



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**Algoritmo basado en agentes para la simulación
computacional de un modelo simple de morfogénesis**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

P R E S E N T A:

JOSÉ ANTONIO VILLARREAL MARTÍNEZ



**DIRECTOR DE TESIS:
DR. GERMINAL COCHO GIL**

2011



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Datos del alumno

Villarreal
Martínez
José Antonio
56102714
Universidad Nacional Autónoma de México
Facultad de Ciencias
093193090

Datos del tutor

Dr.
Germinal
Cocho
Gil

Datos del sinodal 1

Dr.
Pedro Eduardo
Miramontes
Vidal

Datos del sinodal 2

Dr.
José de Jesús
Galaviz
Casas

Datos del sinodal 3

Dr.
Carlos
Gershenson
García

Datos del sinodal 4

Dr.
David Philip
Sanders

Datos del trabajo escrito

Algoritmo basado en agentes para la simulación computacional
de un modelo simple de morfogénesis.
70 p.
2011

Índice general

Introducción	1
I Preliminares	6
1. Antecedentes	7
1.1. Aspectos biológicos	7
1.2. Aspectos matemáticos	8
1.2.1. Autómata celular	8
1.2.2. Autómata celular unidimensional	9
1.2.3. Autómata celular bidimensional	11
1.2.4. Modelos matemáticos de morfogénesis basados en AC	12
1.3. Aspectos físicos	15
1.3.1. Tensión superficial	16
1.3.2. Del modelo continuo al discreto	17
1.3.3. Formación de cúmulos en el modelo de Vinkovic-Kirman	19
2. El algoritmo de Barnes-Hut	21
2.1. El problema de los <i>N-cuerpos</i>	21
2.2. Reducción del orden de complejidad del algoritmo “simple” .	24
2.3. Implementación en el plano	25
2.3.1. Árbol cuaternario adaptativo	25
2.3.2. Construcción del árbol cuaternario	28
2.3.3. Cálculo de la fuerza	32
2.3.4. La complejidad del algoritmo de Barnes-Hut	33
3. Modelos basados en agentes	37
3.1. El concepto de agente	38
3.2. Los MBA en los sistemas complejos	40
3.3. MASyV	43
3.3.1. Aplicaciones de MASyV en sistemas complejos	44

II	El algoritmo basado en agentes	47
4.	El modelo simple de morfogénesis	48
4.1.	Formación de cúmulos de células	48
4.2.	Cálculo de la fuerza usando el potencial	49
4.3.	Migración dirigida	50
4.4.	Implementación	51
4.4.1.	El ambiente	51
4.4.2.	La célula como un agente	51
4.4.3.	Las reglas del MBA	53
4.4.4.	Escalas de tiempo	53
4.5.	El algoritmo general	53
5.	Resultados	56
5.1.	La función potencial y sus parámetros	56
5.2.	Experimentos	57
5.2.1.	Experimento 1	57
5.2.2.	Experimento 2	59
5.2.3.	Experimento 3	60
5.2.4.	Experimento 4	60
5.2.5.	Experimento 5	61
5.2.6.	Experimento 6	61
5.2.7.	Experimento 7	62
	Conclusiones	65
	Bibliografía	68

Agradecimientos

A mis asesores Germinal Cocho y Pedro Miramontes por compartir sus ideas e impulsarme a escribir esta tesis.

A José Galaviz, Carlos Gershenson y David Sanders por el tiempo que dedicaron a la revisión de este trabajo aportando sus valiosos comentarios y correcciones.

A Catherine Beauchemin por la información proporcionada sobre las aplicaciones de *MASyV* y asesoría técnica en su instalación.

A Víctor Mireles, Lorena Caballero, Alejandro Heredia, Jorge Rosas y Andrew Green que con sus opiniones contribuyeron indirectamente al proceso de construcción del algoritmo.

A mis padres, a mis abuelos y a Audra por su apoyo de siempre.

Resumen

En esta tesis se presenta un algoritmo para simular en una computadora la generación de patrones de coloración. Se toman como ejemplo los patrones de coloración en la piel de la larva de un anfibio como el ajolote *Ambystoma mexicanum*. De manera muy simplificada, desde la perspectiva de la física y las ciencias de la computación, se construye un algoritmo basado en agentes partiendo de la idea de concebir a la célula como un agente susceptible a los estímulos del medio. El algoritmo se presenta como un punto de partida para otros modelos más complejos.

En el modelo se consideran dos tipos de células diferentes que se repelen mutuamente. Esta característica se presenta también en modelos de segregación social como el modelo de Schelling. Se ha mostrado que trasladar el modelo discreto de Schelling a un modelo continuo, equivale a transformarlo en un modelo físico de tensión superficial, esto sugiere que en la generación de patrones de coloración la tensión superficial juega un papel importante.

Además de incluir la repulsión entre células diferentes en el modelo, también se incluyen fuerzas largo alcance representadas por un potencial. El cálculo directo de dichas fuerzas en un sistema de N cuerpos es $O(N^2)$, se utiliza el algoritmo de Barnes-Hut que es $O(N \log N)$ para realizar dicho cálculo, mejorando así el orden de complejidad algorítmica.

En el algoritmo se definen tres reglas para mover una célula de un lugar a otro; la primera representa la repulsión entre tipos diferentes de células, la segunda regla nos da la dirección de la fuerza representada por la función potencial y la tercera que simula la migración de la célula desde la cresta neural (la parte superior de la malla) hacia abajo. Variando algunas de estas reglas, se presentan los resultados de la simulación computacional, al final se muestran patrones de coloración parecidos a los que se ven en la piel de algunos anfibios o peces.

Introducción

Un fenómeno biológico, el proceso de desarrollo de formas especializadas en organismos celulares conocido como morfogénesis y en particular la formación de los patrones de coloración en la piel de algunos anfibios, es el que sirve de motivación para este trabajo. La comprensión de dicho fenómeno además de ser de gran importancia para la biología por estar relacionado con aspectos evolutivos y de supervivencia de los animales sería de gran utilidad para comprender otros fenómenos aún más complejos de morfogénesis, como por ejemplo la formación de tumores.

Pretendiendo comprender este fenómeno mediante una simulación computacional no se toma más que lo esencial de la biología y concentramos toda nuestra atención en la creación de un algoritmo basado en agentes que pudiera servir como punto de partida para otras simulaciones computacionales y aplicarse a un espectro más amplio de problemas.

En el diseño de algoritmos, el primer paso es especificar el tipo de problema que se pretende resolver, dejando a un lado los detalles y buscando las características que comparten todos los casos particulares del mismo problema, llegando a una *abstracción del problema*. Luego se inicia la construcción del algoritmo para esta *abstracción*, generando una solución o al menos una aproximación a la solución. El algoritmo así construido no resolverá ningún otro problema que no comparta todas las características especificadas.

Análogamente, en las simulaciones computacionales, en su proceso de construcción se tienen que incorporar las características esenciales que determinan la dinámica del fenómeno que se pretende simular. Finalmente, se llega a una *abstracción del fenómeno* y se construye la simulación para este “fenómeno mínimo” excluyendo los detalles irrelevantes.

Motivación biofísica

Se ha elegido para este estudio la formación de los patrones de coloración en la piel de algunos anfibios por su relativa simpleza en comparación con otros procesos de morfogénesis. El proceso involucra interacciones de miles o millones de células pigmentarias, en los casos más simples de dos tipos diferentes de células. El medio se conoce como la matriz extracelular, pero

para simplificar, en este trabajo podríamos pensar que en el desarrollo del embrión, en sus primeras etapas, el medio en donde se mueven las células y los componentes de las células mismas es similar a un líquido sujeto a fuerzas físicas.

Las interacciones de las moléculas de un líquido se dan a través de las fuerzas entre sus átomos que al estar ionizados interactúan mediante fuerzas de Coulomb, pero también hay que describir las interacciones entre dos moléculas que no estén ionizadas como es el caso de las fuerzas descritas por el potencial de Lennard-Jones o alguno parecido, donde se representa una fuerza de repulsión de corto alcance y una fuerza de atracción de largo alcance. Otra de las fuerzas físicas que se manifiestan en un medio líquido es la tensión superficial.

Las células están hechas de moléculas así que podríamos aproximar sus interacciones con las fuerzas entre sus moléculas. Podríamos pensar a un sistema de células, como un sistema de partículas, en donde las fuerzas de corto y largo alcance intervienen en la formación de patrones celulares.

Debido a este hecho, en el presente trabajo hablaremos indistintamente de células, moléculas, partículas, cuerpos o agentes para referirnos a la misma cosa.

Las fuerzas de largo alcance y el algoritmo de Barnes-Hut

Aunque las magnitudes de las fuerzas de largo alcance pueden ser muy pequeñas entre cada par de partículas, al tener miles de partículas, tendríamos miles de pares, donde cada par aporta una pequeña magnitud a la fuerza. Al sumar todas estas pequeñas magnitudes, la suma es una magnitud que ya no es posible despreciar. Por esto es importante calcular las fuerzas de largo alcance en sistemas con un gran número de partículas o cuerpos.

Con la aparición de las computadoras de alto rendimiento, se abrió la posibilidad de llevar a cabo este tipo de cálculos. Pero no ha sido suficiente debido a que los fenómenos para los cuales se requieren este tipo de cálculos además de que necesitan incluir el mayor número posible de partículas, incluyen otros procesos que merecen ser representados en la computadora para obtener una mejor aproximación de su dinámica.

Calcular las interacciones de N cuerpos en una computadora puede representar una gran cantidad de recursos si el número N es del orden de miles o millones de partículas. Es aquí donde el análisis de algoritmos y estructuras de datos proporcionan soluciones adecuadas y eficientes para realizar los cálculos de estas interacciones. Una de estas soluciones es el algoritmo de Barnes-Hut, que es parte de una familia de algoritmos llamados métodos *Treecode*. Estos algoritmos fueron creados para resolver problemas en astrofísica, como calcular fuerzas gravitacionales y la dinámica de las ga-

laxias donde estan involucradas interacciones de largo alcance de millones de cuerpos celestes [27]. Se les llama *Treecode* en inglés porque usan la idea de dividir el espacio de una forma jerárquica que se representa con una estructura de datos llamada árbol.

La idea de subdividir el espacio en el algoritmo de Barnes-Hut proviene del hecho de que al calcular la fuerza que ejerce una lejana galaxia sobre un cuerpo celeste, esta galaxia se puede pensar como un cúmulo de cuerpos celestes y ser tratada como un sólo cuerpo que se encuentra dentro de alguna subdivisión del espacio.

Subdividir el espacio de manera recursiva para dividir el problema en otros más simples, ha sido ampliamente utilizada en el diseño de algoritmos, por ejemplo para encontrar la envoltura convexa de un conjunto de puntos (*Convex Hull*), o también para encontrar el punto más cercano a otro en un espacio con una métrica determinada. Esta técnica genera algoritmos bastante eficientes y en muchos casos genera estructuras de datos de árboles para representar las subdivisiones del espacio.

El algoritmo de Barnes-Hut es un ejemplo de cómo ciencias como la física, la biología y las matemáticas hacen aportaciones a las ciencias de la computación, y viceversa, así cada una de estas ciencias desde sus perspectivas confluyen en el estudio de la naturaleza. Otros ejemplos son los algoritmos inspirados en procesos biológicos, en las ciencias de la computación encontramos a los algoritmos genéticos, los autómatas celulares o los modelos basados en agentes, este último incluso puede ser todo un paradigma de programación.

El algoritmo basado en agentes

Los organismos vivos estan hechos de células. Para llevar a cabo sus funciones especializadas, las células deben usar información genética; sintetizar, almacenar, y transportar biomoléculas; transformar diversas formas de energía; transferir señales; mantener estructuras internas; y responder a ambientes externos. Muchos de estos procesos involucran aspectos mecánicos [1].

Estas características nos hacen pensar a las células como “entes autónomos”, que responden a las condiciones del medio y se comunican entre sí, esto nos indica que bien podemos modelar a la célula como un agente dentro de un sistema *multiagente* o un *modelo basado en agentes* (MBA), como también se le conoce.

El paradigma de los MBA ha sido tomado por la inteligencia artificial para modelar e implementar soluciones a una amplia variedad de problemas, también ha ganado popularidad en la comunidad biomatemática para realizar simulaciones por computadora. Esto se debe a que al igual que los autómatas celulares, los modelos basados en agentes se definen mediante

reglas que deben de seguir individualmente los agentes y que derivan en comportamientos complejos.

Aunque existen otros trabajos que usan las ecuaciones diferenciales ordinarias (EDO) para modelar estos fenómenos, las interacciones entre las células o partículas y su distribución no homogénea en el espacio nos ha sugerido utilizar un MBA y no a las EDO. Las EDO son bien conocidas y nos proveen de un modelo simple y elegante que requiere de pocos parámetros en comparación a los MBA o a las ecuaciones diferenciales parciales. Las EDO son útiles cuando no se tiene la ubicación o distribución de los individuos de la población que se estudia, pero justo esta puede ser su debilidad: las EDO suponen que la población que se estudia es homogénea y uniformemente distribuida en el espacio de la simulación en todo momento.

En nuestro caso es importante considerar la distribución no homogénea que pueden tener la células en el espacio de la simulación y su cambiante distribución en el medio, esto se debe a su naturaleza, ya que las células provienen de la cresta neural y van migrando a diversos sitios en la epidermis. Aunque con las ecuaciones diferenciales parciales se pueden modelar los cambios en espacio y tiempo, entre más complejos sean estos cambios más complejas son las ecuaciones que los modelan y más difícil es calcular sus soluciones, generando un problema extra de computación.

Partiendo de estas ideas en este trabajo se construye un algoritmo basado en agentes para la simulación por computadora del proceso de morfogénesis de los patrones de coloración.

Objetivo

El algoritmo trata de reproducir la configuración de un sistema celular parecido al que forma los patrones de coloración en la piel de algunos anfibios como son los que se presentan en la larva del ajolote *Ambystoma mexicanum*.

En el sistema se incluyen fuerzas de corto y de largo alcance además de la tensión superficial. Con la simulación computacional se pretende estudiar los efectos de estas fuerzas físicas en el proceso de morfogénesis.

Plan de la tesis

La tesis se divide en dos partes, la primera parte trata de los antecedentes y conceptos involucrados en el diseño del algoritmo. Se compone de los capítulos uno, dos, y tres, en el primer capítulo se ven algunos aspectos biológicos, matemáticos y físicos que están involucrados en el estudio de este fenómeno. El apartado sobre los aspectos matemáticos se refiere principalmente a los autómatas celulares ya que considero son precursores del enfoque que aquí se propone, en los aspectos físicos se abordan las fuerzas físicas que se incluyen en el algoritmo.

En el capítulo dos veremos con detalle el algoritmo de Barnes-Hut que forma parte del algoritmo que se presenta en esta tesis. Se estudian sus estructuras de datos y complejidad algorítmica; también exponemos el criterio que usa para calcular las interacciones de partícula a partícula o de partícula a cúmulo, dependiendo del grado de exactitud que requiera el algoritmo.

En el capítulo tres explicaremos cómo se define un modelo basado en agentes y también se dan a conocer algunos de los trabajos que han seguido esta perspectiva. Enseguida en ese mismo capítulo se exponen las características de *MASyV (Multi-Agent System Visualization)*, el *software* que utilizaremos para visualizar la simulación y se dan ejemplos de las simulaciones computacionales en donde se ha utilizado exitosamente. Usar *MASyV* permitió dejar a un lado el problema de la visualización de la simulación y concentrarnos en diseñar el algoritmo.

La segunda parte se compone del capítulo cuatro y cinco. En el capítulo cuatro se expone el modelo simple de morfogénesis que se presenta en este trabajo, se explican cada una de las interacciones que fueron incorporadas como reglas en el algoritmo y finaliza con la implementación del algoritmo.

En el capítulo cinco se presentan los resultados de los experimentos que se realizaron con la simulación computacional, dichos resultados corresponden a patrones de coloración que se obtuvieron modificando algunas reglas del algoritmo. Por último se dan las conclusiones.

Parte I

Preliminares

Capítulo 1

Antecedentes

1.1. Aspectos biológicos

La morfogénesis es el proceso por el cuál un grupo de células dentro del desarrollo constituyen un embrión y cómo este va cambiando hasta dar lugar a una forma determinada.

El estudio de las células pigmentarias, los cromatóforos, desde la biología del desarrollo o embriología nos ha proporcionado las características básicas de estas células, su clasificación y el conocimiento de algunas reacciones fisico-químicas, pero no nos ha dado con especificidad las interacciones, tampoco ha descrito con suficiente exactitud las vías de movimiento de estas.

Se ha sugerido que esto está estrechamente relacionado con las moléculas que se encuentran en la matriz extracelular (el conjunto de materiales extracelulares que forman parte de un tejido) guiando a las células a lugares específicos, se mencionan a las CAM (en inglés *Cell Adhesion Molecules*) que se encuentran en la superficie celular y sirven a las células para unirse con otras células o con la matriz extracelular. Las uniones entre células del mismo tipo se denominan interacciones homotípicas, y las uniones entre células de diferente tipo se llaman heterotípicas. Básicamente, las interacciones celulares en este proceso son las interacciones entre las células pigmentarias y las interacciones entre éstas con el medio, que es la matriz extracelular y que también esta formada por otras células. En resumen interacciones célula-célula.

Con respecto a la migración de estas células se sabe que en el embrión las células pigmentarias empiezan a desplazarse desde la cresta neural, después estas se unen o repelen dependiendo del tipo de célula, al unirse forman conjuntos de células que llamaremos *cúmulos*. Estos *cúmulos* deberán obedecer a las leyes de la física este hecho es una de las hipótesis de nuestro modelo.

1.2. Aspectos matemáticos

Matemáticamente podríamos decir que existen dos enfoques para modelar la morfogénesis de los patrones de coloración: los modelos de Turing basados en sistemas de reacción-difusión descritos a través de las ecuaciones diferenciales y su contraparte discreta. Los modelos discretos suelen ser autómatas celulares (AC) o en su caso más general pueden ser modelos basados en agentes (MBA). Los modelos de Turing consideran que las células se encuentran uniformemente distribuidas y que siguen pasivamente químicos esparcidos en el medio, los modelos discretos toman en cuenta fundamentalmente las interacciones célula-célula, ya sean locales o de largo alcance, pero también tienen la flexibilidad para tomar elementos de los modelos de Turing.

Entre los primeros modelos con autómatas celulares para la morfogénesis de patrones de coloración podemos mencionar el descrito por Cocho, Pérez Pascual y Rius en [3].

Existen otras investigaciones más recientes basadas en datos experimentales y modeladas con autómatas celulares bidimensionales; éstas toman en cuenta interacciones célula-célula incluyendo las interacciones con la matriz extracelular, y también incluyen la forma de migración de las células, (p. ej. Deutsch y Dormann en [10]).

Un conjunto de células también puede pensarse como un sistema físico, sujeto a interacciones mecánicas y buscando configuraciones de mínima energía, vale la pena explorar mediante la computadora cómo influyen estas fuerzas en la configuración de los patrones de coloración. Un ejemplo de este tipo de experimentos *in silico* es la simulación computacional hecha por Caicedo y Shinbrot en [2] donde se presta especial atención a las interacciones mecánicas de las células, incluyendo modelos de tensión superficial célula-célula y simulando fuerzas de largo alcance a través de la matriz extracelular.

Este trabajo sigue el enfoque discreto, así que revisaremos de forma breve algunos de los esfuerzos que han seguido esta línea de investigación, estos modelos usaron a los AC para sus simulaciones, así que comenzamos con su definición.

1.2.1. Autómata celular

Los AC son una idealización matemática de sistemas físicos en los cuales espacio y tiempo son discretos, y los valores de las magnitudes físicas caen en un conjunto de valores discretos [28].

Podemos resumir que un autómata está definido por los siguientes elementos:

1. Una malla de sitios de una o más dimensiones.

2. Todos los sitios son equivalentes (homogeneidad).
3. Un conjunto de estados $S = 0, \dots, k - 1$, de los cuales los sitios pueden tomar solo un valor $\sigma_i \in S$.
4. Condiciones en la frontera. Estas condiciones definen las vecindades para los extremos de la malla. La condición en la frontera propiciará una evolución del AC en un espacio finito y no infinito.
5. Una regla de evolución ϕ que determina el estado que un sitio tendrá en el tiempo t a partir del estado que los sitios de su vecindad tienen el tiempo $t + 1$. La vecindad de un sitio de radio r , estará formada por las celdas de la malla más cercanas a este sitio que estén a distancia r . La actualización de estos sitios suele ser síncrona.

1.2.2. Autómata celular unidimensional

La vecindad de radio r en una dimensión consta de n celdas donde $n = 2r + 1$. Para cada sitio los valores de su vecindad de radio r en el tiempo t determinarán el valor de su estado para el tiempo $t + 1$, de esta forma la regla de evolución del AC es una función de las celdas de la vecindad. Entonces para cada sitio i en el arreglo del autómata (ver cuadro 1.1) tenemos:

$$\sigma_i(t + 1) = \phi(\sigma_{i-r}(t), \dots, \sigma_i(t), \dots, \sigma_{i+r}(t)) \quad (1.1)$$

donde la regla estará completamente determinada una vez que se halla especificado cada una de las k^{2r+1} posibles configuraciones para una vecindad dada, así habrá un total de $k^{k^{2r+1}}$ posibles reglas de evolución.

...	$i - 2$	$i - 1$	i	$i + 1$	$i + 2$...
-----	---------	---------	-----	---------	---------	-----

Cuadro 1.1: Arreglo de un autómata celular unidimensional

Para las celdas que están en la frontera de la malla del autómata se tienen que definir las condiciones de frontera que definen una vecindad extendida para las celdas de los extremos, estas pueden ser periódicas; una celda en un extremo tiene como vecina a la celda del extremo opuesto, en una vecindad de una dimensión obtenemos un anillo, y en una vecindad de dos dimensiones obtenemos un toro; otra condición de frontera es la reflexiva, la celda vecina de una celda en un extremo es ella misma; la última condición de frontera es una fija, la celda vecina de una celda en el extremo queda con un valor determinado. (Ver figura 1.1).

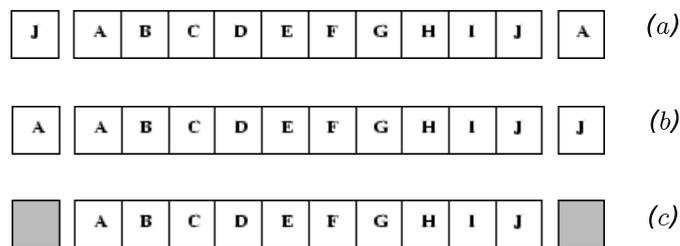


Figura 1.1: En la figura se muestran los ejemplos de las condiciones de frontera en una dimensión. En orden se muestran (a) periódica, (b) reflexiva y (c) fija.

De los trabajos más importantes acerca de los autómatas celulares encontramos el de Wolfram [28] quién centró sus estudios en los autómatas celulares de una dimensión con dos posibles estados en cada sitio 0 ó 1 donde la vecindad de cada celda esta compuesta por ella misma y las dos celdas adyacentes. Las reglas pueden ser expresadas por 8 dígitos binarios de la siguiente manera:

111	110	101	100	011	010	001	000
0	1	0	1	1	0	1	0

Cuadro 1.2: Reglas del autómata celular

En este caso tenemos que $k = 2$ el conjunto de estados es $S = 0, 1$, el radio de interacción es $r = 1$, lo que nos da una vecindad de $n = 3$ sitios con $k^{2r+1} = 8$ posibles configuraciones de esta vecindad y $k^{k^{2r+1}} = 2^8 = 256$ posibles reglas de evolución, la regla que se muestra en el cuadro 1.2 es $\phi = 01011010$ que en su representación decimal es:

$$\phi = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 = 90. \quad (1.2)$$

De los 256 posibles reglas sólo se consideran “válidas” 32 [28], ya que el origen biológico de los AC impone ciertas restricciones. (Ver figura 1.2).

- La regla debe ser simétrica. (110 y 011 deben de generar lo mismo)
- Sólo vida puede generar vida. (Entonces 000 debe generar 0)

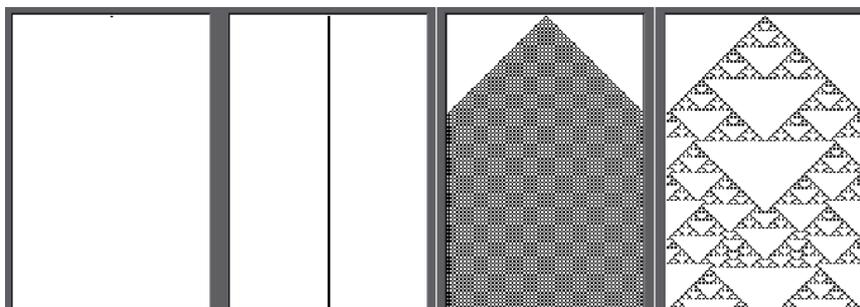


Figura 1.2: Cuando se inicia con un valor distinto de cero, los patrones de las 32 reglas pueden ser divididos en cuatro tipos (patrones que desaparecen con el tiempo, sólo ser copias de la celda inicial por siempre, producir un tipo completamente uniforme de renglones o pares de renglones, o desarrollar patrones no triviales). Los patrones mostrados en esta figura son el resultado de las reglas 128 (10000000), 36 (00100100), 250 (11111010), y 90 (01011010) respectivamente.

1.2.3. Autómata celular bidimensional

Los autómatas celulares en dos dimensiones son importantes ya que pueden representar modelos para la formación de patrones en sistemas físicos. Algunas aplicaciones incluyen crecimiento de cristales, sistemas de reacción difusión y dinámica de fluidos [28].

Al igual que en los autómatas celulares unidimensionales el estado de una de las celdas de la malla toma un valor en el conjunto S y esta determinado por el valor de los estados de sus vecinos más próximos, ahora en dos dimensiones. En el caso de una malla cuadriculada se consideran dos tipos de vecindades, la de von Neumann y la de Moore.

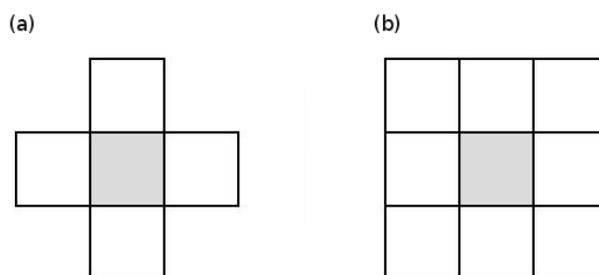


Figura 1.3: En la figura se muestran las vecindades de von Neumann (a), y la vecindad de Moore (b) de la celda marcada en gris.

Por ejemplo, para la vecindad de von Neumann 1.3 (a), la regla para la celda en la posición i, j quedaría definida de la siguiente forma:

$$\sigma_{i,j}(t+1) = \phi(\sigma_{i,j}(t), \sigma_{i,j+1}(t), \sigma_{i+1,j}(t), \sigma_{i,j-1}(t), \sigma_{i-1,j}(t)) \quad (1.3)$$

En la referencia [28] se hace una descripción más detallada de estos autómatas celulares y se encuentran varios ejemplos, más adelante referimos un ejemplo de modelo de autómata celular bidimensional para simular la morfogénesis de los patrones de coloración en el pez cebra.

1.2.4. Modelos matemáticos de morfogénesis basados en AC

Haciendo énfasis en las interacciones célula-célula y en la teoría clonal, uno de los primeros modelos que usan este enfoque discreto es el modelo de Cocho et al. [3] que consiste en un autómata celular unidimensional con las siguientes características:

1. Un número relativamente pequeño de células primarias diferenciadas o conformando una configuración inicial de mínima energía.
2. Esta configuración de un número pequeño de células será la semilla y la condición inicial para un tipo de desarrollo en un autómata celular y la configuración final se congelará.
3. Cada una de estas células en la configuración final se dividirá, siendo el origen de una mancha o región.

En este modelo de autómata celular presentado en el trabajo de Cocho et al. se representan patrones de coloración en animales, principalmente en serpientes y peces. Tomando en cuenta la simetría bilateral de estos animales y las restricciones embriológicas, se usa una malla rectangular, cada estado del autómata unidimensional corresponde a un renglón a lo largo del eje x y se determina que la posición del estado inicial del autómata quede en el renglón que se encuentra a la mitad del eje y y que el autómata celular se propague en ambas direcciones del eje y de una manera simétrica.

Se tienen mallas rectangulares o triangulares, R representa a la malla rectangular y T a la triangular, para cada celda la regla depende de la suma de los valores de la celda y de las celdas vecinas, cada celda puede tomar k valores. Así por ejemplo, $R21$ representa un autómata celular con una malla rectangular, con $k = 2$ y con un radio de interacción $r = 1$, es decir sus dos vecinos adyacentes. Si $r = 0$, quiere decir que el siguiente estado de una determinada celda depende sólo de su estado actual.

Para el caso de las serpientes se presentan las siguientes reglas:

- (a) *R20 Homotípica*. (Regla 10). Esta regla da origen a bandas transversales a lo largo del animal. Un ejemplo consistente con este patrón es la serpiente coralillo. (Figura 1.4 (a)).
- (b) *R21 Homotípica*.
- (I) *Regla 1000*. Triángulos simétricos construyen un rombo. (Figura 1.4 (c))
 - (II) *Regla 1100*. Se obtienen bandas.
 - (III) *Regla 1110*. Es el espejo de la regla 1000.
 - (IV) *Transición de 1100 a 1000*. (Figuras 1.4 (b) y 1.4 (d)).
- (c) *T21 Homotípica*.
- (I) *100*. De nuevo se obtienen triángulos simétricos que construyen rombos.
 - (II) *110*. Espejo de 100.
 - (III) *Transición de 110 a 100*. (Figura 1.4 (d)).

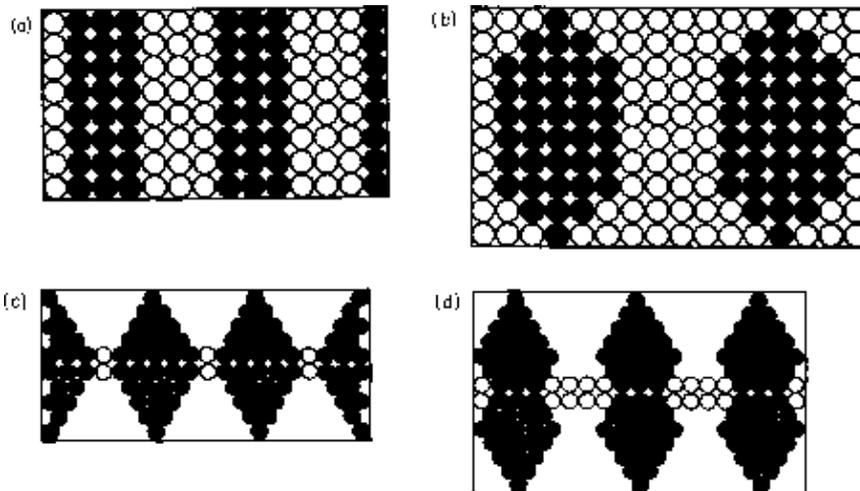


Figura 1.4: En esta figura tomada de [3] el renglón inicial esta en el centro de la figura y empieza a crecer hacia arriba y hacia abajo. (a) R20, 10 ó R21, 1100. (b) Una transición de R21, 1100 a 1000. (c) T21, 100. (d) Una transición de T21, 110 a 100.

En el estudio de los autómatas celulares, la sensibilidad a los cambios de condiciones iniciales es uno de los aspectos que influyen en las configuraciones finales de los patrones que se generan, uno de los estudios que abordan esta problemática es el que realizó Miramontes [29] donde también hace una

clasificación de las reglas de los autómatas celulares de acuerdo a los patrones que producen. Los rasgos en las estructuras de los patrones se proponen como variables subjetivas, estas permitirán visualizar los rasgos que ciertas reglas comparten.

Dentro de los modelos de morfogénesis basados en datos experimentales que usan un autómata celular bidimensional, que ya no sólo estudian interacciones de corto alcance sino también de largo alcance, mencionamos el trabajo de Deutsch y Dorman, 2005, [10].

En los autómatas celulares cada paso se realiza de manera síncrona, actualizando el estado de cada celda de la malla de manera simultánea. La dinámica celular no es exactamente de esta forma así que para hacer simulaciones computacionales más realistas se requiere incorporar comportamientos asíncronos en el modelo, como ejemplo de estas modificaciones a los autómatas celulares podemos mencionar el modelo de (Moreira y Deutsch, 2005) [30]. Este modelo consiste en un autómata celular bidimensional que extiende la naturaleza síncrona del autómata para incorporar movimientos asíncronos, en particular cada célula en una vecindad elegida aleatoriamente se mueve de forma asíncrona dependiendo de si al moverse en cierta dirección puede minimizar su energía. Esta característica es una extensión de la definición clásica de autómata celular. (Ver figura 1.5)

Incluir comportamientos asíncronos en un autómata celular en el cuál cada una de sus celdas se actualiza simultáneamente, aumenta el grado de complejidad del autómata. Los comportamientos asíncronos en las células pigmentarias sugieren la utilización de herramientas más adecuadas para el modelo, herramientas que reflejen con más naturalidad la dinámica de las células. En este trabajo se eligieron los modelos basados en agentes que pueden incluir de forma más natural comportamientos asíncronos.

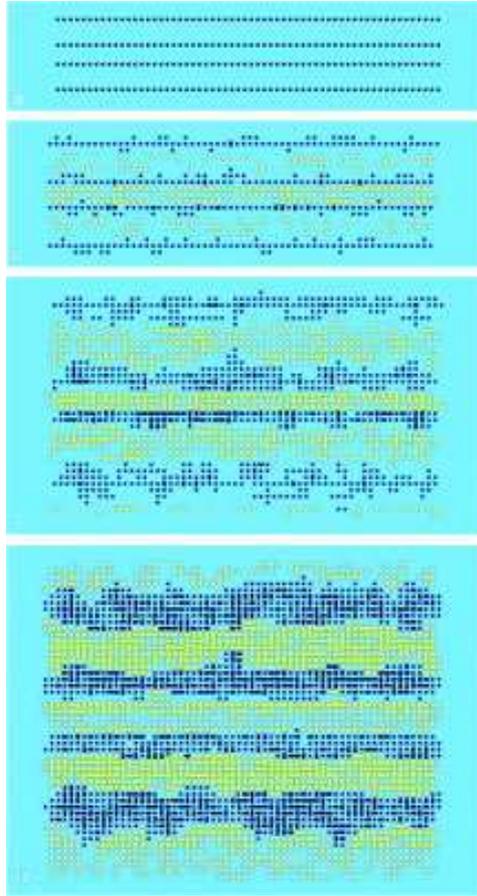


Figura 1.5: En esta figura tomada de [30] se muestra la evolución del autómata celular bidimensional para modelar los patrones de coloración del pez cebra, las fuerzas de atracción entre melanóforos (en azul) es menor que entre xantóforos (en amarillo) y células de distinto tipo se repelen.

1.3. Aspectos físicos

En la simulación se han incluido algunas fuerzas físicas que consideramos importantes para establecer las interacciones entre las células. Primeramente incluimos una de las fuerzas de corto alcance en el modelo que es la de tensión superficial. Luego incluimos las fuerzas de atracción y de repulsión representadas por un potencial que pueden ser de corto y largo alcance y finalmente consideramos a las interacciones con la matriz extracelular como fuerzas externas.

Suponemos que el medio donde interactúan las células es un líquido. Análogamente como pasa con las partículas en un líquido donde podemos suponer que el sistema tenderá a minimizar su energía, en particular cada

estructura o cúmulo reducirá su energía tanto como sea posible. Las células en el interior del cúmulo se repelen y atraen mutuamente, con la misma magnitud y en todas direcciones, entonces la suma de estas fuerzas es cero minimizando su energía y por lo tanto estabilizándose, mientras que las células de la superficie se encuentran en desequilibrio cuando desde fuera del cúmulo se ejercen fuerzas de atracción y repulsión de diferente magnitud a las que se ejercen en el interior, lo cuál las hace susceptibles de cambiar su energía.

Esto indica que para describir el comportamiento del sistema hace falta observar el comportamiento de la fuerza en la superficie de los cúmulos, en física esta fuerza es conocida como *tensión superficial*.

1.3.1. Tensión superficial

La tensión superficial es causada por la atracción entre las moléculas de un líquido por varias fuerzas intermoleculares. En el interior del líquido cada molécula es repelida y atraída igualmente en todas direcciones por moléculas vecinas resultando una red de fuerza cero. En la superficie del líquido las moléculas son atraídas por otras moléculas en el interior del líquido, pero no son atraídas tan intensamente por las moléculas del medio vecino (el vacío, aire u otro líquido).

Por lo tanto, todas las moléculas en la superficie están sujetas a una fuerza de atracción molecular que se ejerce desde el interior del líquido y que es balanceada únicamente por la resistencia del líquido a la compresión, lo que significa que no hay una red de fuerza al interior. No obstante, hay una fuerza para minimizar el área de la superficie, y en este respecto un líquido se asemeja a una membrana elástica en tensión. Así el líquido se comprime a sí mismo hasta que alcanza la menor área de superficie posible, logrando un estado de mínima energía.

En nuestro modelo, tenemos dos tipos de células, las células del mismo tipo se atraen, y células de tipos diferentes se repelen. Y ya que hemos pensado a las células como partículas de un líquido, y tenemos dos tipos de células, esto sería como si tuviéramos dos tipos de líquidos interactuando, que son diferentes y que no se mezclan.

Para que estos líquidos se mantengan en equilibrio, tienen que llegar a minimizar la energía de su superficie, este fenómeno que queremos modelar es un fenómeno continuo de tensión superficial, y pretendemos modelarlo mediante un modelo discreto construido en la computadora, para lo cuál aprovecharemos las heurísticas del modelo de [5], que incluye la fuerza de tensión superficial.

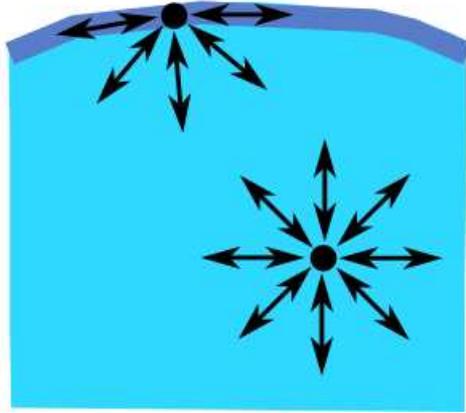


Figura 1.6: Diagrama de las fuerzas sobre las moléculas de un líquido. *Wikipedia*

1.3.2. Del modelo continuo al discreto

Recientemente se han encontrado similitudes entre modelos discretos con modelos continuos de tensión superficial como el caso del modelo de Schelling [5], en este caso Vinkovic y Kirman demuestran la analogía del sistema discreto con un modelo continuo, una ecuación diferencial que describe la tensión superficial.

El modelo de Schelling, es un modelo socioeconómico que surgió a finales de la década de 1960, Tom Schelling introdujo un modelo simple de segregación entre dos tipos de individuos, de color diferente, religión o clase social, cada individuo prefiere estar con personas de su mismo tipo, puede cambiar su lugar de acuerdo a una función de *utilidad*, si el individuo está rodeado de las personas de su mismo tipo o color, la función de *utilidad* es 1 , entonces el individuo permanece en su lugar, si no, la función de *utilidad* es 0 y el individuo se mueve a otro lugar, al lugar más cercano que esté disponible, o a donde le permitan sus medios, el espacio en donde se mueven los individuos es modelado por un malla en donde cada celda puede ser ocupada por a lo más un individuo. Este modelo mostró que con sólo una ligera preferencia del individuo a estar con personas de su mismo tipo, y si logra mudarse a otro sitio para satisfacer esta preferencia se producen verdaderas segregaciones en las poblaciones. El modelo ha sido bastante estudiado en las ciencias sociales, la estructura de las áreas de segregación están relacionadas con el espacio disponible y con las reglas con las que se mueven los individuos en este espacio. Para poder llegar a un enfoque analítico que abarque estas interacciones Vinkovic y Kirman recurren a un modelo físico.

En el modelo presentado por Vinkovic y Kirman [5], se transforma la

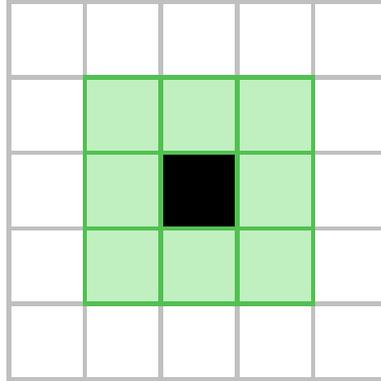


Figura 1.7: Vecindad de Moore con radio 1. Las celdas marcadas en verde son las celdas de interacción para la celda del centro, marcada en negro en esta figura.

En la malla discreta del modelo de Schelling en un medio continuo, se define el área ΔA de una celda de la malla y se reemplaza por la diferencial dA , en el caso discreto se define ΔN que es el número de partículas que pueden ocupar una celda, la celda puede encontrarse vacía o contener sólo una partícula. En el caso continuo este número se reemplaza por dN , así obtenemos la densidad de las partículas en el punto \vec{r} expresada por $n(\vec{r}) = dN/dA$, este número es constante ya que cada celda en la malla puede ser ocupada por a lo más una partícula.

Así que, en términos del cálculo infinitesimal tenemos:

$$\begin{aligned}\Delta A &\approx dA \\ \Delta N &\approx dN \\ n(\vec{r}) &= \frac{dN}{dA}\end{aligned}$$

En el modelo de Schelling para cada individuo se considera una vecindad de Moore¹ y se determina el valor de su *utilidad* (0 ó 1) dependiendo del número de vecinos diferentes. Por ejemplo si consideramos a un individuo de cierto tipo X , y su vecindad de Moore, este tendrá ocho posibles vecinos y si suponemos que no tolera más de tres del tipo Y , si resulta que tiene cuatro o más vecinos del tipo Y , la *utilidad* es 0 , y se desplazará buscando un mejor lugar, en otro caso la *utilidad* es 1 permaneciendo en su lugar.

En la construcción del modelo continuo de Vinkovic-Kirman se reemplaza a los individuos por partículas y la función de *utilidad* es reemplazada por una función de variable continua para medir la energía, esta última función se basa en determinar la medida del ángulo θ que ocupan partículas del

¹La vecindad de Moore de una celda consiste en las ocho celdas más cercanas. Ver figura 1.7

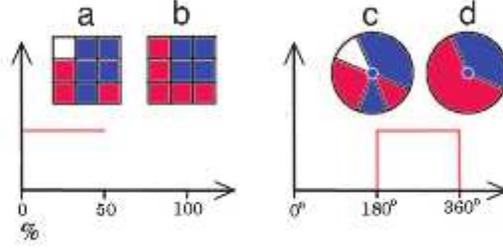


Figura 1.8: La función de utilidad en el caso discreto (izquierda), cuenta los vecinos del tipo contrario en la vecindad de Moore, en la gráfica corresponde al porcentaje, en el caso continuo (derecha) equivale a medir el ángulo θ que ocupan las partículas del tipo contrario alrededor de la diferencial dA . Figura tomada de [5].

tipo contrario alrededor de la diferencial dA (Ver figura 1.8). Así la *utilidad* es reemplazada por la energía $\epsilon(\theta)$ que es mínima cuando la *utilidad* es máxima y viceversa. Con esto se obtiene la energía total $dE = n\epsilon(\theta)dA$ contenida en dA o la energía por unidad de área $dE/dA = n\epsilon(\theta)$. Puesto que la repulsión entre cúmulos de partículas diferentes se manifiesta en la superficie de cada cúmulo, se toma la diferencial de la superficie dL y por lo tanto la energía por unidad de longitud es $d\Sigma = n\epsilon(\theta)dL$. El total de la energía superficial de un cúmulo es la integral de $d\Sigma$ sobre toda la superficie del cúmulo. La dinámica del cúmulo esta determinada por su tendencia a minimizar la energía superficial. Entonces Σ es la *tensión superficial* del cúmulo.

$$\int_S d\Sigma \quad (1.4)$$

Donde S es la superficie del cúmulo. Así tenemos un campo gradiente que es conservativo y por lo tanto tenemos que la fuerza de tensión en \vec{r} es:

$$F(\vec{r}) = -\hat{L} \cdot \nabla_{\vec{r}} \Sigma(\vec{r}) = -n\epsilon(\theta)\hat{L} \quad (1.5)$$

Donde \hat{L} es el vector unitario en la dirección dL . Así tenemos que la fuerza depende de la curvatura local $\theta(\vec{r})$. Entonces la función de *utilidad* convertida a un modelo continuo equivale a la *fuerza de tensión de un cúmulo*. Esto nos permite predecir el comportamiento del cúmulo.

1.3.3. Formación de cúmulos en el modelo de Vinkovic-Kirman

En el modelo Schelling la función de *utilidad* se expresa como la siguiente función por pasos:

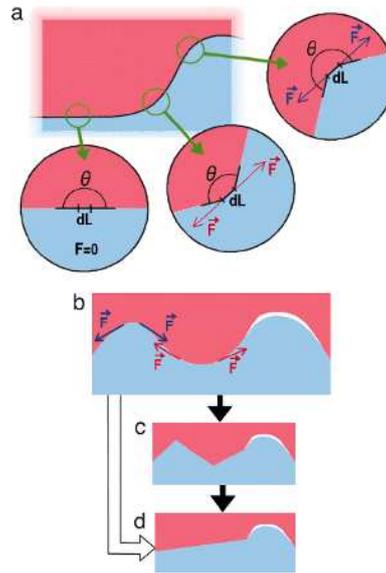


Figura 1.9: Evolución de los *cúmulos* bajo la fuerza de tensión superficial. Figura tomada de [5]

$$utilidad = \begin{cases} 1 & \text{Si el número de vecinos de distinto tipo } \leq x \\ 0 & \text{Si el número de vecinos de distinto tipo } > x \end{cases} \quad (1.6)$$

Como mencionamos arriba en el modelo de Vinkovic-Kirman la función de *utilidad* se traduce en medir la energía midiendo el ángulo θ que ocupan las partículas de distinto tipo alrededor de una partícula determinada, así tendríamos una expresión para la energía

$$\epsilon = \begin{cases} \epsilon_0 & \theta > 180^\circ \\ 0 & \theta \leq 180^\circ \end{cases} \quad (1.7)$$

De la ecuación 1.5 que expresa la fuerza F se concluye que para cúmulos convexos se tiene que $F = n\epsilon_0$ y esto indica que la superficie de los cúmulos tiende a aplanarse debido a la tensión superficial. (Ver figura 1.9).

Capítulo 2

El algoritmo de Barnes-Hut

Las simulaciones computacionales de fenómenos físicos que involucran un sistema de partículas, donde cada partícula interactúa con todas las demás de acuerdo a las leyes de la física, aparecen en astrofísica y en mecánica celeste (con la fuerza de gravedad), en la simulación de plasma y en la dinámica de fluidos (con la fuerza de atracción o repulsión electrostática) o en la dinámica molecular (con fuerzas de Coulomb) [27].

Las primeras cuestiones que saltan a la vista en las simulaciones computacionales son los recursos físicos que utilizaremos, que tiene que ver con el número de operaciones que haremos y su eficiencia. La representación de *N-cuerpos* en una computadora y sus interacciones implican operaciones computacionales, y espacio en la memoria de la computadora, que pueden ser una gran cantidad de recursos si N , el número de partículas, es del orden de miles o millones. A este problema de calcular eficientemente las interacciones de un sistema de N cuerpos o partículas se le conoce como el problema de los *N-cuerpos*.

2.1. El problema de los *N-cuerpos*

Matemáticamente, en su expresión más simple el problema de los *N-cuerpos* puede ser formulado como:

$$U(x_i) = \sum_{j=1}^N \Phi(x_i, x_j) \quad (2.1)$$

donde $U(x_i)$ es la suma de los valores del potencial en función de la posición de la partícula i , x_i y cada una de las posiciones de las otras partículas, estos valores representan fuerzas de largo alcance que no pueden ser despreciadas en la simulación.

Esencialmente están involucradas las interacciones entre pares de partículas, lo cual nos dice que en una computadora tenemos que ejecutar a lo más

tantas operaciones como pares de partículas. Por ejemplo, si representáramos las interacciones de cada par de partículas en una computadora con la matriz \mathbf{U} :

$$\mathbf{U} = \begin{pmatrix} \Phi(x_1, x_1) & \Phi(x_2, x_1) & \dots & \Phi(x_n, x_1) \\ \Phi(x_1, x_2) & \Phi(x_2, x_2) & \dots & \Phi(x_n, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(x_1, x_n) & \Phi(x_2, x_n) & \dots & \Phi(x_n, x_n) \end{pmatrix}$$

la suma anterior equivale a sumar los elementos de su columna i -ésima, y puesto que tenemos N partículas tenemos que tener una suma de estas por cada partícula.

La matriz \mathbf{U} , que representa los pares de un sistema de partículas, es una matriz de $N \times N$ es decir de N^2 elementos. En el análisis de algoritmos, se define el orden de complejidad de un algoritmo como una función discreta que depende del número de elementos que manipula el algoritmo y que regresa el número de operaciones que se ejecutan, la función es una forma de evaluar la eficiencia del algoritmo de acuerdo al número de operaciones realizadas, independientemente de la plataforma o equipo en donde este se ejecute [26]. Así, podemos decir que el algoritmo “simple” para calcular las interacciones entre las partículas es del orden de N^2 , o que tiene una complejidad de $O(N^2)$, esto puede representar una gran cantidad de recursos si consideramos que los problemas a resolver generalmente se modelan con cientos, miles o millones de partículas.

Cuando se ejecutan las simulaciones usando el algoritmo “simple” se tienen dos opciones; la primera, es considerar las N^2 interacciones, haciendo que el tiempo de ejecución sea tan largo que prácticamente sería imposible ver la evolución del sistema para intervalos de tiempo razonables; la segunda, es sólo considerar para cada partícula las interacciones de otras partículas vecinas que estén a su alrededor en un radio de distancia determinado, sin considerar sus interacciones con partículas lejanas y despreciando las fuerzas de largo alcance, lo cuál deriva en obtener grandes imprecisiones en los cálculos para este tipo de problemas, donde las interacciones de largo alcance son importantes.

Para poder considerar de alguna manera todas las interacciones de largo alcance sin afectar el desempeño de la simulación se han creado algoritmos que reducen el tiempo de ejecución a un orden de complejidad de $O(N \log N)$, dichos algoritmos utilizan un árbol como estructura de datos para representar al espacio de la simulación y sus subdivisiones en donde las partículas del sistema están distribuidas, a esto se debe que les hayan llamado *Treecodes*, entre ellos está el algoritmo de Barnes-Hut [27].

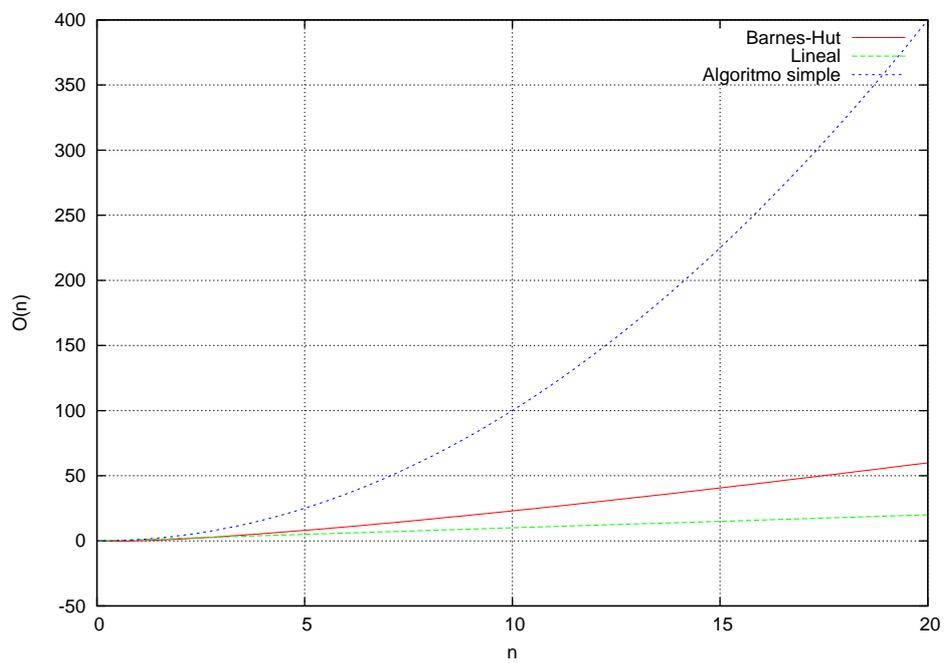


Figura 2.1: Comparación de los órdenes de complejidad de los algoritmos “simple” $O(n^2)$, el de Barnes-Hut $O(n \log n)$ y un algoritmo ideal de orden lineal, $O(n)$. En la gráfica podemos apreciar que $n \log n$ es bastante menor que n^2 cuando n empieza a crecer.

2.2. Reducción del orden de complejidad del algoritmo “simple”

La idea principal del algoritmo de Barnes-Hut basada en la experiencia en cálculos astronómicos para determinar la fuerza gravitacional, consiste en subdividir el espacio en donde se encuentran distribuidas las partículas y para cada una de estas subdivisiones calcular la fuerza recursivamente. Cada partícula quedará en alguna subdivisión.

Si la fuerza sobre la partícula p ejercida por la partícula q esta definida por la función $f(p, q)$, entonces, para calcular las interacciones de una partícula p con todas las demás, necesitamos calcular la fuerza de los pares que se forman con ésta partícula p y todas las demás partículas q .

El criterio para calcular estas interacciones es el siguiente: para cada partícula p , si la partícula q es cercana a p , se calcula directamente $f(p, q)$. Si no, se elige alguna de las subdivisiones B , tal que q este dentro de la subdivisión, y luego se consideran a todas las partículas que estan dentro de B en conjunto con q como una sólo partícula x . Por último calculamos $f(p, x)$.

La partícula x será un sólo cuerpo, representante del conjunto de partículas en la subdivisión B , estará ubicado en el centro de masa de todas las partículas dentro de la subdivisión y su masa será igual a la suma de las masas de las partículas en la subdivisión. A estos conjuntos de partículas les llamaremos *cúmulos*.

En conclusión, para partículas cercanas calculamos las interacciones de partícula a partícula y para partículas lejanas, hacemos una aproximación calculando las interacciones de partícula a cúmulo. Cabe mencionar que esto sólo se hará en ciertos casos, cuando haya cúmulos de partículas suficientemente “lejos”, es decir que su distancia a la partícula sea mayor que un cierto número.

Más formalmente, los casos en donde se aplicará este criterio para calcular la fuerza sobre la partícula p , son los casos en los que $D/r < \delta$, donde r es la distancia de la partícula p al centro de masa x del conjunto de partículas que se encuentran dentro la caja B de lado D y donde δ es un número menor que la unidad, asegurando que la caja B se encuentra a una distancia considerable. Así en vez de calcular la fuerza para cada una de las partículas dentro de esta caja, se consideran todas en su conjunto como una sólo partícula. (Ver figura 2.2).

La forma de subdividir el espacio en el algoritmo de Barnes-Hut esta diseñada para que dada una partícula cualquiera, los cúmulos de partículas lejanos a ésta queden contenidos dentro de alguna de las subdivisiones. Para representar esta configuración se utiliza como estructura de datos un árbol. En el caso bidimensional estas subdivisiones son cuadrados, y en el caso de tres dimensiones el espacio se subdivide en cubos.

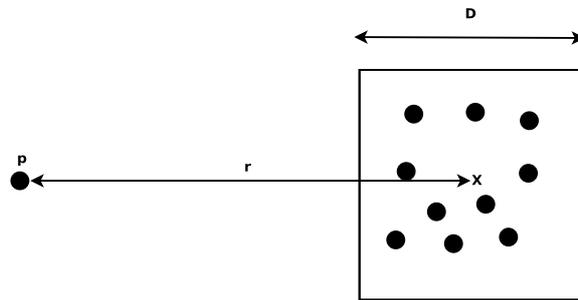


Figura 2.2: Las partículas dentro de la caja de lado D se consideran como una sola partícula ubicada en el centro de masa x

De esta forma, se disminuye el número de operaciones en el algoritmo a costa de perder exactitud en el cálculo de la fuerza, pero en la mayoría de los casos el algoritmo de Barnes-Hut nos da una buena aproximación y siempre es posible estimar el error debido a su estructura jerárquica, esto le da ventaja sobre otros algoritmos que reducen el número de operaciones pero no se puede determinar el error en la aproximación [27].

2.3. Implementación en el plano

2.3.1. Árbol cuaternario adaptativo

Los árboles cuaternarios son utilizados para resolver problemas de búsqueda de objetos situados en un plano o de regiones que contienen un conjunto de objetos. En el caso del algoritmo de Barnes-Hut la meta es identificar las regiones que contienen cúmulos de partículas y para el caso de dos dimensiones el algoritmo es implementado con árboles cuaternarios adaptativos [8].

Definición 1. *El árbol cuaternario adaptativo es un árbol en donde cada nodo tiene a lo más cuatro nodos hijos.*

En el caso bidimensional el árbol cuaternario adaptativo representa al plano y sus subdivisiones en cuatro cuadrantes, los representantes de estos son el nodo raíz y sus nodos hijos de ahí el nombre en inglés de *QuadTree*.

El algoritmo de Barnes-Hut nos indica mediante esta estructura de datos en cuál subdivisión se encuentra cada partícula, así las subdivisiones que representa el árbol reflejan la distribución de las partículas. (Ver figura 2.3).

El nodo raíz representa el plano que contiene a las partículas del sistema, restringiéndonos a un cuadrado suficientemente grande que pueda contener a todas, así podemos dividir este cuadrado en cuatro cuadrados cada uno con la mitad del perímetro y la cuarta parte del área del original. Estos nuevos cuadrados son representados como cuatro hijos del nodo raíz.

Así cada nodo en el árbol representa un cuadrado, al subdividir cada cuadrado en cuatro en el árbol se desprenden cuatro hijos del nodo representante del cuadrado.

Se hacen tantas subdivisiones como sea necesario, hasta que cada partícula quede en una sólo subdivisión, todas las subdivisiones quedan reflejadas en la estructura del árbol.

Las hojas del árbol deben de contener o tienen apuntadores a la posición y las masas de cada partícula que representan, en el caso de los nodos que no son hojas, representan alguna de las subdivisiones y cada uno de estos guarda información necesaria como la posición del centro de masa de las partículas contenidas en la subdivisión y la suma de las masas de las partículas que estan dentro.

Los árboles adaptativos no tienen hojas vacías, así que las subdivisiones donde no haya partículas serán descartadas.

Ya que hemos definido la estructura de datos que usaremos enunciaremos los pasos del algoritmo de Barnes-Hut y posteriormente explicaremos detalladamente cada uno de ellos.

En el caso 2-D, para cada intervalo de tiempo el algoritmo consiste en los siguientes pasos:

1. Construcción del árbol cuaternario adaptativo T , cuya raíz es el nodo R
2. Recorrido sobre el árbol T , calculando las masas y centros de masas para cada nodo
3. Para cada partícula p calcular la fuerza ejercida por todas las demás partículas sobre ésta, usando la función $TreeForce(p, R)$ que recorre el árbol T desde su raíz

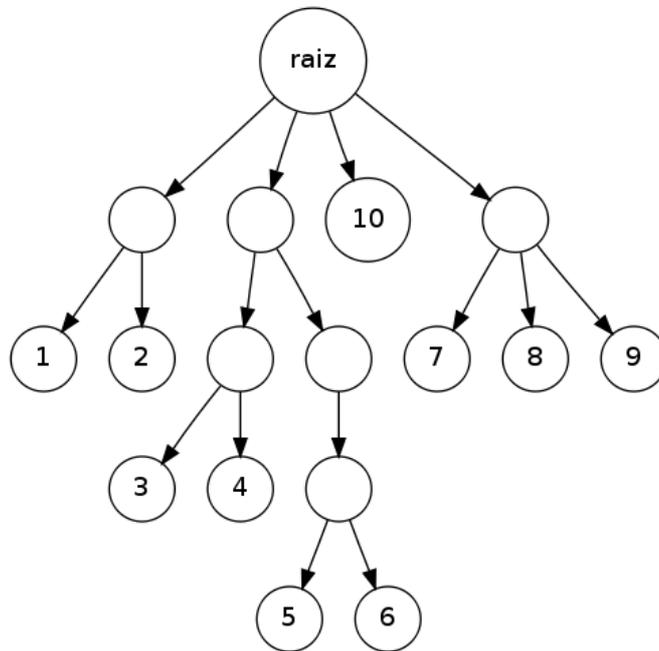
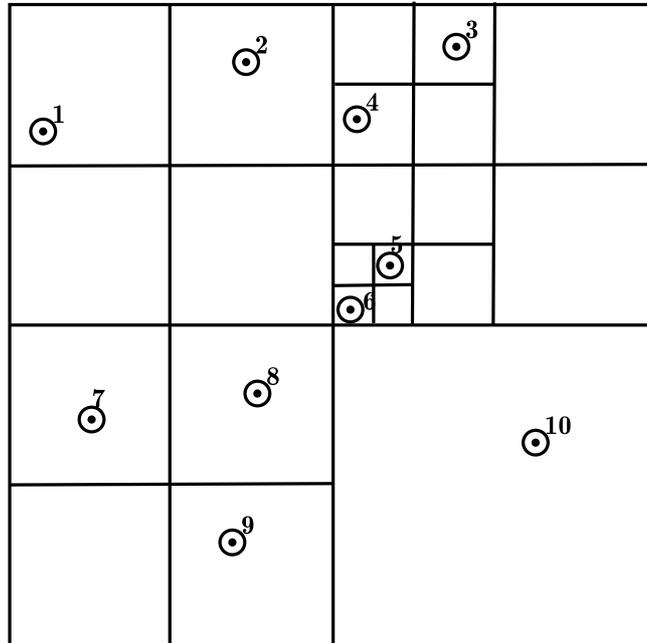


Figura 2.3: Árbol cuaternario adaptativo para una configuración de diez partículas, como se puede apreciar en la figura, el espacio es dividido en cuatro partes iguales recursivamente hasta dejar sólo una partícula en cada subdivisión. Las hojas corresponden al número de partículas del sistema, los nodos están ordenados en el sentido de las manecillas del reloj empezando desde la esquina superior izquierda.

2.3.2. Construcción del árbol cuaternario

El árbol cuaternario adaptativo se construye a partir de las partículas del sistema y su distribución en el espacio, se usa para calcular la fuerza. El algoritmo de Barnes-Hut usa La función *construye_quadtree* para construir el árbol. (Ver el algoritmo 2.1).

Algoritmo 2.1 Función para construir el árbol cuaternario

void *construye_quadtree*

```
1: QuadTreetree  $\leftarrow$  QuadTree.nuevo {Se crea un árbol vacío}
2: for  $i = 0$  to num_particulas do
3:   quadtree.inserta(particulas[ $i$ ], tree.raiz)
4: end for
5: return construye_adaptive_quadtree(tree) {Se eliminan las hojas vacías
del árbol cuaternario}
```

En nuestro caso usamos BFS (*Breadth-first search*) [26] para recorrer los nodos y eliminar las hojas que no contienen ninguna partícula y construir el árbol cuaternario adaptativo.

La función *quadtree.inserta*(p, n) es la función que decide qué nodo o cuadrado, representante de alguna subdivisión, le corresponde a la partícula p de acuerdo a sus coordenadas. Por construcción cada hoja del árbol terminará con una o cero partículas. El algoritmo empieza desde el nodo n , si n es un nodo vacío, aloja en éste la partícula p , si no, busca su lugar recursivamente en alguno de los hijos de n , si n , no tiene hijos, los crea y determina en cuál de los hijos de n quedará alojada la partícula, de nuevo, de acuerdo a las coordenadas de la partícula.

Algoritmo 2.2 Función para insertar la partícula p en el nodo n

void *quadtree.inserta*(p, n)

```
1: if esHoja( $n$ ) then
2:   if esVacio( $n$ ) then
3:      $n.data \leftarrow p$ 
4:   else
5:      $hijos \leftarrow crea\_hijos(n)$  {Crea los hijos de  $n$ }
6:      $pos \leftarrow busca\_posicion(n, hijos)$ 
7:     quadtree.inserta( $p, hijos[pos]$ )
8:   end if
9: else
10:   $pos = busca\_posicion(p, n.hijos)$  {El nodo no es una hoja}
11:  quadtree.inserta( $p, hijos[pos]$ )
12: end if
```

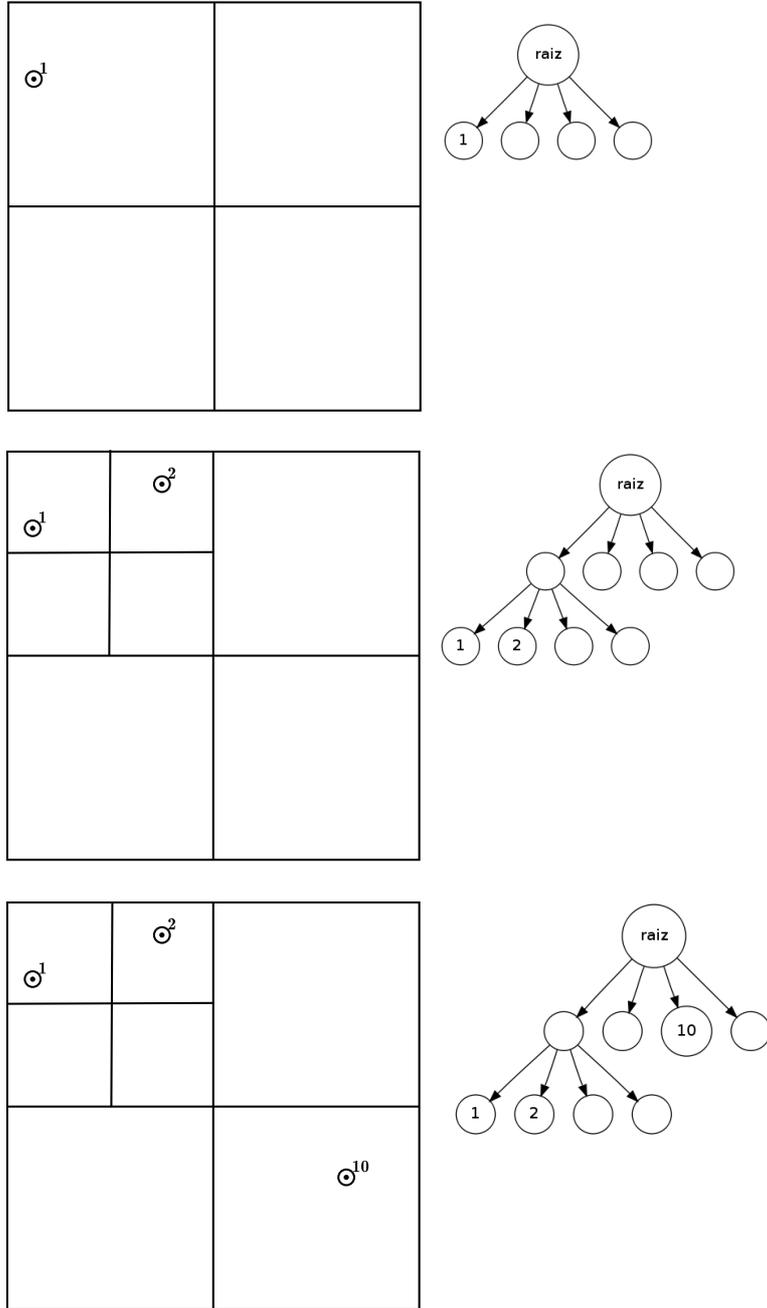


Figura 2.4: Primeras tres etapas de la construcción del árbol cuaternario.

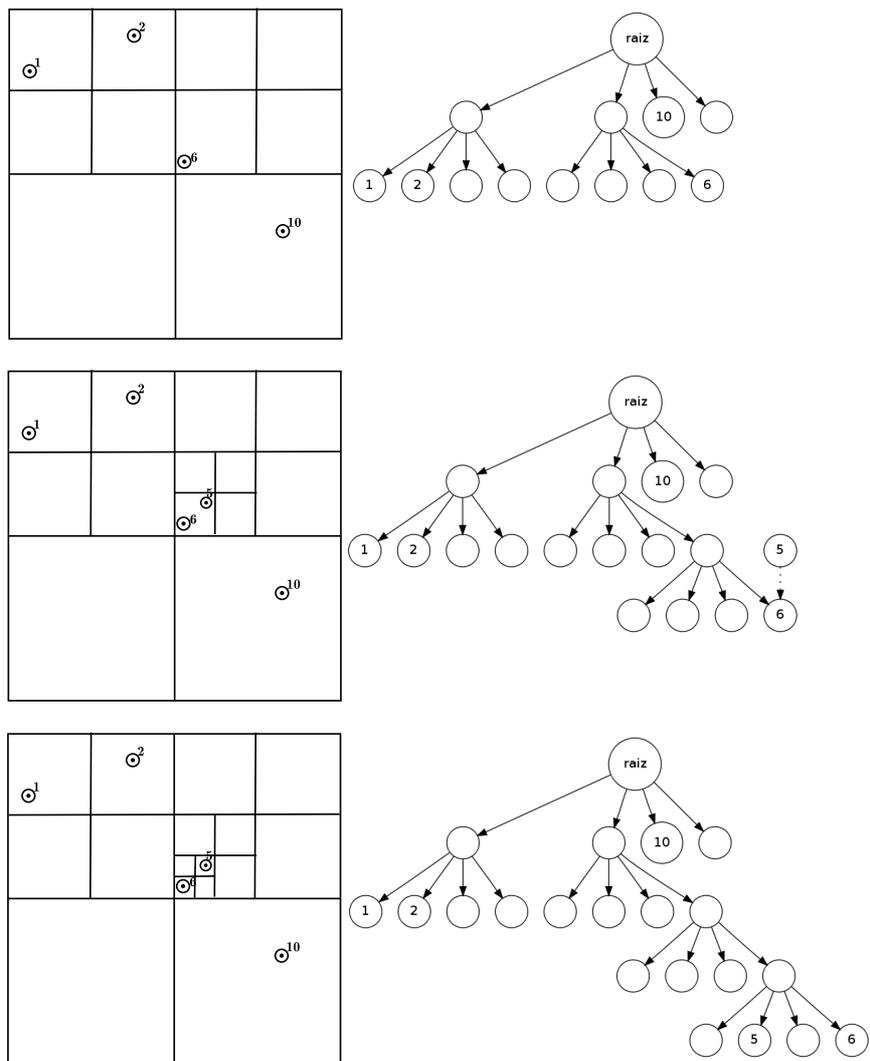


Figura 2.5: Etapa 4 y 5 de la construcción del árbol cuaternario. En la última etapa la partícula 5 cae en la misma subdivisión que 6 generando una nueva subdivisión.

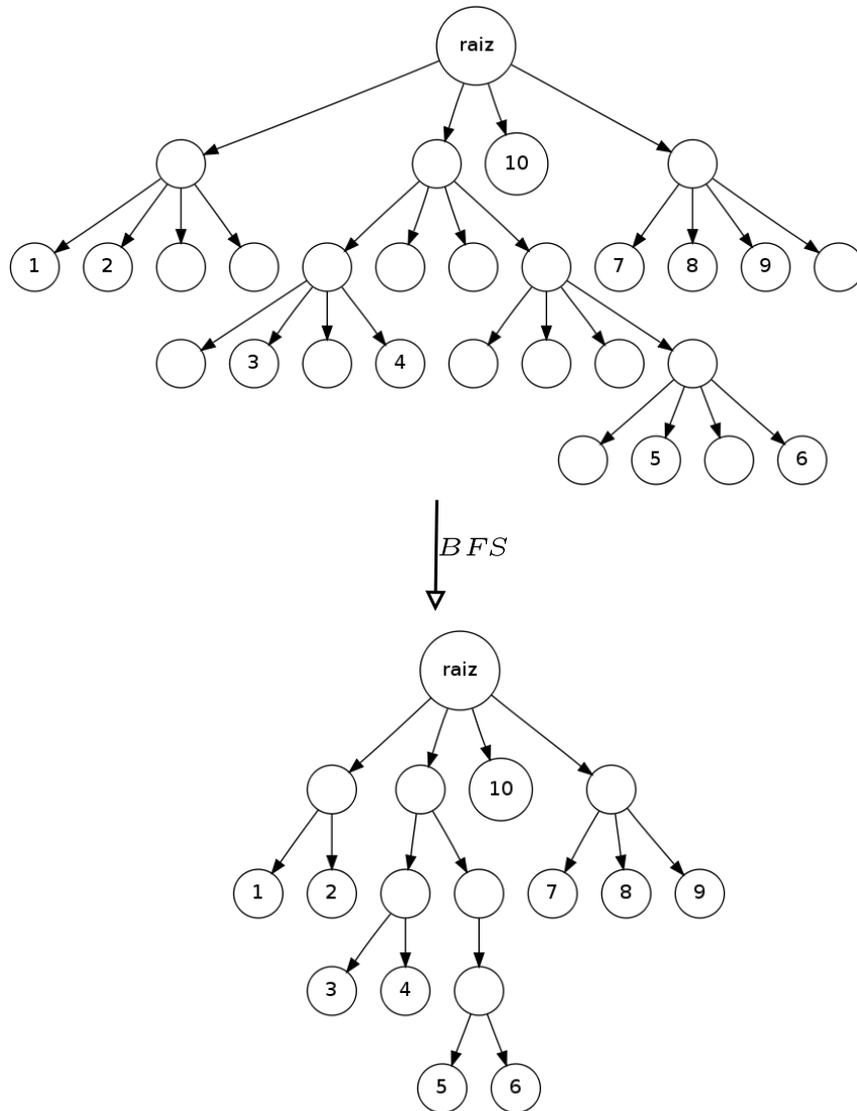


Figura 2.6: Después de agregar todas las partículas, ejecutamos *BFS* sobre el árbol cuaternario, eliminando las hojas vacías, este paso es del orden del número de nodos del árbol más el número de aristas, i.e $O(\# \text{ de nodos} + \# \text{ de nodos}-1) = O(N)$

2.3.3. Cálculo de la fuerza

Ya que tenemos la estructura de datos para cada intervalo de tiempo necesitamos ahora sí calcular la fuerza ejercida en una partícula p por todas las demás partículas en el sistema, aplicando el criterio de la razón D/r mencionado anteriormente y recorriendo el árbol cuaternario desde su raíz.

El criterio consiste en calcular la distancia r de la partícula p al nodo n cuando este no es una hoja del árbol. El nodo n es un representante de un cúmulo de partículas, se calcula D la longitud del lado de la subdivisión donde se encuentra contenido este nodo, la posición del nodo n esta en el centro de masa del cúmulo que representa, se compara la razón D/r con un número positivo δ menor que la unidad, entonces se tiene que:

- Si $D/r < \delta$ se calcula directamente la fuerza donde $0 < \delta < 1$
- en otro caso se calcula recursivamente la fuerza para cada hijo del nodo representante de p .

Cuando el nodo es una hoja entonces representa una partícula y se calcula la fuerza directamente.

Algoritmo 2.3 Función para calcular la fuerza sobre la partícula p

float *TreeForce*(p, n)

```
1: if esHoja( $n$ ) then
2:    $F \leftarrow f(p, n)$  {donde  $f$  es la función de la fuerza}
3: else
4:    $r \leftarrow d(p, n)$  { $d$  es la distancia de  $p$  a  $n$ }
5:    $D \leftarrow longitud.lado.subdivision(n)$ 
6:   if  $D/r < \delta$  then
7:      $F \leftarrow f(p, n)$ 
8:   else
9:      $hijos \leftarrow hijos(n)$ 
10:    for  $i = 0$  to  $hijos.size$  do
11:       $F \leftarrow F + TreeForce(i)$  {para cada hijo  $i$  del nodo  $n$ }
12:    end for
13:  end if
14: end if
```

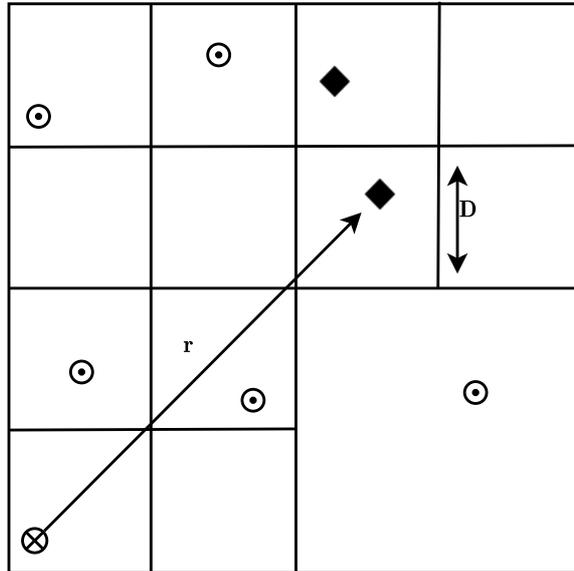


Figura 2.7: Las partículas dentro de la caja de lado D se consideran como una sola partícula ubicada en el centro de masa representado con un rombo, para las demás partículas se calcula directamente la fuerza. Aplicando el criterio D/r con $\delta = 0,5$

2.3.4. La complejidad del algoritmo de Barnes-Hut

Ahora analizaremos la complejidad del algoritmo en el caso de tres dimensiones ya que la versión original del algoritmo de Barnes-Hut esta pensada en el espacio y la demostración en el plano es análoga a ésta. Comenzaremos con la complejidad de la construcción del árbol octanario, para cada tiempo t se construye el árbol octanario que determina su estructura jerárquica de acuerdo a las posiciones de las partículas en el tiempo t .

Análogamente que en el caso de dos dimensiones, trazamos un cubo suficientemente grande para contener a todo el sistema, después se agrega partícula por partícula a esta celda raiz, si dos partículas caen en la misma celda se divide la celda en ocho nuevas celdas del mismo tamaño, es claro que al agregar la segunda partícula al sistema se realiza la primer subdivisión en ocho subceldas cúbicas.

De igual forma, cada celda es representada por un nodo del árbol, una estructura que registra la información acerca de las subceldas que estan contenidas en ella, parte de esta información es la suma de las masas y las coordenadas del centro de masa de las subceldas, también tiene un apuntador a los nodos que representan cada subcelda. Continuamos este proceso de subdividir el espacio hasta la profundidad que se requiera. Cuando todas las N partículas se han agregado el espacio se ha subdividido en celdas cúbicas de diferente tamaño con a lo más una partícula por celda. Cada celda que

contiene a una partícula puede ser agrupada con otras celdas en celdas más grandes que las contienen y estas en otras más grandes hasta llegar a la celda raíz que contiene a todo el sistema. El tamaño promedio de cada celda es aproximadamente del orden del espacio entre las partículas, entonces la altura del árbol, es del orden del número de subdivisiones requeridas para alcanzar a una celda típica, empezando desde la raíz, es de $O(\log_2 N/3) = O(\log N)$, y el tiempo requerido para construir el árbol es de $O(N \log N)$.

En el caso promedio, cuando la distribución de las partículas es uniforme, las partículas se ubican en las hojas del árbol quedando a la misma altura, tenemos que el número de subdivisiones es del orden de la altura del árbol $O(N \log N)$ pero en el peor de los casos, cuando todas las partículas se acumulan en un sólo lugar, el número de subdivisiones es del orden $O(Nb)$, donde b es el número de bits para representar las posiciones de las partículas.

Demostración. Para cada subdivisión en el tiempo t se tiene que el número de hojas en el árbol es:

$$n(0) = (2^3)^0$$

$$n(1) = (2^3)^1$$

$$n(2) = (2^3)^2$$

$$n(3) = (2^3)^3$$

...

$$n(t_i) = (2^3)^{t_i}$$

Supongamos que en el tiempo t_i se tiene que

$$n(t_i) = (2^3)^{t_i} = N$$

$$n(t_i) = 2^{3t_i} = N$$

$$3t_i = \log_2 N \Leftrightarrow t_i = \log_2 N/3 \Leftrightarrow t_i = \log_2 N^{1/3}$$

entonces podemos decir que la altura del árbol es t_i y que esta es $O(\log N)$, y como para cada partícula tenemos que recorrer la misma altura entonces esta operación es $O(N \log N)$. \square

El último paso en la construcción del árbol es anotar en las celdas subdivididas la masas y el centro de masa de las partículas que contienen, esto se hace propagando la información desde las hojas/partículas a la raíz del árbol, esto se puede lograr en el tiempo de $O(N \log N)$. En nuestro

caso utilizamos el algoritmo *pos-orden* para recorrer el árbol, este algoritmo es del orden del número de nodos más el número de aristas que es $O(\# \text{ de nodos} + \# \text{ de nodos}-1) = O(N)$.

Para calcular la fuerza ejercida sobre una partícula p , con la función *Tree-Force* descrita anteriormente el número de interacciones para este proceso $O(\log N)$ para N grandes.

Demostración. Supongamos que las partículas están uniformemente distribuidas en la celda raíz, es decir en el espacio inicial, ahora supongamos que aumentando el número de partículas ocho veces el número inicial, sería como equivalente a juntar ocho celdas raíz.

Las siete celdas que no contienen la partícula p contribuyen en un número pequeño de términos para el cálculo de la aproximación de la fuerza, digamos Δt .

Entonces el número de partículas se va incrementando exponencialmente, es decir, $8^t N_0$ entonces cada que se incrementa N , el número de operaciones se incrementa en un número constante Δt , lo cuál quiere decir que el número de operaciones es $O(\log N)$

$$N = 8^t N_0$$

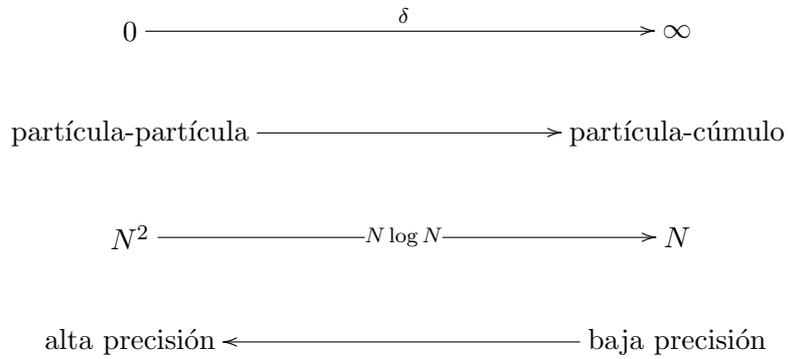
$$N = \exp(t \ln 8) N_0$$

$$\ln N = t \ln 8 + \ln N_0$$

$$t = (\ln N - \ln N_0) / \ln 8$$

entonces se tiene que t es $O(\log N)$ y como para cada t se tiene un número constante de operaciones adicionales Δt , entonces el número de operaciones para esta función es aproximadamente de $O(\log N)(\Delta t)$ esto es $O(\log N)$. \square

El tiempo de ejecución no sólo depende del número de partículas involucradas, también depende fuertemente del parámetro $\delta = D/r$. En el caso en que $\delta = 0$ es equivalente a calcular todas las interacciones de partícula a partícula donde obtenemos máxima precisión pero no es muy eficiente ya que es de $O(N^2)$. De hecho es más ineficiente que el algoritmo “simple” ya que tenemos que construir el árbol y recorrerlo para calcular la fuerza. En el otro extremo, si hacemos que $\delta \rightarrow \infty$, puede producir una muy baja precisión ya que el espacio casi no estará subdividido. Este conflicto para determinar el valor de δ puede resumirse como:



Lo mejor siempre es obtener un algoritmo rápido y eficiente, con buena precisión. Eligiendo δ entre 0,1 y 1,0 se han obtenido buenos resultados en varios casos. Afortunadamente también existe una forma de mejorar la precisión del algoritmo sin la necesidad de una resolución mayor (más subdivisiones, δ más pequeño), la cuál incluye los momentos multipolos de los cúmulos de partículas en vez de tratarlos como un punto de las masas (o cargas) [27].

Capítulo 3

Modelos basados en agentes

Los modelos basados en agentes (MBA) se presentan como una alternativa a las ecuaciones diferenciales ordinarias (EDO), son modelos estocásticos usados para describir poblaciones de agentes que interactúan entre sí, como personas, insectos, o células, usando reglas simples que dictan el comportamiento de los agentes, pueden llegar a describir sistemas complejos. Originalmente estos modelos fueron introducidos por John von Neumann y Stanislaw Ulam bajo el nombre de *espacios celulares* como una posible idealización de los sistemas biológicos. Buscaban demostrar que los procesos biológicos tales como la reproducción y evolución de formas organizadas podían ser modeladas por reglas locales aplicadas a células simples. Una de las virtudes de los MBA es su capacidad para generar comportamientos emergentes, inesperadamente complejos, a partir de reglas muy simples, estos comportamientos pueden ser periódicos o intrincados patrones espaciales y temporales. Los agentes responden dinámicamente y de manera independiente a los ambientes cambiantes. Por lo tanto, aspectos como dinámicas no lineales, o un control en el tiempo pueden ser modelados en la práctica incorporándose en las reglas de los agentes o pueden emerger espontáneamente de la misma dinámica colectiva del sistema. Además otra ventaja de los MBA es que son paralelizables ya que su estructura computacional es inherentemente paralela, así se pueden implementar en computadoras paralelas de una forma muy eficiente.[4]

La capacidad de procesamiento de las computadoras se ha incrementado considerablemente en la última década, haciendo posible el uso de los MBA para los sistemas complejos.

La implementación de los MBA partiendo de las reglas que gobiernan al agente tiende a imitar el comportamiento de los agentes en el sistema real, la descripción se convierte más física que matemática. Por ejemplo los modelos que usan ecuaciones diferenciales consideran tasas de natalidad, tasas de mortalidad, tasas de adherencia, tasas de difusión, mientras los MBA consideran sólo las reglas que determinan las acciones de los agentes.

El lenguaje simple de los MBA hace que estos sean accesibles no sólo a los programadores sino también a científicos de otras disciplinas, que sin ser expertos en programación, pueden involucrarse en el proceso de la creación del modelo.

Los MBA son inductivos, es decir el comportamiento emergente resultante de los agentes es la unidad básica del análisis. Estos agentes responden a interacciones con sus vecinos y con el medio siguiendo reglas simples y finalmente llegando mediante la exploración a definir comportamientos globales.

Por el contrario las EDO son deductivas, pueden describir propiedades globales del sistema a estudiar y que son previamente conocidas, pero no pueden explicar interacciones locales o propiedades heterogéneas, en donde tal vez no existan tasas constantes para construir el modelo.

Aunque también se pueden usar autómatas celulares para simular este tipo de fenómenos encontramos que los MBA son de mayor utilidad para modelar el movimiento de las células o partículas. Los AC (autómatas celulares) son casi siempre homogéneos, uniformemente poblados en el espacio de la malla, mientras que los MBA son heterogéneos y no necesariamente ocupan todos los espacios de la malla. En los AC cada individuo está representado por una celda de la malla, en los MBA puede que un agente esté representado por un conjunto de individuos, es decir ocupando varias celdas en la malla, lo que nos permite modelar a estos conjuntos como partes del sistema y estudiar las interacciones entre ellos. En los AC, los agentes frecuentemente no interactúan con agentes fuera de alguna vecindad determinada, en los MBA además de interactuar con sus vecinos más próximos se pueden representar interacciones largo alcance.

A continuación daremos una definición no formal de los MBA, a partir del concepto de agente y de su arquitectura básica.

3.1. El concepto de agente

Aunque no existe una definición consensuada para el término agente podemos decir que un agente es un sistema autónomo sensible a su entorno, tomando decisiones y ejecutando acciones de manera autónoma. Una de las definiciones de agente más general y más citadas, proviene de la inteligencia artificial y fué establecida por Russell y Norvig [12]:

Definición 2. *Un agente es cualquier cosa capaz de percibir su entorno a través de sensores y actuar en ese medio utilizando accionadores.*

Por ejemplo,

- Un agente humano tiene ojos, oídos y otros órganos como sensores y manos, piernas, boca y otras partes del cuerpo como accionadores.

- Un robot tiene como sensores cámaras, infrarojos, termómetros etc, y motores, o brazos mecánicos como accionadores.
- Un agente de software tiene como entradas contenidos de texto, o paquetes de datos que pasan por algún canal de comunicación, por ejemplo la internet, usando algún protocolo de comunicación como TCP/IP, estos datos son procesados por el software generando una respuesta, o sea otros datos, que vuelven a pasar por algún canal de comunicación.

En la célula, la membrana celular contiene proteínas receptoras que le permiten detectar señales moleculares como las hormonas o detectar nutrientes, siendo esta su interfaz con el medio, responde a estos estímulos con procesos metabólicos efectuados en su interior. Por ejemplo un bacteria como la *Escherichia coli*, es sensible a las concentraciones químicas del medio, que transforma en señales moleculares intracelulares a través de sus organelos, automáticamente se echa a andar una maquinaria interna que hace que sus flagelos puedan girar e impulsar a la bacteria de un lugar a otro.

Así que un agente puede ser un humano, un programa de software, un robot, una célula o incluso una molécula.

En el modelo más simple posible un agente se compone de:

Percepciones Es la información que percibe de su medio ambiente a través de los sensores.

Procesamientos Es el proceso de las percepciones, donde radica la “inteligencia” del agente. En un agente simple, es aquí donde se definen las reglas del comportamiento del agente.

Acciones Se ejecutan ciertas acciones dependiendo de los procesos que se llevaron a cabo. Estas pueden ser respuestas simples a los estímulos del medio o acciones bien planificadas.

También podríamos decir que las acciones van encaminadas a cumplir ciertas metas y que el desempeño del agente se mide en base a que tanto se cumplieron dichas metas o que tanto se adapta al medio ambiente.

Podríamos decir matemáticamente hablando que el agente tiene una función que para cada secuencia de percepciones le asigna una acción, para luego ejecutar esta acción. Esto es lo que se refiere a los procesamientos de la información en el agente. Las secuencias de percepciones tienen que ser finitas y como es natural no para todas las secuencias posibles de percepciones necesariamente se debe realizar una acción, las secuencias pueden ser muchísimas, así que para muchas simplemente el agente responde no haciendo nada y sólo responde a las percepciones que le sean de utilidad para alcanzar alguna de sus metas.

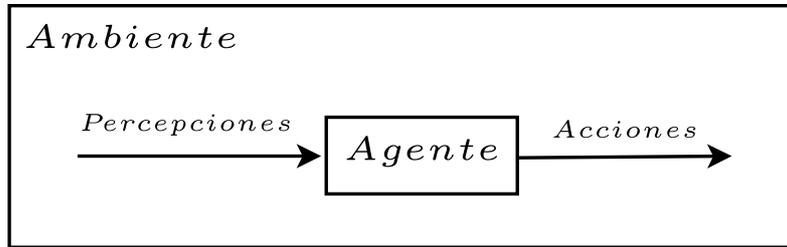


Figura 3.1: Modelo abstracto de un agente.

Los llamados *agentes inteligentes* tienen la capacidad de aprender y explorar las condiciones de su entorno y modificar su comportamiento de manera autónoma para adaptarse al ambiente, estos agentes pueden ser muy sofisticados, con componentes para llevar a cabo inferencias y planear tareas para luego ejecutarlas y evaluar si sus metas fueron completadas satisfactoriamente.

El grado de autonomía y de adaptabilidad o inteligencia puede variar mucho de una aplicación a otra, para nuestro caso usaremos las propiedades básicas de los agentes, con un grado mínimo de autonomía y más que nada nos enfocamos a la emergencia de patrones a nivel global, emergencia que se desprende de interacciones locales entre los agentes del sistema e interacciones con el medio a través de interacciones de largo alcance.

Un modelo basado en agentes es el comportamiento colectivo de un conjunto de agentes en un medio ambiente determinado, se determinan las propiedades del medio y del agente. En este modelo, se definen los estados de cada agente, para ello, se registra en su estado las reglas de comportamiento del ambiente, que son independientes a las acciones del agente, y por otro lado se registran las interacciones del agente con el ambiente, los otros agentes y si es posible el resultado de estas acciones sobre el sistema.

Las reglas del modelo definen las acciones que se tienen que tomar de acuerdo al estado actual del sistema o a su historia, es decir los MBA pueden tener estado o memoria. Así cada uno de los agentes interactúa con los otros agentes y con su entorno siguiendo las reglas especificadas en el modelo.

3.2. Los MBA en los sistemas complejos

Como hemos mencionado las interacciones locales de los agentes, que pueden ser de origen estocástico, producen los patrones y la dinámica global del sistema. Entonces los MBA son particularmente útiles para estudiar sistemas complejos en los cuáles la heterogeneidad de los individuos y las in-

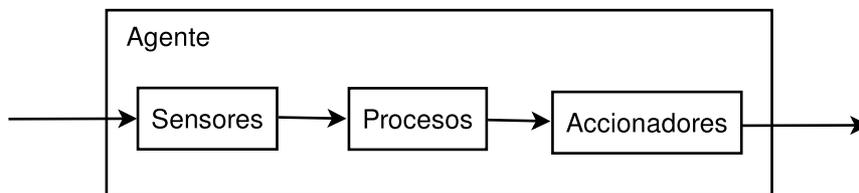


Figura 3.2: Estructura básica de un agente.

teracciones espaciales son importantes. Los modelos basados en agentes son una extensión natural de los autómatas celulares, éstos últimos ampliamente utilizados para simular sistemas biológicos.

Actualmente los MBA han sido utilizados más extensamente en el campo de la inteligencia artificial [16], donde se estudian los agentes de software, desde una perspectiva de la ingeniería de software, los agentes son partes de software autónomas de un sistema distribuido, pero que resuelven algún problema de manera colectiva. Los sistemas basados en agentes tienen como objetivo resolver problemas complejos del mundo real, con cada uno de sus agentes ejecutando tareas por separado para lograr sus metas y así de manera modular, estos sistemas resultan ser robustos, escalables y capaces de adaptarse a los cambios. Se han especificado métodos formales y arquitecturas para este tipo de sistemas [24].

En las ciencias sociales, los individuos son modelados como agentes, siendo actores de algún fenómeno social, se determinan sus propiedades y las interacciones que tienen con otros agentes, y se estudia el fenómeno colectivo que surge de estas interacciones, podemos citar algunos ejemplos, en economía [17], en ciencias políticas [18] y en la sociología [19].

Dentro de la biología podemos mencionar a la ecología [22], [23], en donde los individuos de ciertas especies se representan como agentes, estos agentes pueden ser representados como en los algoritmos genéticos, como una secuencia codificada de material genético, para después utilizar algunas técnicas de los algoritmos genéticos, como mutación y cruza. Se definen las interacciones entre especies e individuos, estas pueden ser relaciones de cazador-presa, de apareamiento o cooperación, y también se puede definir la competencia por los recursos del medio entre individuos de la misma especie, se han descrito redes de cadenas alimenticias con este tipo de modelos y se estudian los comportamientos globales del sistema. En la microbiología se han modelado microorganismos tales como las bacterias, [20], [21], estos organismos son representados como agentes sensibles a los componentes químicos del medio, llevando a cabo procesos bioquímicos en su interior, incluso usando componentes basados en reglas que utilizan la lógica difusa

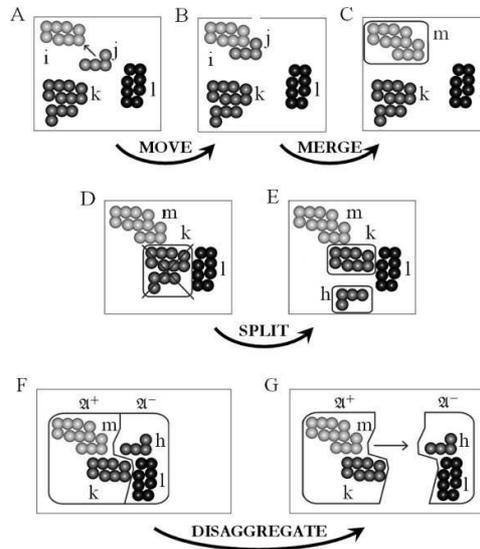


Figura 3.3: En la figura tomada de [25], se muestran las acciones básicas de los agentes (conjuntos de partículas). El agente j puede ejecutar un simple movimiento en el espacio (de A a B). Los agentes i y j se pueden mezclar en un nuevo agente m si su energía de interacción es favorable (de B a C) y evolucionar en el espacio como un sólo cuerpo. Si la energía de un agente k es demasiado alta comparada con la de otros agentes en la simulación, entonces se divide, produciendo un nuevo agente h (de D a E). Si el conjunto de agentes m, k, l, h forman un cúmulo estático, ellos se distancian (de F a G).

para determinar las acciones de la bacteria. También se estudian las poblaciones de bacterias y su comportamiento. En la inmunología, las células del sistema inmunológico son modeladas como agentes [13], de igual forma se representan los microorganismos patógenos y las células que son infectadas. Como en los casos anteriores, se definen las interacciones entre agentes o entre agentes y el medio y se estudian los comportamientos globales del sistema (ver la siguiente sección).

En química se han construido algoritmos inspirados en los modelos basados en agentes, para predecir configuraciones de mínima energía en un conjunto de moléculas idénticas [25]. En estos algoritmos un agente puede ser una molécula o un conjunto de moléculas, se registra la energía mínima de los cúmulos de N moléculas y se toma como parámetro para alcanzar este valor en todos los demás cúmulos. Este parámetro es considerado como un elemento de aprendizaje en el modelo. Durante el algoritmo se ejecutan ciertas acciones como mezclar dos cúmulos, separarlos, o distanciarlos, con el objetivo de alcanzar su mínima energía.

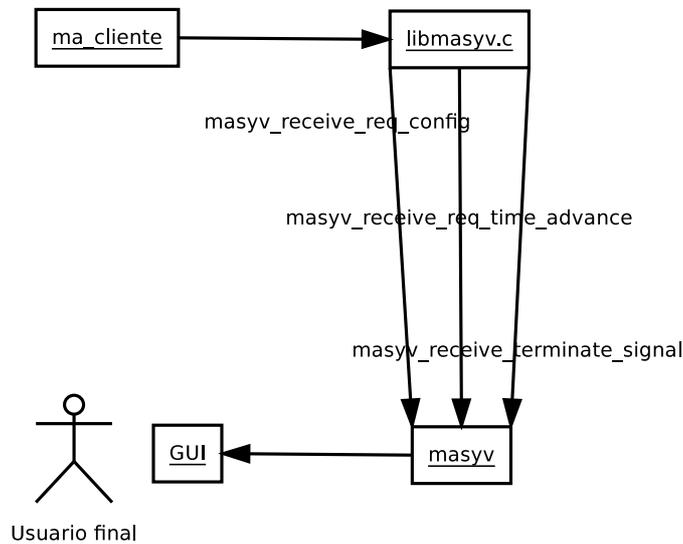


Figura 3.4: Arquitectura cliente-servidor de MASyV

3.3. MASyV

Para implementar la simulación se eligió el software libre llamado *A Multi-Agent System Visualization*¹, MASyV por sus siglas en inglés, es un software creado por Catherine Beauchemin que permite visualizar autómatas celulares y modelos basados en agentes usando una arquitectura cliente-servidor. De acuerdo con esta arquitectura MASyV proporciona una interfaz gráfica de usuario (*Graphic User Interfaces, GUI*), como servidor, así el usuario no tiene la necesidad de implementar una interfaz gráfica y puede concentrarse en programar la simulación usando el lenguaje C, este programa figura como cliente de la interfaz gráfica. MASyV también cuenta con un servidor de línea de comandos, que no es gráfico, logmasyv, también incluye una biblioteca llamada libmasyv, que contiene las funciones que usará el cliente para comunicarse con el servidor.

La biblioteca libmasyv que maneja la entrada y la salida de datos, también proporciona servicios para supervisar la ejecución de la simulación. El sistema ha sido utilizado exitosamente simulando la dinámica del virus de la influenza que explicaremos a continuación.

¹<http://masyv.sourceforge.net>

3.3.1. Aplicaciones de MASyV en sistemas complejos

MASyV ha sido utilizado para estudiar el efecto de la distribución del virus de la influenza en la dinámica de la infección *in vitro* en las células epiteliales del pulmón [14]. Este modelo basado en agentes en donde las células interactúan con su entorno, otras células o virus, usa una malla hexagonal al estilo de los autómatas celulares para representar dichos agentes. Usando datos experimentales las primeras simulaciones muestran ser bastante coherentes con los resultados *in vitro* demostrando la posibilidad de usar el modelo *in silico* para explorar nuevos parámetros en la dinámica de la infección o probar hipótesis de posibles tratamientos antes de aplicarlos en seres humanos.

En la misma línea de trabajo, investigando los efectos de las variaciones en las configuraciones espaciales de las células infectadas por el virus, podemos citar [15] donde MASyV fué utilizado para representar el modelo, una malla cuadrada en 2-D donde cada celda representa una célula susceptible de ser infectada, y todo el espacio es recorrido por las células del sistema inmunológico que pueden matar a las células infectadas. Las distribuciones de las células infectadas incluyen tres casos, un número de células distribuidas uniformemente en el espacio, otro puede ser varios grupos aislados de células infectadas, y el tercero es un sólo cúmulo de células infectadas. Se muestra como la distribución inicial de las células infectadas afecta el desarrollo de la infección, debido a que una célula infectada al estar en un cúmulo aislado de células infectadas no puede esparcir la infección tan rápidamente como una que esta rodeada de células sanas esparciendo la infección en ellas. El autor también explora la regeneración de las células. En los modelos que usan EDO la razón de regeneración celular depende del número de células vivas en el modelo, como una función de división celular, o es una constante, en cambio en el modelo de Beauchemin, un espacio vacío es ocupado por una nueva célula si esta esta junto a una célula sana y con capacidad de dividirse. La onda de la infección deja una secuela de células muertas que sólo podrán reemplazarse por células nuevas al estar junto a células sanas. Esto pasa cuando el sistema inmunológico logra romper la ola de infección, que separa las células muertas de las sanas. Este modelo produce una curva de la razón de células muertas con respecto al tiempo que es muy similar a la curva de datos experimentales y contrasta con la afilada y delgada curva del modelo donde la razón de la regeneración de células depende sólo de las células sanas y no de su ubicación.

Generalmente las simulaciones 2-D en inmunología son simulaciones de la infección *in vitro*, y donde las células susceptibles a infectarse son estáticas, éstas simulaciones tienen sus limitaciones ya que la infección *in vivo* puede no desarrollarse igual, por ejemplo las células T no son estáticas y están en constante movimiento, lo cuál se tiene que considerar en el modelo

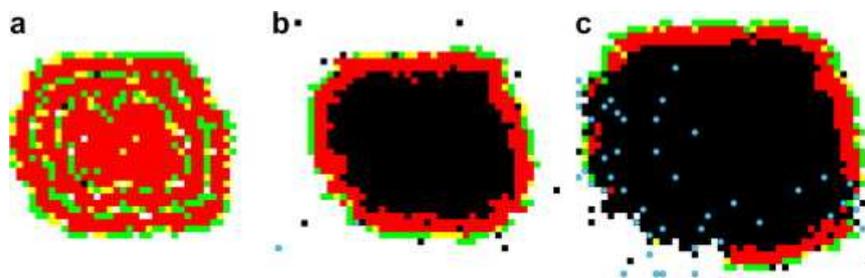


Figura 3.5: En la figura tomada de [4] se muestra el efecto de la regla de regeneración de células en la dinámica de la infección de la influenza. De izquierda a derecha: (a) Usando una regla global en donde las células muertas son regeneradas a una razón proporcional al número de células sanas; (b) usando una regla local donde las células muertas son reemplazadas sólo si alguna de las células vecinas esta sana y dividiéndose; (c) usando una regla local en donde las células muertas sólo pueden ser regeneradas una vez que la ola de la infección ha sido penetrada por las células del sistema inmunológico. Usando la misma razón de mortalidad e infección, la regla local resulta en un número mayor de células muertas pero en un número menor de células infectadas en comparación con una regla global no espacial. Las imágenes fueron tomadas de la ejecución del cliente `ma-immune` de MASyV, y representan en 2-D un tejido epitelial que es recorrido por células del sistema inmunológico (en azul), donde cada celda de la malla corresponde a una célula del tejido que puede estar sana (en blanco), muerta (en negro) o en algún estado de infección (verde, amarilla, o roja).

para obtener un comportamiento de la simulación semejante al sistema real. MASyV nos permite mover a los agentes del modelo o dejar otros estáticos debido a que podemos definir varias capas en el espacio de la simulación.

Existen infecciones *in vivo* que pueden ser modeladas en 2-D como en [15]. En este caso la infección no degenera en neumonía y sólo ataca las células que recubren las vías respiratorias, las cuáles están fijas, lo cual hace lógico pensar la simulación en 2-D. Los autores mencionan en [4] que cuando escribieron el artículo existía poca información accesible que proporcionara datos experimentales acerca de la dinámica de la influenza y que era difícil justificar cuestiones espaciales del modelo. Sin embargo lograron fijar algunos parámetros y hacer relaciones realistas con el sistema biológico. El MBA fué capaz de reproducir algunas características importantes de la influenza A como determinar que el máximo de células infectadas se da a las 48 h y que la enfermedad dura de 5 a 7 días. Ya que se demostró que el modelo era lo suficientemente confiable para simular el sistema biológico, el modelo fué utilizado para determinar los valores de otros parámetros que fueron usados en otros modelos de influenza.

Parte II

El algoritmo basado en agentes

Capítulo 4

El modelo simple de morfogénesis

No simularemos exactamente el movimiento de las células desde su estado embrionario, haremos una aproximación con las características que consideramos más importantes. Suponemos que las interacciones que se dan entre las células pueden modelarse como se modelan las interacciones entre partículas. Así que usaremos indistintamente las palabras, célula, partícula, molécula o agente en nuestro modelo para referirnos a la misma cosa.

En un sistema de partículas la fuerza que se ejerce sobre una partícula generalmente es la suma de otras tres fuerzas [8]:

1. Fuerza externa
2. Fuerzas de corto alcance
3. Fuerzas de largo alcance

A continuación explicamos como se incluyeron estos tres tipos de fuerzas.

4.1. Formación de cúmulos de células

Como se menciona en la primera parte de este trabajo, usaremos algunas heurísticas de [5] que son compatibles con nuestro modelo y son de gran utilidad para simular las interacciones que se dan entre las partículas en un modelo de tensión superficial.

Recordando al modelo citado, tanto en el modelo de Schelling como en nuestro problema encontramos dos tipos de partículas o agentes (células para nuestro caso) que se repelen mutuamente, en este modelo discreto la repulsión de un tipo a otro consiste en el siguiente criterio: dada una partícula de un tipo determinado, si dicha partícula esta rodeada por un número menor o igual a x de partículas de distinto tipo su *utilidad* es 1 por lo cuál

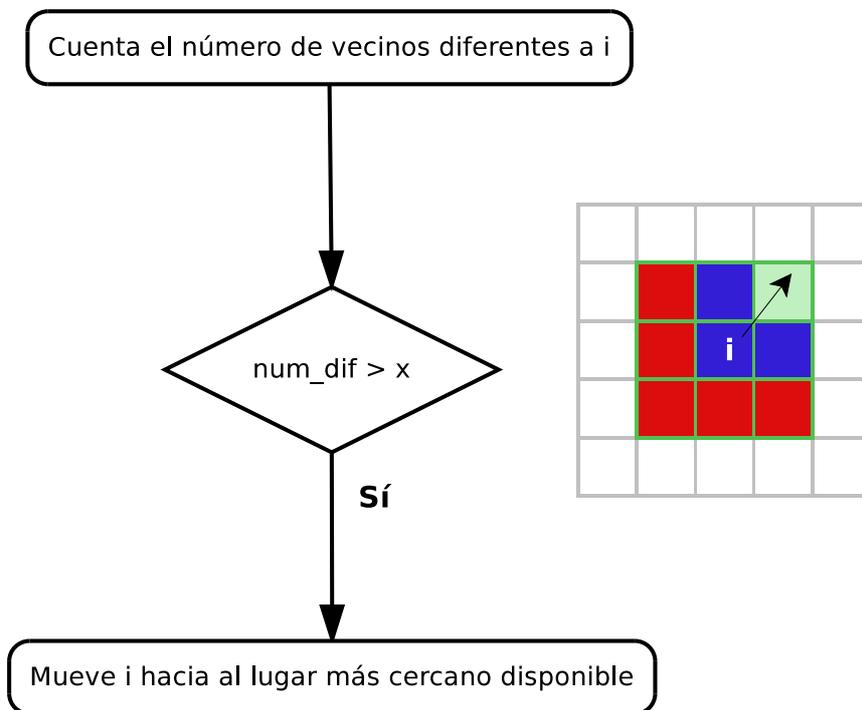


Figura 4.1: En este ejemplo, después de contar para cada partícula i , cuántas partículas del tipo contrario se encuentran en su vecindada de Moore, movemos la partícula si el número de vecinos diferentes es mayor a $x=3$.

permanecerá en su sitio, y en otro caso es θ , por lo cuál dicha partícula se deberá desplazar hacia el lugar más cercano disponible buscando una mayor *utilidad*.

Las células estan sujetas a la tensión superficial, y de acuerdo con la descripción del modelo de Vinkovic-Kirman [5] la superficie de los *cúmulos* de células tiende a aplanarse.

4.2. Cálculo de la fuerza usando el potencial

Las partículas además de estar sujetas a fuerzas de corto alcance, también estan sujetas a fuerzas de largo alcance, estas interacciones se dan de partícula a partícula, así que tenemos que considerar todos los pares de partículas.

El gradiente de la función potencial nos proporciona la dirección en la que una célula deberá moverse para alcanzar el equilibrio con otra célula. Para la célula i , localizada en x_i , el negativo de la suma de los gradientes de esta con todas las otras células determina el término de la fuerza que se ejerce sobre ella.

$$F_i = - \sum_{j=1}^N \nabla \Phi(r_{i,j}) \quad \text{donde} \quad r_{i,j} = |x_i - x_j| \quad (4.1)$$

Entonces, haciendo este cálculo movemos la célula i hacia donde apunta F_i sumando el vector F_i a las coordenadas de la célula i .

Como vimos en el capítulo dos, este cálculo es $O(N^2)$, ya que para cada partícula i tenemos que calcular el potencial con las otras $N-1$ partículas j , esto es $N(N-1)$ operaciones, pero gracias al algoritmo de Barnes-Hut[7] se reducen a $O(N \log N)$. Hemos implementado este algoritmo para el cálculo del potencial en la simulación.

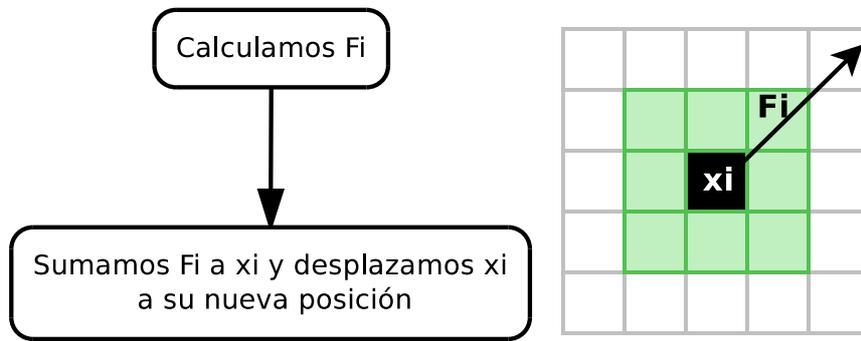


Figura 4.2: Sumamos el vector de la fuerza F_i a las coordenadas de la célula i

4.3. Migración dirigida

También existen fuerzas externas que intervienen en el desarrollo del sistema, estas fuerzas externas pueden ser algunos puntos de fijación u orientación para estas células en la matriz extracelular de la epidermis.

En el desarrollo de la simulación determinamos incluir otro de los aspectos biológicos en el desarrollo embrionario, si bien no consideramos factores de crecimiento, observamos que dirigir la migración de las células pigmentarias es otro de los aspectos que producen cambios en la formación de patrones de coloración. Otro modelo de simulación que consideran la migración de células más detalladamente es [10].

Como una primera aproximación a este fenómeno en la simulación, en alguno de los ensayos *in silico* que mostraremos más adelante, para cada intervalo de tiempo dejaremos avanzar las células de arriba hacia abajo, después aplicaremos el potencial y por último la heurística para simular la tensión superficial.

4.4. Implementación

La implementación del algoritmo basado en agentes para la simulación se compone de los siguientes elementos:

1. El *ambiente* en donde residen los agentes que representa una sección del tejido cutáneo.
2. Los *agentes*, en este caso las células pigmentarias.
3. Las *reglas* que gobiernan la dinámica del sistema representando las interacciones físicas y biológicas del agente con otros agentes o el medio.
4. Las *escalas de tiempo* en donde las reglas son ejecutadas.

A continuación describiremos cada uno de estos componentes.

4.4.1. El ambiente

La simulación se ejecuta en una malla homogénea rectangular en el plano, la malla esta compuesta de celdas cuadradas del mismo tamaño. La vecindad de interacción de *corto alcance* es la vecindad de Moore. (Ver figura 1.7).

Las condiciones de frontera que definen una vecindad extendida para las celdas de los extremos se pueden definir de igual forma que en los *AC*, pero también podemos definir fronteras *absorbentes*, esto es, que cuando un individuo traspase los límites de la frontera de la malla, este individuo desaparece. En nuestro caso no hemos determinado ninguna condición de frontera debido a la naturaleza del problema y porque queremos evitar periodicidades artificiales.

En la simulación el tamaño de la malla donde se mueven las células es constante.

4.4.2. La célula como un agente

La célula responde a ciertos estímulos provenientes del medio ambiente en donde se encuentra, puede percibir estos estímulos con algunos de sus componentes. También responde a los estímulos de otras células.

Para modelar estos aspectos y las interacciones entre células se define a la célula, en este caso un cromatóforo, como un agente simple. La respuesta de la célula a los estímulos del medio es moverse hacia alguna dirección determinada por las reglas definidas en el sistema.

En nuestra simulación consideramos dos tipos de células y cada célula es del tamaño de una celda de la malla y puede moverse a cualquier otra celda que este disponible en la malla.

Las reglas incluyen las percepciones del ambiente.

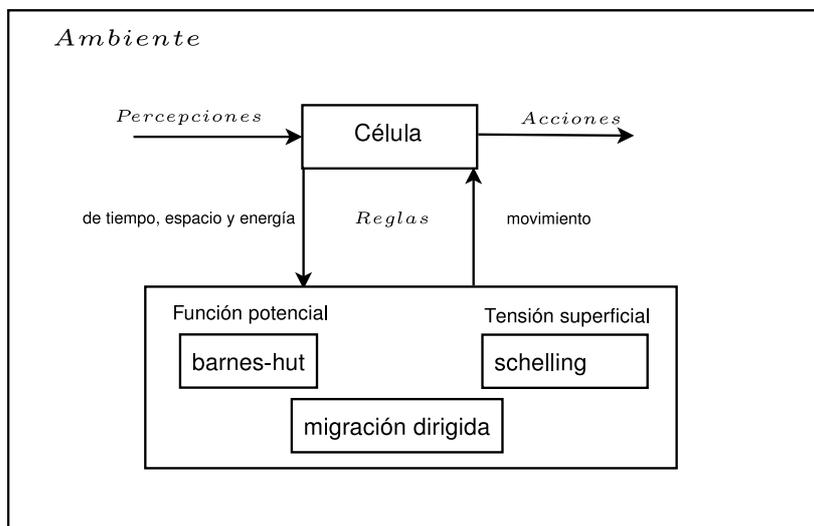


Figura 4.3: Se concibe a la célula como un agente sensible al medio, puede determinar las fuerzas de largo alcance que otras células ejercen sobre ella, la tensión superficial del cúmulo en donde se encuentra o equivalentemente el grado de tolerancia a células contrarias y también el tiempo de evolución del sistema como una percepción de su ambiente, después determina su dirección de movimiento de acuerdo a las reglas del modelo.

4.4.3. Las reglas del MBA

De acuerdo al modelo que describimos en las secciones anteriores los tres aspectos fundamentales que tratamos de simular son las fuerzas de corto y de largo alcance que hemos modelado con la función potencial y utilizando el algoritmo de Barnes-Hut, la tensión superficial modelada con la heurística de Vinkovic-Kirman para el modelo de Schelling, y la migración dirigida.

La decisión de cuál acción tomar está basada en la configuración actual del sistema. Para describir estas situaciones usamos reglas. Una regla es una combinación de condiciones y acciones, entonces tenemos tres reglas para el movimiento de la célula. (Ver figura 4.3).

Las siguientes reglas se aplicarán secuencialmente para cada tiempo t :

1. La regla para calcular las fuerzas de corto y largo alcance. Esto consiste en calcular la fuerza a partir de la función potencial y luego mover la célula hacia a donde apunta esta fuerza. Esta regla está representada por la función `barnes-hut`.
2. La regla para simular la tensión superficial equivalente a la repulsión entre células de distinto tipo, la célula se moverá a un lugar donde esté con células del mismo tipo y los cúmulos tenderán a redondearse buscando la forma más estable. La función que representa esta regla es la función `schelling`. (Ver figura 4.1).
3. La función `migracion_dirigida`, representa una regla que simula la migración de los cromatóforos desde la cresta neural, para este modelo simplemente es mover las células hacia abajo, habiendo iniciado su movimiento desde la parte superior de la malla.

4.4.4. Escalas de tiempo

Cada tiempo t en la simulación tuvo que ser dividido en tres subintervalos, uno para aplicar la regla `barnes-hut`, otro para `schelling`, y por último la `migracion_dirigida`.

Existen tres tiempos que son importantes en la ejecución de la simulación el primero es `evo_t` que es donde el sistema empieza aplicar algunas o todas las reglas. En la simulación primero se agregan células de un tipo y luego del otro, así que el segundo tiempo es el que marca el cambio de cuando pasar de agregar células de un tipo a otro `c_t`. El tercero es el tiempo para la terminación de la simulación que llamaremos `f`, que es cuando detenemos todas las reglas a excepción de la regla `schelling` que sigue ejecutándose.

4.5. El algoritmo general

Establecidas algunas de las fuerzas que intervienen en el fenómeno biológico y representándolas como reglas del MBA daremos paso a describir el al-

goritmo en lo general donde estas reglas se implementan como funciones del sistema.

La evolución del sistema empieza a partir del tiempo $t = \text{evo_t}$ donde empezaremos a aplicar estas funciones. Para cada tiempo t agregamos un número fijo n de células, hasta llegar a un número determinado N de células que deseamos que contenga el sistema, después no desaparecen las células manteniéndose estas N células en el sistema.

Entonces para cada tiempo t :

Agregamos un número n de células en la parte superior de la malla de la simulación, si num_cel es menor a N .

- Si el tiempo $t < \text{evo_t}$:
 1. Sumamos n a num_cell
 2. Movemos las células hacia abajo, las células migran hacia abajo encontrando los lugares disponibles.

- en otro caso:
 1. Aplicamos consecutivamente las funciones para cada una de las fuerzas físicas que establecimos para modelar el sistema.
 - Si $t \bmod 3 = 0$. Aplicamos la migración dirigida. (`migracion_dirigida`)
 - Si $t \bmod 3 = 1$. Aplicamos el potencial usando el algoritmo de Barnes-Hut. (`barnes-hut`)
 - Si $t \bmod 3 = 2$. Aplicamos la tensión superficial, usando la heurística de Vinkovic-Kirman para el modelo de Schelling. (`schelling`)
 2. Si t es igual al tiempo deseado de enfriamiento del sistema f , detenemos la simulación.

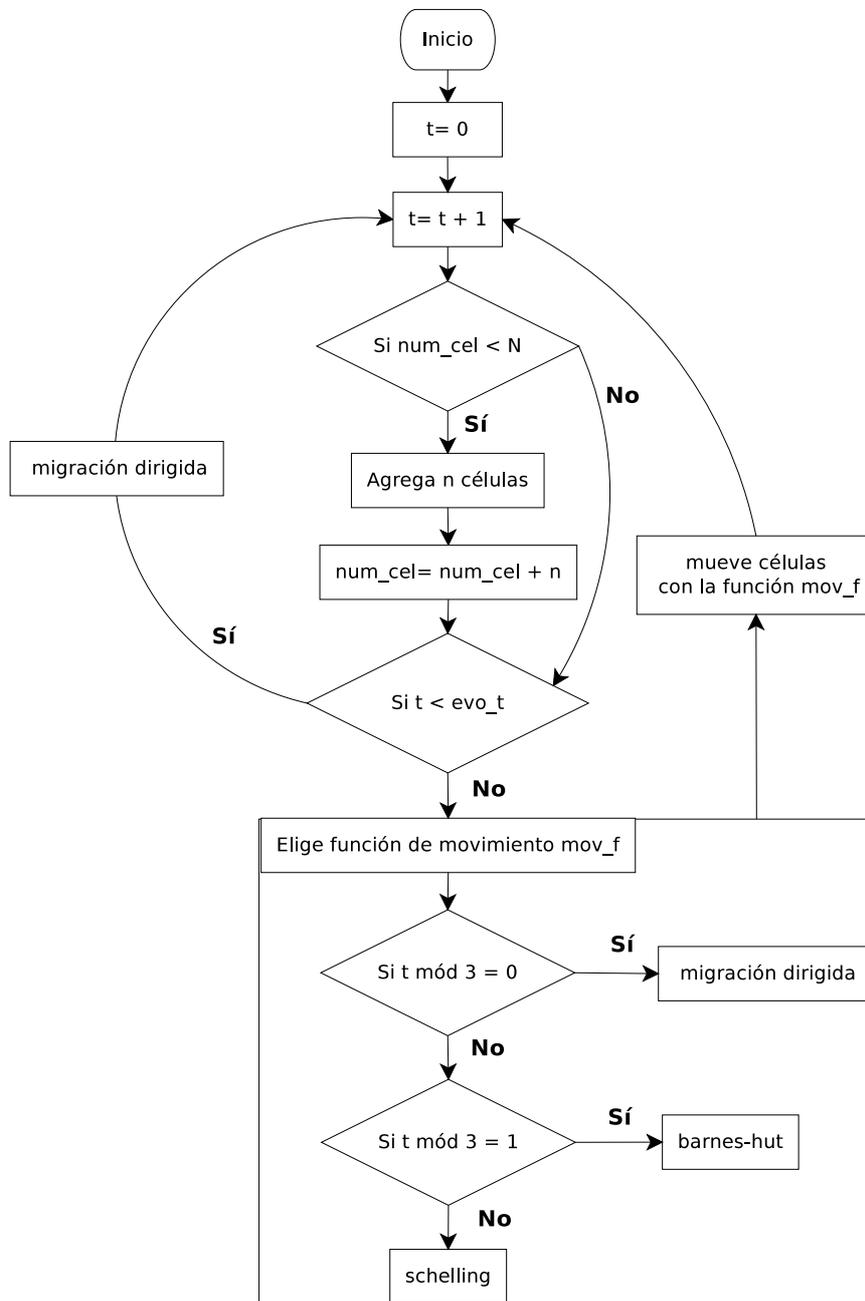


Figura 4.4: Diagrama general del algoritmo implementado para la simulación

Capítulo 5

Resultados

De acuerdo a la definición del algoritmo, recordamos que el número de células se incrementará constantemente, agregaremos un número n de células en cada tiempo t hasta llegar a un número máximo N , que será el total de células en el sistema. Antes del tiempo de evolución del sistema, que llamamos evo_t , las células se agregan en la parte superior de la malla, simulando su nacimiento desde la cresta neural del embrión moviéndose hacia abajo. Después del tiempo evo_t aplicaremos alguna o todas las funciones que representan las interacciones celulares que describimos anteriormente, también en esta fase de evolución del sistema si es que se siguen agregando células, puede darse el caso que estas no se agreguen exactamente en la parte superior, sino un poco más abajo para evitar el estancamiento de las células en la parte superior.

El tiempo llamado c_t indica el momento en el que pasamos de agregar células de un tipo a agregar células del tipo contrario. El tiempo f , que representa el tiempo de “enfriamiento” del sistema, que es donde la ejecución del sistema se detiene.

Con el propósito de tener un buen desempeño de la simulación, hacemos $n = 100$. También cuando movemos una célula y esta no encuentra un espacio disponible intenta sólo una vez reacomodarse, el proceso de reacomodo consiste en que aleatoriamente se elige un espacio en su vecindad de Moore, si este está disponible lo ocupa, si no se queda donde estaba.

5.1. La función potencial y sus parámetros

Hemos decidido usar la función potencial presentada en [31] pero con signo contrario. La razón de haber elegido esta función es que sus parámetros representan con claridad las magnitudes de las fuerzas de atracción y repulsión.

Para cada par de células localizadas en x_i y x_j definimos el potencial como:

$$\Phi(r_{i,j}) = C_R e^{\frac{|r_{i,j}|}{L_R}} - C_A e^{\frac{|r_{i,j}|}{L_A}} \quad \text{donde } r_{i,j} = |x_i - x_j|. \quad (5.1)$$

donde C_A y C_R son magnitudes constantes de las fuerzas de atracción y de repulsión respectivamente. L_A y L_R son longitudes constantes del alcance de las fuerzas de atracción y repulsión.

Un valor negativo de $\Phi(r_{i,j})$ indica un potencial atractivo entre las dos células mientras que un valor positivo representa un potencial repulsivo. Si el valor del potencial es cero indica que las células se encuentran a una distancia en donde alcanzan el equilibrio. El gradiente de la función potencial nos proporciona la dirección en la que una célula deberá moverse para alcanzar el equilibrio con otra célula.

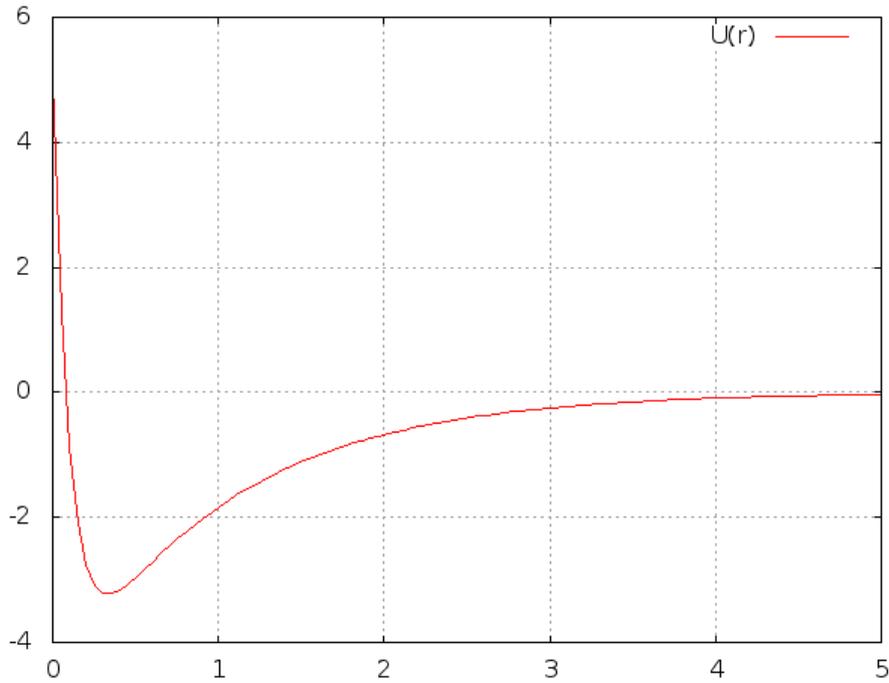


Figura 5.1: Gráfica de la función potencial, con los parámetros $C_A = 5$, $L_A = 1$, $C_R = 10$, $L_R = 0,1$

5.2. Experimentos

5.2.1. Experimento 1

En el siguiente experimento el tiempo de evolución es `evo_t=50`, el tiempo donde la simulación pasa de agregar células de un tipo a otro es `c_t=60`

y el tiempo de congelación es $f=120$ y el número de células $N = 5700$.

Los parámetros del potencial son $C_A = 5$, $L_A = 1,0$, $C_R = 10$, $L_R = 0,1$. Después del tiempo de evolución agregamos las células sólo en la parte superior.

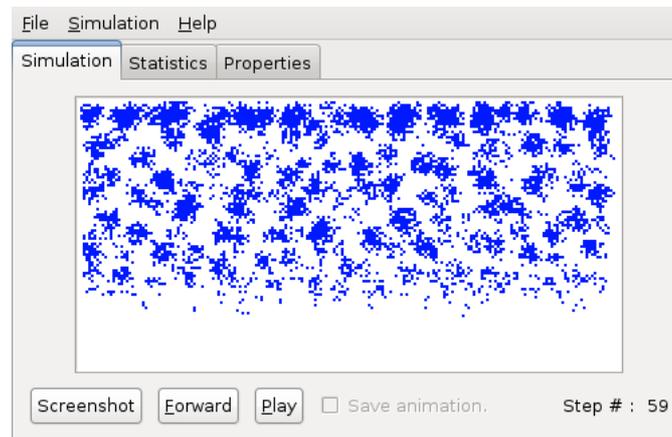


Figura 5.2: Evolución en el tiempo $t = 59$.

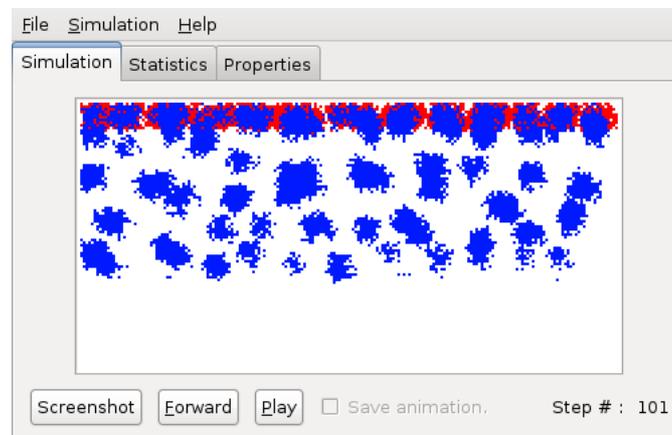


Figura 5.3: Evolución en el tiempo $t = 101$.

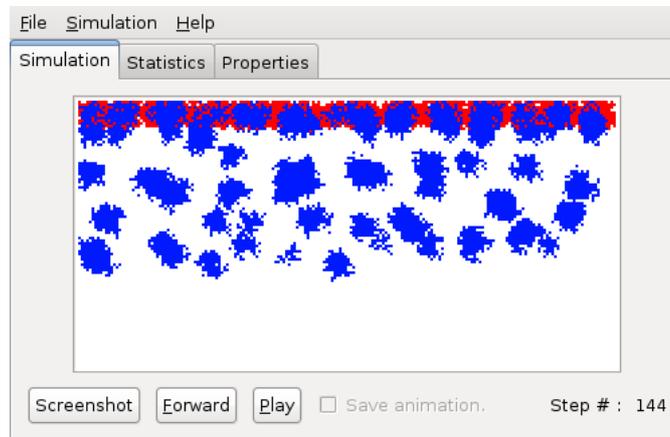


Figura 5.4: Evolución en el tiempo $t = 144$.

Como podemos ver en estos resultados el hecho de agregar las células en la parte superior después del `evo_t` hace que las células del segundo tipo en rojo queden fijas en la parte superior.

5.2.2. Experimento 2

Ahora si aumentamos el tiempo de cambio de un tipo de célula a otro `c_t=80` y el número total de células $N=7400$ y además permitimos que después de `evo_t` las nuevas células se agreguen hasta la mitad de la malla, obtenemos el siguiente patrón.

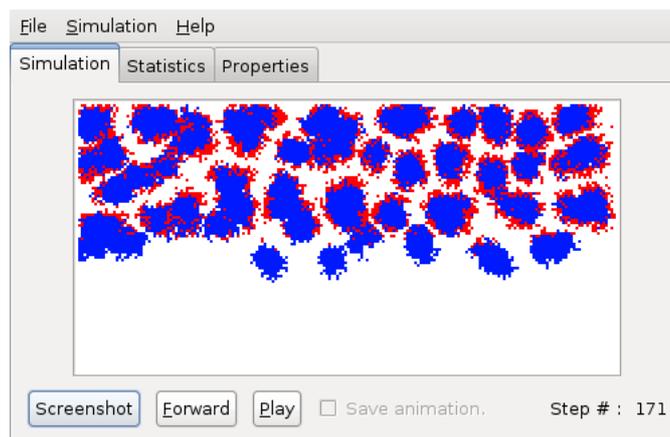


Figura 5.5: Los cúmulos presentan una forma más redondeada cuando las células del segundo tipo quedan en la superficie ya que se genera una fuerza de tensión superficial entre ambos tipos de células.

5.2.3. Experimento 3

Observamos que si aumentamos la magnitud del coeficiente de atracción C_A podemos obtener cúmulos más grandes. El siguiente resultado lo demuestra si hacemos $C_A = 10$ y $c_t=80$.

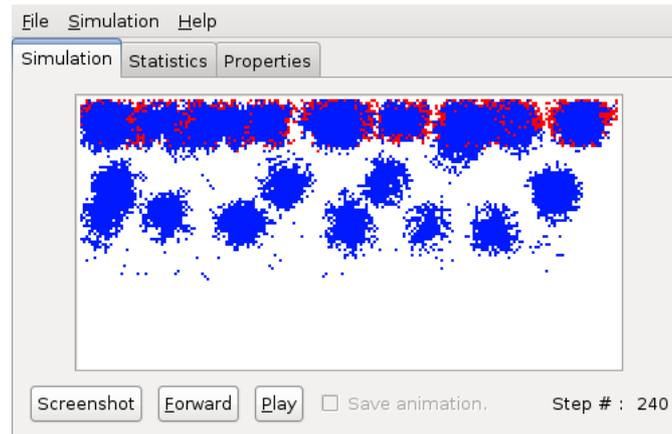


Figura 5.6: Resultado final aumentando el coeficiente de atracción del potencial y el tiempo donde se cambia de un tipo de célula a otro.

5.2.4. Experimento 4

Al igual que en el caso anterior, de nuevo permitimos que las nuevas células se puedan esparcir en toda la mitad superior de la malla y definimos $N=7400$.

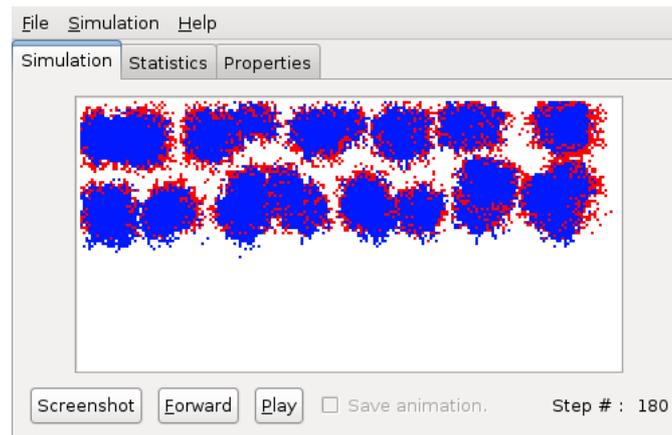


Figura 5.7: Aquí también por la tensión superficial, los cúmulos presentan una forma más redondeada cuando las células del segundo tipo quedan en la superficie.

5.2.5. Experimento 5

También observamos que al aumentar el radio de atracción L_A se puede aumentar el tamaño de los cúmulos, pero en este caso ya no son tan redondeados como los anteriores. Los parámetros del potencial del siguiente resultado son $C_A = 10$, $L_A = 2,0$, $C_R = 10$, $L_R = 0,1$ y $c_t=80$ y $N=5700$.

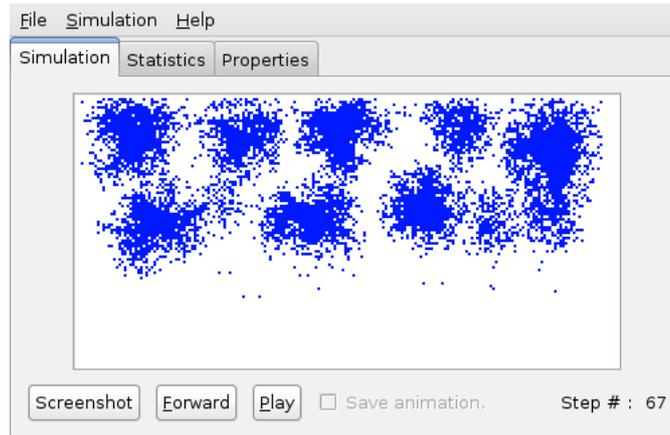


Figura 5.8: Evolución en el tiempo $t = 67$.

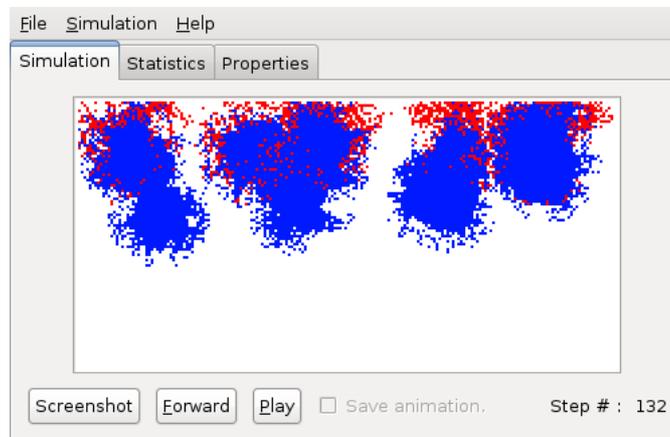


Figura 5.9: Evolución en el tiempo $t = 132$.

5.2.6. Experimento 6

Si iniciamos la evolución del sistema en una etapa temprana, digamos $evo_t=10$, las células se empiezan a acumular en la parte superior del plano

de la simulación, después de este instante, las células sólo se agregan en la parte superior si es que hay espacio.

En el siguiente resultado, donde $evo_t=10$, y las células se agregan en la parte superior y buscan un lugar disponible en su vecindad de Moore, aunque $n=4000$ se acomodaron un poco más de 3000 células debido a que no siempre encontraban un lugar disponible, $c_t=70$ y $f=150$.

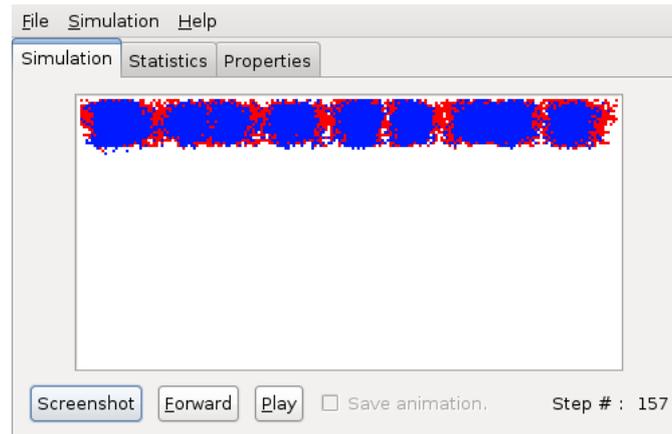


Figura 5.10: Se puede notar que las células del segundo tipo en color rojo fijan y redondean los contornos de los cúmulos de células del primer tipo en azul.

5.2.7. Experimento 7

Ahora basándonos en lo general en experimentos en embriones de ajolote [11], movemos las células desde la cresta neural hacia abajo incluso después del inicio de la evolución del sistema y alternaremos las reglas `barnes-hut`, `schelling` y por último `migracion_dirigida` que es la regla que mueve las células hacia abajo. Las células nacen desde la parte superior de la malla de la simulación.

Los parámetros del potencial para este experimento son $C_A = 10$, $L_A = 1,5$, $C_R =$, $L_R = 0,1$, los parámetros de la simulación son $evo_t=15$, $c_t=80$, $N=8000$ y $f=250$.

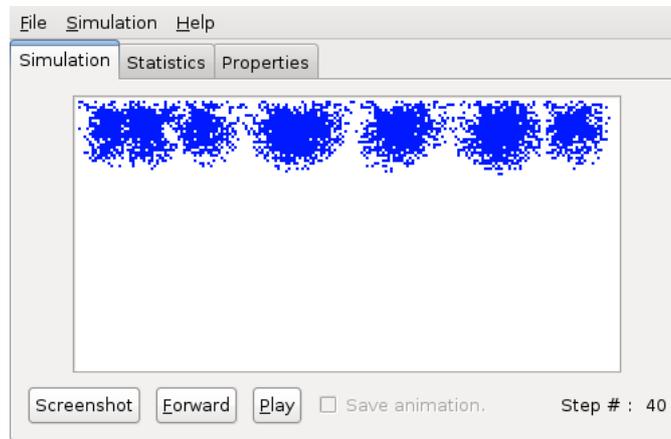


Figura 5.11: Evolución en el tiempo $t = 40$.

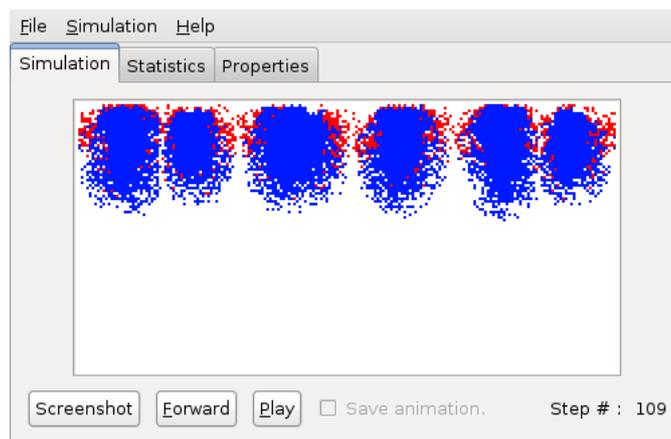


Figura 5.12: Evolución en el tiempo $t = 109$.

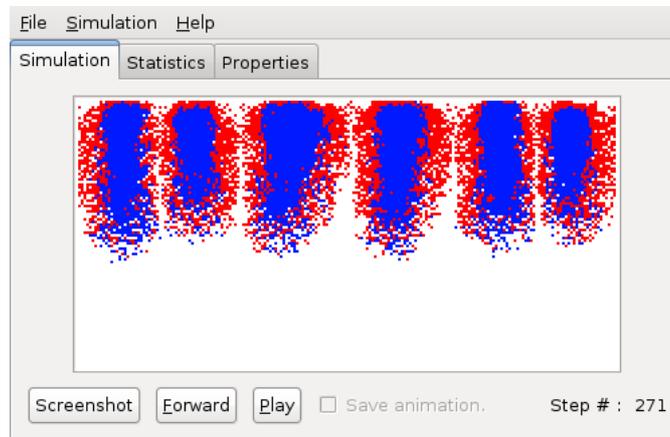


Figura 5.13: Evolución en el tiempo $t = 271$.



Figura 5.14: Puesto que el presente trabajo tiene como motivación un fenómeno biológico, presentamos esta imagen tomada de [10] que muestra la larva del ajolote *Ambystoma mexicanum* para comparar su patrón de coloración con el resultado anterior.

Aunque estos resultados corresponden a un modelo simple implementado en una simulación computacional, ellos se aproximan a patrones de coloración que realmente suceden en la naturaleza. Con algunos elementos mínimos de la biología del desarrollo que se han considerado; como lo son la existencia de dos tipos de células diferentes que se repelen, o suponer que las células migran desde la cresta neural hacia abajo, pero también sin ignorar las fuerzas físicas más importantes, se puede llegar a obtener desde manchas redondeadas o irregulares a bandas verticales alternadas que podemos ver en varios peces o anfibios.

Conclusiones

En el problema específico de tratar de predecir los patrones de coloración en la piel de algunos anfibios y peces, las ciencias de la computación pueden aportar no sólo algoritmos eficientes para realizar los cálculos y proveer las herramientas de software para llevarlos a cabo, sino también una forma de modelar los problemas de esta naturaleza, los modelos basados en agentes han demostrado ser bastante útiles para modelar con simpleza a las células involucradas en este tipo de problemas y ser representadas en una computadora de manera modular y extensible, también pudimos definir algunas interacciones básicas que se dan en el sistema biológico a través de reglas del modelo.

En el modelo se destaca la importancia de la definición de las reglas, que como en los AC, los MBA pueden generar comportamientos muy distintos en la modificación de éstas, ya sea cuando existe una perturbación en los parámetros de las reglas o quitando o incluyendo alguna regla. Por ejemplo, pudimos observar diferentes patrones cambiando los parámetros de la función potencial, o cuando retrasábamos el momento en que se aplica la regla que representa la fuerza de tensión superficial, o cuando las células migran o no desde la parte superior de la malla de la simulación.

Las variaciones pueden ser muchísimas, pero con las que se han presentado es suficiente para entender los cambios que pueden producir cada una.

Al presentar las variantes en las reglas, y sus diferentes resultados, este trabajo sugiere que cambiando estas reglas o sus parámetros y tal vez incluir algunas más que sean claves en el proceso se puede llegar a construir modelos bastante cercanos al proceso real de morfogénesis.

Se pudo observar con claridad la presencia de las fuerzas de corto y largo alcance ya que dichas fuerzas se pudieron manipular cambiando los parámetros asociados a su magnitud y alcance. Por ejemplo para aumentar el tamaño de los cúmulos bastó con aumentar la magnitud de la fuerza de atracción o el alcance de la fuerza de atracción.

Aunque en todas las variantes que se usaron de la función potencial se representa *una fuerza de repulsión de muy corto alcance y una fuerza de atracción de largo alcance*, las distancias menores que la unidad no cuentan para el caso discreto que proponemos, es decir, no se puede dar una dis-

tancia entre dos células menor a la unidad. También cabe mencionar que al determinar la dirección y el lugar (la celda de la malla) hacia donde se moverá cada célula, si no existe una celda disponible en la vecindad de Moore del lugar elegido, simplemente la célula se queda en su lugar provocando un modelo de *esfera dura* o sugiriendo que una vez que los cúmulos son de determinado tamaño sólo cambiarán en su superficie que es donde puede haber lugares disponibles.

Las células viviendo en un medio físico no pueden estar ajenas a las fuerzas físicas que gobiernan el universo. La fuerza de tensión superficial juega un papel importante ya que al incluir la regla que la representa, los cúmulos formados por células de un tipo presentan contornos más redondeados cuando se incluyen células del tipo contrario que cuando no se incluyen. Esto quiere decir que esta regla contribuye a minimizar la energía de los cúmulos, minimizando su superficie, haciéndolos más estables.

En los experimentos en donde se aplicó la regla `migracion_dirigida`, que consiste en mover las células hacia la parte inferior de la malla y aplicamos también las otras dos reglas `barnes-hut` y `schelling` se empiezan a notar como las manchas redondas de células azules, se transforman en bandas verticales más o menos regulares, y que las células de color rojo, que migran después que las células azules empiezan a ocupar los espacios entre estas bandas creándose un patrón que empieza a formarse alternando las bandas azules con bandas rojas. Este patrón se parece al que se puede observar en algunos animales, como en la larva del ajolote *Ambystoma mexicanum* o en algunos peces.

En la construcción del modelo, al tratar de incluir las fuerzas de corto y largo alcance, surge el problema de involucrar miles de células y calcular todas las interacciones de las cuáles el cálculo de manera directa era muy ineficiente, pero gracias al algoritmo de Barnes-Hut se pudieron aproximar muy bien las interacciones de miles de células y no sólo de unas cuantas. El algoritmo dividiendo el problema en varios pequeños problemas al subdividir el espacio en donde se encuentran partículas o cuerpos, generando estructuras de datos jerárquicas como los árboles, hace uso de esta jerarquía para aproximar las interacciones y mejora grandemente el desempeño de la simulación computacional reduciendo el número de interacciones de $O(N^2)$ a $O(N \log N)$. Así, el algoritmo nos permitió explorar la influencia de este tipo de interacciones, que sin ellas se tendría en un modelo incompleto.

Trabajo futuro

Por último podemos decir que aunque en este trabajo no se han explotado al máximo las virtudes de los MBA sí podemos ver varias de estas ventajas, una de ellas es que el modelo nos va a permitir llevar esta implementación del algoritmo a una implementación en paralelo y que utilizando los métodos

de este enfoque computacional se podría optimizar aún más el desempeño de la simulación computacional, con la posibilidad de incluir más interacciones.

Otra de las posibles extensiones de este MBA es extender la definición de agente ya no sólo a células simples, sino a cúmulos de células, aunque se ha dado una aproximación a esta característica a través del algoritmo de Barnes-Hut, no se calculó la energía para cada cúmulo de células ni tampoco se determinó el tamaño de los cúmulos, éstas y otras características podrían incluirse en nuestro modelo como parte de la definición de los agentes y controlar dichas magnitudes.

Para ejemplificar una de estas ventajas, podríamos incluir un elemento de *aprendizaje* para minimizar la energía del sistema, supongamos que podemos calcular la energía de cada *agente cúmulo* y que registramos el tamaño del *cúmulo* del sistema cuya energía es la menor de las energías de todos los cúmulos. Este cúmulo con energía mínima tendría un tamaño de M células, entonces podemos incluir una regla que controle el tamaño de todos los *agentes cúmulos* para que el número de las células que los componen se acerque lo más posible a M y actualizar este valor para cada intervalo de tiempo.

También sería interesante estudiar la migración celular más detalladamente, que igualmente se podrán usar los MBA explorando más posibilidades de movimiento de las células, cada agente puede tener caminos dirigidos por algún químico en la superficie dependiendo del tipo de agente.

Otro de los aspectos que sugieren un estudio más detallado sería determinar los parámetros de las posibles funciones potenciales para cada patrón de coloración específico, por ejemplo las relaciones entre estos parámetros y el tamaño de las manchas.

Aunque quedan muchísimos detalles y variantes por estudiar creo que este trabajo sirve como punto de partida para otros más complejos.

Bibliografía

- [1] C. Zhu, G. Bao, N. Wang. Cell Mechanics: Mechanical Response, Cell Adhesion, and Molecular Deformation. *Annu. Rev. Biomed. Eng.* **02**. pp. 189-226. (2000).
- [2] C. E. Caicedo-Carvajal, T. Shinbrot. In silico zebrafish pattern formation. *Developmental Biology* **315**. pp. 397-403. (2008).
- [3] G. Cocho, R. Pérez-Pascual, J.L. Rius, F. Soto, Discrete Systems, Cell-Cell Interactions and Color Pattern of Animals II. Clonal Theory and Cellular Automata. *Journal of Theoretical Biology* **125**. pp. 437-447. (1987).
- [4] A. L. Bauer, C. A. Beauchemin, A. S. Perelson, Agent-based modeling of host-pathogen systems: The successes and challenges. *Information Sciences*. **179**(10). pp. 1379-1389. (2009).
- [5] D. Vinkovic, A. Kirman. A Physical Analogue of the Schelling Model, *PNAS*. **103**(51). pp. 265-265. (2006).
- [6] S. Mossa, F. Sciortino, P. Tartaglia, E. Zaccarelli. Ground-State Clusters for Short-Range Attractive and Long-Range Repulsive Potentials. *Langmuir* **20**. pp. 10756-10763, (2004).
- [7] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*. **324**(4). (1986).
- [8] J. Demmel. Fast Hierarchical Methods for the N-body Problem, Part 1, <http://www.eecs.berkeley.edu/demmel/cs267/lecture26/lecture26.html>: 12.06.2010
- [9] C. Beauchemin. A Multi-Agent Visualization Program, <http://masyv.sourceforge.net/>: 12.06.2010
- [10] A. Deutsch and S. Dormann. *Cellular Automaton. Modeling of Biological Pattern Formation*. Springer Verlag, 2005.

- [11] H. Epperlein, J. Löfberg, L. Olsson. Neural crest cell migration and pigment pattern formation in urodele amphibians. *Int. J. Dev. Biol.* **40**. pp. 229-238. (1996).
- [12] S. Russell, P. Norving. *Artificial Intelligence: a modern approach*. Segunda edición, Prentice-Hall, Pearson Education International, 2003.
- [13] J.L. Segovia-Juárez, S. Ganguli, D. Kirshner. Identifying control mechanisms of granuloma formation during M. tuberculosis infection using an agent-based model. *Journal of Theoretical Biology.* **231**. pp. 357-376. (2004).
- [14] C. Beauchemin, S. Forrest, F.T. Koster. Modeling influenza viral dynamics in tissue. Proceedings of the Fifth International Conference on Artificial Immune Systems. *Lectures Notes in Computer Science.* **4163**. pp. 23-36. Springer Verlag. (2006).
- [15] C. Beauchemin. Probing the effects of the well-mixed assumption on viral infection dynamics. *J. Theor. Biol.* **242**(2). pp. 464-477. (2006).
- [16] N.R. Jennings. On agent-based software engineering. *Artificial Intelligence* **117**. pp. 277-296. (2000).
- [17] L. Tesfatsion. Agent-based computational economics: modeling economies as complex adaptive systems. *Information Sciences.* **149**(4). pp. 263-269. (2003).
- [18] R. Axelrod. *The complexity of cooperation: Agent based models of competition and collaboration*. Princeton NJ: Princeton University Press. 1997.
- [19] N. Gilbert. *Agent-Based Models. Quantitative Applications in the Social Sciences*. London: SAGE Publications, 2007.
- [20] R. Paton, C. Blachos, J. Saunders, H. Wu. Evolve social agents for bacterial systems modeling. *IEEE Transaction on Nanobioscience.* **3**(3). pp. 208-216. (2006).
- [21] T. Emonet, C. Machal, M. North, C. Wickersham, P. Cluzel. Agent-Cell: a digital single-cell assay for bacterial chemotaxis. *Bioinformatics.* **21**(11). pp. 2714-2721. (2005).
- [22] P. Hraber, T. Jones, S. Forrest. The Ecology of Echo. *Artificial Life* **3**(3). pp. 165-190 (1994).
- [23] O. Schmitz, G. Booth. Modelling food web complexity: The consequences of individual-based, spatially explicit behavioural ecology on trophic interactions. *Evolutionary Ecology.* **11**. pp. 379-398. (1997).

- [24] C. Rouff, M. Hinchey, J. Rash, W. Truszkowski, D. Gordon-Spears (Eds). *Agent Technology from a Formal Perspective*. Springer-Verlag London Limited 2006.
- [25] S. Fortuna, A. Trossi. An Artificial Intelligence Approach for Modeling Molecular Self-assembly: Agent-based Simulations of Rigid Molecules. *J. Phys. Chem. B* **113**. pp. 9877-9885. (2009).
- [26] J.H. Kingston. *Algorithms and Data Structures. Design, Correctness. Analysis*. Addison-Wesley. Segunda edición, 1998.
- [27] S. Pfalzner, P. Gibbon. *Many-body tree methods in physics*. Cambridge University Press. 1996.
- [28] S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley. 1994.
- [29] O. Miramontes. Algunos aspectos de la teoría de autómatas celulares y sus aplicaciones en biofísica. *Revista del Seminario de Enseñanza y Titulación*. Año V Num. 35. Facultad de Ciencias, UNAM, México. (1989).
- [30] J. Moreira, A. Deutsch. Pigment Pattern Formation in Zebrafish During Late Larval Stages: A Model Based on Local Interactions. *Developmental Dynamics*. **232**. pp. 33-42. (2005).
- [31] A. Abajian (Advisor: J. Lowengrub). An Agent-Based Hybrid Model for Avascular Tumor Growth. *The UCI Undergraduate Research Journal*. (2008).