgánulos denominados gránulos laminares. Las células espinosas contienen filamentos de queratina organizados en torno al núcleo.

La interacción de las cadherinas en la matriz extra-celular, junto con el calcio, garantiza la adherencia selectiva de células que poseen el mismo nivel de diferenciación, promoviendo la adhesión de las células epidérmicas y la resistencia al estrés mecánico.

4.2.1.4 Estrato Granuloso

La capa granulosa es la zona de transición entre las capas inferiores de queratinocitos "vivos" y capas superiores sin núcleo "inertes". Las células en esta capa desarrollan gránulos de queratohialinos basófilos (orgánulos que han sufrido una destrucción enzimática parcial), y a esto debe su nombre la capa. Además se desarrollan diferentes componentes estructurales que permitirán desarrollar una barrera epidérmica. Esta estructura contiene proteínas insolubles no queratínicas, que permiten identificar a las células como pertenecientes a este estrato. Aunque no se conocen todos los componentes proteicos que se integran en los

gránulos de queratohialina, algunos de ellos son profilagrina, involucrina, oricina, queratolinina [115].

Cuando las células epidérmicas entran en la última fase de su proceso de diferenciación se llenan de lamelares de naturaleza lipídica. Es en este estrato donde los núcleos se reabsorben y se pierden, este proceso se puede definir como una forma especial de apoptosis. Poco a poco, el espacio citoplasmático de la célula granulosa se transforma en un conjunto de proteínas de fibras, queratina y filagrina. La célula del estrato granuloso se transforma en un corneocito a través de un proceso complejo de transformación cualitativa y cuantitativa de los componentes proteicos y lipídicos.

4.2.1.5 Estrato Córneo

El estrato córneo es la capa más externa de la epidermis, donde la diferenciación es completa. Está constituida por varias capas de células escamosas sin núcleo ni orgánulos. Se encuentran células con forma hexagonal, aplanadas y cornificadas, suspendidas en una matriz extra celular de lípidos. Mantienen los filamentos de queratina y la matriz de filagrina, que junto a los fosfolípidos sintetizados en los cuerpos multilaminares forman una barrera impermeable y ofrecen protección mecánica [115].

Los queratinocitos con diferenciación total se denominan corneocitos, y son los únicos constituyentes de esta capa.

El espesor del estrato córneo varía de una zona a otra dentro del cuerpo humano, siendo más grueso en zonas muy expuestas a rozamiento como, por ejemplo, palmas y plantas de los pies.

El estrato corneo proporciona protección mecánica a la piel sirviendo de barrera para evitar la pérdida de agua y la penetración de las sustancias solubles del medio ambiente [116] .

4.2.2 Dermis

La dermis es la zona ubicada entre la hipodermis y la epidermis. La dermis es un sistema compuesto por diferentes elementos fibrosos, filamentosos, difusos y

celulares del tejido conjuntivo, en el que se localizan las redes vasculares y nerviosas y los apéndices derivados de la epidermis. Los principales tipos de células que se pueden encontrar en la dermis son los fibroblastos, macrófagos, células mastoideas y células circulantes transitorias del sistema inmunitario. Es un tejido resistente, flexible y muy elástico, con un espesor de 1 a 5 mm.

La dermis es un componente primordial para el cuerpo, y desarrolla diferentes funciones: actúa como sostén de la epidermis; es responsable de la resistencia a la tracción de la piel; proporciona nutrientes y soporte inmunológico; juega un papel importante en la regulación de la temperatura, presión y dolor de nuestro organismo entre otros [110].

Estructuralmente se divide en dos capas:

a) Dermis papilar (superior): está compuesta casi exclusivamente por la zona de papilas dérmicas que llegan a las crestas papilares. Tiene muchas fibras de colágeno y elastina. Las terminaciones nerviosas y capilares que alimentan la epidermis se encuentran en la capa papilar superior.

b) Dermis reticular (inferior): es más gruesa que la papilar, sus bandas de colágeno son más gruesas y se disponen horizontalmente.

En la dermis reticular el tejido conectivo está compuesto principalmente por colágeno y fibras elásticas. La degradación del colágeno es orquestada por la familia de enzimas Los componentes de la matriz extra celular, como el colágeno y la elastina, proporcionan una resistencia y elasticidad a la piel y son secretados por los fibroblastos [117].

La dermis posee una gran cantidad de vasos sanguíneos y es responsable de proveer nutrientes a la epidermis. Un entramado de arteriolas se extiende desde la grasa subcutánea de la dermis, que constituyen el plexo vascular profundo. Estos vasos transportan sangre a las diversas partes de la dermis y el tejido subcutáneo. Además, las pequeñas ramas de este plexo nutren los folículos pilosos y las glándulas sebáceas. Las arteriolas y vénulas conectan el plexo vascular profundo con otra red de vasos sanguíneos cutáneos llamado plexo vascular superficial, que proporciona sangre a la epidermis. Además de la alimentación, el flujo sanguíneo cutáneo es también crucial para la termorregulación.

Varias glándulas y folículos pilosos son también componentes importantes de la dermis. Hay dos tipos de glándulas de la dermis: glándulas sudoríparas y glándulas sebáceas. Las glándulas sebáceas son especialmente abundantes en la cara, cuello y cuero cabelludo, y a menudo están conectadas a los folículos pilosos. Estos secretan una solución grasienta llamada sebo que cubre el cabello y la piel y la protege del secado. También controla la pérdida de agua epidérmica, y se cree que tiene algunas propiedades antimicrobianas [118].

Las glándulas sudoríparas segregan el sudor, que es una combinación de agua y minerales, ácido láctico y urea. Existen dos tipos de glándulas sudoríparas: ecrinas y apocrinas. Las glándulas ecrinas se encuentran repartidas por toda la piel, pero son particularmente abundantes en el área de la cara, las palmas y las plantas. Las glándulas apocrinas se encuentran en las axilas, el pecho y la zona genital, segregando sudor espeso, se desarrollan en el periodo de la pubertad en el individuo.

El pelo se produce en los folículos pilosos que se encuentran en la dermis o en la capa de grasa subcutánea, y están presentes en casi todas las superficies del cuerpo humano. En los folículos se encuentra otra fuente de células madre de la piel. Las células madre del folículo piloso están en las regiones abultadas del folículo y son responsables de la renovación ininterrumpida de los folículos pilosos.

4.2.3 Hipodermis

Debajo de la dermis se encuentra la hipodermis, también conocida como tejido subcutáneo, el cual conforma la capa más profunda de la piel. En muchas regiones del organismo este tejido celular es de tipo conectivo laxo, de tipo graso que se divide en pequeños lóbulos desarrollados por paredes o septos fibrosos. Las células que se encuentran en esta capa incluyen adipocitos, fibroblastos y macrófagos.

Existe más de un millón de fibras nerviosas sensoriales en la piel que se distribuyen de forma distinta en el cuerpo, permitiendo la mayoría de ellas la transmisión de las sensaciones táctiles, principalmente en la cara, manos y pies, y encontrándose en inferior cantidad en el resto del cuerpo. Los principales troncos nerviosos que se pueden hallar en la hipodermis se dividen en ramas más pe-

queñas que penetran la dermis y que, en algunos casos, incluso llegan a la capa basal de la epidermis [119].

Las neuronas de la piel se dividen según su modalidad: existen táctiles, de presión y vibración, de detección del frío y calor y sensibles al dolor.

La función de la hipodermis es la de actuar como unión entre la piel y otros tejidos (por ejemplo, los músculos). Protege el cuerpo contra impactos, está involucrado en la termorregulación y sirve como un dispositivo de almacenamiento nutricional.

4.3 Homeostasis de la Epidermis

En la capa basal se suceden los procesos que permiten mantener el equilibrio del tejido epidérmico, controlándose el nacimiento de nuevas células para reemplazar aquellas células que se pierden día a día, producto de diferentes heridas sobre la piel o también por la descamación normal de las células de la capa superior de la epidermis. Este equilibrio constante entre el nacimiento y la muerte celular se conoce como homeostasis, y el conjunto de sucesos que ocurren a nivel celular es el que permite que el tejido mantenga su correcto funcionamiento al mismo tiempo que mantiene constante el número de células.

Principalmente existen dos procesos que se suceden en las células de la epidermis para mantener este equilibrio: la proliferación y la diferenciación. Para que se den estos procesos es necesario que se inicien varios mecanismos moleculares en el interior de cada. En la actualidad todavía no se conocen todos los detalles del porqué se habilitan o no tales procesos.

El comportamiento proliferativo de las células de la capa basal es heterogéneo. Algunas células proliferan lentamente, otras proliferan más rápidamente y otras sencillamente no proliferan. La observación de estos comportamientos ha permitido el desarrollo de modelos para explicar los comportamientos de las células en la epidermis.

Hay dos teorías principales para explicar el proceso de renovación de la epidermis. La primera teoría asume que la piel está formada por una colección de es-

estructuras de elementos finitos llamada "unidades epidérmicas de proliferación" (EPU)[120]. La segunda teoría, más actual [121], sostiene que la epidermis se mantiene usando una única población de células madre ubicada en la capa basal, donde las divisiones celulares son simétricas para mantener dicha capa basal, y asimétrica cuando generan una progenie diferenciada. En contradicción con el modelo de EPU, que predice un tamaño finito para el grupo de células, estos conjuntos pueden variar con el tiempo.

También hay un debate sobre el mecanismo responsable que permite que las células se muevan hacia arriba. Probablemente, la razón fundamental es la actividad mitótica, lo que provoca un movimiento pasivo de las células impulsadas por presión debido a la proliferación de las capas más bajas [122, 123].

4.3.1 Unidades De Proliferación Epidérmica

Las células madre tienen como característica principal el poder dar lugar a nuevas células que conformen un tejido adulto. En la epidermis estas células residen en sitios específicos y tienen como rasgo que no se dividen con frecuencia con un ciclo celular lento, pero tienen una capacidad de proliferar de manera ilimitada durante la vida del organismo. Por otro lado existen otro tipo de células, las cuales se caracterizan por tener una división rápida (células de amplificación de tránsito), originándose a partir de células madre que migran de la capa basal e inician la diferenciación terminal después de un número finito de rondas de división celular [120].

Al igual que todos los epitelios, la epidermis se somete a un proceso constante de proliferación para completar su auto-renovación, que es iniciado por una célula madre. El programa de diferenciación terminal es un proceso perfectamente orquestado que se ejecuta en aquellos queratinocitos que salen de la capa basal, culminando cuando alcanzan la capa más superficial.

En todos los epitelios existe un sistema homeostático encargado de reemplazar las células diferenciadas y de reconstruir el tejido después de una lesión. Este sistema jerárquico implica típicamente varias etapas en las que poco a poco las

células disminuyen su capacidad de reproducción y aumenta su diferenciación. La célula que encabeza la jerarquía es la célula madre y tiene la capacidad de reproducción durante toda la vida del organismo. También es relativamente indiferenciada, se divide con poca frecuencia. No existe claridad sobre la ubicación de las células madre dentro de la epidermis. Algunos estudios sugieren que puede estar presente en dos áreas: en la región folicular (bulbo capilar) y en la capa basal, en contacto con la lámina basal ya sea en la parte inferior o en la cresta de la papila dérmica. [124, 125].

Después de la división, una célula madre da lugar a una célula madre-hija que permanece en el nicho o grupo y otra que denomina célula amplificadora de tránsito (TAC), diferenciándose de las células madre ya que éstas tendrán la posibilidad de iniciar un programa de diferenciación. Estas células proliferan rápidamente y se someten a divisiones al igual que sus descendientes, permitiendo expandir la población.

Las células amplificadoras de tránsito y las células madre constituyen la capa basal proliferativa. Los queratinocitos post-mitóticos son los que migrarán hacia las capas suprabasales, perdiendo su capacidad proliferativa e iniciando una transformación funcional hasta llegar a las capas más altas de la epidermis. La densidad de los diferentes tipos de células en las capas se ha estudiado en varios trabajos [46] para determinar el número aproximado y su distribución.

Las células madre epidérmicas residen en la base interfolicular y están organizadas en columnas hexagonales denominadas unidades proliferativas epidérmicas. Esta organización de la epidermis en forma de columnas se observó por primera vez en 1970 en piel de ratones [126], en la que se descubrió que las escamas y capas superficiales se alineaban de forma precisa en columnas orientadas perpendicularmente a la superficie cutánea, adaptándose regularmente a las columnas vecinas. En 1974 Potten [127], determinó que cada unidad proliferativa tiene un número regular de células las cuales son capaces de reproducirse aunque exista una que se ubique en el centro y tenga un ciclo diferente, siendo esta la célula madre de la unidad y posiblemente la responsable de la regeneración homeostática y también como un contribuyente al cierre de heridas.

En la capa basal cada unidad funciona de manera independiente, estando constituida por aproximadamente 10 células. En el centro la célula madre es rodeada

por células amplificadoras de tránsito. En condiciones homeostáticas normales, la proliferación epidérmica y la migración por parte de las células amplificadoras se producen en la capa basal. A un nivel macroscópico el tamaño de las EPU en la piel humana es de aproximadamente 2mm[128].

4.3.2 Moléculas Celulares Epidérmicas

Al igual que los seres humanos, para que un organismo pluricelular se pueda desarrollar y mantener, las células deben comunicarse.

La comunicación entre las células está mediada principalmente por moléculas extracelulares. Las células se ven influenciadas por diferentes moléculas que se encuentran en la matriz extracelular que activan o inhiben varios comportamientos celulares dentro del tejido epidérmico: algunas operan a larga distancia o en entorno definido por sus vecinos más próximos. Estos cambios permiten que el proceso homeostático o el cierre de heridas se den de manera ordenada.

Dependiendo de la señal, de la naturaleza y el estado de la célula receptora, estos efectores pueden ser proteínas de regulación de genes, canales de iones, componentes de una ruta metabólica o partes del citoesqueleto. Cualquier alteración de estas comunicaciones puede acarrear problemas en el tejido o en el organismo.

La mayoría de las células en los organismos multicelulares emiten y reciben señales. La recepción de las señales depende de las proteínas del receptor, por lo general ubicadas en la superficie celular.

Factores de Crecimiento.

En las células epidérmicas se ha descubierto que su crecimiento y diferenciación está influenciado por diversas citocinas. Entre éstas están el factor de crecimiento epidérmico (EGF) y la interleucina-1 alfa (IL-1), los cuales actúan como estimulantes.

La expresión de estos factores de crecimiento (EGF o su homólogo TGF α) se observa en las capas suprabasales, mientras que la expresión del EGF se limita a la capa basal del epitelio adulto [129].

El receptor de IGF (factor de crecimiento similar a la insulina) también se expresa sólo por las células de las capas basales [130]. Los factores de crecimiento IGF son producidos por los fibroblastos dérmicos y las células de capas suprabasales [131], siguiendo el patrón de expresión de EGF - (TGF α) .

Factores de diferenciación.

El factor de crecimiento transformante (TGF) suprime la proliferación y la diferenciación del queratinocito. La expresión de las tres isoformas conocidas de la familia de TGF es diferente en la piel humana y de piel de ratón [132]. TGF - 2 se expresa en la epidermis humana mientras que TGF - 1 y 3 se expresan en la epidermis de ratón embrionario. TGF - 3 y TGF - 2 se expresan por los fibroblastos de las capas de la dermis relacionados con la membrana basal, respectivamente, en los seres humanos y ratones (TGF - 1 casi no se detecta en la epidermis humana) .

La mayoría de las células de la capa basal expresa los tres tipos de receptores de TGF (RI, RII y RIII).

Moléculas de adhesión.

Como ya se ha mencionado en anteriores apartados de la tesis, la adhesión celular es muy importante en la morfogénesis, en el mantenimiento de su estructura final y en los órganos y tejidos ya formados. Las moléculas de adhesión celular actúan como un pegamento, juntando células que forman organismos multicelulares y tejidos. Las moléculas de adhesión celular están situadas en la superficie de las células y se unen a moléculas de la ECM. Las integrinas y e-cadherinas son las moléculas de adhesión celular más importantes que se expresan en la epidermis. La alteración de la expresión de estas moléculas es característica de células tumorales [133].

La mayoría de las moléculas de adhesión se agrupa en cuatro grandes grupos:

- Independientes del Calcio
 - Inmunoglobulinas.

- Dependientes del Calcio
 - Integrinas.
 - Selectinas.
 - Cadherinas.

En los queratinocitos se expresan diversas moléculas de las familias de las inmunoglobulinas, integrinas y cadherinas.

Diversas moléculas de la familia de las cadherinas intervienen en la formación de las uniones intercelulares de los queratinocitos, y son elementos integrales de éstas.

Las integrinas presentes en la capa basal determinan la proliferación de los queratinocitos. Las integrinas $\beta 1$ tienen dos funciones principales: organizar la ECM, incluyendo la activación de expresión necesaria para la fabricación de la ECM [134] ; permitir a las células y a la membrana basal mantener un equilibrio estable en la fase G1, donde un pequeño cambio en el estado del medio ambiente de la célula es suficiente para iniciar el proceso de replicación [135].

4.4 Heridas en la Epidermis

Cuando la piel se daña de alguna forma, como por ejemplo después de una quemadura o una abrasión, la piel necesita ser reparada para restaurar la homeostasis del tejido. Dependiendo de la magnitud de los daños, la piel puede ser reparada reconstruyendo su estructura y volviendo a su función normal o casi normal.

Varios factores pueden retrasar el proceso de cicatrización de la herida: infecciones, presión en el tejido blando del cuerpo, trauma repetido en la herida, edema, necrosis del tejido local, edad de una persona, estado nutricional y salud general del individuo. Un paciente diabético puede tener problemas de cicatrización de las heridas debido a una reducción en la circulación de la sangre.

Hay varios trabajos en la literatura que revisan los procesos que intervienen en la cicatrización de una herida. Estos incluyen los documentos de [136, 137].

Dos procesos fundamentales suceden en la cicatrización de una herida epidérmica: la migración de células dentro de la herida y la proliferación celular para reemplazar las células a partir de las células cerca del borde de la herida.

Cuando se ha producido una herida epidérmica, existe un período de latencia de varias horas antes de que las células basales que rodean la herida se separen de la membrana basal. Estas células luego aumentan su tamaño y migran a través de la herida. Las células basales inician un proceso de mitosis para reemplazar las células que han migrado. La migración se completa cuando las células de la epidermis cubren la herida y se forma una nueva membrana basal.

El mecanismo de migración epidérmica en la superficie de la herida no se conoce completamente. Las células de la epidermis por lo general carecen de movilidad, pero cuando se ha producido una lesión, las células sufren un cambio fenotípico y empiezan a moverse. Esto permite que las células se muevan a través de lamelipodios.

Folkman y Moscona [138] propusieron que la actividad mitótica aumentada en el borde de la herida se asocia con cambios en la forma celular. Con la migración de las células epidérmicas en el borde de una herida, las células epidérmicas se vuelven más planas y esto parece aumentar su tasa de división.

Se cree que un factor que controla el movimiento de células y la proliferación es la inhibición por contacto [139, 140]. Se supone que la inhibición por contacto se produce cuando las células dejan de proliferar porque se vuelven tan numerosas que las células están muy cerca unas de otras. Se cree que existen elementos bioquímicos entre las células que comunican una señal para inhibir la división celular. Se ha propuesto que la ausencia de contacto activa la migración en una herida [136].

La migración y la proliferación de las células también se activan por una combinación de moléculas de señalización, llamadas factores de crecimiento. Estos pueden ser producidos por las células de la epidermis. Los fibroblastos de la dermis también pueden secretar factores de crecimiento que pueden afectar al crecimiento epidérmico y a la migración. Tales factores incluyen el factor de crecimiento-1 (EGF) similar a la insulina (IGF-1) y factor de crecimiento de los queratinocitos (KGF).

Hay tres formas principales en las que los factores de crecimiento controlan los procesos celulares asociados con la curación de heridas. Los factores de crecimiento atraen a las células diana específicas y estimulan a migrar a la zona de la herida (quimiotaxis). Los factores de crecimiento también inducen a las células diana a dividirse y proliferar (mitogénica) y estimulan la actividad celular, por ejemplo, mediante la inducción de los fibroblastos para la síntesis de colágeno.

Existen numerosos factores de crecimiento que se han identificado, pero no todos están implicados en la cicatrización de heridas. Las principales fuentes de factores de crecimiento durante la cicatrización de heridas son las plaquetas y macrófagos. Los factores de crecimiento que están implicados en la curación de heridas son: factor de crecimiento epidérmico (EGF), KGF, factor de crecimiento transformante α (TGF α), factor de crecimiento transformante - β (TGF β), factor de crecimiento derivado de plaquetas (PDGF), y el factor de crecimiento de fibroblastos (FGF).

En el capítulo 6 de esta tesis se extenderá más a cerca de este tema, que hace parte central de esta tesis.

*-Rápido, Campanita —susurró—, cierra la ventana. Con cerrojo. Muy bien, y ahora tú y yo salimos por la puerta.
James Matthew Barrie. – Peter Pan*



Capítulo 5 Aceleración del Modelo de Ising y del Modelo de Potts a Través de Implementaciones Paralelas

Este capítulo está dividido en dos partes basadas en los artículos realizados durante el desarrollo de esta investigación [141, 142]. Como ha sido mencionado en el capítulo primero de la Introducción de la tesis, uno de los principales intereses en el desarrollo del trabajo es ofrecer un entorno de simulación que permita a experimentadores desarrollar y probar hipótesis de trabajo, aplicando una novedosa traducción sencilla al modelo, modificando o añadiendo nuevos parámetros.

Dentro de este ámbito, como también se ha mencionado en los capítulos precedentes, uno de los desafíos de los modelos basados en la mecánica estadística es su coste computacional. El GGH-CPM es un modelo discreto lo suficientemente expresivo para implementar diferentes tipos de modelos biológicos [5, 143], sin embargo es muy costoso computacionalmente y por esto su uso puede verse limitado en modelos compuestos por gran número de componentes. Los modelos de un tejido completo requieren grandes cantidades de memoria y procesamiento. Por ello, la necesidad de desarrollar simulaciones con un enfoque secuencial donde se use tan sólo un único procesador limita los resultados buscados. En este caso particular la simulación de un modelo completo de la homeostasis de

la epidermis exige una gran cantidad de número de células e interacciones, que se traduce en mayor gasto de recursos de la homeostasis de la epidermis exige una gran cantidad de número de células e interacciones, que se traduce en mayor gasto de recursos.

Claramente se hace necesaria la adición de mecanismos que hagan que el algoritmo se pueda ejecutar de manera paralela, donde la lógica del modelo CPM-GGH se pueda distribuir en diferentes tipos de dispositivos de multiproceso (clústeres de ordenadores de alto rendimiento, procesadores multi-núcleo o GPUs).

Dentro de esta investigación se proponen dos estrategias de paralelización. Por un lado, hemos desarrollado un algoritmo paralelo de descomposición espacial basado en el paso de mensajes usando MPI, que permite grandes simulaciones del modelo de Potts, orientado a ser ejecutado en clústeres de ordenadores, que no modifica la estrategia del modelo de Montecarlo. Por otro, se propone un mecanismo alternativo de actualización de la rejilla para acelerar el número de evaluaciones de cambio de *spin* del modelo de Ising.

5.1 Una Implementación Concurrente Usando Java E Hilos Para El Algoritmo De Metrópolis - Monte Carlo En El Modelo De Ising

Desde hace varias décadas se han estudiado mediante simulaciones de Monte Carlo las diferentes propiedades y el cálculo de exponentes críticos en el modelo de Ising [28], con la mejora continua de su precisión gracias al beneficio ofrecido por el aumento de los recursos computacionales. Tales recursos informáticos se pueden aplicar a diferentes áreas de la física, incluyendo: simulaciones de dinámica molecular [144] u optimización estocástica [145], que exigen una gran cantidad de recursos de computación de alto rendimiento. Estos recursos de computación incluyen arquitecturas de varios núcleos basado en memoria compartida, pudiendo utilizarse mecanismos de paralelización como OpenMP o MPI.

Además, para poder agilizar la ejecución de las aplicaciones, se utilizan clústeres de ordenadores, bien sea usando nodos homogéneos o heterogéneos, computación distribuida o procesadores con múltiples núcleos. Una tendencia reciente en la informática es la paralelización usando la capacidad de cómputo de las tarjetas gráficas o GPU.

Este trabajo está enfocado a mostrar un tipo de paralelización que hace abstracción de la máquina física donde se ejecuta el algoritmo, delegando esta labor a un lenguaje de alto nivel y sin tener que recurrir a directivas adicionales que el desarrollador deba usar. El resultado es una implementación concurrente del modelo de Ising usando Java como lenguaje de programación para el algoritmo de Metrópolis - Monte Carlo. El método utiliza características propias que ofrece Java, como hilos, monitores, variables compartidas y constructores que encapsulan y ocultan los detalles de bajo nivel que pueden ser requeridos en otro tipo de enfoques a la hora de paralelizar un algoritmo.

Adicionalmente, el uso de este enfoque hace que el algoritmo no dependa de la plataforma donde se ejecuta. El intérprete del lenguaje tiene el control de la comunicación con el sistema operativo, maximizando la concurrencia tanto como sea posible y evaluando a cada momento los recursos disponibles ofrecidos por el equipo.

Como se ha explicado en capítulos anteriores, el modelo de Ising fue desarrollado por Ernst Ising en los años veinte y en su forma original permite simular el comportamiento estadístico de los dominios ferromagnéticos (o anti-ferromagnéticos). Después fue extendido a sistemas compuestos por varias partículas, donde cada una de ellas puede contener diferentes valores de *spin* (modelo de Potts). En 1992 Glazier y Graner [8] determinaron que las células biológicas se pueden representar como dominios adyacentes que tienen el mismo valor de *spin* o índice. Con esta suposición, fueron capaces de simular con éxito el proceso de reorganización de las células con diferentes energías de adhesión. Posteriores extensiones y modificaciones del modelo de Potts llevaron a la creación del modelo de Graner Hogeweg Glazier -GGH, conocido inicialmente

como *Celular Potts Model* (CPM), [8, 9]. El GGH describe fenomenológicamente la célula, extrayendo la mayor información para describir su comportamiento y reproducir su evolución dentro de los sistemas biológicos complejos, como un tejido vivo.

La simplicidad matemática del modelo GGH y su capacidad de simular sistemas compuestos de hasta 10^7 células (en un solo procesador), hace que sea una de las herramientas más prometedoras en el campo de la modelización biomédica de tejidos, órganos y organismos.

La característica común del modelo de Ising, Potts y GGH es su conexión en su origen común con la mecánica estadística. Los sistemas reales que simulan, por lo general están contruidos por múltiples copias de elementos microscópicos (cambio de *spin* o de dominios celulares). El comportamiento de todo el sistema es originado por los cambios locales a nivel microscópico.

Por ejemplo, esto ocurre en la simulación de la organización celular o "*cell sorting*", donde se presenta una reorganización espontánea de los objetos celulares para llegar a la configuración más favorable energéticamente. En cada cambio microscópico, la configuración de una sola célula depende sólo de la variación local de la energía debido a un cambio mínimo de la configuración del sistema. El comportamiento complejo de todo el sistema es, por lo tanto, derivado de las normas locales. Computacionalmente, la simulación se implementa mediante técnicas de Monte Carlo y se basa en el algoritmo Metrópolis - Monte Carlo.

Típicamente el algoritmo de Metrópolis se implementa como un algoritmo que evalúa las actualizaciones de la rejilla en serie. La óptica actual sufre de una ineficiencia computacional, desperdiciando la capacidad de las arquitecturas multi-procesador o el uso de clústeres de ordenadores. Adicionalmente, al poner en práctica una versión paralela del algoritmo, el investigador puede encontrarse con el inconveniente de que dichos algoritmos sean específicos para la aplicación y su portabilidad se encuentre limitada a la arquitectura donde fueron implementados.

El primer objetivo y enfoque de la tesis es proponer una técnica general para implementar el paralelismo en la simulación, utilizando el número máximo de los recursos disponibles del sistema, de manera que no dependa de los detalles de la plataforma utilizada. Nuestro enfoque parece natural y simple, y añade una cantidad mínima de sobrecarga computacional.

5.1.1 Autómatas Celulares, Modelo de Ising y sus Variantes

Los autómatas celulares se concibieron en una época en que las únicas herramientas informáticas que estaban disponibles para la mayoría de los investigadores eran el lápiz y el papel. El uso de tiempo y espacio de estado discreto se adapta perfectamente a las características de los ordenadores y parecen haberse diseñado para usarse en estos.

El comportamiento global o colectivo en un sistema complejo emerge de la evolución y la interacción de muchos elementos. Por lo tanto, deben existir modelos de programación acordes a tales modelos que permitan expresar el comportamiento e interacción de un número muy grande de elementos individuales.

Debido a su paralelismo inherente, los autómatas celulares pueden ser explotados para modelar sistemas emergentes de gran escala en los ordenadores paralelos. En este escenario, los modelos paralelos celulares proporcionan herramientas útiles para la programación de cálculos complejos.

Un enfoque clásico para la simulación de los fenómenos complejos que se base en la programación tradicional usando herramientas y lenguajes de programación convencionales es difícil, y el resultado es de código largo y complejo. Esto ocurre debido a que estos enfoques de programación se basan en el diseño de un sistema en su conjunto. Diseño y programación no parten de elementos básicos o componentes del sistema, sino que representan un sistema mediante el modelado de sus características generales.

El comportamiento global de estos sistemas se desprende de la evolución y la interacción de un número masivo de elementos simples, por lo que no necesita ser codificado explícitamente.

Las arquitecturas paralelas son muy adecuadas para ser usadas por modelos abstractos de computación (como los autómatas celulares) que representan nuevas formas de describir los fenómenos y sistemas complejos. Todas las células de un autómata celular se actualizan en paralelo. Así, el estado de la totalidad de los avances del autómata en pasos de tiempo discretos y el comportamiento global del sistema se determina por la evolución de los estados de cada célula como resultado de múltiples interacciones locales.

Los autómatas celulares proporcionan un marco global para la ejecución de aplicaciones paralelas que aportan soluciones naturales a fenómenos complejos y dinámicos basados en el uso de tiempo discreto y espacio discreto.

Hay varios entornos de *software* que están disponibles para el uso de los autómatas celulares en paralelo. Para una revisión, véase, por ejemplo [146, 147]. En conclusión, es conveniente subrayar que la relación entre CA y el cómputo paralelo no es sólo tecnológica, sino también conceptual. Los autómatas celulares representan una forma de describir el mundo que se adapta de forma natural a los ordenadores paralelos. En este sentido, hay que tener en cuenta que las versiones paralelas de los algoritmos de simulación de AC deben equilibrar el coste de comunicaciones con el trabajo que se realiza en paralelo.

El objetivo de nuestro enfoque es proponer una técnica general para implementar el paralelismo en la simulación de tejidos o sistemas celulares a partir de un modelo modificado del CPM-GGH. Nuestra implementación paralela parte del modelo de Ising, que puede ser extendida al modelo CPM-GGH.

5.1.2 Modelo de Ising

Resumamos aquí las principales características del modelo, explicadas ya en otros puntos de la tesis, para evidenciar las características de nuestra aproximación.

Originalmente el Modelo de Ising es un modelo discreto de ferromagnetismo, sobre la base de los momentos magnéticos o *spines* σ de los átomos individuales y sus energías de interacción (J) que contribuyen al comportamiento energético global a través de una función de energía denominada Hamiltoniano H .

El modelo de Ising considera al material como una rejilla donde los *spines* de éste se distribuyen regularmente. Los *spines* están representados por dos valores discretos $\sigma \in \{1, -1\}$. Para reducir la energía del sistema, la siguiente expresión regula el modelo:

$$H(\sigma_i) = -J \sum_{i,j} s_i s_j \quad (5-1)$$

La suma define la interacción de los *spines* teniendo en cuenta los cuatro vecinos más cercanos para una posición seleccionada. Si $J > 0$ el sistema es de tipo ferromagnético, donde las interacciones del *spin* y sus vecinos prefieren ser paralelos, es decir: $s_i = +1$ o $s_{i+1} = +1$ o $s_i = -1$ o $s_{i+1} = -1$. En los materiales no ferromagnéticos ($J < 0$) los enlaces prefieren ser contrarios o anti-paralelos, es decir: $s_i = +1$ o $s_{i+1} = -1$ o $s_i = -1$ o $s_{i+1} = +1$.

El modelo de Ising y sus derivados utilizan con frecuencia en sus simulaciones una aproximación indirecta, ya que una solución analítica o numérica es difícil. El método de Metrópolis - Monte Carlo [26] es uno de los algoritmos más populares usados para simular, no sólo el modelo de Ising, sino también sus variantes (por ejemplo, Potts y GGH). Este método se puede describir por medio del siguiente esquema:

1. Elegir un sitio de la rejilla al azar.

2. Calcular el cambio energético (ΔH) según la función de energía de acuerdo a la configuración actual y sus vecinos.
3. El cambio del valor del *spin* en el sitio actual se realiza de acuerdo con la siguiente función de probabilidad:

$$P(\sigma \rightarrow \sigma') = \begin{cases} e^{\frac{-\Delta H}{kT}}, & \Delta H > 0 \\ 1, & \Delta H < 0 \end{cases} \quad (5-2)$$

Donde k es la constante de Boltzman y T la temperatura.

4. Volver al paso 1 tantas veces como sea necesario para obtener una configuración estable que permita calcular los exponentes críticos.

En una rejilla bi-dimensional, X_{max} y Y_{max} (X e Y son los máximos valores obtenibles para cada dimensión) definen un paso de Monte Carlo (MCS), y $X_{max} \times Y_{max}$ son el número de intentos de cambio del valor del *spin*.

Las instantáneas de una simulación cualquiera (tomadas a intervalos regulares, medidos por el número de intentos de cambios en el sistema en el algoritmo de Metrópolis) representan la evolución temporal del sistema estudiado.

Las principales variantes del modelo de Ising son el modelo de Potts y el CPM-GGH. El modelo de Potts fue propuesto en los años cincuenta por Renfrey Potts [35]. Esta extensión permite el uso de más de dos tipos de valores que pueden asumir los *spines*. El modelo de Potts simula con éxito diferentes procesos físicos, tales como el crecimiento de granos o modelos de burbujas. Hay similitudes entre el comportamiento de estos sistemas y el que intenta explicar el modelo de Ising: existe una temperatura crítica, por debajo del cual los *spines*, dominios o células se disocian y por encima desaparecen.

Glazier et al. [8, 9] han ampliado estos modelos a partir del modelo de Potts. Ellos observaron que las células biológicas se pueden representar como dominios espacialmente extendidos, caracterizados por el mismo valor de índice por ana-

logía con los de los dominios ferromagnéticos. Además usando simples interacciones *spin* a *spin* el modelo GGH abarca muchos comportamientos celulares observados en la naturaleza: volumen y su forma, la adhesión célula a célula, la respuesta de las células a las señales químicas, la secreción de algún químico por una célula, la división celular, la muerte celular y la diferenciación celular entre otros.

La combinación de rejilla, interacción de la celda y sus vecinos más próximos, y una función de transición regida por la dinámica de Metrópolis, pueden entenderse según el modelo de Ising como un cuasi-autómata celular [148, 149]

Como se ha explicado anteriormente, la manera natural de implementar este algoritmo suele actualizar una posición de la rejilla sólo en cada instante de tiempo, lo que va en contra de un enfoque paralelo de su implementación.

5.1.3 Implementación Paralela del Modelo de Ising

Típicamente el algoritmo de Metrópolis - Monte Carlo se implementa como un algoritmo secuencial lo que conlleva cierta ineficiencia computacional [25, 150].

Las simulaciones por ordenador o bien complementan los modelos analíticos o ayudan a encontrar un solución cercana. La simplicidad matemática de los modelos numéricos combinados con potentes ordenadores puede, y en muchos casos es, un sustituto de la inexistencia de la solución analítica elegante y compacta para un problema dado. Sin embargo muchas implementaciones olvidan las arquitecturas multi-procesador.

Los fenómenos físicos y biológicos simulados utilizando Ising o modelos GGH caen, en la mayoría de los casos, en la categoría de los modelos donde no existe una solución analítica. El atractivo de los modelos GGH (o Ising) podría incrementarse en gran medida si las simulaciones se pudiesen hacer en un tiempo

relativamente corto, donde el modelo pueda modificarse cambiando valores sobre varios parámetros y ver sus resultados de manera más rápida.

Para diseñar una versión paralela que incorpore el algoritmo de Metrópolis para el modelo de Ising, hemos evaluado tres tipos de técnicas [25]: *paralelización Trivial*, donde el problema se divide en tareas completamente diferentes sin la necesidad de la inter-comunicación entre procesos; *descomposición funcional*, donde un problema se rompe en una gran variedad de trabajos ejecutados por diferentes procesadores, cuyos resultados deben ser puestos juntos; *la descomposición por dominios*, donde el problema se divide en regiones para su procesamiento. Los enfoques de paralelización del modelo de Ising son en su mayoría de la última clase. Barkema y MacFarland [150] proponen una versión paralela donde la rejilla se descompone en diferentes partes (dominios) que se asignan a procesadores diferentes, donde se introduce un método de descomposición del tipo tablero de ajedrez. Korniss G [151] describe una versión paralela usando el algoritmo *n-fold* para ser ejecutado a bajas temperaturas y otros enfoques propuestos por [152] aplicados a simulaciones de crecimiento epitaxial de cobre.

El enfoque de esta tesis abstrae y oculta los detalles de bajo nivel, delegando al sistema operativo los módulos de distribución de carga de procesos. Desde el punto de vista del desarrollo de *software*, un lenguaje orientado a objetos como Java soporta hilos concurrentes. Ésta es una opción razonable como herramienta para implementar el algoritmo, a pesar del hecho que la máquina virtual estándar de SUN-ORACLE de Java (JVM) no se puede ejecutar en clúster. Es posible usar otras distribuciones de Java modificadas para ser ejecutadas en un clúster de ordenadores por medio de una máquina virtual distribuida (DJVM), que permite la ejecución en paralelo de hilos de Java. De esta manera, una aplicación multiproceso Java se ejecutaría en un clúster como si se estuviera ejecutando en una sola máquina.

Hay diferentes tipos de DJVM, tales como: Java/DSM [72], JavaParty [73], cJVM [74], Jackal [75], Hyperion [76], JSDM [77], Kaffemik [78], J/Orchestra [79], JESSICA2 [80], JESSICA3 [81], JESSICA4 [82], Terracotta Clustered JVM [85]

5.1.4 Métodos e Implementación

En la descripción del modelo que se realiza a continuación se utilizan los siguientes nombres:

- X_{max} , y_{max} definen la dimensión de la rejilla en X e Y
- *Main* es el nombre del hilo principal del algoritmo.
- Los hilos que permiten relacionar la ubicación de cada *spin* en la rejilla, se denomina como espín *threads*.

5.1.4.1 Pseudo-Código del Main Thread

El hilo principal monitoriza la ejecución de cada uno de los *pasos de Monte Carlo* (MCS) de la rejilla. A su vez el hilo principal es el encargado de crear un conjunto de hilos "hijos" de *spines* donde se evalúa la probabilidad de cambio de la posición seleccionada, siendo ejecutados en paralelo. El hilo principal gestiona y espera hasta que todos ellos terminen y verifica la actualización correcta de la rejilla para que el siguiente paso de Monte Carlo pueda iniciarse.

Es fácil darse cuenta de que la única condición de sincronización es en el momento en que el conjunto de hilos de *spines* terminan de ser evaluados y se da paso al siguiente MCS, pues no se puede iniciar si no se cumple tal condición.

El pseudo-código del hilo principal se describe a continuación:

1. Se empieza inicializando aleatoriamente la configuración inicial de la rejilla.
2. La siguiente configuración es inicialmente una copia de la primera configuración.
3. Para cada MCS:
 - a) Para cada familia de hilos de *spines* y mientras que el sistema admita la creación de un nuevo hilo, hacer:

- I. Seleccionar una posición de la rejilla aleatoriamente.
 - II. Mientras la posición sea vecina de la ya seleccionada, se selecciona otra posición aleatoria (el máximo número de intentos es un parámetro del algoritmo).
 - III. Crear un nuevo hilo para la posición seleccionada e iniciar su ejecución.
4. Se selecciona una nueva configuración.
 5. Se copia la configuración previa como la configuración siguiente.

5.1.4.2 Pseudo-Código del Spin Thread

Todos los hilos de *spines* son creados por el hilo principal. Ellos comparten la configuración actual y la siguiente. Todos ellos tienen que obtener los valores de sus vecinos de la misma copia de la configuración actual, y si es necesario deben cambiar el valor del *spin* en la misma copia de la siguiente configuración.

El código siguiente resume la idea anterior:

1. Calcular el cambio energético ΔH en posición seleccionada de la rejilla.
2. Cambiar el valor del spin según la ecuación (5-2). El algoritmo tiene en cuenta que todos los hilos de *spines* cambian la misma copia de la siguiente configuración. Un monitor de Java tiene la responsabilidad de controlar la configuración por medio de un método de sincronización, para asegurar que tan sólo un hilo de *spines* actualiza a la vez la configuración.

Este enfoque reúne las siguientes características importantes:

- El diseño del algoritmo hace que sea imposible intentar cambiar simultáneamente dos posiciones contiguas.
- La forma en que se seleccionan estas posiciones no introduce factores asimétricos, es decir, todos los *spines* de la rejilla pueden ser visitados, posiblemente en un orden diferente en cada MCS, por las siguientes razones:

- Sea n el número medio de *spines* evaluados simultáneamente. Se necesitan, por lo tanto, $(x_{\max} \times y_{\max}) \div n$ familias de *spines* que simultáneamente se evalúan para cada MCS. Evitando en la evaluación de espines contiguos asegura que $n \leq (x_{\max} \times y_{\max}) \div 2$. Algunos algoritmos, por ejemplo el autómata celular de Creutz [96] o el Q2R [95], son capaces de alcanzar este máximo valor. Por lo tanto, es necesario más de una familia para completar cada MCS.
- Las familias no dependen una de otras.

Nuestro enfoque acepta cambios de *spines* con el mismo criterio que en el método de Metrópolis - Monte Carlo, asegurando que ambos enfoques comparten las mismas características.

Una simulación tradicional del modelo de Ising usando el método de Metrópolis - Monte Carlo obedece a la condición de balance detallado: si para cada proceso interno en la evolución del sistema existe un proceso inverso que se produce a la misma velocidad, esto asegura que una secuencia de configuraciones converge con la distribución de equilibrio correcta, independientemente de la condición inicial. La cadena de Markov en un proceso de Monte Carlo genera una secuencia de configuraciones en las que la probabilidad $x_j(t+1)$ de un sistema observado el sistema en el estado j en el momento $t+1$ se determina tan sólo por el estado anterior i en el paso t .

Esto puede ser expresado por:

$$x_j(t+1) = x_i(t)P_{ij} \quad (5-3)$$

Donde $x_i(t) = \{x_1(t), x_2(t), \dots, x_t(n)\}$ es la distribución de probabilidad en el paso t . Los elementos P_{ij} de la matriz de transición representan la probabilidad que el sistema en transición desde el estado i al estado j . Si existe una distribución de equilibrio de los estados de x^* tal que $x^* = x^* P$, entonces $x(t)$ garantiza la

convergencia de x^* y $t \rightarrow \infty$ cuando P satisface la condición de balance detallado:

$$x_i^* P_{ij} = x_j^* P_{ji} \quad (5-4)$$

Aunque el balance detallado de una cadena de Markov es condición suficiente para asegurar la convergencia, no es del todo necesario [153]. En este trabajo se opta por no hacer cumplir dicha condición ya que nuestro interés difiere del cálculo de exponentes del modelo de Ising o Potts y se orienta a la aceleración de la evolución celular en el CPM-GGH. Se puede concluir que nuestro algoritmo es una implementación paralela alternativa al modelo estándar del algoritmo de Metrópolis. Sin embargo, el algoritmo no garantiza que vaya a producirse perfectamente la condición del balance detallado. Al igual que en [98], la condición no debe ser estricta pues no es necesario tenerla en cuenta.

5.1.5 Resultados

Las siguientes imágenes muestran las diferentes configuraciones resultantes generadas a través del algoritmo. Las simulaciones ilustran el correcto comportamiento que se muestra en el modelo estándar:

- La imagen 1 (de arriba abajo y de izquierda a derecha) de la Figura 5.1 muestra una configuración inicial de una rejilla de 100 x 100.
- Las imágenes 2 a 6 muestran las configuraciones alcanzadas después de 1000 MCS para diferentes valores de T (respectivamente 0,2, 0,5, 1, 1,5 y 2,5). Se puede observar la presencia de dominios claros con dichos valores de T , análogo al uso del método estándar.
- Las imágenes 7 a 9 muestran las configuraciones alcanzando los valores más altos de T (respectivamente 3,5, 4,5 y 6). Ellos muestran una distribución más aleatoria de la configuración de los *spines*, que es esperable de la conducta en el modelo estándar de Metrópolis.

Una de las principales características de nuestro enfoque es el hecho de que podemos ajustar el grado de paralelismo (número de hilos o *spines* evaluados al mismo tiempo) para llegar a la máxima soportada por el *hardware*.

Hay dos factores críticos en el modelo: la memoria y el número de procesadores. Nuestro algoritmo puede usar toda la memoria disponible, a lo sumo $(x_{\max} \times y_{\max}) \div 2$ hilos. El sistema operativo es el responsable de controlar y gestionar correctamente los hilos, haciendo que nuestro código no dependa de ninguna plataforma concreta.

En nuestro algoritmo, el mecanismo que selecciona las posiciones que serán cambiadas puede resultar más costoso frente al algoritmo de Metrópolis - Monte Carlo, ya que el algoritmo debe comprobar y garantizar que una posición seleccionada no es vecina de cualquier otro *spin*, esto lleva a que se produzcan esperas o evaluaciones innecesarias.

Una vez que se han creado los n hilos de la configuración actual, es posible (dependiendo de la plataforma) que todos los hilos puedan ser evaluados simultáneamente. En el mejor de los casos, el tiempo empleado para su evaluación será igual al que se emplearía para analizar toda la configuración como si fuese un cambio de un único *spin*.

De esta manera y en la mayoría de los casos, nuestro enfoque mejora el rendimiento del método original, ejecutándose en múltiples procesadores y no en serie como hace tradicionalmente una línea de procesamiento.

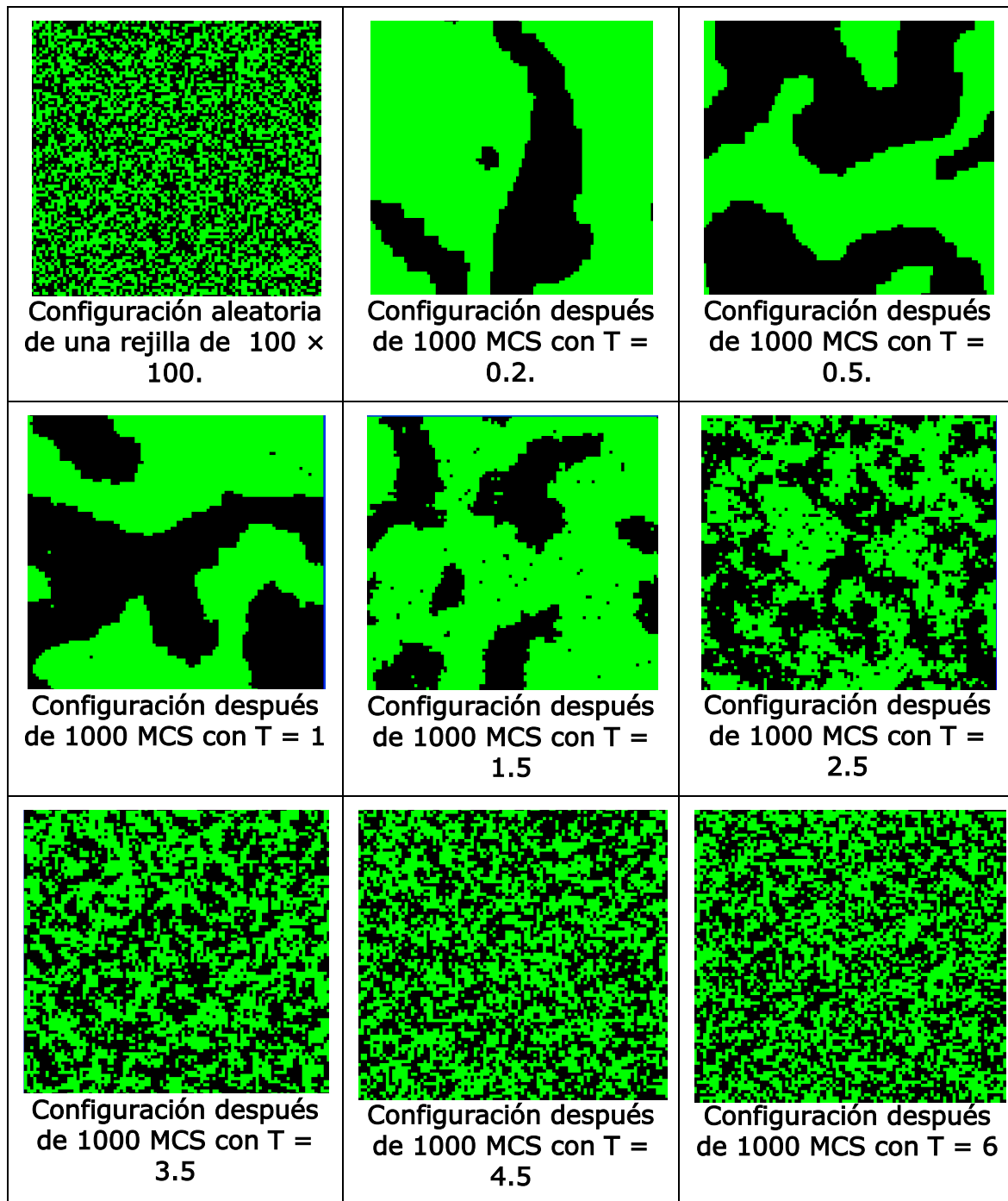


Figura 5.1. Configuraciones a diferentes temperaturas usando el modelo de Ising

5.2 Paralelización del Método de Metrópolis – Monte Carlo Para el Modelo De Potts Usando una Partición Dinámica de la Rejilla

En los últimos años, los ordenadores han aumentado su capacidad de cálculo. También las redes han mejorado las tasas de transferencia de datos para la interconexión de estas máquinas.

Los programas que hoy en día tratan de sacar ventaja a estas nuevas tecnologías, deben ser escritos utilizando otras técnicas a las tradicionales de programación. La mayoría de los algoritmos fueron diseñados para ser ejecutados en un solo procesador, en una forma no concurrente, pues en su momento no se contaba con las ventajas que ofrecen al ser ejecutados simultáneamente en un conjunto de procesadores o distribuidos en una red de ordenadores.

En el caso particular de modelos de mecánica estadística basados en rejilla, en una estrategia de paralelización usando MPI, cuanto mayor sea el número de cálculos por píxel en la rejilla, más pequeña es la relación de paso de mensajes, y por tanto se produce una mayor eficiencia. En los modelos de Ising o de Potts, usando una actualización secuencial, la carga computacional por actualización de *spines* es pequeña frente a la actualización en modelo de CPM-GGH, determinado por las diferencias definidas en el *hamiltoniano* vistas en capítulos anteriores. Adicionalmente en el CPM-GGH, la proporción de intentos fallidos de actualización frente a las aceptadas puede ser muy grande (típicamente 10^4 o más) [98].

La principal dificultad de la paralelización CPM-GGH es que la energía efectiva no es local. Por ejemplo, el cambio de un lugar de la rejilla puede afectar al volumen de dos células y por consiguiente a la energía asociada con todos los píxeles en ambas células. Si los píxeles de una célula se dividiesen entre subdominios y las actualizaciones que afectan a una célula estuviesen compartidas entre diferentes zonas donde no se garantiza la correcta comunicación, podría presentarse como resultado acerca del estado de la célula información obsoleta entre las partes.

Si se usa una paralelización en clúster, donde cada procesador calcula un subdominio rectangular predefinido basado en la rejilla completa, la no-localidad del cálculo energético aumenta en gran medida dada la frecuencia de ajuste para la sincronización. Esto causa latencia en la comunicación entre unidades de proceso debido al tiempo que cada procesador pasa esperando información en vez de estar calculando.

Para resolver este problema, realizamos una propuesta de paralelización usando el modelo de Potts como punto de partida, el cual luego se aplicará en el CPM-GGH, mejorando la estructura de datos para describir la posición y descomponer el subdominio de manera dinámica, asignando a cada nodo subrejillas más pequeñas para que cada dominio calcule una porción de la simulación.

Durante más de dos décadas, el uso de la computación paralela en máquinas distribuidas para resolver problemas de cálculo ha sido el foco de la comunidad de computación de alto rendimiento. Los avances en microprocesadores y las redes de computadoras han provocado el uso de clústeres de estaciones de trabajo como una alternativa a los cada vez más costosos superordenadores. Sin embargo, la demanda de potencia de cálculo continúa creciendo mientras la mayoría de las máquinas disponibles están infrautilizadas. Debido a este crecimiento continuo de las capacidades, se han planteado varios problemas de optimización asociados con algoritmos relacionados con el diseño de redes y topologías.

Para una optimización general en los clúster de ordenadores, se necesita una metodología dinámica de asignación de recursos [154-156]. En la actualidad, esta es una de las principales razones de la existencia de un interés especial en la búsqueda de nuevos algoritmos, lo que permitirá la sustitución de los métodos tradicionales. La eficiencia y la posibilidad de escalado que ofrecen las arquitecturas paralelas hacen que el uso de los métodos tradicionales no sea aplicable en muchos casos [157].

El principal problema en el diseño de un algoritmo paralelo o distribuido reside en la comunicación y la sincronización de los procesos para su ejecución concurrente en diferentes procesadores. Este es un problema de optimización NP-complejo, por el que no siempre se obtiene la mejor solución: en muchos casos sólo una aproximación puede ser calculada.

Desde el punto de vista de la multiplicidad, una buena aplicación paralela sobre un clúster heterogéneo (HNoC) debe ser capaz de distribuir de manera desigual los cálculos, teniendo en cuenta, al menos, la velocidad de los procesadores, la topología de la red y los recursos ofrecidos por cada procesador. La eficiencia de la aplicación paralela también depende de la exactitud de la estimación de la velocidad de los procesadores sobre el clúster. Calcularla es difícil pues la velocidad real puede estar supeditada a las diferencias en el conjunto de instrucciones, el número de unidades de ejecución de instrucciones, el número de registros y la estructura de la jerarquía de memoria, entre otros.

Varios algoritmos se han usado para paralelizar el modelo de rejilla [92, 93, 144] desarrolló para el modelo de Ising, usando para el algoritmo de la tablero de ajedrez, que a su vez es la base de la paralelización del CPM-GGH mediante MPI [150]. Esencialmente, el algoritmo utiliza un mecanismo de actualización asíncrona de subcuadrículas en diferentes momentos.

Desde el punto de vista de las comunicaciones, uno de los principales problemas en la implementación de redes de datos es la elección de la topología de la red que podría verificar ciertas características de confiabilidad, suponiendo esto como una medición que evalúa la probabilidad de éxito en la comunicación entre pares de unidades de procesos. Este es un factor no trivial en la calidad de los servicios ofrecidos a todos los equipos. La evaluación de los parámetros exactos que determinan la confiabilidad de la red es también un problema NP – completo. Por esta razón, hay que tener en cuenta una optimización de la topología de la red. Hay que asegurarse de que el retraso temporal producto de la comunicación y la sincronización de los procesos, sea menor al retraso por motivo del tratamiento de los datos por cada procesador en cada ordenador que compone el clúster. Además, es importante tener en cuenta que cuando el problema se reparte, podría producirse que el tiempo empleado para transmitir datos entre ordenadores y sincronizarlos supere el tiempo que utiliza la CPU y presente los resultados.

La red de comunicación es generalmente heterogénea. La velocidad y ancho de banda de la red entre los diferentes procesadores es un elemento que incide significativamente. No es lo mismo realizar una transferencia de datos sobre el mismo *bus* de datos en la misma máquina que cuando se realiza un intercambio

de datos en la red. Otro problema sobre la red es la eficiencia del protocolo de comunicación pues también incrementa la latencia. Una buena paralelización debe ser capaz de utilizar varios protocolos de red entre diferentes nodos [158].

5.2.1 Definiciones y Antecedentes

Un grupo de ordenadores trabajando de forma conjunta constituyen un clúster, que puede ser vistos como un grafo dirigido ponderado, en el que se define el coste de los recursos en los vértices o nodos (recursos computacionales) y en el que las aristas son la comunicación entre un par de nodos. De una manera discreta, un nodo es un ejemplo de unidad de procesamiento. Cuando se aplica el concepto de grafos para un entorno distribuido y heterogéneo de ordenadores (HNoC), los costes asociados y ponderados con sus pesos dependen de diversas características del sistema, tales como la velocidad de procesamiento y la latencia en la comunicación entre los nodos de la red. Cada nodo del clúster puede asumir el rol de proceso *maestro* o *esclavo*, donde los maestros se encargan de enviar las tareas a realizar por los nodos esclavos y de gestionar la sincronización entre los procesos.

Los esclavos desempeñan el papel de ejecución de las tareas enviadas por el proceso maestro. En el presente caso, todos los nodos son indistintamente maestro o esclavo.

Con el fin de construir un entorno de computación para las clúster de ordenadores, es necesario contar con un algoritmo que permita predecir el rendimiento y el consumo de recursos de diferentes configuraciones de clúster.

Este algoritmo se denomina *Resource Management System (RMS)*. El problema es la dificultad de predecir el tiempo de cálculo por un nodo antes de que reciba una cierta carga arbitraria. Además, esto será aún más difícil si se tiene en cuenta la variación de la topología de la comunicación en el clúster. Las tareas básicas del RMS son aceptar las peticiones de recursos realizadas por las aplicaciones y asignarlos de acuerdo a su disponibilidad.

Como todos los nodos tienen la misma oportunidad de actuar como maestro o esclavo, se necesita un nodo especial (un "*supermaster*") donde el RMS se ejecute. El RMS utiliza una estimación predictiva basada en una función matemática (función heurística) con el fin de asignar a las tareas paralelas un conjunto de recursos.

Esta función heurística será necesaria para definir la partición del grafo y la asignación de la carga de trabajo a cada nodo. En la actualidad, estas técnicas para predecir el rendimiento del sistema dinámico se basan en técnicas de colas y/o en técnicas históricas. Haciendo una comparación entre estos dos modelos, el de colas requiere mayor duración en la CPU para hacer la predicción de tiempo de respuesta, mientras que en el caso del modelo histórico sus predicciones son casi instantáneas.

Sin embargo, en la técnica de gestión de colas resulta menos costosa la puesta en práctica que en el modelo histórico. Esto se debe a que el diseño de un modelo histórico implica que se especifique y se valide cómo se determinarán las previsiones, mientras que el modelo de cola puede ser resuelto de una forma automática.

Ambas técnicas se pueden combinar para sacarles ventaja [159]. El modelo de gestión de colas define una red de colas de aplicación. La estrategia de solución consiste en dividir el problema en tareas en función de los recursos y el número de servidores que pueden atender, proporcionando una topología inicial y, posteriormente, a través de la evolución en el tiempo, usando un método histórico para su ajuste (determinando la *re*-dimensión de las colas en cada paso del algoritmo, hasta que la solución converja en una distribución óptima de los recursos, tareas y esperas en las comunicaciones).

Para el uso del modelo de colas es necesario definir una estructura para cada nodo que pertenece al clúster, que será compartida por todas las peticiones de entrada Figura 5.2. Los nodos pueden ser tanto clientes (solicitud de información) como servidores (procesadores de información). La cola puede ser gestionada usando diferentes políticas de atención FCFS (*primero en llegar, primero en ser atendido*), LCFS (*último en llegar, primero en ser atendido*) o SIRO (*atención en orden aleatorio*).

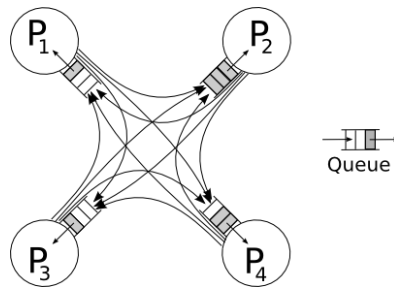


Figura 5.2. Ejemplo de un clúster heterogéneo con 4 nodos. Cada servidor tiene su propia cola para recibir las solicitudes entrantes de los restos de los nodos de la granja.

Usando la notación de Kendall [160], el modelo de colas se define por los siguientes factores:

A / B / C / X / Y / Z, donde:

- A: distribución de llegada de los procesos. Distribución de probabilidad en la llegada de procesos.
- B: distribución del tiempo de atención. Distribución de probabilidad que muestra la distribución de tiempo de servicio de un cliente.
- C: número de servidores.
- X: número máximo de clientes permitidos en el sistema, incluidos aquellos que están activos. Cuando este número sea el valor máximo, las llegadas son rechazadas.
- Y: tamaño de la población (número de procesos).
- Z: disciplina de la cola. Orden de prioridad de los trabajos en la cola.

Por lo tanto, la solución del problema de la optimización de los recursos globales de un clúster heterogéneo, puede ser abordada si se resuelve el problema de encontrar el camino mínimo de la gráfica asociada [161], teniendo en cuenta las esperas derivadas de la comunicación en la red y el tiempo de ejecución de las tareas por cada procesador.

5.2.2 Modelo Matemático

El sistema puede ser modelado como un grafo dirigido ponderado G_s , denotado por $G_s(P, L, \tau, \delta)$, denominado como el **SystemGraph**; P denota un conjunto finito de procesadores que representa los nodos o vértices de la gráfica G_s ; L es el conjunto finito de vértices que representa los enlaces de comunicación entre un par de unidades de proceso: los bordes de la gráfica G_s ; cada vértice $p_i \in P$ se define por un conjunto de parámetros del sistema (memoria, velocidad, sistema operativo...) sobre la base de los recursos disponibles de la HNoC. Debido a esto, cada procesador tiene un peso $\tau(p_i)$ que denota el coste de procesamiento por unidad de cálculo. Cada enlace o arista entre dos procesadores p_i y p_j , denotado por $l_{i,j} \in L$, donde el enlace tiene un peso $\delta_{i,j}$ es la latencia en la comunicación entre dos nodos por unidad de transferencia. Si dos nodos $(p_i, p_j) \in P$ no están conectados uno con otro, entonces $l_{i,j} = \infty$. Se supone que todos los nodos de la gráfica están conectados con al menos otro nodo de la HNoC, pero no se imponen restricciones sobre la topología de la red, ya que esto no está completamente definido y puede variar entre dos momentos en la simulación del problema que se está ejecutando.

Así, es necesario definir una función de vecindad que determinará el conjunto de nodos que están vinculados con cualquier nodo del HNoC,

$$\forall p_i \in P: \text{neig}(p_i) = \{p_k\} \mid l_{i,j} \neq \infty \quad (5-5)$$

Junto a la misma, se define otra función $\text{path}(p_i, p_j)$, así:

$$\forall p_i, p_j \in P: \text{path}(p_i, p_j) = \{p_k\}^* \quad (5-6)$$

Donde p_k^* es un conjunto ordenado de nodos, en el que cada nodo $p_k \in \text{neig}(p_i)$ o $p_k \in \text{neig}(p_j)$ o $p_k^n \in \text{neig}(p_k^{n-1})$.

Es necesario definir la matriz de latencias C_L que contiene la red de latencias entre dos unidades de procesos, $\forall p_i, \forall p_j \in P, C_{L_{i,j}} = lat(p_i, p_j)$, que depende también del enlace físico y el enlace de datos en la capa de red.

Para dos nodos adyacentes $p_i, p_j \in P, C_{L_{i,j}} = lat(p_i, p_j) = \delta_{i,j}$ (pero si p_i y $p_j \in P$ no son adyacentes en el HNo), la latencia se define como la suma de los pesos de los enlaces en el camino más corto entre ellas Figura 5.2:

$$C_{L_{i,j}} = lat(p_i, p_j) = \sum (\delta_{p_k} | P_k \in \min(path(p_i, p_j))) \quad (5-7)$$

La matriz C_L podrá ser simétrica o no, ya que todas las comunicaciones podrían ser diferentes, también en una comunicación dúplex, debido a la propiedad dirigida del grafo.

El grafo de caminos mínimos para un nodo específico del grafo G_s denotado por $G_m^a(P, L, \tau, \delta)$, se define como el grafo que contiene el mínimo de caminos para el nodo $a \in P$ para cualquier nodo del grafo Figura 5.3.

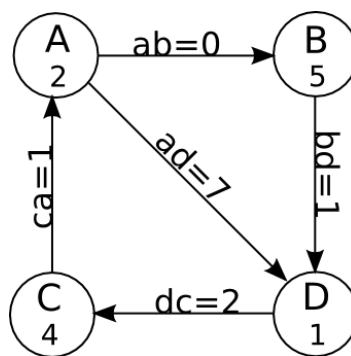


Figura 5.3. Ejemplo para el sistema G_s . Los elementos del grafo son: Nodos: $P = \{A, B, C, D\}$, enlaces $L = \{ca, ab, ad, bd, dc\}$, pesos de procesamiento = $\{2, 5, 4, 1\}$ latencia en la comunicación $\delta = \{1, 0, 7, 1, 2\}$ y vecindario (A) = $\{B, D\}$, $path(C, D) = \{C, A, B, D\} \cup \{C, A, D\}$ y $pathmin(C, D) = \{C, A, B, D\}$

La C_L matriz para este ejemplo es:

$$G_L = \begin{matrix} A \\ B \\ C \\ D \end{matrix} \begin{pmatrix} A & B & C & D \\ 0 & 0 & 3 & 7 \\ 4 & 0 & 3 & 1 \\ 1 & 1 & 0 & 2 \\ 3 & 3 & 2 & 0 \end{pmatrix}$$

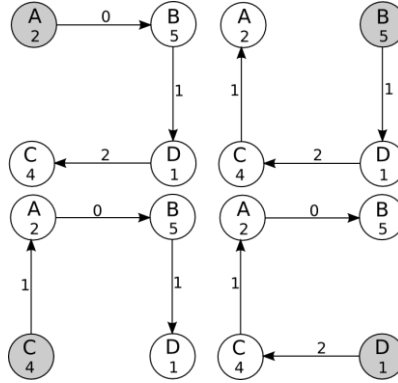


Figura 5.4. Grafos G_m^a obtenidos para cada nodo para el grafo de la Figura 5.3.

La aplicación también se puede modelar usando un grafo dirigido ponderado G_a , denotado por $G_a(T, D, \omega, \lambda)$ definido como el *ApplicationGraph*. T denota un conjunto de vértices del grafo que representa las tareas a realizar y D representa un conjunto finito de bordes del grafo donde $\{(t_i, t_j) | t_i, t_j \in T\}$.

Cada vértice tiene un peso de computación $\omega(t_i), \forall t_i \in T$, que representa la cantidad de cálculo requerido por la tarea de t_i para llevar a cabo un paso del algoritmo. Cada borde tiene un valor de $\lambda_{i,j}$ que representa la cantidad de datos a ser enviados desde v_i a v_j .

Por lo tanto, el tiempo de ejecución Γ de una tarea $t_i \in T$ en un procesador $p_j \in P$, suponiendo el peor de los casos en los que no hay solapamiento entre la computación y la comunicación, se define como:

$$\begin{aligned} \Gamma(t_i, p_j) = & \omega(t_i) \times \tau(p_j) \\ & + \sum_{t_l \in \text{neig}(p_k)} \sum_{\substack{p_k \in P \\ k \neq j}} \lambda(t_i, t_l) \times \delta(p_j, p_k) \end{aligned} \quad (5-8)$$

Donde $\omega(t_i) \times \tau(p_j)$ representa la cantidad de cálculo requerido para la tarea t_i por costo de procesamiento por unidad de cálculo.

Dado un grafo del sistema denotado por $G_s(P, L, \tau, \delta)$ y una aplicación gráfica $G_a(T, D, \varpi, \lambda)$, el objetivo es proyectar las características de $\Gamma: (T, D) \mapsto (P, L)$ para minimizar la función Γ , basado en los requisitos de la aplicación y las limitaciones del sistema, tales como la topología del grafo del sistema.

Desde el punto de vista de la técnica de predicción de colas, se puede asumir que el HNoC puede ser descrito por un modelo M/G/1/ ∞ / ∞ /SIRO, en el que la distribución del proceso de llegada es un proceso de Markov [162]. El tiempo de servicio es una función de distribución general de las llegadas independientes al sistema y sólo hay un servidor para tal cola [163].

El algoritmo permite a cualquier servidor tomar de la cola elementos en un orden arbitrario, de acuerdo con una prioridad determinada. Esta característica fuerza al parámetro SIRO de la cola. Se debe mantener una sola cola por cada servidor pero, desde el punto de vista de todo el sistema, las colas se pueden modelar como un M/G/c/ ∞ / ∞ /SIRO, donde c es el número de nodos del HNoC. La longitud de la cola (número máximo de trabajos en cola) L , dependerá de la tasa de llegada λ , y sobre la tasa de servicio μ y será calculado por la expresión:

$$L = \frac{\lambda^2 E[S^2]}{2(1 - \rho)} + \rho \quad (5-9)$$

Donde $\rho = \frac{\lambda}{\mu}$ y $E[S^2]$ es el segundo momento del valor esperado de la tasa de servicio de distribución S .

5.2.3 Modelo de Potts

El modelo de Potts [35] fue descrito por Renfrey B. Potts en 1952 y es una generalización del modelo de Ising [28]. Este modelo se utiliza para estudiar los *spines* y el comportamiento de los materiales ferromagnéticos, y se ha extendido a la simulación para la formación de burbujas [164] y simulación celular [8, 9],

donde la dinámica incluida busca minimizar la energía global del sistema. La elección de este algoritmo es debida a la imposibilidad de resolver de forma directa el modelo, requiriendo el uso de métodos estadísticos para generar configuraciones que se distribuyan correctamente de acuerdo a la función de distribución de Boltzman, aunque se pueden usar otras [45]. Al obtener N configuraciones posibles, se puede aproximar al comportamiento del sistema.

El modelo de Potts consta de un conjunto de *spines* o índices que se ubican en una rejilla, generalmente euclidiana rectangular de dos dimensiones, pero a menudo se generaliza a otras dimensiones u otros tipos de rejillas. La función que gobierna el modelo es la del *Hamiltoniano*:

$$H = \sum_{\substack{i,j \\ \text{Vecinos}}} J(\tau(\sigma_i), \tau(\sigma_j))(1 - \delta(\sigma_i, \sigma_j)) + \sum_{\sigma} [\lambda_{\text{vol}}(\sigma)(v(\sigma) - V_i(\sigma))^2] \quad (5-10)$$

Donde δ es la función *delta de Kronecker*, de modo que: $\delta(x, y) = 0$ si $x \neq y$ y 1 si $x = y$. Para efectos de la simulación se utiliza un *Hamiltoniano* modificado donde se incluye la restricción de volumen para la simulación de burbujas [164].

El método de Metrópolis – Monte Carlo usa la distribución de Boltzmann como función de aceptación de los cambios, que se define como:

$$P(\sigma \rightarrow \sigma') = \begin{cases} e^{\frac{-\Delta H}{kT}}, \Delta H > 0 \\ 1, \Delta H \leq 0 \end{cases} \quad (5-11)$$

Donde K_B es la constante de Boltzman y T es un valor de temperatura determinado.

La evolución del modelo se hace de la siguiente manera:

1. Se elige al azar un *spin* de la rejilla.
2. Se elige al azar un *spin* vecino de la posición actual.
3. Si el *spin* vecino es igual al espín actual, hay que volver al paso 1.
4. Se cambia el valor actual del *spin* elegido con el valor del *spin* de su vecino de acuerdo con la probabilidad P de la expresión 2.
5. Volver al paso 1.

La rejilla se representa de manera toroidal y se actualiza continuamente: para cada punto de la rejilla se propone un estado de *spin* diferente, y se calcula el valor energético si el cambio se diese. Depende de las interacciones de los vecinos y de la temperatura global. Si la nueva energía es más pequeña que la anterior, se acepta el nuevo estado (minimización de la energía). Si no es así, todavía hay una cierta posibilidad de que sea aceptada, lo que lleva a la selección de un *spin* de manera aleatoria.

5.2.4 Modelo de División Dinámica de Cargas de Trabajo de la Rejilla en un Hnoc

El modelo de Potts tiene tres desventajas:

1. A nivel de su eficiencia computacional, el modelo se implementa usando una simulación de Monte Carlo con un algoritmo de Metrópolis estándar. Como se puede observar, tan sólo se intenta actualizar la rejilla una vez en cada momento durante la evolución del modelo: es decir, la parte crítica del algoritmo reside en el punto 1.
2. El algoritmo de Metrópolis también posee una segunda desventaja documentada [25]. Cuando se acerca a la transición de fase, el número de intentos de cambio de *spin* que se requieren para generar una configuración estadísticamente independiente es muy alto, es decir, se deben realizar un mayor esfuerzo y por ende gasto de CPU.
3. Generación de números aleatorios.

Estos inconvenientes son los factores que consumen el tiempo de procesamiento y ralentizan el cálculo de la rejilla.

5.2.5 Implementación del Algoritmo y Balanceo de Carga en un Clúster Heterogéneo

La ejecución del modelo de Potts en serie considera a la rejilla estática e invariable en el tiempo. La ejecución de nuestro modelo propone una descomposición por dominios que pueda ser atendida por varios nodos. Esta distribución no se define necesariamente antes de la ejecución en el HNoC. El algoritmo permite una estructura dinámica que ajusta la repartición de los datos de acuerdo a la capacidad de cada nodo y de acuerdo a las diferentes latencias de la red y su dependencia con las diferentes conexiones entre los nodos.

El modelo permite la reconfiguración de los nodos de un HNoC basado en un mecanismo de paso de mensajes. El uso de este entorno permite a cualquier nodo ser capaz en tiempo real de obtener información sobre el contenido de cualquier otro nodo del sistema, sin tener que comunicarse primero con el nodo maestro del sistema. Esto también permite a los nodos modificar su rol de maestro/esclavo en función del valor o la estructura de datos requerida Figura 5.3.

El HNoC se representa como un grafo dirigido dinámico Figura 5.2, en el que las diferentes conexiones tendrán pesos en los bordes y en los vértices del grafo. La información de cada nodo en el grafo contendrá la carga efectiva del procesador, teniendo en cuenta la disponibilidad para ejecutar nuevos procesos. También el nodo contendrá un valor estadístico proporcional al tiempo de ejecución de los procesos en las etapas anteriores del algoritmo. Por lo tanto, las aristas de la gráfica representan las conexiones entre los procesadores disponibles, de acuerdo con un cierto instante del algoritmo.

Estas conexiones se ponderan estadísticamente de acuerdo con los retrasos de la red y el tiempo de transmisión de los datos que dependen del protocolo de red de y la capa física utilizada para esta comunicación. Estos dos valores serán importantes para el cálculo de la optimización de la función histórica.

Las tramas de datos de control también permiten a cualquier nodo unirse o salir del clúster dinámicamente, así como el envío de un comando específico para el nodo *supermaster*, para ser incluido o ser suprimido del conjunto total de los nodos. Cuando un equipo se une al grupo, el *supermaster* calculará de nuevo el grafo de caminos mínimos para que cada nodo del clúster reconozca al nuevo nodo. En caso contrario, este cálculo se realiza con los nuevos parámetros del sistema. Cuando un nodo abandona la agrupación que le aconsejará el nodo *su-*

permaster, el proceso de anulación del registro hará nuevamente una reorganización de los nodos de la HNoC.

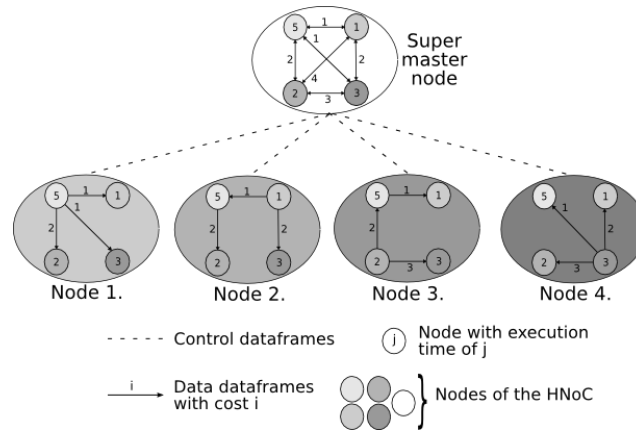


Figura 5.5. HNoC formado por 5 nodos, 4 maestro/esclavo y 1 nodo *supermaster*. El supernodo contiene el grafo completo con toda la comunicación posible entre cada nodo y el valor estadístico de las esperas y la carga efectiva de cada procesador.

La principal ventaja de este sistema consiste en evitar el diseño de un HNoC por parte del administrador de la agrupación. Esta persona, sin este sistema, debe diseñar la arquitectura y la topología de HNoC en función de los accesos que se necesitan, dependiendo del problema considerado. Además, este modelo será muy útil para eliminar la necesidad de distribuir los recursos entre los diferentes nodos, en función de los recursos disponibles.

La motivación principal consiste en generar de forma dinámica una partición (un conjunto de dominios) que cubra la rejilla de datos y se ajuste de manera óptima al problema Figura 5.6. Para esto, el tamaño de cada dominio generado (número de tareas asignadas por el maestro a cada proceso esclavo) dependerá del resultado de la función estadística determinada por el **RMS**. Estas funciones se basan tanto en las propiedades físicas del nodo y la topología de red, como en las propiedades relacionadas con el tiempo (valores históricos).

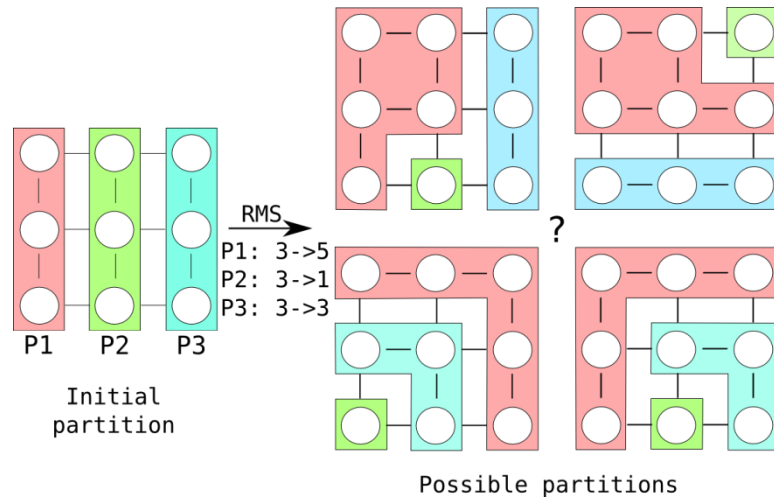


Figura 5.6. A partir de una distribución de datos arbitraria de una cuadrícula de 3×3 (cuadrícula con 3 procesadores).

El valor RMS nos da sólo el número de tareas que se asignarán a cada procesador, pero no sabemos nada acerca de la mejor distribución de los datos de la aplicación.

El número total de las tareas asignadas a cada procesador depende del valor RMS calculado por el nodo supermaster de la HNoC. Pero las diferencias en el tamaño del dominio entre los pasos de la simulación implican una variación de la distribución de los datos en el desarrollo de la simulación. Por lo tanto, es necesaria para determinar la mejor topología de datos para cada paso del algoritmo, teniendo en cuenta los requisitos de datos y simetrías que también dependen del problema resuelto en la HNoC.

El algoritmo genera tantos grupos como procesos esclavos estén involucrados en la simulación, y cada uno tendrá la mayor cantidad de tareas de acuerdo al valor de la función RMS obtenida para tal nodo o procesador. Cada grupo contiene algunos vértices de la *ApplicationGraph* y sus dependencias. El algoritmo de búsqueda puede basarse en las funciones de anchura, profundidad u otras. La exactitud de esta solución dependerá de este algoritmo.

Debido a la partición de datos del problema, si un determinado procesador no cumple sus requisitos de datos, éste enviará un mensaje al proceso maestro del clúster (no se permiten mensajes entre esclavos), el cual recibe el valor solicita-

do y es capaz de continuar con su cálculo. El maestro asignará esta dependencia en una estructura local. Cuantos más mensajes envíe el esclavo al maestro, más lento será el algoritmo y la simulación. Por lo tanto, dependiendo de la asignación de la rejilla de datos en los nodos del clúster será mejor el rendimiento obtenido. Una vez que el algoritmo de búsqueda ha sido ejecutado y el orden de los procesadores determinados en la matriz, la descomposición de dominios determina la mejor configuración que controle la mayor cantidad de tareas específica para cada procesador, el algoritmo de agrupamiento determinará la mejor topología de datos después de cada paso del algoritmo, teniendo en cuenta sus dependencias.

5.2.6 Resultados

Todos los experimentos en este trabajo se han realizado utilizando el modelo definido en [164]. Los parámetros de simulación utilizados fueron: temperatura = 2 y temperatura = 15. Rejilla (DxD) $D = 8$, $D = 16$, $D = 32$, $D = 64$ y $D = 128$. Energía de adhesión entre burbujas $J = 3$ y la energía de adhesión entre las burbujas y el medio = 1. El número de burbujas aproximado de acuerdo con la medida de la rejilla fue: 10 para $D = 8$, 40 para $D = 16$, 80 para $D = 32$ y 200 para $D = 128$. Un ejemplo del comportamiento de la simulación se puede ver en la Figura 5.7 para $D = 128$. La simulación después de 5000 MCS.

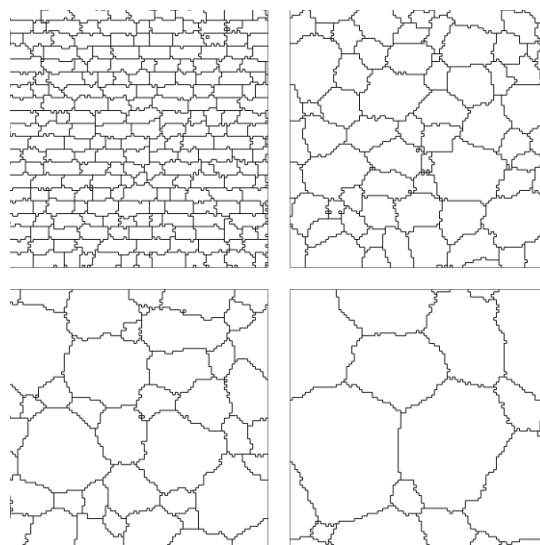


Figura 5.7. Simulación de burbujas usando el modelo de Potts

El paralelismo de nuestro algoritmo se puede obtener fácilmente con la explotación de la naturaleza concurrente del problema a través de la ejecución simultánea de tantos procesos esclavos como se deseen.

Si dos dominios necesitan información de una cierta posición en la rejilla para la ejecución de un paso en su simulación, y dicho proceso no tiene los datos requeridos, este le pedirá al maestro este valor. El envío por parte del maestro de una solicitud al esclavo lo hará de manera recursiva). Teniendo en cuenta que el número de lanzamientos es una función de la temperatura de la simulación, la probabilidad de que dos dominios calculen los mismos datos es baja, y si este valor es bajo la probabilidad de que dos esclavos deban comunicarse entre sí también es baja. Por lo tanto, nuestra simulación será útil cuando la temperatura de la rejilla sea baja. Por otro lado, si la temperatura es alta, el tiempo empleado para comunicar dos esclavos, también lo será, en este caso el maestro actuará como proxy.

El modelo simulado se ha resuelto mediante la ejecución de tres algoritmos diferentes: la versión monoprocesador, la versión en paralelo (sin necesidad de utilizar este *middleware*) y la versión paralela (usando el sistema de reconfiguración de datos propuesto). Si se compara estos tres algoritmos programados, el tiempo medio empleado (en segundos) para la ejecución de 5000 MCS, variando la dimensión de la red de D , se muestra en las tablas 1 y 2. El incremento de tiempo entre ambas simulaciones, variando el valor de la temperatura, provoca un incremento de los paquetes enviados entre los procesos esclavos y maestro, disminuyendo el rendimiento de la simulación.

Algoritmo	D=8	D=16	D=32	D=64	D=128
Monoprocesador	2,15	6,62	13,48	30,46	62,17
Sin reconfiguración	1,41	3,12	6,28	13,02	38,76
Con reconfiguración	1,29	2,85	5,18	10,46	29,29

Tabla 5.1. El tiempo de ejecución, en segundos, de la versión de tres del mismo algoritmo, 5000MCSs. El algoritmo de agrupación utilizado en esta ejecución fue

la DFS. Temperatura = 2. La mejora alcanzada con este ejemplo era casi el 112%

Algoritmo	D=8	D=16	D=32	D=64	D=128
Monoprocesador	2,15	6,62	13,48	30,46	62,17
Sin reconfiguración	3,23	5,97	9,71	17,42	41,89
Con reconfiguración	3,19	5,15	8,16	15,06	38,29

Tabla 5.2. El tiempo de ejecución, en segundos, de la versión de tres del mismo algoritmo, 5000MCSs. El algoritmo de agrupación utilizado en esta ejecución fue la DFS. Temperatura= 15

*-Nunca ponga en duda la verdad de lo que usted no puede entender,
porque el mundo está lleno de maravillas."*

L. Frank Baum , Rinkitink en Oz



Capitulo 6 Simulación de la epidermis Humana Usando GGH-CPM

La modelización de los fenómenos celulares lleva en continua evolución durante los últimos años, debido a sus aplicaciones en diversos campos de la ciencia médica [44, 165]. Los modelos informáticos y las simulaciones son útiles para probar hipótesis experimentales, rápidos de implementar y fáciles de modificar. La identificación y comprensión de los mecanismos a nivel celular que dan lugar a la formación de estructuras complejas, tanto en el crecimiento normal y anormal (tumoral), hacen que sea posible proporcionar ayuda básica a la búsqueda de nuevos tratamientos para el cáncer y otras enfermedades. Las simulaciones por ordenador también hacen que sea más fácil evaluar o interpretar los datos clínicos, ayudan a desarrollar nuevas hipótesis y provocan el avance en la investigación experimental.

A pesar de su complejidad molecular interna, las células como unidades globales son capaces de realizar un conjunto de tareas colectivas, tales como agruparse entre ellas, moverse en una dirección dada, cambiar formas, interactuar con otras células, secretar y absorber materiales, diferenciarse, crecer, regenerarse y morir. Se propone un modelo de cicatrización epidérmica (GGH-SKIN) basado en el modelo celular de Potts o GGH [8, 9, 45, 166], implementados en el entorno de la modelización celular CompuCell3D. GGH-SKIN implica varios parámetros, tales como: (1) diferentes estados del ciclo celular de células proliferativas; (2) diferentes tipos de división celular (simétrica o asimétrica) de células en proliferación; (3) mecanismos para la activación y la inhibición de la migración celular durante el cierre de la herida; (4) mecanismos de pérdida de células en la capa superior (escamosas).

Esta propuesta no es el único enfoque de la aplicación de técnicas de simulación discreta para modelar la dinámica de la epidermis humana.

En [167] se describe un modelo multi-agente que incluye diferentes elementos continuos (en la forma de resolver ecuaciones diferenciales) utilizado tanto para imitar el comportamiento de células epidérmicas en la homeostasis, como para estudiar el linaje de células madre. Nuestro modelo, sin embargo, es único, ya que encarna la forma de las células individuales, proporcionando de este modo al investigador una representación morfológica realista del tejido.

La piel está compuesta por tres capas principales o componentes de tejido de acuerdo con su profundidad: epidermis (epitelio), dermis (tejido mesenquimal) e hipodermis (grasa). La epidermis humana (la capa externa de la piel), está provista de mecanismos homeostáticos que mantienen la estructura de la piel y un equilibrio constante entre la multiplicación celular y la pérdida de células [111].

La epidermis se compone de un conjunto de capas de células (queratinocitos) sujetos a un proceso de auto-renovación continua. En su viaje desde la capa más profunda a las capas superiores de la epidermis, los queratinocitos se someten a un proceso de diferenciación que incluye tanto los cambios estructurales como morfológicos [111].

La epidermis se divide en varias capas de acuerdo a la morfología de los queratinocitos y en función de su ubicación: capa basal, capa espinosa, capa granulosa y capa escamosa (córnea), como ya se ha descrito de manera introductoria en el capítulo 4.

La **capa basal** es responsable de la regeneración permanente, como resultado de su división mitótica constante. Se cree que los queratinocitos sólo se dividen en la capa basal, y la división celular es suprimida ya que sufren el proceso de estratificación en las capas suprabasales [122]. Normalmente se compone de una sola fila de queratinocitos que dependen de la interacción con la membrana basal para dividirse [168]. Sin embargo, en condiciones hiperproliferativas, tales como la curación de heridas, la capa proliferativa parece expandirse.

La epidermis está sujeta a una continua auto-renovación permitida por las células madre epidérmicas (Stem Cell - SC) que residen en la capa basal [169]. Es-

tas células residen en sitios específicos, no se dividen con frecuencia y tienen una capacidad ilimitada para proliferar, pero su ciclo celular es generalmente lento. Las células de división rápida (denominadas clásicamente células amplificadoras de tránsito o TAC) se originan a partir de células madre, pueden migrar a lo largo de la capa basal y son las responsables de iniciar la diferenciación terminal después de unas cuantas rondas de división celular [120]. Existen pruebas experimentales que muestran que hay diferentes tipos de células proliferantes en la epidermis, es decir, las células madre y las células de amplificación transitoria (TA)[170]. Las células madre no son terminales diferenciadas y pueden proliferar sin límite a lo largo de toda la vida del organismo. Las células TA derivan de las células madre, han limitado el potencial proliferativo y se someten a eventualmente a la diferenciación terminal. Las células SC y TAC componen la capa basal proliferativa. Se han realizado diversos estudios para determinar la densidad de los diferentes tipos de células [46, 171].

Existen dos teorías actuales que explican el proceso de renovación epidérmica. La primera teoría asume que la piel está formada por una colección de estructuras de elementos finitos llamados "unidades" de proliferación de la epidermis (EPU) [120]. Dicha teoría consiste en que una célula madre se divide ilimitadamente y simétricamente o asimétricamente. Las células resultantes de estas divisiones, denominadas células amplificadoras de tránsito (TAC), se rigen por un número limitado de divisiones asimétricas.

La segunda teoría más reciente [121] sostiene que la epidermis se mantiene gracias a una única población de células madre en la capa basal, donde las divisiones celulares son simétricas cuando tienen la intención de mantener esta capa, y asimétricas cuando generan una progenie diferenciada. En contradicción con el modelo de EPU, que predice un tamaño finito para el EPU, estos conjuntos pueden variar con el tiempo.

También hay un debate sobre el mecanismo responsable del movimiento celular hacia arriba dentro de la epidermis. Probablemente, la razón fundamental es la actividad mitótica, la cual provoca un movimiento pasivo de las células al estratificar, debido a la pérdida de la adhesión celular y debido a que las células basales se ven impulsadas por presión proliferativas [123, 172]. Asimismo, el estado

del ciclo celular de los queratinocitos durante la diferenciación epidérmica es objeto de debate [122, 173, 174].

6.1 Heridas en la Epidermis

Cuando la epidermis sufre un daño producto de algún suceso, por ejemplo, después de un corte, una quemadura o abrasión, la piel inicia un proceso de reparación para restaurar la homeostasis del tejido. Dependiendo de la magnitud de los daños, la piel puede ser reparada y devuelta a su estructura y función antes de la lesión. Dos procesos principales tienen lugar durante la cicatrización: la migración de células dentro de la herida y el aumento de la tasa de proliferación celular, para reemplazar células perdidas, donde se aumenta la actividad de las células madre en los bordes de la herida [136]. Esto estimula el poder mitótico de las células en el borde, posiblemente se puede explicar por medio de la pérdida de coacción física sobre la mitosis en ausencia de células vecinas [122, 175].

La activación de las células madre epiteliales y de mecanismos de control eficientes de sus descendientes, son críticos para la curación de heridas. En este sentido, los mecanismos moleculares de las células multipotentes, la naturaleza del microambiente de la herida, su composición celular y factores usados durante la señalización, determinarán el proceso de cierre. Una comprensión más profunda del proceso de re-epitelización de heridas normales proporcionará sin duda un mayor conocimiento de diversas patologías que pueden hallarse en la cicatrización de heridas.

Uno de los pasos clave en el proceso de cicatrización de la herida es el momento en el que las células madre del folículo piloso cambian su destino y migran hacia la epidermis, asegurando así una rápida afluencia de progenitores que se difunden alrededor de la herida. Aunque se han mostrado múltiples vías de señalización que ayudan a la proliferación y la migración de los queratinocitos de la epidermis en la herida, están empezando a ser descubiertos los factores que desencadenan, de manera coordinada, el destino de las células madre de conmutación hacia el linaje epidérmico en el inicio de la cicatrización de heridas [176].

Existe incertidumbre sobre la ubicación de las células madre. Algunos estudios sugieren que pueden estar presentes en dos áreas: en la región folicular (abultamiento) y en los grupos formados en la capa basal de la epidermis entre folículos, ya sea en la parte inferior o en la cresta de la papila dérmica, dependiendo del sitio del cuerpo [124, 125].

Cuando la epidermis está herida, se produce una latencia de varias horas antes de que las células situadas en la capa basal más cercana a la herida inicien su migración hacia este lugar. Durante el cierre de la herida, la población de células forma una "onda viajera" continua, con un estrecho contacto entre las células detrás del frente de onda [177]. Las células cambian de tamaño y desarrollan lamelipodios para moverse, tratando de llenar el espacio vacío en la herida. Las células basales comienzan un proceso de división acelerado para reemplazar las células migradas. La migración es completa cuando las células de la epidermis cubren la herida y se forma una nueva capa exterior. Durante la curación, la epidermis sufre diversos fenómenos, tales como un período de inflamación que inicia un proceso antibacterial. Esto da lugar a un cambio en la profundidad de las diferentes capas en la epidermis, cuyas células cambian de forma y se hacen más grandes. La capa basal, por ejemplo, puede hincharse varios micrómetros. Al final del proceso, las diferentes capas recuperan su tamaño anterior y su estado.

La migración y la proliferación celular también se activan mediante una combinación de moléculas de señalización conocidas como factores de crecimiento, que pueden ser producidos por las células en la epidermis o la dermis. Existen tres formas principales por las que los factores de crecimiento controlan los procesos celulares relacionados con la cicatrización de heridas[136, 138]:

1. Los factores de crecimiento que actúan como atrayentes, estimulando a las células para migrar a la zona de la herida (quimiotaxis).
2. Los factores de crecimiento que inducen a las células a dividirse y proliferar.
3. Los factores de crecimiento que estimulan la actividad celular.

Numerosos factores de crecimiento se han identificado para regular la cicatrización de heridas epidérmicas [137, 178], principalmente: factor de crecimiento epidérmico (EGF), factor de crecimiento de queratinocitos (KGF), factor de crecimiento de fibroblastos (FGF), factor de crecimiento transformante alfa (TGF- α), factor transformador del crecimiento beta (TGF- β) y factor de crecimiento derivado de plaquetas (PDGF).

Diferentes modelos se han desarrollado para explicar el comportamiento de cierre de heridas epidérmicas. Varios de ellos se basan en la ecuación de Fisher, que ayuda a describir y capturar, en términos de motilidad celular o difusión, la migración y la proliferación celular. Uno de los modelos más relevantes de este grupo fue propuesto por Sherratt y Murray [179, 180]. Usando un modelo muy simple, se pueden realizar predicciones de la velocidad de la curación y el efecto de la ubicación de factores de crecimiento y la forma inicial de la herida. El modelo predice cómo la velocidad de la curación se ve afectada por la adición de un factor de crecimiento específico.

Como alternativa a estos modelos continuos existe una propuesta discreta [181], que representan a cada célula individual de forma explícita mediante el uso de un camino aleatorio en la dirección de un estímulo.

Nuestro modelo, sin embargo, es único, ya que encarna la forma de las células individuales, proporcionando de este modo al investigador una representación morfológica realista del tejido. Nuestro enfoque está centrado en la célula: modelamos con la mayor precisión posible las características de las células reales porque consideramos que el comportamiento de la epidermis completa debe surgir de las células particulares. Estas características incluyen el tamaño, la forma y la ductilidad de la célula.

6.2 Descripción Informal de un Modelo para el Cierre de Heridas Epidérmicas

Una red de células bioquímicas puede ser considerada como un conjunto de unidades de cómputo (agentes) que interactúan y cooperan para lograr un objetivo común. Durante el análisis y el diseño del modelo que presentamos aquí, deci-

dimos elegir una aproximación discreta donde las células se distribuyen en una rejilla en la que los procesos macroscópicos tienen lugar.

El modelo incorpora las siguientes características, tomadas de la bibliografía:

- La epidermis es un tejido epitelial compuesto por diferentes tipos de células distribuidas en varias capas. Las diferentes capas tienen distintos tamaños y también pueden variar de acuerdo al estado del tejido: normal o de recuperación de una herida.
- En la epidermis, las células madre, junto con las células amplificadoras de tránsito, son responsables de mantener el equilibrio entre la renovación y muerte, tanto en condiciones normales como en las heridas. Están ubicadas en lugares concretos en la capa basal. Suponemos que las normas que regulan la renovación de tejidos son las propuestas en el modelo descrito anteriormente [120].
- La epidermis muestra diferentes comportamientos de proliferación. Las células madre y las células de amplificación transitorias siguen un ciclo celular constante, moviéndose a través de diferentes estados (G2/S/G2/M) antes de dividirse. Las células madre pueden dividirse sin límite, mientras que las células amplificadoras transitorias pueden dividirse de tres a cinco veces. Las células diferenciadas no pueden dividirse.

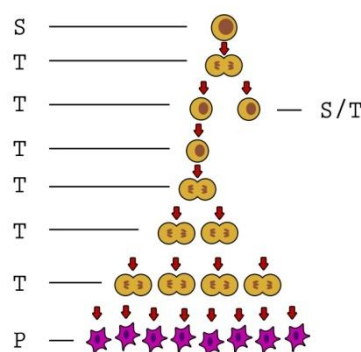


Figura 6.1. Ilustración 1. Mitosis Asimétrica / Simétrica

- Las células capaces de dividirse pueden experimentar la mitosis simétrica o asimétrica. Los ciclos de SC y TAC tienen una duración diferente, siendo más larga para las células madre. Mientras que se está reparando una herida, la tasa de proliferación de ambos tipos de células puede aumentar quince veces.
- Los diferentes tipos de células pueden adherirse unas a otras, mientras que algunas células pueden ser atraídas o repelidas más que otras. Por ejemplo, las células madre tienden a adherirse a la capa basal y entre ellas mismas, creando de este modo grupos de células madre.
- La membrana extracelular juega un papel importante en el tejido epidérmico. Cada tipo de célula debe tener diferentes niveles de fluctuación en su membrana, que a veces pueden hacer que sea más fácil el movimiento de la célula a través del tejido, o provocar que sea rígido y fijo en una posición dada.
- La actividad mitótica es el mecanismo responsable del flujo normal de las células en el proceso homeostático del tejido, debido a la presión que genera fuerza en las células para moverse de forma pasiva.
- En el proceso homeostático normal las células migran hacia las capas superiores siguiendo un camino vertical único. Cuando la epidermis sufre una herida (se pierde una parte del tejido), se inicia un nuevo proceso que estimula las células cercanas, empujándolas para llenar el espacio vacío. El espacio libre de las células estimula las cercanas a moverse, y se dividen más rápidamente (inhibición por contacto). En este caso, las células se mueven en ambas direcciones (horizontal y vertical), manteniendo su camino hacia arriba junto con el llenado normal de la herida. El movimiento de las células en las inmediaciones de la herida se convierte en ondulado.
- Una vez que las células amplificadoras transitorias han alcanzado su número máximo de divisiones comienza un proceso de diferenciación, mientras que avanzan hacia las capas superiores del tejido. En su ascenso

hacia la capa escamosa (estrato córneo), estas células aumentan su volumen y superficie.

6.3 Materiales y Métodos

6.3.1 COMPUCELL 3D

Es muy complejo aplicar el modelo CPM-GGH cuando los investigadores expertos pretenden jugar con diferentes parámetros para obtener resultados que permitan poner a prueba sus hipótesis. El equipo de J. Glazier, en el centro de bio-complejidad de la Universidad de Indiana, ha desarrollado un *software* de simulación de múltiples celdas llamado CompuCell3D (CC3D) [143, 182], que incorpora en su base el modelo de CPM-GGH. Los usuarios pueden especificar sus modelos utilizando lenguajes meta-etiquetados XML, y ejecutar sus simulaciones en un entorno gráfico. CompuCell3D hace que sea posible el uso de lenguajes de script para extender sus características y construir simulaciones más complejas.

Las simulaciones se han implementado usando CompuCell3D, que es una herramienta de *software* de código abierto diseñado para simular modelos celulares basados en el modelo (GGH-CPM)[9, 45].

Un desafío clave en la biología moderna es entender cómo las máquinas a escala molecular llevan a estructuras funcionales complejas en la escala de los tejidos, órganos y organismos. Mientras el experimento proporciona la verificación final de hipótesis biológicas, los modelos y simulaciones por ordenador subsecuentes son cada vez más útiles para sugerir ambas hipótesis y experimentos para probarlos. La identificación y cuantificación de las interacciones a nivel de células que desempeñan un papel vital en la formación de patrones, ayudará a la búsqueda de tratamientos para enfermedades como el cáncer y para las técnicas de desarrollo de nuevas estructuras celulares.

A diferencia de los experimentos, los modelos son rápidos de desarrollar, no requieren aparatos costosos y son fáciles de modificar. Sin embargo, la abstrac-

ción de la complejidad de las células o los tejidos vivos en un relativamente simple formalismo matemático/computacional es difícil. A través de CompuCell3D se pueden asignar propiedades o atributos a las células (atributos de células). CompuCell3D permite a los usuarios crear modelos sofisticados con facilidad y rapidez usando un código personalizado, permitiendo extender funcionalidades al usar lenguajes de alto nivel como C++ o *Python*. También facilita el hecho de que el enfoque del modelo pueda ser reutilizado y a la vez ser susceptible de ser compartido.

Un modelo CC3D consta de *scripts* CC3DML (formato basado en XML), *scripts* de *Python* y archivos específicos de configuración inicial de las células en la rejilla. El fichero CC3DML especifica los parámetros básicos: como las dimensiones de la rejilla GGH, los tipos de células, los mecanismos biológicos y la información auxiliar (tales como rutas de archivos). Los *scripts* de *Python* monitorizan el estado de la simulación e introducirán los cambios en los comportamientos celulares.

CompuCell3D es modular, cargando sólo los módulos necesarios para un modelo en particular. Los módulos que modifican el *Hamiltoniano* (modifican la función de energía) o gestionan los eventos de la rejilla se denominan *plugins*. Debido a que los *plugins* son los que se invocan con mayor frecuencia, están codificados en C++ para no comprometer su velocidad.

Existen unos segundos módulos generalmente denominados *steppables* que realizan operaciones en las células de la simulación. Los *steppables* se llaman a intervalos fijos medidos en pasos de Monte-Carlo. Los *steppables* tienen tres usos principales:

- Ajustar los parámetros de células en respuesta a los eventos de la simulación.
- Resolver las ecuaciones diferenciales en derivadas parciales implicadas en algunos procesos que pueden ser representados de esta manera (como la difusión de quimiotácticos en el medio).
- Simular condiciones iniciales para cargar o guardar los resultados de la simulación.

La mayoría de *steppables* son implementados en *Python*. Gran parte de la flexibilidad de CC3D proviene del uso de este enfoque.

El *kernel* CC3D se apoya la computación paralela de arquitecturas de memoria compartida (a través de OpenMP), que proporciona aceleraciones considerables en equipos multi-core.

CC3D separa la representación gráfica del proceso de simulación. Para ello utiliza el módulo CC3D Player. CC3D Player es una interfaz gráfica que ejecuta modelos CC3D. Permite a los usuarios cambiar los parámetros del modelo durante la ejecución, definir múltiples visualizaciones 2D y 3D de la rejilla y realizar el análisis de la simulación a tiempo real.

6.3.2 Modelo CPM-GGH Para Simulación Epidérmica

El modelo Glazier-Graner-Hogeweg (GGH), también llamado el modelo celular de Potts (CPM), fue presentado por F. Graner y J. Glazier en 1993. El modelo original fue diseñado para simular la adherencia y reordenamiento de las células. El lector puede revisar el capítulo 2 de esta tesis para un mejor entendimiento del modelo y sus fundamentos teóricos.

El CPM-GGH es un método eficiente que se puede utilizar a nivel macroscópico para modelar sistemas y tejidos celulares mediante la reproducción de su dinámica. Básicamente, consiste en una rejilla de sitios (o píxeles) distribuidos en una red, varios de los cuales conforman las células de uno o más tipos. A cada píxel (i, j) se le asigna un número entero x que determina si pertenece a una célula (x) o no y, a su vez, $\tau(\sigma(x))$ denota el tipo de célula con σ identificación (x) . Un valor de $x = 0$ representa el medio en el que las células pueden moverse (matriz extracelular - ECM).

Muchos modelos se han desarrollado utilizando el modelo de CPM-GGH: modelos ferromagnéticos, modelos de burbujas y varios sistemas biológicos como el crecimiento de un tumor [177], y el desarrollo del sistema vascular [183], entre otros.

Como se ha mencionado antes, el centro del modelo de CPM-GGH es su *Hamiltoniano*, donde se especifica el comportamiento y las interacciones del sistema a nivel celular y multi-celular.

El siguiente hamiltoniano describe las interacciones de la epidermis del modelo:

$$H = H_{\text{adhesión}} + H_{\text{surface}} + H_{\text{volume}} - H_{\text{External Potentia}} \quad (6-1)$$

Donde,

$$H_{\text{adhesion}} = \sum_{\substack{\bar{i}, \bar{j} \\ \text{neighbors}}} J(\tau(\sigma_{\bar{i}}), \tau(\sigma_{\bar{j}}))(1 - \delta(\sigma_{\bar{i}}, \sigma_{\bar{j}})) \quad (6-2)$$

+

$$H_{\text{surface}} = \sum_{\sigma} [\lambda_s(\sigma)(s(\sigma) - S_t(\sigma))^2] \quad (6-3)$$

+

$$H_{\text{volume}} = \sum_{\sigma} [\lambda_v(\sigma)(v(\sigma) - V_t(\sigma))^2] \quad (6-4)$$

-

$$H_{\text{ExternalPo tential}} = \sum_{\sigma} \bar{\lambda}_{\text{External Potential}}(\sigma)(\vec{x}_{\text{Destinatio n}} - \vec{x}_{\text{Source}}) \quad (6-5)$$

Los parámetros que aparecen en la fórmula (6-2) y (6-3), $s(\sigma)$, $S_t(\sigma)$, $v(\sigma)$, $V_t(\sigma)$, son expresados en número de píxeles.

Como se dijo anteriormente, cada célula tiene un solo descriptor $\tau(\sigma)$, el cual denota el tipo de célula σ con identificador $\sigma(x)$. En el modelo CPM-GGH el parámetro T define la fluctuación de la membrana celular.

Es un hecho bien conocido que no todas las células en la epidermis son igualmente capaces de realizar el movimiento. Para incluir esta observación en la definición del modelo, consideramos diferentes valores de T . Los valores son: $\tau(\sigma) = \{1, 4, 5, 4, 2, 1\}$, donde valores bajos ($\tau(\sigma) \rightarrow 0$) representan la inexistencia de movimiento.

La simulación se lleva a cabo en una rejilla rectangular bidimensional de 200 x 100 píxeles, donde cada píxel es un hexágono. Esta cuadrícula corresponde a una sección de tejido de $20 \times 10 \mu\text{m}^2$ de una herida *in vivo* (como se muestra en la). Para asegurar que la porción de tejido simulado aquí no está aislado, hemos impuesto condiciones de contorno coloidales al eje horizontal de la cuadrícula, de tal manera que los píxeles $x = 0$ es un vecino del píxel $x = 100$.

Hemos dividido la epidermis en cuatro capas diferentes que contienen cinco tipos distintos de células: $\tau(\sigma) = \{SC, TAC, SpinousCells, GranularCells, StratumCells\}$. El tipo de célula y la célula que pertenece a la capa específica se determinan por el centro de masa de la célula en el plano xy .

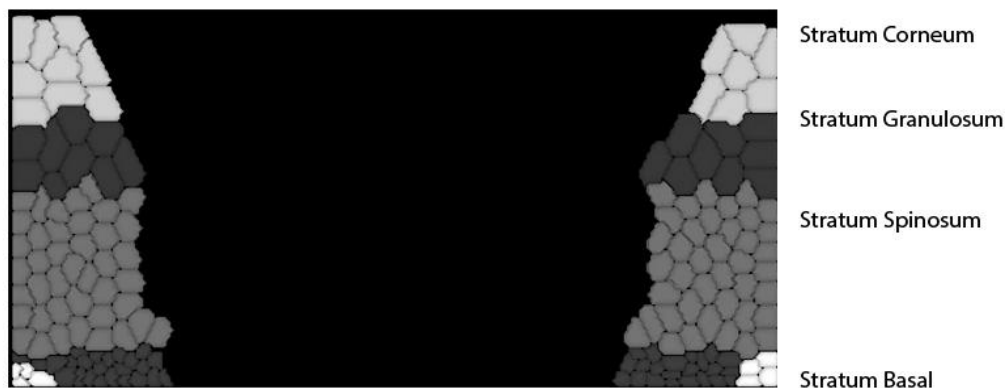


Figura 6.2. Condiciones iniciales de la simulación epidérmica

No se ha impuesto ninguna restricción en el tiempo de simulación en términos de Montecarlo. El número de pasos en la simulación se ha ajustado para observar la evolución de cierre de la herida y, posteriormente, el proceso homeostático normal de la epidermis. Para evaluar los cambios propuestos de píxeles en la

cuadrícula, evaluamos vecinos cercanos del segundo grado, garantizando así la forma isotrópica de las células

Para que el modelo sea consistente a nivel celular, se ha especificado una matriz energética de contacto que incorpora la adhesión y repulsión entre las células; esto es importante para la formación de tejidos en los organismos complejos. Se necesita adhesión y fuerzas de repulsión entre los diferentes tipos de células para asegurar la formación de tejidos en los organismos complejos [184, 185]. Esto se logra por medio de la siguiente matriz de energía de contacto:

Cuando los valores más bajos de energía de contacto corresponden a una adherencia más fuerte, estos valores se han ajustado empíricamente para el comportamiento adecuado del modelo.

	SC	TAC	SpC	GC	CC
SC	1	30	30	30	30
TAC		10	20	10	10
SpC			10	10	10
GC				10	10
CC					10

Tabla 6.1. Valores de unión para J

En la simulación, el volumen de destino de la célula se define como V_{target} . Para cada celda, el volumen de destino varía de acuerdo con su diferenciación: SC, TAC, SpC, GC, CC = {20, 30, 70, 80, 80}.

La elasticidad de la membrana celular se controla por su área de superficie S_{target} . Para mantener su forma, la superficie de todas las células se define como θ^*

$n * V_{\text{target}}$ donde $\theta = \{2,2,2,3,4\}$. En el modelo de GGH, la restricción de volumen evita la desaparición de la célula, y el término dedicado a la superficie controla la forma de la célula (al menos en promedio).

En nuestro modelo, el ciclo celular se incorpora como una extensión del *Hamiltoniano* clásico. El área celular y el perímetro se incrementan por medio de un factor de crecimiento, hasta que se duplican sus valores originales. Entonces, la mitosis se produce siguiendo el eje x, que en nuestro modelo representa la capa basal [186]. Las nuevas células tienen la mitad del volumen de su madre, lo que garantiza una presión interna constante durante la mitosis. Para implementar la mitosis simétrica en caso de SC, nuestro modelo establece a 0,05 la probabilidad de generar una nueva célula madre en una mitosis.

En el modelo, las SC y las TAC tienen un ciclo celular sincronizado, es decir, al final del ciclo se llevan a cabo varias mitosis de las células del mismo tipo. Se asume en la simulación que la mitosis de las células madre es simétrica/asimétrica, mientras que las células de TAC la división es simétrica. Además, el mismo mecanismo de mitosis se aplica tanto en las células madre como en TAC, con la condición de que en el último caso, la TAC no debe haber alcanzado su número máximo de divisiones. De lo contrario, la célula se reclasifica como una célula diferenciada. El algoritmo que describe esta extensión es el siguiente:

```

01  input : Type of Cell, CellVolume
02  initialization;
03  for AllCellInSimulation do
04  if Cell is type StemCell then
05      if Volume of the Cell is the 2*Target Volume then
06          Do MitosisInParrallelToXAxis;
07          Volumen and Surface of the Child Cell is double! of the
Parent Cell;
08          Volumen and Surface of Parent Cell is half of the Parent
Cell;
09      End
10  End
11  if Cell is type TAC then
12      if Volume of the Cell is the 2*Target Volume

```

```

13           If NumOfMitosisOfCell<3 then
14               Do MitosisInParrallelToXAxis;
15               Volumen and Surface of the Child Cell is half of
the Parent Cell;
16               Volumen and Surface of Parent Cell is half of the
Parent Cell;
17           End
18       End
19   End
20 End
    
```

Algoritmo 1. Mitosis de la célula

Esquemáticamente, una migración de la herida se produce de la siguiente manera:

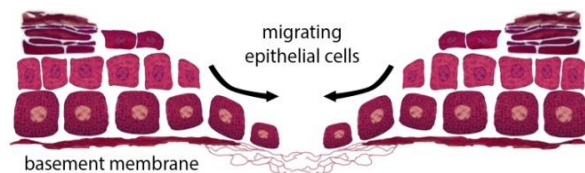


Figura 6.3. La migración celular en la curación de heridas

El espacio dejado por las células perdidas en la herida estimula el resto del tejido para invadir y llenar ese espacio, lo que desemboca en un comportamiento similar al movimiento de las olas. Nuestro modelo incorpora este comportamiento de la siguiente manera:

1. Bajo las observaciones Se cree que existe una fuerza externa que determina el movimiento direccional de las células. El término $H_{ExternalPotential}$ de la ecuación (1) hace posible incorporar tal observación en el *Hamiltoniano*, pudiendo especificar un movimiento efectivo de la célula en la dirección opuesta al vector especificado en $\vec{\lambda}_{ExternalPotential}(\sigma)$, ayudando aplicar una

fuerza adicional a la que ejerce las nuevas células que nacen en la capa basal, que por sí solo no es suficientes para producir el efecto deseado en nuestros experimentos.

2. Las células en proceso de cicatrización de heridas son propensas a mantenerse en estrecho contacto con otras células, de manera que su movilidad se reduce con un aumento de la densidad local (es decir, el contacto de cada célula con sus vecinos). Este fenómeno se denomina inhibición por contacto de la locomoción celular. En nuestro modelo, las células de tipos SC y TAC aumentan su volumen con el tiempo de acuerdo con diferentes umbrales aplicados a su superficie común con las células vecinas:

$$\Delta V_t = \begin{cases} 0,009, & 0,00 \leq C_{\text{surface}} < 0,40 \\ 0,007, & 0,40 \leq C_{\text{surface}} < 0,80 \\ 0,004, & \geq 0,80 \end{cases} \quad (6-6)$$

En el ciclo de renovación continua de la epidermis, las células en la capa escamosa pueden desprenderse y desaparecer debido a los diferentes efectos. En nuestro modelo, este efecto ocurre cuando se sobrepasa un umbral en la parrilla determinada por su centro de masa.

6.4 Resultados

En esta sección se presentan los resultados de varios experimentos *in silico*. La Figura 6 muestra instantáneas de la simulación en varios instantes de tiempo (MCS, pasos Monte Carlo). El modelo comienza con 2 grupos de 4 células madre, unos 66 TAC y 95 queratinocitos diferenciados:

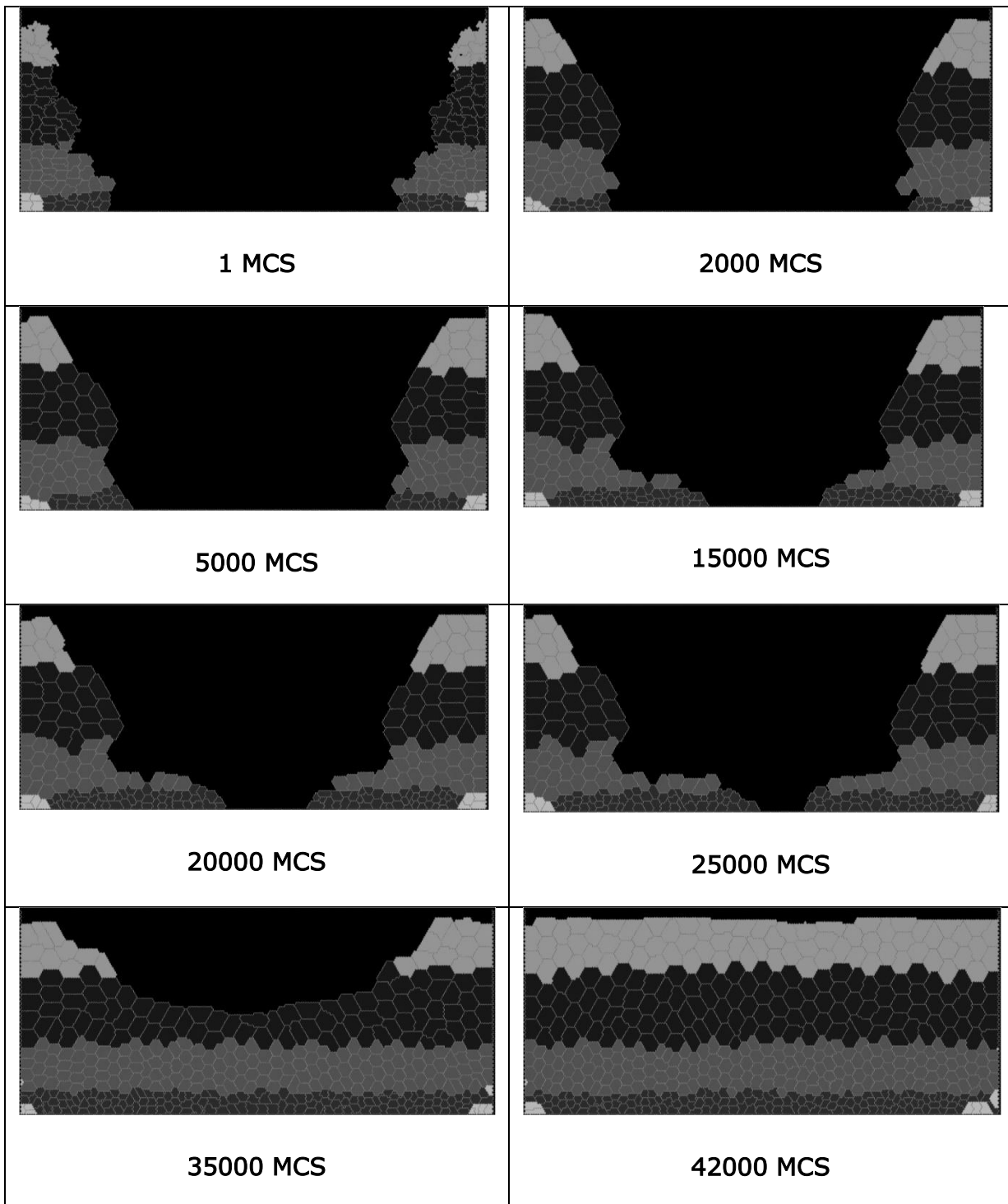


Figura 6.4. Instantáneas de una simulación en diferentes etapas de Monte Carlo

Los resultados de nuestra simulación han sido comparados con los resultados reales de los conejos. La bibliografía no proporciona datos sobre la cicatrización de heridas en seres humanos a parte de algunos datos que simulan ulceraciones crónicas [187-189].

Se observa en los resultados la misma aceleración del proceso mitótico que se muestra en los datos reales. Más adelante, cuando la herida se cierra, esta aceleración se detiene. Este efecto es producido por el factor de crecimiento que controla el proceso mitótico.

La comparación de los resultados de la simulación con los datos reales permite describir el proceso de cicatrización de heridas a través del uso de un modelo discreto donde sus componentes se distribuyen en una rejilla. Los objetos experimentan un comportamiento similar a los datos reales recogidos de la literatura de referencia, excluyendo el comportamiento microscópico.

Los datos proporcionados por [190] muestran cuatro fases diferentes durante el proceso de curación. En la primera hay un importante proceso de engrose de los bordes de la herida dada por una gran división celular. En la segunda el crecimiento disminuye. En la tercera se incrementa de nuevo. Y en la cuarta, donde la curación es casi total y los bordes casi se cierran, la división cae. Nuestro *modelo in silico* es capaz de reproducir el mismo comportamiento **Figura 6.4**.

La Figura 6.5 muestra los resultados de varios ensayos experimentales que han sido realizados para ajustar los valores adecuados de los parámetros, especialmente la tasa de crecimiento de las células en división. Todas las simulaciones se llevan a cabo durante 50.000 MCS, invirtiendo 1 hora y 10 minutos en cada uno y correspondiendo a los 7 días de simulación *in vivo*.

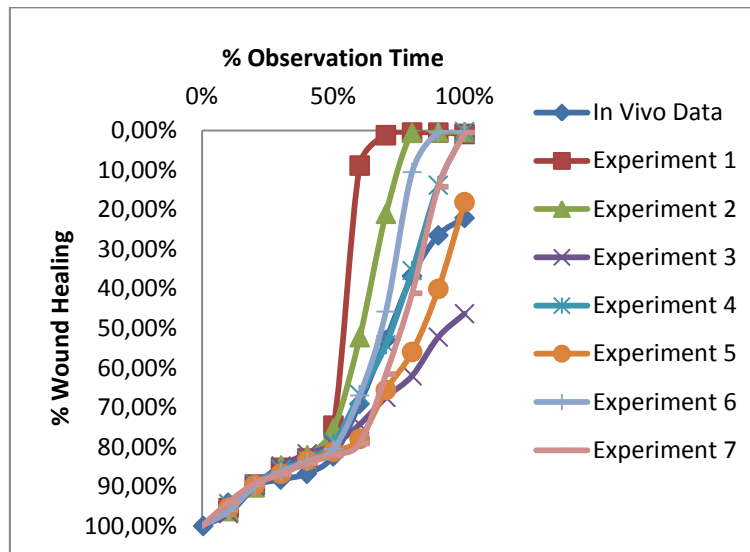


Figura 6.5. Una comparación entre los datos reales (datos *in vivo*) y los resultados de diferentes ejecuciones de nuestra simulación. Las cuatro fases sucesivas en el proceso real son correctamente capturadas por nuestro modelo.

6.5 Discusión

Los autores de los modelos anteriores de la cicatrización de heridas suelen proporcionar una validación biológica, donde:

- Las células se abstraen como esferas rígidas de un radio apropiado.
- Las fuerzas inter-celulares se simulan mediante muelles virtuales colocados entre ellos; es decir, una vez que las células están lo suficientemente cerca unas de otras, las fuerzas de adhesión y reacción de contacto entre ellas se simulan por la fuerza de los muelles situados en el punto de contacto.
- Los únicos parámetros celulares que se tienen en cuenta son el tipo celular (células madre y amplificadoras de tránsito, los queratinocitos diferenciados) y linaje (el conjunto de las células madre que nacen a partir de una célula madre dada). Este modelo no está realmente interesado en una simulación detallada de los queratinocitos.

- Como se ha destacado anteriormente, este modelo está vinculado de algún modo a las técnicas continuas por medio de alguna resolución de ecuaciones diferenciales y los términos, sobre todo en relación con las fuerzas físicas.
- El modelo incorpora el comportamiento 3D.

Nuestro modelo, sin embargo, es único, ya que incluye la forma de las células individuales, proporcionando de este modo al investigador una representación morfológica realista del tejido. Otras diferencias con los modelos anteriores son los siguientes:

- Nuestro enfoque se basa en una aproximación *centrada en la célula*. Usando esta óptica se modela con la mayor precisión posible las características de las células reales, porque se considera que el comportamiento de la epidermis como tejido debe surgir de la interacción de las células individuales. Estas características incluyen el tamaño, la forma y la ductilidad de la célula.
- Se está ampliando un modelo (GGH) diseñado expresamente para resaltar las propiedades de la celda y mostrar el comportamiento emergente de todo el tejido del comportamiento celular. El GGH encapsula la maquinaria involucrada en los tejidos, tales como la garantía de una configuración mínima de energía, a través de sus bases en la mecánica estadística. Proporciona una alta forma de abstracción al incluir el conocimiento de los expertos de una manera más o menos estándar, a través de los términos y parámetros del *Hamiltoniano*. En el que aquí se utiliza prevalece el uso de las propiedades discretas de GGH, aunque también es capaz de añadir información continua en forma de PDE, lo que hace posible extender nuestro sistema para incluir componentes quimiotácticos en el futuro.

Es cierto que un investigador que compara GGH y los sistemas multi-agente similares a los descritos anteriormente, podría considerar que los dos enfoques pueden ser similares en lo relacionado con el proceso de ajuste de sus parámetros, pero el GGH siempre proporciona el mismo tipo de parámetros a todos los fenómenos tisulares. Una descripción más detallada de las ventajas y la idoneidad del GGH para la simulación tisular se puede encontrar en [42].

6.6 Observaciones Finales

- La herida simulada es similar a los datos reales a los que se hace referencia en la literatura. Los datos usados no muestran un cierre total de la herida, mientras que nuestra simulación sí incluye el cierre total.
- Las últimas tres fases del proceso de cicatrización de heridas siguen una función exponencial (logística), donde la tasa de crecimiento se hace más pequeña que el área de la herida se acerca a la curación total.
- La estructura de los tejidos es mantenida correctamente por nuestro modelo, que incorpora la mayoría de las características de la epidermis, tanto en la literatura como en el consenso de especialistas.
- La inhibición de contacto de esas células más cercanas a la herida está incluida.
- La migración de las células de tipo EPU está incluida. Se demuestra que el factor de crecimiento aumenta en varias fases del proceso de cicatrización de la herida. El tamaño y la forma de las células es similar a un tejido *in vivo*. Las teorías acerca de los diferentes tipos de mitosis se tienen en cuenta.
- El modelo aquí desarrollado puede ser una herramienta útil para la simulación de la epidermis. Este modelo es único, ya que encarna la forma de las células individuales, lo que proporciona al investigador una representación realista morfológica del tejido.

-Lo que un hombre puede inventar, otro lo puede descubrir.

Arthur Conan Doyle - Sherlock Holmes – Los Bailarines



Capitulo 7 Conclusiones y trabajo futuro

7.1 Conclusiones

En esta tesis se han estudiado la migración y la proliferación celular en sus comportamientos generales para explicar tanto la homeostasis como el cierre de heridas en la epidermis. Las herramientas utilizadas para su modelado se basan en el modelo CPM-GGH, el cual es un modelo que pertenece a los denominados *Cell-based Models*, siendo este un tipo particular de un autómata celular.

El principal objetivo de esta investigación ha sido el de intentar explicar este fenómeno biológico, combinando diferentes áreas de conocimiento: la física a través de la mecánica estadística con la informática teórica, añadiendo técnicas de computación paralela que permitiesen la aceleración del modelo.

Como se mencionó en la introducción, la motivación principal de esta investigación ha sido proponer un mecanismo usando la informática teórica que permita a investigadores la posibilidad de incorporar rápidamente hipótesis de trabajo para el desarrollo de sus simulaciones, y que a su vez éstas pudieran realizarse en un tiempo razonable. En el capítulo 5 se presentaron los resultados de las dos aportaciones hechas en esta tesis sobre mecanismos de aceleración del modelo GGH-CPM; en el capítulo 6 se muestra el modelo e implementación de la epidermis usando el GGH-CPM.

El motivo de haber escogido este modelo frente a otros reside principalmente en la capacidad que tiene el CPM-GGH de incorporar las formas que puede tener una célula. Otros modelos centrados en modelizado de células describen a las.

células como puntos o esferas [167], omitiendo el hecho de que la forma de las células puede ser un factor determinante y esencial en el desarrollo de un tejido. El uso del concepto de energía total en el sistema se utiliza con el fin de representar las fuerzas físicas de adherencia y elasticidad de las membranas celulares y del medio en que las células se encuentran.

Los efectos de estas fuerzas son muy difíciles de representar usando otras alternativas [42]. En el CPM-GGH, la energía de contacto de una célula con las células circundantes y el ECM determinan la forma de la célula. Bajo esta óptica creemos que se pueden alcanzar modelos más realistas y útiles para biólogos en el desarrollo de sus experimentos.

Nuestro modelo computacional reproduce espacial y temporalmente la formación regular de las capas de la epidermis usando un número limitado de células epidérmicas, a partir de un conjunto de condiciones biológicas iniciales bastante razonables con valores contrastados con la bibliografía. En el modelo CPM-GGH, el proceso de calibración no es sencillo y se determinó por un proceso iterativo de cambio de parámetros, concretamente en aquellos que se incorporan directamente en el *Hamiltoniano*. Las energías de contacto entre los diferentes tipos de células determinan la reorganización de las diferentes capas que constituyen la epidermis, usando para ello diferentes valores de la restricción determinada por la restricción J del *Hamiltoniano*. El cambio y modificación de este parámetro durante el proceso de modelización, posibilita la comprobación de que problemas de una incorrecta relación en la fuerza de adhesión entre diferentes tipos de células pueden articular en el modelo pistas sobre la malformación o constitución de enfermedades dentro del tejido (tales como psoriasis o cáncer). Estas simulaciones se basan principalmente en la hipótesis de Adhesión Diferencial de Steinberg [40], que reproduce en los experimentos los patrones de ordenación de los diferentes tipos de células del tejido epidérmico.

Los mecanismos que se emplean para modificar la estructura del modelo GGH-CPM y así incluir los comportamientos de mitosis y quimiotaxis se hacen indirectamente, provistos por mecanismos auxiliares, que se implementan fuera del *Hamiltoniano*.

En esta tesis se lograron varios objetivos durante el proceso de obtención del modelo final, los cuales comprenden:

- Las propiedades relevantes para la modelización de las células dentro del alcance de esta tesis se extrajeron con éxito. Se incorporó la inhibición por contacto para capturar las interacciones de las células como elemento fundamental para el cierre de heridas. El modelo de interacción fue capaz de reproducir el reordenamiento de las células como se observa *in vivo*. En cada uno de estos casos, la mecánica de células en los modelos tiene que ser lo suficientemente realista para que cualquier modelo tenga éxito, sin duda una tarea no trivial.
- El modelo permite explorar el efecto de los diferentes parámetros que determinan como será la rigidez de la membrana viéndose expresado por el parámetro de fluctuación, energías de contacto y de volumen, siendo independiente una de otra. Por lo tanto, se puede dilucidar el papel que desempeñan las diferentes propiedades que se incorporan en el modelo CPM-GGH ofreciendo un poder de predicción para determinar qué parámetros son experimentalmente más importantes.
- Se encontró, variando la probabilidad de que las células circunden a otras, una manera de simular la inhibición por contacto presente en los queratinocitos de la capa basal y de aquellas células que se encuentran en el borde de la herida, permitiendo desarrollar el efecto de onda invasiva que se observa en el cierre de heridas. Tal comportamiento pone de manifiesto que el modelo de GGH puede predecir de forma natural este tipo particular de invasión.
- La variación de la proliferación de las células de los extremos de la herida es provocado por agentes quimotácticos y factores de crecimiento, que activan las células en el momento de la herida, acelerando o retardando el cierre de la herida. Este comportamiento se incorporó a través de una función de probabilidad que ajusta el modelo para mostrar tal conducta.

Teniendo en cuenta que el modelo de herida aquí mostrado es un proceso simplificado conocido como *tape-stripping*, en el cual no existe flujo de sangre y por consiguiente menor presencia de factores de crecimiento que ejercen control sobre el comportamiento celular, en un modelo que incorpore

heridas más profundas en el que se comprometa la dermis, la función de probabilidad deberá ser revisada y seguramente ajustada.

- El enfoque que se adoptó aquí posee dos ventajas principales: la replicación exitosa de la inhibición por contacto de la mitosis y la capacidad de tener parámetros que modifiquen el comportamiento de la célula mientras se somete al ciclo mitótico.

El mecanismo de proliferación presentado en esta tesis permite asegurar que se produce un comportamiento realista. Se observó que la ubicación de las células madre permite explicar el proceso homeostático y de regeneración del tejido, partiendo de células individuales ubicadas en lugares específicos. El desarrollo del cierre de la herida presentó un crecimiento de tipo logístico recogido experimentalmente y comparado con los datos en el cierre de heridas de conejos. El comportamiento del crecimiento logístico observado determinó la modificación del modelo, aplicando una función de probabilidad para modificar el factor de crecimiento y así recoger los diferentes estadios de proliferación de las células epidérmicas.

Además, se presentaron diferentes valores de adhesión, ya descritos en el capítulo 6, en el cual se expuso cómo las curvas de crecimiento mantenían su forma logística para todos los diferentes regímenes de adhesión, aunque hubo una tendencia de crecimiento continuo lento.

No ha sido fácil la elección de parámetros trasladables a este modelo desde la bibliografía. La disponibilidad de los datos experimentales se limitaba a experimentos hechos en animales como ratones o conejos. En el caso particular del cierre de heridas, básicamente existen pocos o ningún estudio para casos normales en humanos. Los datos más relevantes encontrados a heridas ulceradas [187], lo cual supone un límite a la hora de contrastar los datos.

Una modelización multi-escala de la homeostasis de la piel y de cierre de heridas es un primer paso obligatorio en el desarrollo de un modelo mucho más complejo, donde se pueda incluir mayor tipo de elementos de interés para el experi-

mentador, en el cual se puedan realizar predicciones (por ejemplo: cambios en el ciclo celular que permitan comprobar la relación del tipo de mitosis con el mantenimiento del tejido; también de toxinas o terapias potenciales que afecten el desarrollo del tejido). Incluso se podría pensar en la interacción de un agente químico a escala molecular, prediciendo cómo estas perturbaciones moleculares afectarán al desarrollo normal de la epidermis.

Con el actual aumento de la capacidad computacional y el aumento del número de núcleos por procesador que funcionan en paralelo, surge entonces la cuestión de qué escala pueden llegar a tener los modelos basados en células. Una propiedad importante de estos modelos es su complejidad computacional. Si esta propiedad se sigue manteniendo, se puede suponer que un modelo celular de cien mil millones de componentes (cualquier órgano) requerirá diez mil veces más capacidad de computación y de memoria. La escala asumible en la actualidad está en el orden de magnitud de miles de células. Aunque la capacidad computacional ha crecido exponencialmente desde el desarrollo del primer microprocesador, queda mucho por hacer para contrarrestar esta desventaja.

En general, esta tesis ha contribuido significativamente a la comprensión de la proliferación e invasión de las células en el contexto del cierre de heridas, mediante el uso de un enfoque de la modelización de la célula. El modelo CPM-GGH implementado aquí incorporó un nuevo medio de la generación de la proliferación celular en este contexto. Su rendimiento computacional ha sido debidamente validado en comparación con la implementación paralela desarrollada para CompuCell 3D.

7.2 Trabajo Futuro

Los resultados incluidos en esta tesis posibilitan varias líneas futuras de investigación, tanto en la línea de modelos epidérmicos como en la búsqueda de estrategias de paralelización que pudiesen incluir aun más naturalidad en el modelo.

En conclusión, el modelo podría extenderse a la evaluación de nuevas teorías de proliferación y diferenciación de los queratinocitos, propuestas en [191]. Tam-

bién sería interesante comprobar cuál es el proceso mecánico-biológico que se presenta durante la invasión y metástasis, usando como base los cambios de las propiedades de adhesión y otros parámetros que se incorporan en el *Hamiltoniano* o en sus extensiones. Esta podría ser la sincronización de la progresión del ciclo celular y sus efectos sobre la proliferación de un carcinoma de células escamosas [192].

Sería también interesante contrastar los experimentos e hipótesis que aquí se han expuesto, comparando resultados en laboratorio vs resultados *in silico*, revisando si las propuestas formuladas en esta tesis son capaces de reproducir el mismo comportamiento, esto determinaría la existencia o no de desviaciones. Los ensayos hechos aquí ofrecen una gran cantidad de información tanto para biólogos como para matemáticos, sobre todo porque introduce comportamientos celulares adicionales o equivalentes en parámetros matemáticos, y sin embargo puede generar patrones de comportamientos únicos, macroscópicamente observables. Los cambios graduales en la adhesión celular podrían cambiar la dinámica de reorganización de las células después de la diferenciación de las epidérmicas, mejorando el realismo de la dinámica y dando lugar a más morfologías más realistas.

Aunque se ha comentado en los apartados de esta tesis que una ventaja del modelo CPM-GGH es la reproducción del comportamiento y movimiento de la membrana celular durante su uso, no se ha podido reproducir la forma que poseen las células de la capa escamosa, alargadas y casi estáticas. Este problema está dado por la combinación del uso de la rejilla hexagonal junto con la extensión que modifica el hamiltoniano, que define como será la forma de las células alargadas, haciendo que en ciertos momentos durante la evolución del modelo, se hayan observado la fragmentación de las células.

A mi juicio, los modelos basados en células continuarán siendo utilizados en sistemas cada vez más grandes y complejos. En el capítulo anterior el uso del modelo CPM-GGH se restringió a un limitado modelo 2D, un modelo 3D en el cual se intentase usar en escalas de tiempo y espacio realistas para simular proceso tomaría un tiempo bastante largo, recordemos que la complejidad del modelo crece a razón de 2^n en el caso del modelo en dos dimensiones.

Un modelo más sofisticado a nivel microscópico puede incorporar rutas de señalización del tipo Delta-Notch, que puedan ser expresadas usando ecuaciones diferenciales parciales con el fin de demostrar cómo las suposiciones inherentes a la modelización macroscópica y comportamientos que no están representados fácilmente a esa escala, pueden tener un serio impacto en las predicciones de un modelo, incluso en un contexto biológico del ensayo.

La modelización basada en la célula sin duda tiene muchos desafíos interesantes para el futuro. A corto plazo, muchos investigadores que trabajan en biología desconocen este tipo de herramientas de modelización. El uso de este tipo de herramientas puede ser un mecanismo que podría ayudar a constituir un laboratorio virtual, rápido y barato para ejecutar las simulaciones y ofrecer un control completo sobre todos los aspectos del tejido a simular. Los experimentos virtuales permiten una rápida creación de prototipos de nuevas ideas y ayudar a orientar experimentos eficaces. La modelización es generalmente una experiencia desafiante, en especial en el desarrollo de modelos de organismos más complejos, sin duda debe capturar la imaginación de los investigadores.

Los resultados en esta tesis abren la posibilidad de una multitud de nuevas investigaciones, que incluyen por ejemplo creación de modelos de cáncer y metástasis, usando para ello cambios en las propiedades de adhesión y otros parámetros, como la sincronización de la progresión del ciclo celular.

Por último, otra extensión útil de los experimentos realizados será el desarrollo de un modelo de tres dimensiones, ya que se podría explorar si la dinámica dependiese de la dimensionalidad.

Referencias

- [1] P.S. Frenette and D.D. Wagner, "Adhesion Molecules—Blood Vessels and Blood Cells," *N.Engl.J.Med.*, vol. 335, no. 1, pp. 43-45.
- [2] A.M. Bailey, B.C. Thorne and S.M. Peirce, "Multi-cell agent-based simulation of the microvasculature to study the dynamics of circulating inflammatory cell trafficking," *Ann.Biomed.Eng.*, vol. 35, no. 6, pp. 916-936.
- [3] H. Byrne and D. Drasdo, "Individual-based and continuum models of growing cell populations: a comparison," *J.Math.Biol.*, vol. 58, no. 4-5, pp. 657-687.
- [4] P. Pacheco, "An introduction to parallel programming," 2011.
- [5] S.D. Hester, J.M. Belmonte, J.S. Gens, S.G. Clendenon and J.A. Glazier, "A multi-cell, multi-scale model of vertebrate segmentation and somite formation," *PLoS computational biology*, vol. 7, no. 10, pp. e1002155.
- [6] N.M. Luscombe, D. Greenbaum and M. Gerstein, "What is bioinformatics? A proposed definition and overview of the field," *Methods Inf.Med.*, vol. 40, no. 4, pp. 346-358.
- [7] B. Tanos and E. Rodriguez-Boulan, "The epithelial polarity program: machineries involved and their hijacking by cancer," *Oncogene*, vol. 27, no. 55, pp. 6939-6957.
- [8] F. Graner and J.A. Glazier, "Simulation of biological cell sorting using a two-dimensional extended Potts model," *Phys.Rev.Lett.*, vol. 69, no. 13, pp. 2013.
- [9] J.A. Glazier and F. Graner, "Simulation of the differential adhesion driven rearrangement of biological cells," *Physical Review E*, vol. 47, no. 3, pp. 2128.
- [10] T. Ören, "The many facets of simulation through a collection of about 100 definitions," *SCS M&S Magazine*, vol. 2, no. 2, pp. 82-92.
- [11] M. Minsky, "Matter, mind and models,".
- [12] K. Velten, "Mathematical modeling and simulation," 2009.
- [13] F.E. Cellier, "Continuous system modeling," 1991.
- [14] P.A. Fishwick, "The role of process abstraction in simulation," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 18, no. 1, pp. 18-39.
- [15] B.P. Zeigler, H. Praehofer and T.G. Kim, "Theory of modeling and simulation," vol. 19, 1976.
- [16] R. Lahoz-Beltrá, "Bioinformática: Simulación, vida artificial e inteligencia artificial," 2010.

- [17] L.N. de Castro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1-36.
- [18] J. Von Neumann and A.W. Burks, "Theory of self-reproducing automata,".
- [19] M. Gardner, "Mathematical games: The fantastic combinations of John Conway's new solitaire game "life"," *Sci.Am.*, vol. 223, no. 4, pp. 120-123.
- [20] S. Wolfram, "Universality and complexity in cellular automata," *Physica D*, vol. 10, no. 1, pp. 1-35.
- [21] J. Hardy, Y. Pomeau and O. De Pazzis, "Time evolution of a two- dimensional model system. I. Invariant states and time correlation functions," *Journal of Mathematical Physics*, vol. 14, pp. 1746.
- [22] B.H.Y. Pomeau and U. Frisch, "Lattice-gas automata for the Navier-Stokes equation," *Phys.Rev.Lett.*, vol. 56, no. 14, pp. 1505.
- [23] B. Chopard, R. Ouared, A. Deutsch, H. Hatzikirou and D. Wolf-Gladrow, "Lattice-gas cellular automaton models for biology: from fluids to cells," *Acta Biotheor.*, vol. 58, no. 4, pp. 329-340.
- [24] D. Chandler, "Introduction to modern statistical mechanics," *Introduction to Modern Statistical Mechanics, by David Chandler, pp.288.Foreword by David Chandler.Oxford University Press, Sep 1987.ISBN-10: 0195042778.ISBN-13: 9780195042771*, vol. 1.
- [25] M. Newman and G. Barkema, "Monte Carlo Methods in Statistical Physics chapter 1-4,".
- [26] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, "Equation of state calculations by fast computing machines," *J.Chem.Phys.*, vol. 21, pp. 1087.
- [27] W. Lenz, "Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern," *Physikalische Zeitschrift*, vol. 21, no. 613-615, pp. 2.
- [28] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 31, no. 1, pp. 253-258.
- [29] J.W. Gibbs, "Elementary principles in statistical mechanics: developed with especial reference to the rational foundation of thermodynamics," 2010.
- [30] S.G. Brush, "History of the Lenz-Ising model," *Reviews of Modern Physics*, vol. 39, no. 4, pp. 883.
- [31] L. Onsager, "Crystal statistics. I. A two-dimensional model with an order-disorder transition," *Physical Review*, vol. 65, no. 3-4, pp. 117.
- [32] D.P. Landau and K. Binder, "A guide to Monte Carlo simulations in statistical physics," 2009.

- [33] J.A. Glazier, A. Balter and N.J. Popławski, "Magnetization to morphogenesis: a brief history of the Glazier-Graner-Hogeweg model," pp. 79-106.
- [34] J. Ashkin and E. Teller, "Statistics of two-dimensional lattices with four components," *Physical Review*, vol. 64, no. 5-6, pp. 178.
- [35] R.B. Potts, "Some generalized order-disorder transformations," vol. 48, no. 2, pp. 106-109.
- [36] C. Domb, "Configurational studies of the Potts models," *Journal of Physics A: Mathematical, Nuclear and General*, vol. 7, no. 11, pp. 1335.
- [37] P.S. Sahni, G.S. Grest, M.P. Anderson and D.J. Srolovitz, "Kinetics of the q-state Potts model in two dimensions," *Phys.Rev.Lett.*, vol. 50, no. 4, pp. 263.
- [38] M.S. Steinberg, "Reconstruction of tissues by dissociated cells," *Science*, vol. 141, no. 3579, pp. 401-408.
- [39] M.S. Steinberg, "Adhesion-guided multicellular assembly: a commentary upon the postulates, real and imagined, of the differential adhesion hypothesis, with special attention to computer simulations of cell sorting," *J.Theor.Biol.*, vol. 55, no. 2, pp. 431-443.
- [40] M.S. Steinberg and L.L. Wiseman, "Do morphogenetic tissue rearrangements require active cell movements? The reversible inhibition of cell sorting and tissue spreading by cytochalasin B," *J.Cell Biol.*, vol. 55, no. 3, pp. 606-615.
- [41] M.H. Swat, S.D. Hester, A.I. Balter, R.W. Heiland, B.L. Zaitlen and J.A. Glazier, "Multicell simulations of development and disease using the CompuCell3D simulation environment," pp. 361-428.
- [42] R.M. Merks and J.A. Glazier, "A cell-centered approach to developmental biology," *Physica A: Statistical Mechanics and its Applications*, vol. 352, no. 1, pp. 113-130.
- [43] S. Turner, J.A. Sherratt, K.J. Painter and N.J. Savill, "From a discrete to a continuous model of biological cell movement," *Physical Review E*, vol. 69, no. 2, pp. 021910.
- [44] K.A. Rejniak and L.J. McCawley, "Current trends in mathematical modeling of tumor-microenvironment interactions: a survey of tools and applications," *Exp.Biol.Med.*, vol. 235, no. 4, pp. 411-423.
- [45] A. Balter, R.M. Merks, N.J. Popławski, M. Swat and J.A. Glazier, "The Glazier-Graner-Hogeweg model: extensions, future directions, and opportunities for further study," pp. 151-167.
- [46] G.D. Weinstein, J.L. McCullough and P. Ross, "Cell proliferation in normal epidermis," *J.Invest.Dermatol.*, vol. 82, no. 6, pp. 623-628.
- [47] G.E. Moore, *Cramming more components onto integrated circuits.*

- [48] L.B. Kish, "End of Moore's law: thermal (noise) death of integration in micro and nano electronics," *Physics Letters A*, vol. 305, no. 3, pp. 144-149.
- [49] D. Peleg, "Distributed computing: a locality-sensitive approach," vol. 5, 2000.
- [50] D.B. Kirk and W.H. Wen-mei, "Programming massively parallel processors: a hands-on approach," 2010.
- [51] B.J. Walker and D. Steel, "Implementing a Full Single System Image UnixWare Cluster: Middleware vs Underware.," pp. 2767-2773.
- [52] R. Buyya, "High Performance Cluster Computing: Architectures and Systems (Volume 1)," *Prentice Hall, Upper SaddleRiver, NJ, USA*, vol. 1, pp. 999.
- [53] A. Barak and O. La'adan, "The MOSIX multicomputer operating system for high performance cluster computing," *Future Generation Comput.Syst.*, vol. 13, no. 4, pp. 361-372.
- [54] R. Lottiaux, P. Gallard, G. Vallée, C. Morin and B. Boissinot, "OpenMosix, OpenSSI and Kerrighed: a comparative study," vol. 2, pp. 1016-1023.
- [55] D.J. Becker, T. Sterling, D. Savarese, J.E. Dorband, U.A. Ranawak and C.V. Packer, "BEOWULF: A parallel workstation for scientific computation," vol. 95.
- [56] G. Vallée, S.L. Scott, C. Morin, J. Berthou and H. Prisker, "Ssi-oscar: a cluster distribution for high performance computing using a single system image," pp. 319-325.
- [57] C. Morin, R. Lottiaux, G. Vallée, P. Gallard, G. Utard, R. Badrinath and L. Rilling, "Kerrighed: a single system image cluster operating system for high performance computing," pp. 1291-1294.
- [58] G.M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," pp. 483-485.
- [59] M.J. Flynn, "Some computer organizations and their effectiveness," *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 948-960.
- [60] G. Jost, H. Jin, D. an Mey and F.F. Hatay, "Comparing the OpenMP, MPI, and hybrid programming paradigms on an SMP cluster," vol. 3, pp. 2003.
- [61] W.D. Gropp, E.L. Lusk and A. Skjellum, "Using MPI: portable parallel programming with the message-passing interface," vol. 1, 1999.
- [62] G.E. Blelloch, "Vector models for data-parallel computing," vol. 75, 1990.
- [63] T. White, "Hadoop: the definitive guide," 2012.
- [64] D. Kirk, "NVIDIA CUDA software and GPU parallel computing architecture," vol. 7, pp. 103-104.

- [65] C. Evangelinos and C. Hill, "Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2." *ratio*, vol. 2, no. 2.40, pp. 2.34.
- [66] B. Carpenter, G. Zhang, G.C. Fox and X. Li, "HPJava: Data parallel extensions to java,".
- [67] Anonymous "Ateji PX for Java," , vol. 2014, no. 01/10.
- [68] J. Maassen, R. Van Nieuwpoort, R. Veldema, H. Bal, T. Kielmann, C. Jacobs and R. Hofman, "Efficient Java RMI for parallel programming," *ACM Transactions on Programming Languages and Systems*, vol. 23, no. 6, pp. 747-775.
- [69] S. Allamaraju, K. Avedal, R. Browett, J. Diamond and J. Griffin, "Java server programming," , 2001.
- [70] M. Baker, B. Carpenter, G. Fox, S.H. Ko and S. Lim, "mpiJava: An object-oriented Java interface to MPI," , pp. 748-762.
- [71] M. Baker and B. Carpenter, "Mpj: A proposed java message passing api and environment for high performance computing," , pp. 552-559.
- [72] W. Yu and A. Cox, "Java/DSM: A platform for heterogeneous computing," *Concurrency: Practice and Experience*, vol. 9, no. 11, pp. 1213-1224.
- [73] M. Philippsen and M. Zenger, "JavaParty - Transparent Remote Objects in Java," *Concurrency Pract.Exp.*, vol. 9, no. 11, pp. 1225-1242.
- [74] Y. Aridor, M. Factor and A. Teperman, "cJVM: a Single System Image of a JVM on a Cluster," , pp. 4-11.
- [75] R.S. Cost, T. Finin, Y. Labrou, X. Luan, Y. Peng, I. Soboroff, J. Mayfield and A. Boughannam, "Jackal: a java-based tool for agent development,".
- [76] G. Antoniu, L. Bougé, P. Hatcher, M. MacBeth, K. McGuigan and R. Namyst, "The Hyperion system: Compiling multithreaded Java bytecode for distributed execution," *Parallel Computing*, vol. 27, no. 10, pp. 1279-1297.
- [77] Y. Sohda, H. Nakada and S. Matsuoka, "Implementation of a portable software DSM in Java," , pp. 163-172.
- [78] J. Andersson, S. Weber, E. Cecchet, C. Jensen and V. Cahill, "Kaffemik-a distributed jvm on a single address space architecture,".
- [79] E. Tilevich and Y. Smaragdakis, "J-Orchestra: Enhancing Java programs with distribution capabilities," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 19, no. 1, pp. 1.
- [80] W. Zhu, C. Wang and F.C. Lau, "Jessica2: A distributed java virtual machine with transparent thread migration support," , pp. 381-388.

- [81] K.T. Lam, Y. Luo and C. Wang, "Adaptive sampling-based profiling techniques for optimizing the distributed JVM runtime," pp. 1-11.
- [82] Anonymous "Jessica4," vol. 2014, no. 01/15.
- [83] J.N. Zigman and R. Sankaranarayana, "dJVM-A distributed JVM on a Cluster," *Made available in DSpace on 2011-01-05T08: 29: 32Z (GMT).No.of bitstreams: 4 TR-CS-02-04.pdf.jpg: 1672 bytes, checksum: cf2bb05a2f50a86a8d8076c373cb549b (MD5) 1526-01.2003-06-27T03: 13: 01Z.xsh: 356 bytes, checksum: f78d665ea8ec38edfda01515ed114(TRUNCADO).*
- [84] M. Factor, A. Schuster and K. Shagin, "JavaSplit: a Runtime for execution of monolithic Java programs on heterogenous collections of commodity workstations," pp. 110-117.
- [85] A. Zilka, "The definitive guide to Terracotta. APress, <http://www.apress.com>, 2008.", 2008.
- [86] P. McGachey, A.L. Hosking and J.E.B. Moss, "Classifying Java class transformations for pervasive virtualized access," *ACM Sigplan Notices*, vol. 45, no. 2, pp. 75-84.
- [87] R. Miceli, G. Civario, A. Sikora, E. César, M. Gerndt, H. Haitof, C. Navarrete, S. Benkner, M. Sandrieser and L. Morin, "AutoTune: a plugin-driven approach to the automatic tuning of parallel applications," pp. 328-342.
- [88] C. Navarrete and E. Anguiano, "Architecture based on dynamic graphs for the dynamic reconfiguration of farms of computers,".
- [89] G. Fursin, C. Miranda, O. Temam, M. Namolaru, E. Yom-Tov, A. Zaks, B. Mendelson, E. Bonilla, J. Thomson and H. Leather, "MILEPOST GCC: machine learning based research compiler,".
- [90] C. Navarrete Navarrete, "Platform for automatic parallelisation of sequential codes using dynamic graphs partitioning and based on user adaptable load balancing,".
- [91] M.O. Ball, "Computing network reliability," *Oper.Res.*, vol. 27, no. 4, pp. 823-838.
- [92] J. Wang and R. Swendsen, "Nonuniversal critical dynamics in Monte Carlo simulations," *Phys.Rev.Lett.*
- [93] U. Wolff, "Collective Monte Carlo updating for spin systems," *Phys.Rev.Lett.*, vol. 62, no. 4, pp. 361.
- [94] M. Creutz, "Microcanonical monte carlo simulation," *Phys.Rev.Lett.*, vol. 50, no. 19, pp. 1411-1414.
- [95] G.Y. Vichniac, "Simulating physics with cellular automata," *Physica D*, vol. 10, no. 1, pp. 96-116.

- [96] M. Creutz, "Deterministic ising dynamics," *Annals of physics*, vol. 167, no. 1, pp. 62-72.
- [97] B. Smith, P. Bjorstad and W. Gropp, "Domain decomposition: parallel multilevel methods for elliptic partial differential equations," 2004.
- [98] N. Chen, J.A. Glazier, J.A. Izaguirre and M.S. Alber, "A parallel implementation of the Cellular Potts Model for simulation of cell-based morphogenesis," *Comput.Phys.Commun.*, vol. 176, no. 11, pp. 670-681.
- [99] É Gusatto, J.C. Mombach, F.P. Cercato and G.H. Cavaleiro, "An efficient parallel algorithm to evolve simulations of the cellular Potts model," *Parallel processing letters*, vol. 15, no. 01n02, pp. 199-208.
- [100] J.J. Tapia and R.M. D'Souza, "Parallelizing the Cellular Potts Model on graphics processing units," *Comput.Phys.Commun.*, vol. 182, no. 4, pp. 857-865.
- [101] D. Frenkel and B. Smit, "Understanding molecular simulation: from algorithms to applications," 2001.
- [102] A.B. Bortz, M.H. Kalos and J.L. Lebowitz, "A new algorithm for Monte Carlo simulation of Ising spin systems," *Journal of Computational Physics*, vol. 17, no. 1, pp. 10-18.
- [103] M. Novotny, "Monte Carlo algorithms with absorbing Markov chains: Fast local algorithms for slow dynamics," *Phys.Rev.Lett.*, vol. 74, no. 1, pp. 1.
- [104] A.M. Werling, Y. Doerflinger, J.M. Brandner, F. Fuchs, J.C. Becker, D. Schrama, H. Kurzen, S. Goerdts and W.K. Peitsch, "Homo- and heterotypic cell-cell contacts in Merkel cells and Merkel cell carcinomas: heterogeneity and indications for cadherin switching," *Histopathology*, vol. 58, no. 2, pp. 286-303.
- [105] R.A. Foty and M.S. Steinberg, "The differential adhesion hypothesis: a direct evaluation," *Dev.Biol.*, vol. 278, no. 1, pp. 255-263.
- [106] B. Alberts, "Essential cell biology: an introduction to the molecular biology of the cell," vol. 1, 1998.
- [107] S.J. Singer and G.L. Nicolson, "The fluid mosaic model of the structure of cell membranes," *Science*, vol. 175, no. 23, pp. 720-731.
- [108] R.O. Hynes, "Integrins: versatility, modulation, and signaling in cell adhesion," *Cell*, vol. 69, no. 1, pp. 11-25.
- [109] A. Upadhyaya, "Thermodynamic and fluid properties of cells, tissues and membranes,".
- [110] C. David H, "Overview of biology, development and structure of skin," *Fitzpatrick's Dermatology in General Medicine*, vol. 1.

- [111] E. Fuchs and V. Horsley, "More than one way to skin..." *Genes Dev.*, vol. 22, no. 8, pp. 976-985.
- [112] J. Jorizzo, J. Bologna and R. Rapini, "Dermatology," 2008.
- [113] M. Teunissen, "Dynamic nature and function of epidermal langerhans cells in vivo and in vitro: a review, with emphasis on human Langerhans cells," *Histochem.J.*, vol. 24, no. 10, pp. 697-716.
- [114] J. Dowling, Q. Yu and E. Fuchs, "34 integrin is required for hemidesmosome formation, cell adhesion, and cell survival," *J.Cell Biol.*, vol. 134, pp. 559-572.
- [115] E. Proksch, J.M. Brandner and J. Jensen, "The skin: an indispensable barrier," *Exp.Dermatol.*, vol. 17, no. 12, pp. 1063-1072.
- [116] P. Ehrhardt and J. Jens-Michel, "Skin as Organ of Protection," *Fitzpatrick's Dermatology in General Medicine*, vol. 1.
- [117] A.J. Singer and R. Clark, "Cutaneous wound healing," *N.Engl.J.Med.*, vol. 341, no. 10, pp. 738-746.
- [118] J. Kanitakis, "Anatomy, histology and immunohistochemistry of normal human skin," *Eur.J.Dermatol.*, vol. 12, no. 4, pp. 390-399.
- [119] A.C. Williams and B.W. Barry, "Terpenes and the lipid-protein-partitioning theory of skin penetration enhancement," *Pharm.Res.*, vol. 8, no. 1, pp. 17-24.
- [120] C.S. Potten and M. Loeffler, "Stem cells: attributes, cycles, spirals, pitfalls and uncertainties. Lessons for and from the crypt," *Development*, vol. 110, no. 4, pp. 1001-1020.
- [121] E. Clayton, D.P. Doupé, A.M. Klein, D.J. Winton, B.D. Simons and P.H. Jones, "A single type of progenitor cell maintains normal epidermis," *Nature*, vol. 446, no. 7132, pp. 185-189.
- [122] A. Freije, L. Ceballos, M. Coisy, L. Barnes, M. Rosa, E. De Diego, J. Blanchard and A. Gandarillas, "Cyclin E drives human keratinocyte growth into differentiation," *Oncogene*, vol. 31, no. 50, pp. 5180-5192.
- [123] F. Meineke, C.S. Potten and M. Loeffler, "Cell migration and organization in the intestinal crypt using a lattice-free model," *Cell Prolif.*, vol. 34, no. 4, pp. 253-266.
- [124] U.B. Jensen, S. Lowell and F.M. Watt, "The spatial relationship between stem cells and their progeny in the basal layer of human epidermis: a new view based on whole-mount labelling and lineage analysis," *Development*, vol. 126, no. 11, pp. 2409-2418.

- [125] P.H. Jones, S. Harper and F.M. Watt, "Stem cell patterning and fate in human epidermis," *Cell*, vol. 80, no. 1, pp. 83-93.
- [126] I. Mackenzie, "Relationship between mitosis and the ordered structure of the stratum corneum in mouse epidermis," *Nature*, vol. 226, pp. 653-655.
- [127] C.S. Potten, "The epidermal proliferative unit: the possible role of the central basal cell," *Cell Prolif.*, vol. 7, no. 1, pp. 77-88.
- [128] S. Miller, T. Sun and P. Coulombe, "Epidermal growth and differentiation," *Fitzpatrick's Dermatology in General Medicine*, pp. 357-383.
- [129] R. Vassar and E. Fuchs, "Transgenic mice provide new insights into the role of TGF- α during epidermal development and differentiation." *Genes Dev.*, vol. 5, no. 5, pp. 714-727.
- [130] S.R. Edmondson, S.P. Thumiger, G.A. Werther and C.J. Wraight, "Epidermal homeostasis: the role of the growth hormone and insulin-like growth factor systems," *Endocr.Rev.*, vol. 24, no. 6, pp. 737-764.
- [131] S.M. Rudman, M.P. Philpott, G.A. Thomas and T. Kealey, "The role of IGF-I in human skin and its appendages: morphogen as well as mitogen?" *J.Invest.Dermatol.*, vol. 109, no. 6, pp. 770-777.
- [132] M. Wataya-Kaneda, K. Hashimoto, M. Kato, K. Miyazono and K. Yoshikawa, "Differential localization of TGF- β -precursor isotypes in normal human skin," *J.Dermatol.Sci.*, vol. 8, no. 1, pp. 38-44.
- [133] I. Ramis-Conde, M.A. Chaplain, A.R. Anderson and D. Drasdo, "Multi-scale modelling of cancer cell intravasation: the role of cadherins in metastasis," *Physical biology*, vol. 6, no. 1, pp. 016008.
- [134] S. Raghavan, C. Bauer, G. Mundschau, Q. Li and E. Fuchs, "Conditional Ablation of β 1 Integrin in Skin Severe Defects in Epidermal Proliferation, Basement Membrane Formation, and Hair Follicle Invagination," *J.Cell Biol.*, vol. 150, no. 5, pp. 1149-1160.
- [135] P.H. Jones, "Epithelial stem cells," *Bioessays*, vol. 19, no. 8, pp. 683-690.
- [136] R.A. Clark, "Overview and general considerations of wound repair," pp. 3-33.
- [137] G.C. Gurtner, S. Werner, Y. Barrandon and M.T. Longaker, "Wound repair and regeneration," *Nature*, vol. 453, no. 7193, pp. 314-321.
- [138] J. Folkman and A. Moscona, "Role of cell shape in growth control,"
- [139] J. Bereiter-Hahn, "Epidermal cell migration and wound repair," pp. 443-471.

- [140] R.D. Paladini, K. Takahashi, N.S. Bravo and P.A. Coulombe, "Onset of re-epithelialization after skin injury correlates with a reorganization of keratin filaments in wound edge keratinocytes: defining a potential role for keratin 16." *J.Cell Biol.*, vol. 132, no. 3, pp. 381-397.
- [141] C. Castaneda-Marroquín, de la Puente, Alfonso Ortega, M. Alfonseca, J.A. Glazier and M. Swat, "A threaded Java concurrent implementation of the Monte-Carlo Metropolis Ising model," , vol. 2009, pp. 103.
- [142] C. Castaneda-Marroquín, C.B. Navarrete, A. Ortega, M. Alfonseca and E. Anguiano, "Parallel Metropolis-Montecarlo simulation for Potts model using an adaptable network topology based on dynamic graph partitioning," , pp. 89-96.
- [143] N.J. Popławski, A. Shirinifard, M. Swat and J.A. Glazier, "Simulation of single-species bacterial-biofilm growth using the Glazier-Graner-Hogeweg model and the CompuCell3D modeling environment," *Mathematical biosciences and engineering: MBE*, vol. 5, no. 2, pp. 355.
- [144] T. Preis, P. Virnau, W. Paul and J.J. Schneider, "GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model," *Journal of Computational Physics*, vol. 228, no. 12, pp. 4468-4477.
- [145] J. Schneider and S. Kirkpatrick, "Stochastic optimization," , 2006.
- [146] G. Spezzano, D. Talia, S. Di Gregorio, R. Rongo and W. Spataro, "A parallel cellular tool for interactive modeling and simulation," *Computational Science & Engineering, IEEE*, vol. 3, no. 3, pp. 33-43.
- [147] S. Bandini, G. Mauri and R. Serra, "Cellular automata: From a theoretical parallel computational model to its application to complex systems," *Parallel Computing*, vol. 27, no. 5, pp. 539-553.
- [148] M.S. Alber, M.A. Kiskowski, J.A. Glazier and Y. Jiang, "On cellular automaton approaches to modeling biological cells," , pp. 1-39.
- [149] J. Moreira and A. Deutsch, "Cellular automaton models of tumor development: a critical review," *Advances in Complex Systems*, vol. 5, no. 02n03, pp. 247-267.
- [150] G. Barkema and T. MacFarland, "Parallel simulation of the Ising model," *Physical Review E*, vol. 50, no. 2, pp. 1623.
- [151] G. Korniss, G. Brown, M. Novotny and P. Rikvold, "Hard Simulation Problems in the Modeling of Magnetic Materials: Parallelization and Langevin Micromagnetics," , pp. 134-139.
- [152] Y. Shim and J.G. Amar, "Growth instability in Cu multilayer films due to fast edge/corner diffusion," *Physical Review B*, vol. 73, no. 3, pp. 035423.
- [153] V.I. Manousiouthakis and M.W. Deem, "Strict detailed balance is unnecessary in Monte Carlo simulation," *J.Chem.Phys.*, vol. 110, pp. 2753.

- [154] R. Canal, J. Parcerisa and A. González, "Dynamic cluster assignment mechanisms," pp. 133-142.
- [155] R. Bhargava and L.K. John, "Improving dynamic cluster assignment for clustered trace cache processors," pp. 264-274.
- [156] K. Amiri, D. Petrou, G. Ganger and G. Gibson, *Dynamic function placement in active storage clusters*.
- [157] A. Lastovetsky, "Scientific Programming for Heterogeneous Systems-Bridging the Gap between Algorithms and Applications," pp. 3-8.
- [158] J. Dongarra and A. Lastovetsky, "An overview of heterogeneous high performance and grid computing," *Engineering the Grid: Status and Perspective*, pp. 1-25.
- [159] D.A. Bacigalupo, S.A. Jarvis, L. He, D.P. Spooner and G. Nudd, "A comparative evaluation of two techniques for predicting the performance of dynamic enterprise systems,".
- [160] L. Kleinrock, "Theory, volume 1, Queueing systems," 1975.
- [161] B.R. Arafeh, K. Day and A. Touzene, "A Paradigm for Allocating Parallel Application Tasks to Heterogeneous Computing Resources on the Grid.," pp. 41-48.
- [162] B. Song, C. Ernemann and R. Yahyapour, "Parallel computer workload modeling with markov chains," pp. 47-62.
- [163] D. Gross and C.M. Harris, "Fundamentals of queuing theory," 1998.
- [164] J.A. Glazier, M.P. Anderson and G.S. Grest, "Coarsening in the two-dimensional soap froth and the large-Q Potts model: a detailed comparison," *Philos.Mag.B*, vol. 62, no. 6, pp. 615-645.
- [165] A.R. Anderson, M.A. Chaplain and K.A. Rejniak, "Single-cell-based models in biology and medicine," 2007.
- [166] J.A. Glazier, A. Balter and N.J. Popławski, "Magnetization to morphogenesis: a brief history of the Glazier-Graner-Hogeweg model," pp. 79-106.
- [167] S. Adra, T. Sun, S. MacNeil, M. Holcombe and R. Smallwood, "Development of a three dimensional multiscale computational model of the human epidermis," *PLoS one*, vol. 5, no. 1, pp. e8511.
- [168] J.C. Adams and F.M. Watt, "Fibronectin inhibits the terminal differentiation of human keratinocytes,".
- [169] F.M. Watt, C.L. Celso and V. Silva-Vargas, "Epidermal stem cells: an update," *Curr.Opin.Genet.Dev.*, vol. 16, no. 5, pp. 518-524.

- [170] Y. Barrandon, "The epidermal stem cell: an overview," vol. 4, no. 4, pp. 209-215.
- [171] S.B. Hoath and D. Leahy, "The organization of human epidermis: functional epidermal units and phi proportionality," *J.Invest.Dermatol.*, vol. 121, no. 6, pp. 1440-1446.
- [172] A. Gandarillas, D. Davies and J. Blanchard, "Normal and c-Myc-promoted human keratinocyte differentiation both occur via a novel cell cycle involving cellular growth and endoreplication," *Oncogene*, vol. 19, no. 29, pp. 3278-3289.
- [173] J. Zanet, A. Freije, M. Ruiz, V. Coulon, J.R. Sanz, J. Chiesa and A. Gandarillas, "A mitosis block links active cell cycle with human epidermal differentiation and results in endoreplication," *PLoS One*, vol. 5, no. 12, pp. e15701.
- [174] A. Gandarillas and A. Freije, "Cycling up the epidermis: reconciling 100 years of debate," *Exp.Dermatol.*
- [175] A. Puliafito, L. Hufnagel, P. Neveu, S. Streichan, A. Sigal, D.K. Fygenson and B.I. Shraiman, "Collective and single cell behavior in epithelial contact inhibition," *Proceedings of the National Academy of Sciences*, vol. 109, no. 3, pp. 739-744.
- [176] M.V. Plikus, D.L. Gay, E. Treffeisen, A. Wang, R.J. Supapannachart and G. Cotsarelis, "Epithelial stem cells and implications for wound repair,"
- [177] J.A. Sherratt, "Wavefront propagation in a competition equation with a new motility term modelling contact inhibition between cell populations," *Proceedings of the Royal Society of London.Series A: Mathematical, Physical and Engineering Sciences*, vol. 456, no. 2002, pp. 2365-2386.
- [178] S. Werner and R. Grose, "Regulation of wound healing by growth factors and cytokines," *Physiol.Rev.*, vol. 83, no. 3, pp. 835-870.
- [179] J. Sherratt and J. Murray, "Mathematical analysis of a basic model for epidermal wound healing," *J.Math.Biol.*, vol. 29, no. 5, pp. 389-404.
- [180] J.A. Sherratt and J. Murray, "Models of epidermal wound healing," *Proceedings of the Royal Society of London.Series B: Biological Sciences*, vol. 241, no. 1300, pp. 29-36.
- [181] A.Q. Cai, K.A. Landman and B.D. Hughes, "Multi-scale modeling of a wound-healing cell migration assay," *J.Theor.Biol.*, vol. 245, no. 3, pp. 576-594.
- [182] M.H. Swat, G.L. Thomas, J.M. Belmonte, A. Shirinifard, D. Hmeljak and J.A. Glazier, "Multi-scale modeling of tissues using CompuCell3D," *Methods Cell Biol.*, vol. 110, pp. 325.
- [183] R.M. Merks, E.D. Perryn, A. Shirinifard and J.A. Glazier, "Contact-inhibited chemotaxis in de novo and sprouting blood-vessel growth," *PLoS computational biology*, vol. 4, no. 9, pp. e1000163.

- [184] N.B. Ouchi, J.A. Glazier, J. Rieu, A. Upadhyaya and Y. Sawada, "Improving the realism of the cellular Potts model in simulations of biological cells," *Physica A: Statistical Mechanics and its Applications*, vol. 329, no. 3, pp. 451-458.
- [185] Y. Zhang, G.L. Thomas, M. Swat, A. Shirinifard and J.A. Glazier, "Computer simulations of cell sorting due to differential adhesion," *PloS one*, vol. 6, no. 10, pp. e24999.
- [186] C. Blanpain and E. Fuchs, "Epidermal homeostasis: a balancing act of stem cells in the skin," *Nature reviews Molecular cell biology*, vol. 10, no. 3, pp. 207-217.
- [187] F. Jiménez, C. Garde, E. Poblet, B. Jimeno, J. Ortiz, M.L. Martínez, A. Gutiérrez-Rivera, V. Pérez-López, U. Etxaniz and C. Naveda, "A pilot clinical study of hair grafting in chronic leg ulcers," *Wound Repair and Regeneration*, vol. 20, no. 6, pp. 806-814.
- [188] J. Lademann, U. Jacobi, C. Surber, H. Weigmann and J. Fluhr, "The tape stripping procedure—evaluation of some critical parameters," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 72, no. 2, pp. 317-323.
- [189] C. Paillet-Mattei, C. Guerret-Piécourt, H. Zahouani and S. Nicoli, "Interpretation of the human skin biotribological behaviour after tape stripping," *Journal of The Royal Society Interface*, vol. 8, no. 60, pp. 934-941.
- [190] H. Van den Brenk, "Studies in restorative growth processes in mammalian wound healing," *Br.J.Surg.*, vol. 43, no. 181, pp. 525-550.
- [191] A. Gandarillas, "The mysterious human epidermal cell cycle, or an onco-gene-induced differentiation checkpoint," *Cell Cycle*, vol. 11, no. 24, pp. 4507-4516.
- [192] D. Grossman and D.J. Leffell, "Squamous cell carcinoma," *Fitzpatrick's Dermatology in General Medicine. Seventh edition. Mc Graw Hill, New York*, pp. 1028-1035.