



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

AUTOORGANIZACION EN LA DINAMICA DE UN SISTEMA QUE SOPORTA ONDAS ESPIRALES

# TESIS

QUE PARA OBTENER EL TITULO DE:

F I S I C O

PRESENTA:

FELIPE DE JESUS NEGRETE VERDIGUEL

DIRECTOR DE TESIS: DR. OCTAVIO MIRAMONTES VIDAL

289840



FACULTAD DE CIENCIAS  
SECCION ESCOLAR

2001



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**MAT. MARGARITA ELVIRA CHÁVEZ CANO**  
Jefa de la División de Estudios Profesionales de la  
Facultad de Ciencias  
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:  
"Autoorganización en la dinámica de un sistema que soporta ondas espirales"

realizado por NEGRETE VERDIGUEL FELIPE DE JESUS

con número de cuenta 9455681-8 , pasante de la carrera de Física.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis  
Propietario

Propietario DR. OCTAVIO MIRAMONTES VIDAL

Propietario DR. GERMINAL COCHO GIL

Propietario DR. PABLO PADILLA LONGORIA

Suplente DR. JORGE HUMBERTO ARCE RINCON

Suplente DR. JOSE LUIS MATEOS TRIGOS

*O. Miramontes V.*

*G. Cocho Gil*  
*P. Padilla L.*

*J. Mateos Trigos*

**Consejo Departamental de Física.**

*Patricia Goldstein Menache*

**DRA. PATRICIA GOLDSTEIN MENACHE**

**Coordinadora de Licenciatura**

DEPARTAMENTO DE FÍSICA

El siguiente trabajo sólo tiene sentido en mi vida  
con la paciencia y apoyo de mis padres,  
Lourdes y Felipe, con el amor de mi familia,  
Diana, Diego, Claudia y Adriana, con el tiempo  
ocupado en los amigos, con los brazos abiertos  
de la Universidad. Pero sobre todo con la  
existencia de Gisell, que me ha dado una visión  
única del mundo...

# CONTENIDO

|  |           |
|--|-----------|
| Introducción   | 3         |
| <b>1 Formación de Patrones</b>                         | <b>6</b>  |
| 1.1 Introducción . . . . .                             | 6         |
| 1.2 Sistemas Naturales . . . . .                       | 8         |
| 1.3 Auto-organización, un caso particular . . . . .    | 17        |
| <b>2 Medios Excitables</b>                             | <b>20</b> |
| 2.1 Introducción . . . . .                             | 20        |
| 2.2 Excitabilidad en Fenómenos Naturales . . . . .     | 21        |
| 2.3 Formulación de ME . . . . .                        | 22        |
| 2.3.1 Modelos Continuos . . . . .                      | 23        |
| 2.3.2 Modelos Discretos . . . . .                      | 24        |
| 2.4 Auto-organización en ME . . . . .                  | 26        |
| 2.4.1 Auto-organización en Modelos Continuos . . . . . | 27        |
| 2.4.2 Auto-organización en Modelos Discretos . . . . . | 31        |
| <b>3 Modelo Genérico del Miocardio</b>                 | <b>33</b> |
| 3.1 Introducción . . . . .                             | 33        |

|   |           |
|---|-----------|
| <i>CONTENIDO</i>                                  | 2         |
| 3.2 Actividad Eléctrica . . . . .                 | 34        |
| 3.2.1 Problema Mecánico . . . . .                 | 35        |
| 3.2.2 Problema Eléctrico . . . . .                | 35        |
| 3.2.3 Corrientes Iónicas . . . . .                | 36        |
| 3.3 Propagación de Ondas . . . . .                | 40        |
| 3.3.1 Propagación en el Tejido Cardíaco . . . . . | 41        |
| 3.3.2 Patrones en Superficies . . . . .           | 46        |
| 3.4 Modelización Discreta . . . . .               | 49        |
| <b>4 Simulación</b>                               | <b>52</b> |
| 4.1 Introducción . . . . .                        | 52        |
| 4.2 Autómata Celular . . . . .                    | 54        |
| 4.3 Juego de la Vida . . . . .                    | 55        |
| 4.4 Resultados . . . . .                          | 61        |
| <b>5 Conclusiones</b>                             | <b>63</b> |
| Apéndice  | 68        |
| Referencias                                       | 103       |

# Introducción

El presente trabajo tiene como objetivo explorar rasgos cualitativos en el comportamiento de un modelo computacional con estructura espacial. Este modelo ha sido motivado principalmente por la dinámica que presenta la actividad eléctrica en el tejido cardiaco en humanos. Si bien, esta dinámica ya ha sido ampliamente estudiada, aún no se tienen resultados contundentes y generales en cuanto a la naturaleza de los fenómenos involucrados. En este trabajo se describe el comportamiento de un parámetro físico de importancia básica. Este parámetro es la “*Entropía*”  $S$  o la “*Cantidad de información*”  $-S$ ; la cual indica la cantidad de bits necesarios para especificar un estado en un ensamble de posibles estados. Con ello, se pretende identificar de manera elemental la posible formación de patrones espaciales.

La distribución del contenido de este trabajo tiene la siguiente estructura. El primer capítulo aborda la formación de patrones como un fenómeno presente en una gran variedad de sistemas, para después identificar en cada uno de los casos, una propiedad en común. A esa propiedad en común se le conoce como auto-organización. Es inmediata la necesidad de encontrar rasgos de

auto-organización en otros sistemas que también presentan patrones.

Después, en el segundo capítulo se desarrolla un marco teórico adecuado para modelar sistemas de este tipo, en el que la modelización continua y discreta tiene diferencias metodológicas y enfoques de las cuales extraeremos ventajas. Por ejemplo, la modelización continua está determinada por la solución numérica de un sistema de ecuaciones diferenciales parciales. Por su parte la modelización discreta puede ser obtenida a partir de un modelo computacional simple.

En el tercer capítulo se presenta un panorama general del sistema que motiva básicamente nuestro estudio de la dinámica de un modelo que presenta un tipo particular de patrones espaciales. El fenómeno que nos motiva se presenta en una situación patológica conocida como arritmia cardiaca y consiste en la formación, desplazamiento, fragmentación y aniquilación de ondas espirales en la dinámica eléctrica del miocardio.

Por último se presenta, en el cuarto capítulo, el esquema general bajo el cual se realiza una simulación del fenómeno descrito, que estará basada en el código de un modelo de autómatas celulares simple (Juego de la Vida). Este autómata tiene como rasgo fundamental la simpleza que implican las reglas de transición entre estados.

De forma tal que el problema inicial consiste en identificar el conjunto de reglas de transición que nos permitan obtener las ondas espirales. Después

se presenta el problema de identificación de auto-organización que es abordado con la medición de la entropía y que por comparación con reglas de transición distintas y sin la presencia de patrones, nos ayuda a observar la auto-organización que está involucrada en la dinámica. Por último, se concluirá en el quinto capítulo.

# Capítulo 1

## Formación de Patrones

En este capítulo se expone el concepto de auto-organización. Se dan ejemplos de sistemas reales desde magnéticos hasta sistemas biológicos como lo es el de la amiba *Dictyosteleum discodeum*. Se termina el capítulo mencionando la utilidad del parámetro físico entropía y la definición de información. En este capítulo se siguen muy de cerca las referencias [1,2,3].

### 1.1 Introducción

Es deseable que el tratamiento analítico de la mayoría de los problemas tienda a su forma más simple. Desde los comienzos de la ciencia, éste ha sido un camino adecuado para obtener aproximaciones aceptables a los fenómenos naturales. Sin embargo, tanto en el caso de la física como en muchas otras áreas del conocimiento, la simplificación de los problemas, debido a la gran cantidad de restricciones técnicas necesarias para obtener aproximaciones

adecuadas, tiene un costo elevado. Esto implica restringirse al estudio de pocos sistemas simples, con la única libertad de estudiar sistemas más complicados a través del análisis de un conjunto de subsistemas. El punto aquí, no es despreciar la aproximación tan valiosa que da esta visión de las cosas. El punto es que algunos sistemas quedan fuera de este tratamiento, especialmente los que muestran comportamientos no lineales y con gran interacción con los alrededores. Valga decir, por cierto, que la mayoría de los fenómenos en la naturaleza son de este tipo. Ahora bien, el tratamiento de estos sistemas no puede, en principio, alejarse mucho del tratamiento de los sistemas simples por dos razones importantes. Primero, el lenguaje formal es similar; segundo, la aproximación obtenida por aquellos métodos en muchos de los casos, aporta información valiosa que no debe dejar de ser considerada.

De esta forma se llega a un lugar en donde se elige identificar en los sistemas no lineales y fuera del equilibrio (mecánico y/o termodinámico) rasgos comunes. En estos sistemas se encuentra cierto grado de organización. El carácter subjetivo que puede tomar el grado de organización tiene su origen en la diversidad de los sistemas que la presentan. Por citar algunos ejemplos tenemos que un sólido puede mostrar una estructura cristalina, amorfa o policristalina. Fluidos, bajo ciertas condiciones, muestran flujos con trayectorias bien definidas. Reacciones químicas tienen difusión y catalización con comportamiento prácticamente organizado. Sistemas más complicados, algunos de ellos, membranas que vibran y en general cualquier sistema oscilante mecánico, eléctrico y magnético, forzados. Sistemas biológicos, como órganos y tejidos en donde se distingue organización mecánica, química

y eléctrica. Sistemas biológicos complejos como especies animales o vegetales y sus agrupaciones (poblaciones); alcanzando el extremo en sistemas con estructura social. El rasgo común en el estado de cada uno de los ejemplos mencionados, es la organización en el tiempo o en el espacio.

## 1.2 Sistemas Naturales

En el caso de los sistemas físicos, es conveniente considerar la escala a la cual se analiza un sistema. ¿Cuál es la importancia del tamaño del sistema? La respuesta se encuentra más accesible en el siguiente ejemplo. Una fase gaseosa, vista microscópicamente, muestra un conjunto de interacciones entre vecinos fig [1.1]. Las interacciones muestran una baja correlación entre ellos. Sin embargo pasemos al caso en el que el número de vecinos aumenta; entonces la cantidad de parámetros aumenta irremediamente, de tal manera que no resultaría práctico observar el comportamiento entre vecinos. Más bien, se necesitaría observar algún parámetro, por ejemplo un parámetro de colisiones.

Una vez que se considera un conjunto de parámetros se observan cualidades del sistema que en lo individual no era posible determinar, al menos de forma directa. Y dentro de esas cualidades una que resulta por demás interesante es la de la auto-organización. Es decir, el sistema visto a través de sus componentes, individualmente, sólo permite hablar de interacciones locales. A diferencia de la observación global de algunos de sus parámetros, que permite

observar cualidades colectivas.

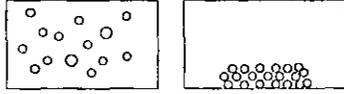


Figura 1.1: Fase gaseosa y solida, en donde se puede diferenciar una baja correlación y un arreglo con mayor orden estructural.

La dirección que tomaremos aquí no es la de determinar esos parámetros; en su lugar, tomaremos uno físicamente trascendente, la entropía en su definición estadística. Ahora bien, continuando con el sistema anterior, tenemos que si se manipula al sistema, se puede pasar de una situación en la cual los elementos constituyentes no guardan ninguna correlación entre ellos a una situación en la que la distancia promedio se reduce y la distancia está correlacionada, y que bien puede corresponder a la fase líquida.

Continuando con la manipulación sobre variables de estado, como la temperatura o volumen, se obtiene un estado para el sistema en el que se puede distinguir un arreglo estructural. Este arreglo estructural puede tener diferentes características. Puede estar formado por arreglos periódicos en rangos amplios, en tal caso lo identificamos como cristal. Si existen diferentes tipos de arreglos, periódicos, se trata con una estructura policristalina. Si el rango de los arreglos se reduce a rangos pequeños, sobre la vecindad de la molécula, entonces se trata de estructuras amorfas.

El problema anterior es un caso muy simple en el que se observa un cambio en la disposición espacial de los elementos constituyentes de un sistema a través de manipulación externa que trasciende, particularmente, en el orden espacial del sistema. Dentro de los sistemas en fase sólida un caso bien identificado de organización, se encuentra en los materiales ferromagnéticos. En estos sistemas se observan regiones bien definidas llamadas "Dominios Magnéticos" fig [1.2], donde se contiene un espín promedio que tiene una dirección y sentido particular. Los espines tienden a orientarse cuando el sistema es expuesto a un campo externo, si este campo sobrepasa un valor crítico, el sistema conservará esta nueva orientación y dará al sistema en general, propiedades macroscópicas importantes. Es decir, el sistema ya no asimila la magnetización externa. En cambio al disminuir la intensidad del campo externo el sistema no sigue la misma trayectoria con la que llegó. Y aún cuando el campo externo sea nulo, el sistema conservará cierta magnetización remanente. Éste es el conocido ciclo de Histéresis. Es en este sentido que dominios adyacentes pueden influenciarse mutuamente llegando a desmagnetizarse en un caso extremo. Otra característica de estos sistemas, y en particular de sistemas ferromagnéticos, es que existe una temperatura bien definida, llamada punto de Curie, por encima de la cual el comportamiento ferromagnéticos desaparece bruscamente, y el material pasa a comportarse como paramagnético.

Un fluido se comporta ordenadamente cuando se le aplican condiciones de frontera particulares. Cabe aclarar que el orden en este sistema se observa globalmente en el flujo. El fenómeno referido es conocido como celdas de

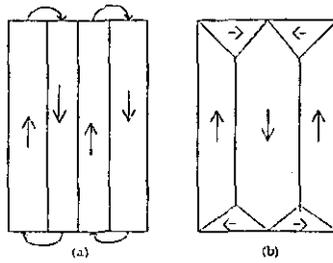


Figura 1.2: Dominios Magnéticos. En cada uno de los casos se observa el espín promedio. Los dos casos se diferencian en la cantidad e intensidad de flujo del campo fuera del material. El material con cuatro dominios tiene un flujo exterior mayor que el que material con siete dominios pues los cuatro triangulares hacen dentro del material lo que el primero tiene que hacer por fuera.

Bénard o de convección; el sistema consta únicamente de un fluido en un recipiente, al cual se le aplica calor en la pared inferior fig [1.3]. El calor provoca un gradiente de temperatura entre la superficie y el fondo del recipiente que contiene al fluido. De esta forma el calor es difundido por transporte convectivo. Cuando el gradiente sobrepasa un valor crítico, el sistema muestra movimiento macroscópico.

El movimiento consta de movimiento ascendente por flotabilidad; el flujo se enfría en el ascenso y después baja nuevamente. Cuando la temperatura se incrementa en la base, el gradiente aumenta más y aparece un nuevo tipo de organización fig [1.4]. Se observa entonces la aparición de una geometría con características hexagonales y cilíndricas, en donde los cilindros adyacentes giran en sentidos opuestos. Si el gradiente aumenta aún más y el flujo es cilíndrico, entonces los cilindros ondularán sobre su eje.

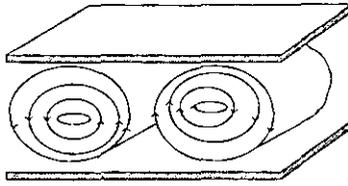


Figura 1.3: Vista cualitativa de convección, Celdas de Bénard. Las flechas indican el flujo convectivo.

El sistema anterior muestra claramente dos rasgos importantes. El primero es la dependencia del comportamiento global a las condiciones iniciales; el segundo rasgo es el de la auto-organización. Por auto-organización se piensa en un proceso en el que los elementos del sistema muestran, en conjunto, tendencia a un comportamiento particular, determinado únicamente por el estado anterior del sistema global y, a diferencia del comportamiento organizado, que depende implícitamente de la determinación de las reglas por parte de una directriz interna o externa, éste no requiere de esa restricción.

Aplicar una temperatura inicial, por ejemplo, similar a la que se presenta la ondulación de los cilindros arroja otro resultado. Esto nos conduce a la inestabilidad global del sistema y a la irreversibilidad de los procesos. La inestabilidad se piensa como una situación en la cual a un sistema que está en equilibrio o en un estado estacionario, fuera del equilibrio, por el contacto con el exterior, le resulta prácticamente imposible controlar cualquier variable de estado.

Respecto a la auto-organización en el fluido tenemos lo siguiente. El flujo

convectivo de algunas regiones, las que contienen un exceso de energía, siguen una trayectoria bien definida. Que es establecida sin la presencia de una directriz externa. Y además de que la trayectoria está bien definida, sus cualidades geométricas hacen resaltar este fenómeno de organización.

En otro sistema se conjuntan compuestos químicos que reaccionan, catalizan y autocatalizan dando lugar a diferentes regiones que soportan distintas reacciones. Algunas reacciones particulares muestran un comportamiento interesante que igualmente es interpretado como organización, tanto espacial como temporal. En el caso de que el sistema este aislado se obtiene, después de un tiempo transitorio, una razón de concentraciones con un valor constante llamada constante de equilibrio. En el caso de que el sistema no esté aislado, se presenta una reacción inversa. Ésta es la manifestación de autocatálisis, donde la presencia de un producto puede inhibir la formación de su propio producto, lo que es interpretado inmediatamente como un rasgo de auto-organización.

Para el caso de formación de patrones, en estos sistemas químicos, se observa la presencia de inhomogeneidades espaciales. Como resultado, se pueden formar ondas circulares o formas cilíndricas que presentan movimientos de traslación. Si el sistema pierde invariancia traslacional en una de sus direcciones, entonces se presenta un rompimiento de simetría. Un caso particular se observa en los sistemas llamados biestables. En concreto, dos o más estados estacionarios coexisten simultáneamente bajo las mismas condiciones de frontera.

En estos ejemplos, se ha hablado del concepto de organización. En particular, se busca identificar un rasgo por demás interesante, la llamada auto-organización, donde la directriz que organiza no existe de manera externa o centralizada sino que surge del sistema mismo. Esto se ha hecho de tres formas distintas. La primera como regularidad en el movimiento en el caso particular de oscilaciones. La siguiente es la de biestabilidad. La última forma de auto-organización se obtiene en la formación de patrones, donde el rompimiento de simetría es debido a la pérdida de invariancia traslacional. Por pérdida de invariancia traslacional, entendemos la no conservación del momento, en un sentido clásico.

Es claro que ni todos los sistemas en su fase gaseosa pueden ser llevados a configuraciones cristalinas y mucho menos todos tienen propiedades magnéticas. Tampoco se encuentra en cualquier fluido tal comportamiento convectivo y ni todas los compuestos químicos reaccionan de esta manera. El punto aquí es observar sistemas simples presentando comportamiento no trivial, el cual puede ser encontrado en sistemas más complejos, como lo son los biológicos.

Los sistemas biológicos inevitablemente se deben ver como sistemas cuyo estado se encuentra fuera del equilibrio y difícilmente de forma estacionaria. El flujo de energía y materia es un hecho que los distancia de la aproximación al aislamiento; estos flujos son básicamente identificados con la radiación solar y obtención de nutrientes. En su mínima escala, nivel celular, se

identifican inhomogeneidades importantes. Por ejemplo, existe una gran diferencia entre la concentración de algunos elementos dentro y fuera de las células. El estado del sistema está fuera del equilibrio, y esto origina procesos de conducción. El sostenimiento de tales procesos lo llevan a cabo reacciones bioenergéticas como la glucólisis y la respiración y el transporte activo. Un problema básico en el tratamiento de estos sistemas es el de identificar alguna consecuencia inducida por restricciones fuera del equilibrio sobre los rasgos del sistema biológico.

Los sistemas pueden ser desde los simples, como una bacteria, hasta sistemas multicelulares como tejidos y órganos. Un ejemplo muy ilustrativo es el de un sistema compuesto de amibas fig [1.5] que, a través de procesos transitorios, pasa de organismos unicelulares a un sólo organismo multicelular.

El comportamiento de este sistema se reduce a un fenómeno de agrupación que se lleva a cabo a partir de un conjunto inicial de amibas (*Dictyostelium discoideum*). El proceso se realiza en condiciones ambientales adecuadas, las cuales son equivalentes a los parámetros de control que restringen el estado que permanece fuera del equilibrio. El resultado, después de la agregación, es un cuerpo multicelular.

Este cuerpo se desplaza hacia una situación con condiciones ambientales más favorables, principalmente nutrientes. Después por diferenciación el sistema queda formado por dos tipos de células. Una que constituye una parte equivalente a un tallo y la otra que soporta la formación de esporas, las cuales se esparcen y en condiciones específicas inician un nuevo ciclo.

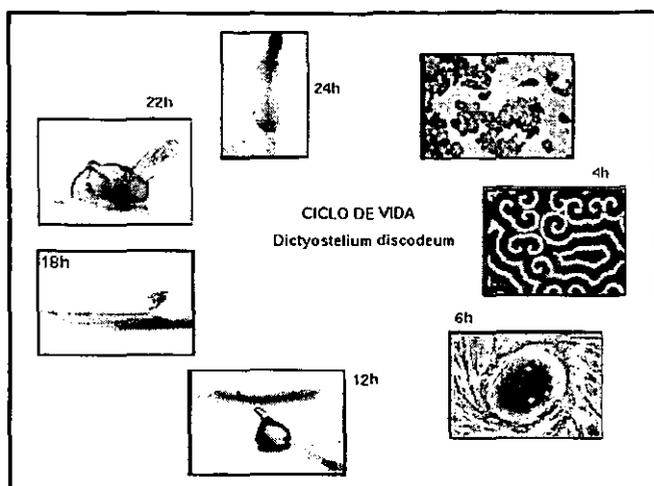


Figura 1.4: Aquí se muestra el ciclo de vida y dentro de él, la agrupación de amibas, *Dictyostelium discoideum*.

El rasgo básico, la organización, se observa cuando la agrupación sufre la agregación y termina cambiando su configuración inicial a una donde el movimiento de amibas se realiza en regiones con mayor concentración de algunas sustancias sintetizadas. Además, la aceleración de este proceso está ligada a la amplificación de las señales químicas. Dando como resultado formas regulares como espirales, las cuales determinan trayectorias que siguen algunas amebas. Es claro que, desde que comienza la agregación y se realiza la síntesis de sustancias químicas la invariancia traslacional se pierde, obteniendo un nuevo ejemplo de rompimiento de simetría y como consecuencia la formación de patrones.

### 1.3 Auto-organización, un caso particular

Un rasgo importante de en los sistemas antes mencionados es la irreversibilidad de estados. Los sistemas anteriores son sistemas abiertos que intercambian energía con los alrededores lo cual provoca que no podamos escribir de ninguna manera, un enunciado de conservación. Entonces, el problema se puede pensar en el siguiente sentido. El intercambio de energía entre el sistema y los alrededores tiene como consecuencia no poder disponer de energía ya transformada, por ejemplo, en calor. Además podemos pensar en términos de la configuración del sistema. Lo que nos deja con el hecho de que cada una de las configuraciones tiene una única descendiente, pero no necesariamente una predecesora [4]. Y según se puede entender, un sistema reversible tiene un número constante de posibles configuraciones, a diferencia de un sistema irreversible donde el número de posibles configuraciones puede cambiar con el tiempo. Y es en este sentido donde la auto-organización puede tomarse de una forma accesible. Si la cantidad de posibles configuraciones disminuye con el tiempo, se llegará a una situación en la que sólo podamos presenciar alguna que otra configuración, y el llegar a esta situación lo podemos entender como auto-organización.

La entropía del sistema puede ser un vehículo adecuado para monitorear el comportamiento del sistema. Y en particular con la definición estadística donde la entropía es el logaritmo del número promedio de estados posibles del sistema

$$S = - \sum_i p_i \log p_i \quad (1.1)$$

$p_i$  es la probabilidad para el estado  $i$ . La entropía puede ser considerada equivalentemente como un número promedio de bits binarios necesarios para especificar un estado en un ensamble de posibles estados. La entropía total del sistema es la suma de las entropías de subsistemas estadísticamente independientes. La entropía es típicamente maximizada cuando un sistema es completamente desorganizado, y el máximo número de subsistemas actúan independientemente.

Cabe aclarar que la entropía no es en si misma la entropía común definida por Boltzmann. Es más bien la definición que fué aportada por Shannon. Quien a partir de la teoría de la información llego a una expresión idéntica a la de Boltzmann para el caso de tener la constante  $K$  igual a 1 ( $S = -K \sum_i p_i \log p_i$ ). Sin embargo el uso de esta definición de medida de información es indistinguible de la definición de entropía para el resto de nuestro análisis. Entonces es parte de esta ambivalencia que los conceptos de entropía y el de la medida de información, compartan ciertas descripciones. Es decir, la interpretación de la medida de información se comparte con la de  $H$  para ciertos sistemas. Por ejemplo, sistemas con alta entropía se asocian con:

- i) Gran cantidad de energía disponible para realizar trabajo
- ii) Desorden
- iii) Igual probabilidad de eventos, baja probabilidad de un evento particular

- iv) Distribución uniforme
- v) Mayor incertidumbre
- vi) Aleatoriedad
- vii) Libertad(gran variedad) de elegir, muchas salidas posibles
- viii) Amplia diversidad
- ix) Mucha información
- x) Gran cantidad de información usada para especificar el estado de un sistema

La primera asociada a sistemas termodinámicos. La segunda a sistemas físicos y a su configuración en estados que pierden energía. La tercera asociada al movimiento de partículas derivada de equilibrio. La cuarta se basa en una distribución uniforme de datos. Tanto la quinta como la sexta, en desorden total, probabilidades idénticas y una distribución uniforme. La séptima se contempla cuando existen un gran número de estados, entonces si elegimos una estado en particular, no existirá desorden ni incertidumbre. Probabilidad mayor,  $p = 1.0$ , entonces, entropía cero. En la octava muchas salidas posibles sugieren diversidad. Para la novena se dice que dado un ensamble o distribución de probabilidad, representamos un valor de información de muchos bits. Un gran número de bits significan una gran cantidad de información. La décima interpreta la entropía como la cantidad de información necesaria para especificar el estado del sistema.

# Capítulo 2

## Medios Excitables

Es en este capítulo donde se expone la excitabilidad de un sistema. Con una formulación continua y discreta. Para después concluir con una visión de auto-organización dentro de la evolución de un sistema que presenta ondas espirales.

### 2.1 Introducción

En los fenómenos identificados como auto-organizados, se presenta una característica fundamental y útil para simplificar el problema del planteamiento y dirigirlo a un tratamiento factible de solución. El rasgo es la capacidad que muestra cada sistema para reaccionar a perturbaciones sobre cierto límite, y a través de un proceso aleatorio, elegir un estado de entre varios posibles. Esta característica se debe al comportamiento individual de cada elemento, que reacciona a las condiciones del medio. La reacción a un estímulo es referida

comúnmente como excitabilidad y los sistemas en los que es identificada, se conocen como medios excitables (ME).

## 2.2 Excitabilidad en Fenómenos Naturales

Dentro de los medios excitables reales algunos son bien conocidos y citaremos algunos ejemplos. Cabe aclarar que la diferencia entre cada uno de estos es a nivel de su representación material, pues como se aclaró antes, procuramos encontrar los rasgos cualitativos comunes de los sistemas con comportamiento complejo.

Tomemos un primer ejemplo. Un Bosque como sistema puede mostrar excitabilidad respecto a un estímulo, como el fuego. En donde se puede identificar un estado inicial, el estado correspondiente a un árbol que pueda quemarse. Después, bajo la acción del estímulo, se puede definir otro estado, el árbol en ignición; seguido por un estado en donde no hay árbol que quemar. El sistema podrá tener algunas variaciones pero el proceso requerirá de la difusión del fuego entre árboles, y en particular, de vecinos cercanos. Entonces un bosque puede considerarse como un medio excitable, bajo el estímulo del fuego; y donde se presentan al menos tres estados. El primero, corresponde a un estado en reposo; el segundo a un estado excitado; el tercero, al estado refractario, el cual permanecerá hasta que un nuevo árbol ocupe ese lugar, y obtenga la capacidad de combustión. Es inmediato reconocer que el estado de cada árbol depende del estado de los vecinos cercanos.

Como siguiente ejemplo se puede considerar un sistema mecánico. Este sistema puede estar compuesto por un arreglo unidimensional de péndulos de torsión acoplados [14]. El estímulo es únicamente mecánico (forzamiento). Ahora bien, al forzar a uno de los péndulos al movimiento, éste ejecutará el estímulo sobre el péndulo inmediato. Sin embargo, la tensión ejercida por el primer elemento necesitará superar un umbral en la tensión. Con lo cual se provocará el movimiento del vecino, el cual sigue siendo estimulado a través de el vecino anterior, primer elemento, hasta poder sobrepasar la tensión mínima y cambiar el estado del siguiente elemento. El proceso es largo, pero el sistema resulta ser muy ilustrativo, por la simplicidad de sus elementos y de el gran detalle que involucra en la transferencia de energía, lo que muestra fundamentalmente el comportamiento de un sistema abierto. Y por demás, el concepto básico que soporta al medio excitable. Esto es, un estímulo aplicado, donde el sistema en reposo responde con algún estado ya sea, excitado, refractario, y con la transferencia de esa información, en su interior.

## 2.3 Formulación de ME

El uso de sistemas dinámicos permite encontrar bajo ciertas condiciones y para sistemas particulares una solución a la ecuación de evolución del sistema que presenta formación de patrones. La ecuación de evolución para cada variable macroscópica es una ecuación que tiene la forma (2.1). Después (2.2) muestra una forma diferente para la evolución de la misma variable, bajo una

nueva idea, la de involucrar en la evolución a constricciones externas.

$$\frac{d\chi_i}{dt} = F(\{\chi_i\}, \bar{r}, t) \quad (2.1)$$

$$\frac{d\chi_i}{dt} = F(\{\chi_j\}, \lambda) \quad (2.2)$$

para (2.2) tenemos a  $F$  como una función que depende del resto de variables ( $\chi_j$ ) y de parámetros de control ( $\lambda$ ). Los parámetros de control son parámetros que pueden ser determinados en el exterior del sistema, y como lo muestra la ecuación de evolución, determinan el comportamiento de éste. Considerando que las variables de estado pueden comportarse como campos, en el sentido de que dependen continuamente de las coordenadas del sistema, se presenta acoplamiento de diferentes variables. La función  $F$  puede tener elementos diferenciales como laplacianos, en el caso de temperaturas y concentraciones y, gradiente en la velocidad.

### 2.3.1 Modelos Continuos

De este modo obtenemos la representación de un sistema continuo y extendido que sufre de fluctuaciones pero que es controlado por parámetros que permiten soluciones estables, con la posibilidad de linealización. Entonces, considerando variables macroscópicas y pensando que también el procedimiento es aplicable en el caso de variables microscópicas, se denotan cantidades físicas  $U = (U_1, U_2, \dots)$ .  $U$  puede ser la velocidad, densidad, temperatura, en el caso de hidrodinámica; en química, densidad de moléculas, concentración. En general,  $U$  obedece una ecuación, conocida como ecuación de difusión, y

tiene la forma

$$\frac{\partial U_\mu}{\partial t} = G_\mu(\nabla, U) + D_\mu \nabla^2 U_\mu + F_\mu(t) \quad (2.3)$$

Donde  $G$  es una función no lineal de  $U$  y su gradiente.  $D$  se refiere a difusión, en caso de ser real, si por el contrario es imaginario, se refiere al tipo de onda que se propaga.  $F$  son fluctuaciones causadas por causas externas y disipación;  $\mu$  es el índice que una de las variables del conjunto  $U$ . El objetivo de este planteamiento es el de obtener, a partir de la ecuación (2.2), soluciones estables y de ahí determinar la dinámica del sistema alrededor de un punto de inestabilidad.

### 2.3.2 Modelos Discretos

Volviendo a los rasgos fenomenológicos, tenemos que en los sistemas de amibas descrito anteriormente que encontrarán condiciones favorables, no sufrieron el proceso de agregación. Las que no, se agregaron para procurarse, por si mismas, las condiciones adecuadas para reproducción. Así, podemos suponer que en los elementos de los sistemas existe un estado, el cual distingue dos clases de configuraciones del sistema global. A este límite entre estados lo identificamos como umbral.

Entonces, considerar al conjunto completo de variables de estado, complica innecesariamente al sistema, como consecuencia usar una forma discreta y no una continua es preferible, analítica y computacionalmente en ocasiones para los sistemas dinámicos discretos, obtenemos una expresión para la ecuación

de evolución con la forma

$$\chi_{t+1} = F(\chi_t) \quad (2.4)$$

Para esta ecuación es más sencillo definir el estado de umbral al cual el sistema global presenta un cambio dada la función  $F$  (la forma depende de cada problema). Así, esta función determina en que clase de estados se encuentra la variable  $\chi$  por lo que se infiere que ella también presenta un valor crítico que es equivalente al umbral de estado. Tales estados no se prolongan, al contrario, corresponden a la discretización del tiempo, por lo que el estado que se presenta en la ecuación (2.3) es equivalente a un punto fijo para un sistema dinámico continuo. Ahora bien, considerando que la variable de estado sólo puede pertenecer a uno de las clases de estados existentes, es conveniente que esta variable sea representada en términos lógicos. Esta lógica permite alcanzar la máxima simpleza del sistema; cuando la función sea distinta de cero, la variable también lo será; en caso contrario, también coincidirá. De esta forma se obtiene un modelo con la habilidad de asimilar transiciones entre estados estacionarios. Este modelo puede extenderse a un arreglo cuadrangular o cúbico, cuyas celdas representan los elementos del sistema original y que a su vez juegan el papel de componentes lógicos con interacciones locales fig [2.1]. Nos referimos a este sistema como un Autómata Celular (AC).

Por ejemplo, un AC unidimensional se escribe de la forma:

$$\chi_{i,t+1} = F_i(\chi_{i-r,t}, \chi_{i-r+1,t}, \dots, \chi_{i+r,t}) \quad (2.5)$$

Esta ecuación indica que la variable  $\chi$  del sitio  $i$ , depende de los estados de

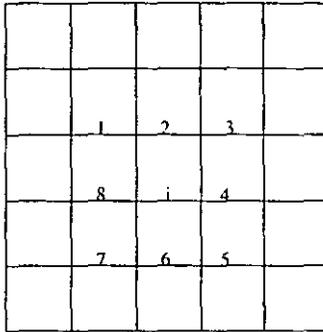


Figura 2.1: Arreglo cuadrículado que muestra elementos que interactúan (vecinos cercanos) alrededor del elemento  $i$

los  $2r+1$  vecinos en el momento anterior.

## 2.4 Auto-organización en ME

Dentro de las propiedades de los fenómenos que presentan auto-organización se observan rasgos oscilatorios. Pero, ¿cómo vincular propiedades oscilatorias con organización y sistemas dinámicos?. Bien, dentro del tratamiento analítico de los medios que soportan los fenómenos de auto-organización, se vincula la organización espacial con la propagación de ondas y la organización temporal con el establecimiento de uno de los estados disponibles por el medio, como lo es excitabilidad, reposo y refractariedad. En concreto, dentro de los dos tipos de modelo, ya mencionados, se observa la auto-organización espacial a través de la propagación de ondas, generalmente por medio de una ecuación de difusión, en un medio continuo. En un modelo

discreto la propagación está determinada por una función de transición.

Identificar el vínculo entre el concepto de auto-organización y oscilaciones permite dar forma al problema de formación de patrones en un medio excitable, al que podemos definir como un medio espacialmente extendido que soporta la propagación de ondas. Entonces, es recomendable abordar de una manera esquemática la evolución ondulatoria sobre el medio, y de ahí identificar los rasgos más relevantes, dinámicos y geométricos, con que cuentan los ME.

El esquema toma en cuenta la formación de un impulso ondulatorio provocado por un forzamiento externo; inmediatamente después se cuenta con posibles geometrías del frente de onda, frentes planos, circulares y ondas espirales. A partir del frente de onda formado se debe tomar en cuenta la propagación en un medio homogéneo y generalizar con la aplicación de inhomogeneidades. Y por último, cual es el fin de la evolución de los frentes en propagación, convergencia, divergencia, refracción, fragmentación o aniquilación.

### 2.4.1 Auto-organización en Modelos Continuos

Comenzando con la formación de ondas, se plantea un problema fundamental que es la formación de patrones, ya mencionado en el primer capítulo. Aquí se mencionarán algunos aspectos dinámicos.

Entonces, para la formación de las ondas sólo nos referiremos a resultados publicados cuyo objetivo principal es el de describir la propagación de

ondas en ME a través de parámetros bien definidos. Se comenzará con el efecto que provoca la fuente de perturbaciones. Entonces, el efecto que tiene lo equivalente a una fuente en un sistema clásico pero en un ME, fuerza externa, que cuenta con parámetros como periodo y amplitud no es necesariamente asimilada por un ME. Esto es, un impulso en el medio tendrá parámetros distintos. Un buen ejemplo de la relación que pueden guardar ambos parámetros, internos y externos, se encuentra en la referencia [7]. En donde los parámetros ( $\nu$  y  $A$ ) son relacionados a través de cuatro relaciones estadísticas que involucran la intensidad de ruido externo y el periodo del forzamiento. Estas relaciones estadísticas no son relevantes en nuestro objetivo pero de ellas se infiere que se puede optimizar el número de impulsos con el número de ciclos de forzamiento, esto quiere decir que se puede saber si es que la "fuente" producirá algún efecto en el medio.

Dentro del proceso de selección del frente de onda [8] se plantean diferentes problemas de interés, ya que sólo dos estructuras topológicas son comunes en un ME, circulares y espirales. Ahora bien, después de ubicar el efecto que tiene un posible forzamiento en la excitación del medio, se pasa inmediatamente a la cuestión, esta forma de excitación provoca los dos frentes distintos. La respuesta es negativa y se debe a que el forzamiento externo se ve desde adentro del sistema de dos formas diferentes: Como un conjunto de inhomogeneidades distribuidas aleatoriamente, cada una con periodos diferentes, y como una inestabilidad local.

Como resultado de tal distribución y de la diferencia de periodos se obtiene

un conjunto de trenes de onda circulares cuyo periodo varía de ciclo a ciclo. Entonces, se forma sólo un tipo de frente, lo cual permite esperar que la estructura espiral sea el resultado de algún proceso en donde intervienen solamente frentes circulares y parámetros propios del medio. En cierta forma podemos predecir el fin que tienen los frentes circulares, éste no podría ser otro que la aniquilación por colisión, consecuencia directa de la no linealidad del medio y de las dimensiones finitas de un ME, y de un sistema real.

Sin embargo, antes de que los frentes resulten ser aniquilados existe una tendencia asintótica a un estado coherente. Por su parte, el problema que presentan las espirales es el de su formación. Sobre la formación de espirales se plantea un problema de carácter general, y es el de la fragmentación de los frentes de onda, que dan lugar a la formación de las espirales. El mecanismo que se presenta en el proceso de fragmentación de frentes de onda está relacionado con la existencia de bifurcaciones [11]. Este mecanismo relaciona el valor crítico de un parámetro de control con la fragmentación del frente, este valor crítico no coincide con el punto de bifurcación, pero este valor se acerca demasiado, es ahí, donde las inestabilidades, por más pequeñas que sean, son significativas. La naturaleza de la perturbación no es fundamental en este enfoque. Sin embargo, la perturbación puede ser externa por ejemplo, un campo aplicado, o interna, variación en la curvatura del frente. En ambos casos la inestabilidad se reduce al análisis de inestabilidades transversales y longitudinales.

Es claro que la presencia de inhomogeneidades en el medio puede provocar la

perturbación interna a la que nos referimos anteriormente. En la formación de las ondas espirales en particular, son otros los parámetros de la onda que resultan ser relevantes. Por ejemplo, la longitud de onda. Esta longitud que tiene la franja en estado excitado, es fundamental para asimilar la perturbación. Detallando este proceso, se puede considerar al frente al localizar el obstáculo (homogeneidad) de modo que cuando la vecindad del obstáculo sea rodeado por el frente excitado, éste tendrá un retraso en los extremos. Y si la longitud de onda es lo suficientemente grande, la primera línea de excitación coincidirá, o se retrasará, respecto a la última línea del frente de excitación. Éste es un mecanismo que permite la formación de ondas espirales, pero es claro que la geometría y dimensiones del obstáculo resultan determinantes en el mecanismo.

Esto esboza la formación de ondas espirales simples, pero esto no explica que algunas ondas espirales presenten más de un brazo. Bien, características como el número de brazos, tienen una dependencia importante con la dinámica del frente, en [10] se ha aclarado este punto. Específicamente, el número de brazos es proporcional al radio de la espiral y a la refractariedad del medio. Reduciendo la excitabilidad se obtiene un incremento en el periodo, el cual es proporcional, indirectamente, al número de brazos. La variación de la excitabilidad se obtiene con la variación de parámetros de control. Por su parte la formación se entiende como un proceso de colisión en el que se unen espirales con el mismo sentido de rotación; mientras que espirales con sentido opuesto se aniquilan. Como aspecto final de propiedades dinámicas de frentes de onda se considera la velocidad de

propagación, la cual puede ser una función de parámetros de la fuente [9]. Estos parámetros son valores de la derivada temporal de la fuente considerando los bordes y el valor del umbral, en el sistema.

Esta formulación apunta hacia dos aspectos importantes de los ME. La interacción entre variables de excitación y recuperación, y mecanismos de difusión. Desafortunadamente el mecanismo preciso de formación permanece abierto [12], se puede considerar que el modelo continuo tiende a ocultar mecanismos simples de interacción local, los cuales dan razón a la formación de frentes de onda. Es este punto, donde el modelo discreto permite un mejor tratamiento, utilizando el concepto de organización a través de una función de transición que aplica simultáneamente en todos los elementos de la red discreta, una configuración, consecuencia de la información contenida un momento anterior.

### 2.4.2 Auto-organización en Modelos Discretos

Al igual que en el esquema ondulatorio seguido en el modelo continuo, aquí, un aspecto de gran importancia se ubicó en el proceso de transición de estados, excitado, reposo y refractario [12]. El proceso es dirigido por la función de transición ( $F$ ) formada en general, por la composición de la función de conexión y la de umbral; donde se aplica la configuración de la vecindad y un rango en algunos parámetros, respectivamente. La dinámica está determinada por la aplicación sincronizada de la  $F$  local en todos los lugares del espacio.

Una forma de ver la formación de ondas en un modelo discreto es considerando que en ciertas regiones activas se crean dominios ordenados, esto es, con un estado particular, si estos dominios crecen, la interfase provee la posibilidad de formar estructuras espirales simples. De cualquier modo, la formación de ondas en estas interfaces simplemente es una adaptación metodológica que nos conduce a un sistema que contiene un conjunto de ondas coexistentes, espirales y circulares. Lo que resalta es el mecanismo sincronizado que hace que los elementos cooperen y juntos presenten comportamiento colectivo.

# Capítulo 3

## Modelo Genérico del Miocardio

Para este tercer capítulo se muestra como, para un sistema biológico, se busca responder el comportamiento de una patología, a través de la descripción del mismo por medio de conceptos físicos y un lenguaje matemático. El concepto físico más importante es el de auto-organización y el lenguaje matemático es el de los sistemas dinámicos. Esto dentro de una pequeña parte de un esquema descriptivo del sistema completo, el corazón. En otras palabras, la auto-organización en la actividad eléctrica del corazón.

### 3.1 Introducción

El corazón como órgano y sistema biológico en humanos, presenta una serie de comportamientos patológicos. Uno de ellos consiste en el cambio del ritmo cardiaco, el cual desencadena una compleja variedad de ritmos que van desde una forma ordenada hasta una completamente impredecible. Dos

de estos ritmos están identificados como Taquicardia y Fibrilación, para ambos, la caracterización resulta particularmente importante. Existe una secuencia en la presencia de los ritmos; un ritmo normal puede pasar a ser una taquicardia, la cual, a su vez, puede llegar a ser fibrilación. En el caso de la taquicardia, el corazón late rápidamente, con un aparente traslape de actividad entre un latido y otro. En la Fibrilación diferentes áreas del corazón se contraen fuera de sincronía y resulta prácticamente imposible distinguir entre latidos "beats".

Entendemos por fibrilación la contracción de las paredes auriculares de manera rápida, descoordinada e ineficaz. Y por Taquicardia a la aparición de pulsos súbitos.

## 3.2 Actividad Eléctrica

Poder obtener una descripción matemática de los fenómenos mencionados arriba puede parecer simple. Sin embargo, este sistema, como la gran mayoría de los sistemas reales, no es tratable directamente a través de reducciones lineales. Para entender el sentido de esta limitación, se ubicará en el contexto de una parte de los fenómenos que presenta el corazón y que son de dos tipos principales: Mecánicos y Eléctricos.

### 3.2.1 Problema Mecánico

En el caso mecánico se presentan tres tipos de problemas. El primero trata el tejido en un estado estático; el segundo considera la contracción y relajación; el tercero, considera la estructura espacialmente. Por ejemplo, el primer problema incluye la descripción de la estructura muscular (microscópicamente). Además de que busca una liga entre esta microestructura y la función del órgano como un todo. En el segundo, se incluyen metodologías analíticas y experimentales para identificar propiedades que sean dependientes del tiempo, estudios de elasticidad, y modelizaciones computacionales. El tercero y último incluye análisis de esfuerzos y elasticidad. Ambos análisis son espaciales, y se aplican a patologías particulares. Además de considerar las consecuencias funcionales de las heterogeneidades que presenta el tejido.

### 3.2.2 Problema Eléctrico

En lo que respecta al problema eléctrico, se cuenta principalmente con la modelización de las corriente iónicas presentes en el tejido. En segundo lugar, están los aspectos fisiológicos; por último, la simplificación de los modelos ya existentes con propósitos de eficiencia computacional y de mayor realismo.

Se pueden considerar algunos ejemplos en cada caso. Inicialmente se encuentran sistemas de ecuaciones no lineales que describen la variedad de corrientes iónicas, donde algunos de ellos consideran fuertemente la experimentación. En el segundo caso, se encuentra la consideración fisiológica del proceso de excitación y la propagación de ondas. En tercer lugar, se

introducen modelos con sistemas dinámicos discretos, autómatas celulares; además de la consideración de patologías particulares; y también procesos de supresión de algunos fenómenos también particulares, como la desfibrilación (cardioversión eléctrica). La cual se entiende como el proceso en el que, por medio de un pulso, se cambia la frecuencia del sistema restaurando el ritmo normal.

### 3.2.3 Corrientes Iónicas

Dado el tipo de problemas que involucra el comportamiento eléctrico, nos detendremos brevemente en la fenomenología de la membrana celular y corrientes iónicas. Entonces, la membrana celular cumple con funciones como la formación de compartimentos, la regulación del movimiento de sustancias, la transferencia de información, la interacción intercelular, además de ser un lugar para las funciones bioquímicas y transferencia de energía. Y por supuesto sirve como una barrera entre el exterior e interior de la célula que mantiene dentro de ella Potasio( $K_+$ ), Magnesio( $Mg^{++}$ ) y Fosfato( $PO_3$ ), y fuera Sodio( $Na^+$ ), Cloruro( $Cl^-$ ), Bicarbonato( $HCO_3$ ), Oxígeno( $O_2$ ), Glucosa, ácidos grasos y Aminoácidos. En todas las células existe una diferencia entre su concentración iónica intracelular y externa, así como un potencial eléctrico a través de la membrana. Ambos se encuentran íntimamente relacionados, ya que el potencial eléctrico depende de la distribución desigual de los iones dentro y fuera de la célula.

La modelización de la membrana celular está basada en la idea de una

bicapa de lípidos y proteínas que flotan en ella libremente. De estas proteínas hay algunas que atraviesan completamente la membrana y es a través de ellas que se tiene una vía de comunicación entre el exterior e interior. Estas proteínas forman los conocidos canales iónicos por donde pasan los iones de sodio, potasio y cloro. El transporte de iones a través de la membrana celular se presenta principalmente en dos procesos: La difusión y el transporte activo.

La difusión es el movimiento continuo y al azar, de las moléculas de las sustancias. Dentro de esta forma de transporte se pueden identificar dos diferencias, una simple y otra facilitada. La simple es un tipo de difusión en donde no interviene ningún tipo de fuerza externa para dirigir el movimiento de las moléculas. De modo que la difusión se presenta sólo por la diferencia en las concentraciones entre el exterior e interior. La difusión facilitada consiste en el transporte de sustancias que por si solas no pueden pasar a través de la membrana, pero que con la ayuda de la combinación de un portador se pueden difundir.

Por otra parte, el transporte activo es el movimiento de sustancias producido por procesos químicos, glucólisis y respiración, que proporcionan la energía necesaria para originar el movimiento, el cual depende de procesos enzimáticos ocurridos en la membrana. Durante el proceso del transporte activo se desplazan moléculas contra un gradiente de concentración, de presión o eléctrico. Entre las diversas sustancias que son transportadas activamente se encuentran iones de sodio, potasio, calcio, fierro, hidrógeno,

cloruro, yoduro, urato, azúcares y aminoácidos. Pero dentro del transporte de todas estas sustancias el que resulta más importante es el de iones de sodio y potasio que recibe un nombre particular, bomba de sodio. Este mecanismo permite mantener los gradientes de concentración a pesar de que existe movimiento neto de sodio hacia el interior de la célula y, de potasio hacia la parte externa de la misma.

El potencial de la membrana es el potencial entre ella debido a la distribución asimétrica de las cargas eléctricas, la cual a su vez es el resultado de la diferencia de concentración iónica determinada por el transporte activo. En general se acumula un exceso de iones negativos dentro de la membrana celular, y un número igual de iones positivos fuera de ella. El resultado es el desarrollo de un potencial de membrana y cuando el flujo de iones es cero, la membrana solo se considera polarizada.

Se puede considerar que el potencial se desarrolla a través del transporte activo, mediante la bomba de sodio potasio con difusión de iones a través de la membrana como consecuencia de la diferencia en la concentración de iones dentro y fuera, y los aniones que se encuentran dentro y colaboran en la diferencia de potencial. La analogía entre la membrana y un condensador es bien conocida.

Ahora bien, la propagación del potencial de acción se entenderá a partir del comportamiento que tenga el mismo potencial pero en una célula nerviosa. Éste se relaciona directamente con la resistencia que se oponga transversalmente o longitudinalmente en ella. Si la resistencia longitudinal

es poca, entonces la tendencia a que el potencial se traslade a lo largo de la membrana será mayor. Este comportamiento podría significar una atenuación a lo largo de la longitud de la membrana de forma exponencial. Y lo que pasa en realidad es que cada vez que el potencial pasa por un sector de la membrana el potencial se amplifica. Es intuitivamente claro que este comportamiento es no lineal y de forma inmediata pasamos a que en la generación de rotores, conducción en bloque, sistemas de osciladores, etcétera, se usan sistemas de ecuaciones no lineales. Como consecuencia de esto, es posible obtener el comportamiento al que modelos lineales no accederían.

Entre este conjunto de comportamientos está, como más relevante, la excitabilidad. Sin embargo, a diferencia de lo que se podría esperar. El aumento en detalles de un sistema, nos aleja de la descripción deseada. Esto en el sentido de que el sistema de ecuaciones se complica enormemente. Para este tipo de problemas se ha sugerido como una alternativa, simplificar los modelos, a partir de la disminución de variables. Aún así, el modelo continua conservando limitaciones en su precisión, sencillamente porque la modelización continua no corresponde con los rasgos discretos en el cambio de estado de los elementos básicos, como lo es la clausura y abertura de los canales iónicos. Todo esto debido a que se buscó inicialmente la evolución de la corriente iónica, en lugar de incidir directamente en comportamiento de estados. Así, terminamos hablando de modelos con una actividad promedio de un número infinito de células.

### 3.3 Propagación de Ondas

De acuerdo con la idea de relacionar la auto-organización y medios excitables a través de la formación, y propagación de ondas. Será necesario considerar la dinámica ondulatoria. La dinámica ondulatoria en los medios excitables resulta sumamente atractiva. En particular por sus propiedades no lineales, las cuales soportan una gran variedad de fenómenos, como por ejemplo la existencia de solitones. Para nuestro caso, la propagación se referirá a la propagación de información que, entre célula y célula, activa secuencialmente a los elementos restantes a lo largo del medio (tejido muscular). Inicialmente la descripción para la propagación de ondas se obtiene con un modelo simple pero no lineal, a través de una ecuación conocida como la ecuación diferencial del cable eléctrico.

Alternativamente se reduce el problema a uno con menos grados de libertad. En este nuevo modelo se pueden encontrar diferentes versiones, y que en general es conocido como un sistema de ecuaciones de reacción-difusión.

Ahora bien, para considerar la propagación del pulso eléctrico en el medio no lineal, anisotropico y complejo, se considera como objetivo, encontrar las propiedades de la misma propagación, y de la distribución espacial de las células [27] Para lo cual, comenzar con el caso unidimensional resulta lo más conveniente. En este caso, la propagación unidimensional muestra una gran cantidad de cualidades dentro de las cuales se puede encontrar, dependencia de velocidades con resistencias y capacitancias. También una relación entre geometría (longitud) y velocidades. Y además, dependencia de la amplitud del potencial de acción con la velocidad. Por ejemplo, en caso de que la propa-

gación se realice en una superficie, la velocidad de propagación resulta ser distinta en direcciones ortogonales (propagaciones unidimensionales ortogonales) Además de que la propagación de ondas muestra un comportamiento particular, la formación de patrones. Este comportamiento colectivo y su manifestación en la forma de espirales, es el que nos interesa.

### 3.3.1 Propagación en el Tejido Cardíaco

Al considerar más arriba algunas de las propiedades de los sistemas que soportan la propagación de ondas. Estamos considerando un esquema útil para describir la propagación de un pulso unidimensional. Y cuya interpretación fué concebida en el comportamiento eléctrico de un axón. Pero, ¿que pasa con la propagación en el tejido cardíaco?. Para acceder a este problema se requiere de una adecuada simplificación, según se entiendan los rasgos fundamentales del comportamiento. Esto se refiere a que, la propagación al ser soportada por la existencia de las corrientes iónicas, la polarización de la membrana, y la conductividad entre el espacio intracelular, entre otras cosas. Podrá ser analizada en base a algunos de estos elementos. Por ejemplo. En este medio, tejido cardíaco, se identifica una propiedad interesante. La excitabilidad que consiste en lo siguiente.

Cuando las células nerviosas son excitadas el potencial de acción sufre una alteración rápida, la cual dura una fracción de segundo (aprox  $\frac{1}{1000}$  seg) para regresar de inmediato a su valor en el reposo. La forma que presenta el potencial en el tejido cardíaco es de una meseta fig [3.1].

Los mecanismos que desencadenan el potencial de acción son: estimulación

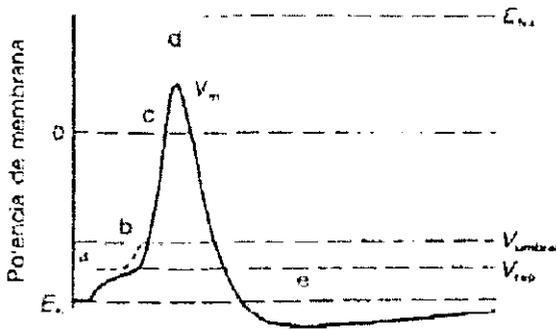


Figura 3.1: Potencial de acción tipo meseta característico del músculo cardíaco. a: Debido al retraso en la disminución de la permeabilidad del  $Na^+$ ; b: Por las corrientes de  $Ca^{++}$  hacia el interior; c: Disminuye la permeabilidad al  $K^+$

eléctrica de la membrana, aplicación de productos químicos, lesión mecánica y cambios térmicos. Todos ellos producen un cambio transitorio en las permeabilidades de la membrana. El potencial de acción se presenta en dos etapas: Despolarización y Repolarización.

**Despolarización** . Cuando es excitada la célula, la permeabilidad de la membrana para los iones de Sodio aumenta bruscamente por lo que muchos estos iones penetran al interior de la fibra y así causan la desaparición del potencial normal en reposo.

**Repolarización** . Casi inmediatamente después de producida la despolarización, se presenta nuevamente la impermeabilidad y la permeabilidad a los iones de Potasio aumenta bruscamente. Esto provoca el movimiento de cargas positivas desde el interior de la membrana hacia

el exterior, lo que restaura la carga negativa en el interior de la membrana. En consecuencia se restablece el potencial normal de membrana en reposo.

Ahora bien, la despolarización parcial de una membrana excitable inicia un potencial de acción, sólo cuando la fuerza del estímulo es suficiente para despolarizar el potencial de membrana hasta un nivel crítico, denominado *potencial de umbral*; los estímulos superiores o iguales a este umbral provocan potenciales de acción. Además los potenciales aparecen en forma máxima o se dan en absoluto.

El periodo refractario se refiere a que en este periodo la membrana no responde. Esto significa que la aplicación de un estímulo de máxima intensidad, antes de que haya desaparecido por completo el potencial anterior, no necesariamente desencadenara un segundo potencial. Pero si se aplica un estímulo supraumbral se distinguirán dos periodos refractarios separados que se encuentran asociados con un potencial de acción.

**Periodo refractario absoluto** . Es un intervalo de inexcitabilidad de la membrana ante la aplicación de cualquier estímulo, corresponde al período de cambios de permeabilidad al Sodio.

**Periodo refractario relativo** . Después del período refractario absoluto hay un intervalo durante el cual puede producirse una segunda respuesta, pero sólo si la intensidad del estímulo es superior al nivel umbral. Corresponde al período de incremento de la permeabilidad al

Potasio. Generalmente este período es seguido de un breve período de excitabilidad supernormal.

El período refractario limita el número de potenciales de acción que puede producir una membrana excitable en cierto tiempo.

De este modo el problema se reduce a considerar los dos tipos de corrientes para, de ahí, contemplar un modelo inicial que pueda reproducir la excitabilidad del tejido. Sobre este problema existe una solución que ha sido la base de toda una serie de modelos que logra reproducir el comportamiento eléctrico. El modelo inicial es conocido como la ecuación de FitzHugh-Nagumo [27]. Y es, al tejido cardiaco, lo que la ecuación del cable eléctrica a un axón. Se pueden encontrar en la literatura diferentes modelos sobre la actividad eléctrica en el corazón para distintas partes, como las fibras de Purkinje, etc.

Volviendo al modelo, la consideración inicial y más importante, es que existe una relación entre las dos corrientes referidas más arriba. Esta relación es simple y se refiere a que el cambio de la corriente con la distancia es proporcional a la corriente que se intercambia entre el interior y exterior de la célula. Estos modelos son modelos de Reacción-Difusión. Y son escritos usualmente con ecuaciones diferenciales parciales:

$$\frac{\partial u}{\partial t} = f(u, v) + D \frac{\partial^2 u}{\partial x^2} \quad (3.1)$$

$$\frac{\partial v}{\partial t} = g(u, v) + \Delta D \frac{\partial^2 v}{\partial x^2} \quad (3.2)$$

Donde  $x \in R^2$  y  $t$  son medidos en unidades espaciales y temporales determinadas. El propagador, variable  $u$ , representa un potencial eléctrico o una concentración química;  $D$  es un coeficiente de difusión con unidades espaciales  $\frac{M^2}{T}$ , y el controlador  $v$  representa la apertura de canales iónicos o alguna concentración química local difundiéndose a la razón  $\Delta D$ . Para aplicaciones fisiológicas se toma  $\Delta = 0$ . El parámetro  $\varepsilon$  es un factor adimensional extraído de  $f(u, v)$  y/o  $g(u, v)$  que cuantifica las razones relativas de excitación y recuperación.

Ahora veamos al problema globalmente. Considerando que el modelo anterior es una descripción útil del comportamiento local, no se debe perder de vista que el objetivo, es el comportamiento del tejido cardiaco. Esto es, un comportamiento global, colectivo, resultado de la aportación cuantitativa de todos los procesos locales. Entonces, entre algunas de las deficiencias del modelo de Nagumo se puede reconocer la consideración continua y local del medio. La consecuencia de esto es que dentro de este modelo inicial, la propagación eléctrica no es muy clara. Sin embargo, con un análisis de la misma ecuación de Nagumo [27]. Puede ser observada la condición de que el estímulo inicial debe ser suficientemente grande. De no ser así, el estado original del sistema es restablecido. Además que, como consecuencia, se encuentra una velocidad de propagación única [27].

Cabe mencionar que el modelo no es el único modelo del que se ha valido el trabajo de los últimos años sobre el comportamiento eléctrico del sistema. Otro modelo que tiene resultados importantes, es el que considera las dos

regiones más importantes en el Miocardio. De ellos se puede realizar una simplificación al considerar dos osciladores acoplados [17]. Ambos con parámetros distintos. Y como resultado relevante se considera relacionar parámetros en las ecuaciones acopladas con el potencial de acción. Aun así, el sostenimiento de la propagación no es muy claro.

### 3.3.2 Patrones en Superficies

El comportamiento emergente, como uno de los fenómenos con más interés dentro del comportamiento del corazón tiene como manifestación, la formación de patrones. Para reducir el problema que involucra la descripción de tales patrones, se consideran los casos particulares como son las arritmias cardiacas.

Un aspecto básico de los modelos que simulan este comportamiento, basados en variaciones del modelo de Nagumo, es que muestran la evolución de los sistemas a partir de condiciones iniciales conocidas. De ellos se obtiene un resultado importante. La propagación de un frente de onda, que si bien puede ser plano, se deforma en el proceso evolutivo, al encontrarse con heterogeneidades físicas o funcionales. Se entiende por las primeras, por ejemplo, la conexión entre diferentes tejidos y, por funcionales se entiende la diferencia local en propiedades refractarias, y excitables.

Las anomalías en el medio provocan la deformación del frente de varias maneras. Dentro de las cuales se pueden identificar las ondas circulares

(reentrates), las espirales y los rotores [18]. Las ondas circulares se observan cuando un frente de onda circula alrededor de un obstáculo. Las espirales son observadas cuando el frente de onda ha encontrado un obstáculo y uno de los extremos encuentra dificultades para continuar. De modo que si consideramos al frente como una franja compuesta de elementos que se encuentran en estado excitado. Se considerara que la parte frontal de esta franja se retrasa en su avance y la parte posterior, la alcanza. Provocando que a lo largo del frente, las partes más cercanas al obstáculo se muevan más despacio que el extremo; y como consecuencia se presente una onda espiral, como mencione antes. Por su parte, los rotores son una forma de referirnos a ondas espirales que se desplazan de dos formas distintas. Una de ellas sigue una trayectoria cerrada; mientras que la otra no.

La deformación del frente es un proceso secuencial y múltiple. Tomando dos frentes que rodean dos obstáculos y que se encuentran en la región entre defectos. Lo que se observará es una nueva deformación debida a la aniquilación de ambos frentes. Con este ejemplo se nota que la evolución es un proceso que por un lado mantiene la propagación; y por otro, una secuencia de deformaciones en el frente de onda. La evolución es progresiva, no se observa la transición de un estado con un frente aislado, a un estado con un número de fragmentos que se sostenga [24].

El objetivo de este trabajo no contempla la descripción de cada uno de los parámetros que están implicados en el fenómeno. Más bien se intenta exaltar la auto-organización, como propiedad. Que bien puede ser parte de un

esquema que complementa al esquema clásico reduccionista. Por ejemplo, para el caso particular de las arritmias, se sugieren dos postulados básicos que explican la fibrilación Ventricular [18]:

- i) El comportamiento irregular eléctrico que es respuesta de la fibrilación puede ser explicado en base a la auto-organización de rotores.
- ii) La formación de ondas rotacionales es el resultado de fragmentación de frentes de onda propagándose.

Observar la auto-organización en uno de estos modelos cuantitativa o cualitativamente puede llevarse a cabo de varias maneras: El alto grado de organización puede ser visto desde la correlación entre las distancias de cada uno de los componentes o como se sugiere en [15], con un análisis sobre el comportamiento del espacio fase. Con una idea semejante a la de observar el cambio del estado de una dinámica periódica que, al aplicarle una fuerza externa, puede mostrar a través del cambio de la fase inicial, el cambio en el comportamiento. Del mismo modo, ahí se sugiere observar el espacio fase y asociar una fase al rotor que se presenta alrededor de un defecto topológico (punto fijo) o singularidad de fase (lugar con fase arbitraria). Con un análisis teórico que sugiere la utilización de variables angulares, para mostrar el cambio cíclico de estados de elementos que pertenecen la vecindad de la singularidad.

Si consideramos que los modelos mencionados, sólo pueden darnos a una aproximación determinista, incompleta y desentendida de la actividad real de los elementos básicos del sistema. Esto, aún cuando ha quedado claro que

la formación de los patrones es una propiedad emergente del sistema, y que no basta el considerar solo algunos de los rasgos, ya sean dinámicos o no; sino, que esa propiedad más la interacción entre elementos (células) debe ser necesaria para una descripción realista. Entonces tenemos que al considerar un modelo que incluya las reglas internas locales de transición de estado, se puede considerar un modelo realista en el sentido de que el estado de un elemento está dado en función del estado de la vecindad y que la contribución de todas las localidades, forman rasgos colectivos. Para este caso, se cuenta con un modelo que discretiza el tiempo, el espacio y las interacciones.

### 3.4 Modelización Discreta

El modelo debe cumplir con el requerimiento de un sistema espacialmente extendido con rasgos de excitabilidad y refractariedad, además de soportar la propagación de ondas. Es un Medio Excitable como el que ya ha sido presentado anteriormente. Si se pide además que los detalles locales sean considerados a través de parámetros involucrados en los procesos locales. Entonces la modelización puede realizarse con un autómata celular. Un Autómata celular (AC) es un modelo discreto en el tiempo y espacio que está compuesto por un número finito de elementos. Donde los elementos son ordenados en un arreglo generalmente plano, también puede ser un arreglo tridimensional. Los elementos dentro del arreglo toman su lugar, generalmente, en el centro de cuadrados. Aunque éstos pueden ser sustituidos por otros polígonos [25]. A cada elemento se le asocia un estado físico. Y el número de estados posibles de un elemento, también es finito. El estado que

presenta un elemento es el resultado de la interacción de la vecindad en un instante anterior. Es en este punto donde la discretización del tiempo tiene sentido. Pues los instantes están bien definidos.

Es necesario aclarar que la dimensión de los elementos del *AC* no corresponden a un elemento básico del sistema real. En este caso, a una célula. Más bien se refiere a una localidad. De este hecho se pueden surgir algunas observaciones. Por ejemplo, un arreglo con un gran número de elementos debe ofrecer una mejor aproximación. De ser así, cuanto significaría suficientemente grande. Y es de aquí de donde surge su contraparte. Cual es el tamaño mínimo que puede presentar una buena aproximación a los diferentes tipos de comportamiento. Por ahora se verá que esquema sigue el modelo *AC*.

Entonces, cada elemento tendrá influencia en el estado que adoptará un elemento vecino. Y el modelo más sencillo que se puede concebir, es uno donde se presenten dos estados básicos. El estado en reposo y el excitado. Cada uno corresponde a un estado de conducción e inactivación (reposo y excitación).

En caso de incrementar en un estado más, se obtiene un modelo con mayor detalle. El nuevo estado es sugerido como un estado refractario donde la inactivación no se lleva a cabo de manera espontánea. Sino más bien de una forma monótona, que excluye la posibilidad de excitarse nuevamente sin haber obtenido el reposo previamente, como ya se mencionó. Hay

propiedades muy interesantes que adquiere el autómata, al variar los periodos, y distribución, tanto de excitación y refractariedad, como se verá más adelante. Cabe destacar que de los primeros esfuerzos realizados para dar bases a una biología matemática fueron por parte del Dr. Arturo Rosenblueth, el cual es citado en algunos trabajos de referencia en este trabajo.

# Capítulo 4

## Simulación

Este cuarto capítulo expone brevemente el juego de la vida como un sistema discreto que permite la elaboración de una simulación que, bajo una regla de transición, muestra ondas espirales binarias. Además se incluye un análisis simple del sistema a través de la medición del parámetro entropía.

### 4.1 Introducción

En este capítulo se trabaja el problema central de esta tesis el cual tiene como motivación la generación de ondas espirales en el tejido cardiaco. Y en particular en el proceso de evolución dinámica por el cual las ondas espirales se crean, interfieren y aniquilan. De modo que se decidió observar un proceso de evolución de ondas espirales en uno de los modelos más simples posibles. Al modelo se le asignarán dos parámetros independientes y se observará como repercute el parámetro Entropía al respecto. ¿Que se puede

esperar del comportamiento de este parámetro? Pues como se mencionó en el capítulo uno, la entropía de un sistema es máxima cuando un sistema tiene demasiada información, y el número máximo de subsistemas actúan independientemente. En un autómata celular la entropía toma su valor máximo de un bit por celda para un ensamble equiprobable [4]. Además de que para un autómata irreversible, la entropía decrece con el tiempo. Para el autómata ese decremento en  $S$  denota irreversibilidad. Como se mencionó anteriormente, la irreversibilidad se entiende en un sentido de obtener una configuración a partir de varias posibles, lo que significa disminuir la variedad de estados y, por tanto, aumentar la probabilidad y al mismo tiempo reducir la entropía.

Ahora bien, ya que se ha mencionado que el método para observar el comportamiento cualitativo del medio excitable que soporta un proceso de formación y aniquilación de ondas espirales, es el uso de un autómata celular, cuya dinámica es una de las más simples posibles; y el estudio de sistemas fisiológicos tiene como opción más viable, el lenguaje que ofrece la teoría de sistemas dinámicos, las otras están limitadas que sólo sirven como aproximaciones simples. La razón para que la dinámica no lineal sea adecuada a la dinámica fisiológica es que organismos vivos compuestos de múltiples sistemas despliegan oscilaciones complejas que interactúan con otras y con el medio ambiente y que son retroalimentadas de manera no lineal.

## 4.2 Autómata Celular

Con la idea de que un modelo de autómata celular es una forma computacionalmente accesible para representar ondas en medios excitables. Partamos de la noción de excitabilidad [29]. Para hacer esto uno asume que el estado de un sistema puede estar descrito por variables simples que toman valores de  $0, 1, \dots, N$ . Donde al valor cero se le llama estado en reposo, mientras que los estados del 1 al  $D$  ( $D < N$ ) son estados excitados y los estados de  $D + 1$  al  $N$  son estados refractarios.

Después se define un arreglo de celdas donde el estado de cada célula en el arreglo es actualizado en tiempos discretos de acuerdo con el estado presente de la célula y de su vecindad. Una celda en reposo permanece en reposo hasta que existe un número suficiente de celdas excitadas. La vecindad es tomada comúnmente con los cuatro u ocho vecinos más cercanos en un arreglo cuadrangular. Un número suficiente de células vecinas excitadas es usualmente tomada como 1. Una célula en estado “ $i$ ” cambia al estado “ $i + 1$ ” en el siguiente intervalo de tiempo, y una célula en el estado  $N$  regresa al estado 0.

Con estas reglas se han realizado muchos trabajos en los que se analiza la propagación de ondas. En estos modelos se deja a un lado un hecho importante, la dispersión, la cual se puede incluir si se consideran diferentes velocidades de propagación en el medio.

Una vez descrita la idea básica detrás de un autómata celular se describirán

las reglas que sostienen la transición y que producen un patrón particular.

### 4.3 Juego de la Vida

El juego de la vida es un autómata celular bidimensional con reglas de transición simples. Los estados que presentan cada una de las celdas son reposo y excitado. Y la función de transición determina un cambio de estado al tiempo " $t + 1$ " dependiendo del estado de la celda en el tiempo " $t$ " y de los vecinos al tiempo " $t$ " y " $t - 1$ ".

El código del programa que simula al autómata ha sido tomado como software libre [30] y se muestra en el apéndice. El autómata consta de un ciclo de 8 tiempos. La regla de transición se presenta en la fig [4.1].

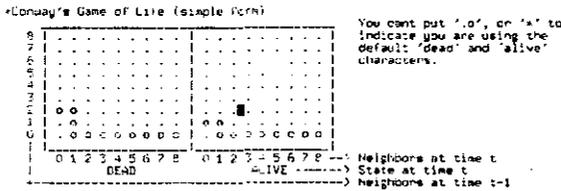


Figura 4.1: La regla que determina la transición en el autómata celular dentro del código del juego de la vida

De donde se puede ver que si hay dos celdas en reposo durante dos tiempos, en el siguiente, la celda central estará en reposo y así continuará hasta los ocho tiempos. Además si hay una celda excitada durante dos tiempos,

en el siguiente la celda central estará excitada. Esta regla permite observar el patrón que contiene espirales a partir de condiciones iniciales aleatorias.

Después de ejecutar el programa se encadenan procesos que nos permitirán obtener a partir del número de celdas vivas, la probabilidad asociada a cada una de las configuraciones que determinan la evolución del sistema. Inmediatamente después se puede calcular la entropía  $S = \sum p_i \log_2(1/p_i)$  asociada al proceso. Se considerará la entropía en relación al tamaño del autómatas. Es decir, la entropía será una medida espacial del sistema y será obtenida de la siguiente manera.

En la evolución dinámica de un autómatas celular se registrará el número de celdas vivas que aparecen en cada configuración. Es decir se registrará el número de celdas vivas y cuantas veces se obtuvo ese registro a lo largo de la dinámica.

De esta forma podemos tener dos opciones. La primera es continuar la dinámica por siempre; y la segunda tomar un autómatas finito que tendrá su configuración final elegida arbitrariamente; pero dentro de un estado en el que los patrones buscados permanezcan. De esta forma se obtiene una densidad de probabilidad respecto a la aparición de un número determinado de celdas vivas. O si se prefiere, se obtiene una densidad de probabilidad del número de bits necesarios para soportar la dinámica que presenta los patrones espirales.

De esta manera se ha elegido obtener un número de 2000 configuraciones a partir de un transiente de 1000. Es decir, el autómata alcanzó la configuración 3000 y nosotros comenzamos a contar a partir de la configuración 1001. Cada uno de estos análisis se realizó con condiciones iniciales distintas. En particular nos referimos a que el porcentaje inicial de celdas vivas lo variamos entre 5% y 100%. La diferencia entre cada uno de estos porcentajes no afectó la dinámica de los autómatas. Por lo que se decidió continuar con el análisis exclusivo de la dinámica para un autómata con 50% de celdas vivas inicialmente.

El tamaño espacial del autómata se toman en un rango de 5 a 50 elementos(celdas) de longitud por cada lado. El proceso de análisis en la entropía se realizó con diferentes condiciones. Es decir, comenzando con un autómata definido como un arreglo de 5 celdas por lado se observa la evolución de 3000 pasos que se mencionan arriba. Para este proceso se obtiene un valor en la entropía total. De modo que para obtener una mejor idea de la evolución del proceso, se toman 100 de estas corridas para obtener una varianza en el valor de la entropía. Este método se repite para cada uno de los autómatas,  $10 \times 10$ ,  $15 \times 15$ ,  $20 \times 20$ , etc...

Además se procede con el mismo método para obtener la entropía en autómatas con dinámicas que no presentan los patrones espirales. Sin embargo es difícil asegurar que la dinámica de estos autómatas era una dinámica completamente desordenada. Es decir, sin correlaciones entre los elementos. De modo que sólo nos limitaremos a decir que la dinámica no presentaba

las espirales. Por otra parte, la comparación del parámetro  $S$  entre los distintos autómatas, puede identificar la presencia de las espirales. Y con esto diferenciar grados de organización, que como ya lo dijimos, al carecer de una directriz, hablamos de auto-organización. La figura [4.2] muestra una imagen del autómata donde el patrón muestra tres espirales

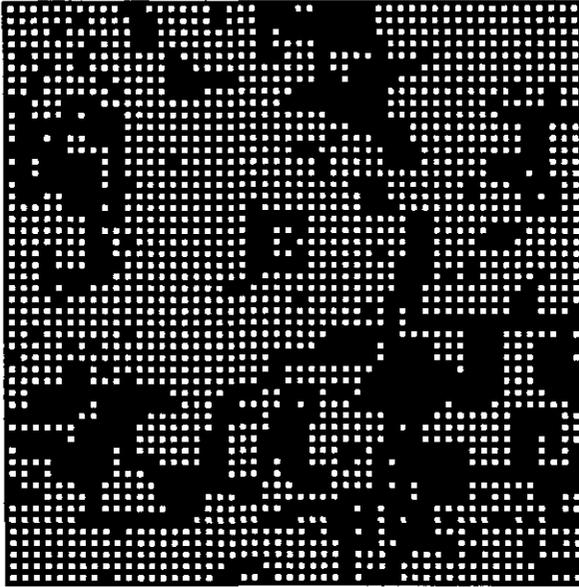


Figura 4.2: Patrón generado por el autómata celular donde se identifican tres espirales. Las cuales están formadas por celdas en negro que representan el estado excitado, mientras que las celdas blancas representan al estado en reposo. El automata mostrado tiene condiciones periódicas a la frontera y el arreglo cuenta con  $50 \times 50$  celdas.

Este autómata en particular tiene 50 celdas por lado. Las condiciones a la frontera son periódicas y las celdas negras representan al estado excitado mientras que las blancas representan las celdas en reposo. Además, la regla

que generó este patrón es la misma regla que se muestra en la fig [4.1]. En ella se identifican dos espirales en el centro (patrón en forma de B). Ambas se encuentran de frente; en ocasiones alguna de ellas subsiste. También se puede encontrar una espiral en la parte izquierda. En la figura de abajo se puede ver con mejor claridad una espiral fig [4.3].



Figura 4.3: Patrón generado por el autómata celular donde se identifican una espiral. Este arreglo cuenta con  $50 \times 50$  celdas, condiciones a la frontera periódicas, condiciones iniciales aleatorias y con celdas en estado excitado en negro.

En la figura [4.4]; se muestran los resultados obtenidos del análisis simple de entropía que se realizó en diferentes sistemas. El programa que realizó la estimación del parámetro  $S$  está dado en el apéndice (ENTROPY PROGRAM). La curva de entropía obtenida muestra que existe actividad a partir

de  $N = 10$  en el autómata  $N \times N$ . Y una clara diferencia entre ella y las curvas que corresponden a procesos que en su evolución no presentaron un patrón evidente.

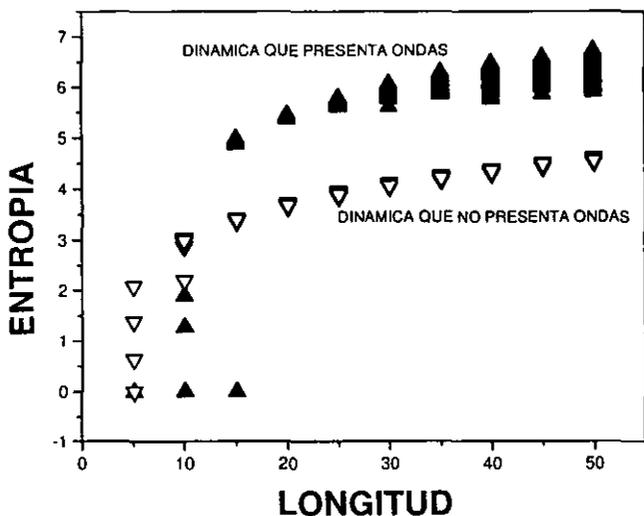


Figura 4.4: Relación entre la entropía obtenida para el modelo que presenta ondas espirales y la entropía obtenida en una dinámica aleatoria. La curva muestra 100 puntos por dinámica realizada a autómatas con  $N \times N$  celdas, los cuales están representados por la *longitud* $N$ . Las figuras blancas representan al parámetro medido para autómatas que no presentan patrones espirales. Las figuras negras indican el comportamiento del parámetro en los autómatas que soportan ondas espirales.

Se han discriminado los tamaños a partir de  $50 \times 50$  por la razón de que el comportamiento de la entropía es monótono a partir de ese tamaño.

En la curva anterior se puede observar la regularidad de la entropía en todos los casos. Dentro de este comportamiento regular el que resulta más interesante es el de la curva superior que es la que corresponde a la regla principal fig [4.1] y que es máximo en comparación con el de otras reglas. La dispersión en cada uno de los casos nos habla de la regularidad con la que se presenta la dinámica. Debe aclararse que esta curva muestra la relación entre el parámetro  $S$  y el tamaño del sistema. Por lo que no debe confundirse el comportamiento de la entropía con su relación temporal, o cualquier otra. De esta forma podemos observar que, según se entiende por  $S$ , éste es máximo cuando el sistema tiende a uno de los estados con mayor probabilidad de entre varios de un ensamble. Ahora, en la gráfica se tienen presentes dos reglas. Sin embargo el comportamiento de la curva nos puede indicar que esta dinámica está muy cerca de la dinámica que presenta la regla principal. Esta regla no presenta rasgo alguno, visible, de organización. Esto lo podríamos inferir, sin necesidad de observar la película, solamente de comparar la diferencia con la curva superior.

## 4.4 Resultados

Partiendo de la idea de auto-organización y del concepto de medio excitable para llegar después al fenómeno auto-organizado que presenta la actividad eléctrica del corazón, el autómatas celular que se ha elegido, ha cumplido con las expectativas en un sentido práctico.

La regla de transición es fácilmente manipulable. Aprovechando esto se han probado diversas reglas. Con ellas se ha identificado que existe un conjunto de

las mismas que permite la observación de patrones ordenados; y que después de aplicar el mismo análisis de entropía cada una de ellas, el comportamiento del parámetro no ha tenido diferencias significativas, mientras que el mismo parámetro ha mostrado diferencias entre estas reglas y las que producen una dinámica sin presencia de patrones (ondas espirales identificables).

Sobre la varianza del parámetro en la curva, podemos decir que debido a la iniciación aleatoria del autómata, la dinámica de las regiones que aportan información a la formación de las espirales es distinta en cada una de las evoluciones (corridas) que se realizaron. De modo que la varianza va aumentando conforme la cantidad de estas regiones aumenta. Como puede verse en la fig [4.4] que mientras aumente el tamaño del autómata, en particular, desde 30 celdas por lado, la varianza aumenta.

# Capítulo 5

## Conclusiones

A lo largo de este trabajo se han citado algunos de los aspectos que involucran los fenómenos no lineales en sistemas específicos (sistemas que presentan patrones) Y es en este sentido en el que se dirigió el modelo que implementamos. Una vez realizado el modelo podemos decir que en él se han presentado rasgos de auto-organización que coinciden, en cierta forma, con el resto de los fenómenos citados inicialmente. ¿De que forma? La respuesta es simple si se observa que el sistema está integrado por unidades bien definidas (celdas). Ellas se han organizado bajo una lógica muy simple y que coincide con la regla de transición que determina la dinámica del sistema. Sin embargo sabemos por la referencia del primer capítulo que esto no coincide con la idea de auto-organización directamente. Pues mientras la regla de transición es local, el efecto que interesa es colectivo. Y de esta manera se pueden observar autómatas con regla de transición bien definida pero sin ningún efecto global organizado. Así es que ha de buscarse el rasgo que bajo la nula participación de una directriz permita la evolución

de el sistema hacia una situación específica. Es así que, como parte de la auto-organización del sistema, la regla de transición permite a cada uno de los elementos adquirir un estado de entre dos posibles. Y con ello contribuir en la configuración global. La cual resulta ser en este caso espacial y temporal.

La contribución a la auto-organización del sistema se realiza en diferentes partes. Es decir, la formación de patrones, toma sentido al considerar la excitabilidad del autómatas y la contribución de cada una de las regiones que componen al arreglo, las cuales no fueron analizadas en este trabajo. Pero que mostraron su existencia en la curva de entropía.

Comenzando con la excitabilidad del medio, según se comentó en el segundo capítulo, los elementos del sistema tendrán una capacidad individual para asimilar los estímulos externos. Es de esta forma que la excitabilidad del medio está directamente relacionada con la regla de transición. Es así que para interpretar el estímulo externo se asume que mientras más elementos "vivos" rodeen a una celda, mayor será la magnitud de éste. Por lo tanto, la magnitud del estímulo aplicado en nuestro sistema, es débil. Sin embargo el estímulo persiste el tiempo suficiente como para que a la celda se le de tiempo de asimilar la información. Es decir, estímulo o información, no deben de ser ni breves ni demasiado prolongados. De otra forma el sistema no tiene la capacidad para generar patrones con esta característica.

Una vez que fué visto un rasgo importante dentro de la dinámica local del

autómata celular. En el siguiente paso se buscó algún indicio de contribución local. Y fué considerada la opción de probar el autómata con tamaños diferentes. A partir de 5 elementos por lado, en arreglos cuadriculares. De esto se obtuvo un comportamiento ya esperado debido a los reportes que se encontraron en la literatura. Indican que, la presencia de espirales de actividad eléctrica en tejido cardiaco sólo se observa para órganos que sobrepasan cierto tamaño. Cabe aclarar que no se pretende extrapolar el resultado fisiológico de esos reportes con el trabajo de esta tesis. Sino dar un antecedente de que los patrones presentados en algunos sistemas, muestran cierta limitación espacial. Entonces, el comportamiento era esperado. Sin embargo, de las dinámicas obtenidas, se identificó un tamaño aproximado para el cual el autómata carece de rasgos organizados.

Es a partir de los 10 elementos por lado, que el autómata comienza a mostrar patrones definidos. Esto es lo que nos hace pensar en que el autómata que presenta los patrones espirales se divide en localidades que aportan información y que su tamaño aproximado es de 10 celdas por lado. Bajo esta misma idea se esperaría que en un autómata con un gran número de elementos, la cantidad de localidades también aumente. Y de ser así, la información o los efectos que se puede tener entre vecindades, estará limitada espacial y temporalmente. Es decir, la información entre localidades muy distantes tendrá que pasar por otras localidades intermedias para comunicarlas. Esto implica un cambio de la información original. Es por esta razón que la curva de entropía, para arreglos grandes, tiende a aumentar la varianza. Pues considerando que las condiciones iniciales son

aleatorias. La información inicial en cada una de las dinámicas, que pudiera presentar un autómata de estos tamaños. Implica que la evolución de ésta y la forma en la que se distribuye siempre sea distinta, pero semejante. Y al aumentar el tamaño, esto se magnifica más.

Dentro de las implicaciones de separar al autómata en localidades. No se puede incluir la predicción del número de espirales que aparecerán en autómatas con tamaños grandes. Lo que si podemos decir es que, por debajo de 10 celdas por lado, no habrá ninguna.

Sobre el comportamiento del parámetro entropía, en comparación con el parámetro asociado a otro autómata, con su respectiva regla de transición. Tenemos por un lado, el aumento del parámetro "S" respecto a la longitud de los lados del arreglo cuadrangular; por el otro tenemos la diferencia entre "S" para cada uno de los sistemas, los que presentan y los que no presentan las espirales. Comenzando por el primero, podemos encontrar la coincidencia con la idea de las localidades e incluso, sin necesidad de considerarlas. Pues como lo mencionamos en el capítulo uno, "S" puede significar la cantidad de información que requiere nuestro sistema para presentar tal configuración. De esta manera es inmediato esperar un aumento del parámetro con el aumento del tamaño.

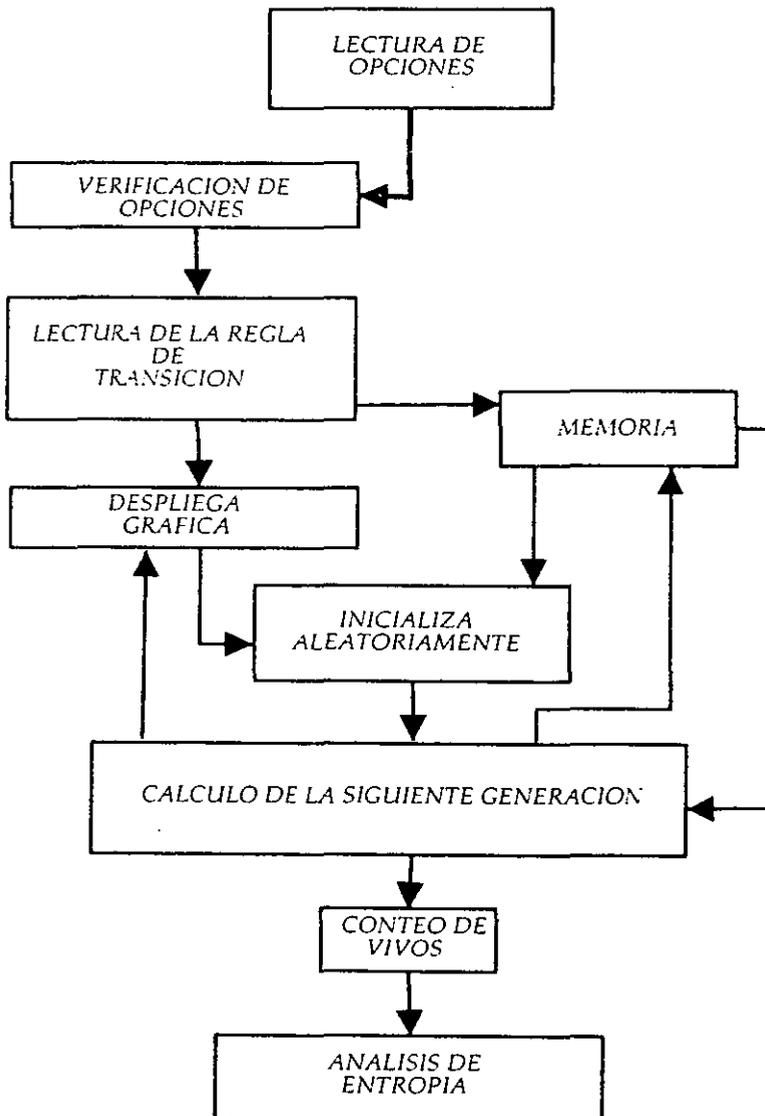
En segundo lugar se tiene la comparación entre "S" de cada una de las reglas de transición. Entonces, que autómata requiere de más información para conseguir la configuración dada. La respuesta nos la otorga la curva de

“S”. Y la explicación la encontramos en que un autómata ordinario guarda, por definición, en cada una de las celdas una correlación con las celdas de la vecindad inmediata. Y para un autómata que presenta patrones eso no es suficiente. Claro, en segundo orden. Es decir, el autómata que presenta las espirales tiene, al igual que cualquier otro, una regla de transición que correlaciona los estados de una vecindad. Pero para que un patrón se manifieste, se deben de relacionar no sólo elementos de una vecindad, sino vecindades. Entonces, esta correlación entre distintas localidades, no ha sido un rasgo cedido intencionalmente al sistema, como lo podría ser la regla de transición. Más bien, es parte de lo apasionante del comportamiento de este tipo de sistemas no lineales y auto-organizados.

Como último punto se aclarará que este trabajo está básicamente motivado por el comportamiento real de sistemas biológicos, en particular, el corazón. Esto sin ningún afán de resolver el problema fisiológico en sí, sino con el fin de dar un sentido tangible, familiar al fenómeno.

# Apéndice

El siguiente diagrama muestra el flujo de datos. Este flujo incluye al programa fuente y a las funciones que son invocadas en la ejecución del programa, las cuales son mostradas en este apéndice. El diagrama concluye con el análisis de entropía que es realizado por el programa ENTROPY. Al final de este apéndice se incluyen los scripts que controlan todo el proceso.



.....

xesca-1.0, XWindow Extended Simulator of 2D Cellular Automata.

(AKA tugli, The Unleashed Game of Life)

Copyright (C) 1997 Francesc Rocher

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

-----

Francesc Rocher can be found by

snail-mail:

University of Girona, Dept. of Economics

Campus Montilivi s/n

17071 Girona

SPAIN

e-mail: rocher@gnomics.udg.es

www: <http://gnomics.udg.es/~rocher/>

See also the xesca homepage, <http://gnomics.udg.es/~rocher/xesca/> where you will find the latest release of the program and more information.

.....

File xesca.c

.....

```

#include "data.h"      /* Global definitions      */
void faxit(int);      /* Fatal exit              */
#include "options.c"   /* Options parser          */
#include "ygames.c"   /* Games file reader      */
#include "xfunca.c"   /* XWindow functions      */
#include "genera.c"   /* The generator & initializer */
void
main(int argc, char **argv)
{
    int i;
    int j;
    read_options(argc,argv);
    read_games();
    if (listg)
        exit(1);
    x_init();
    game = lgame[igame-1];
    initialize();
    if (endgen == -1)
        while (TRUE)
            {

```

```
next_gen();
sleep(swait);
}
else
    while (generation < endgen)
    {
next_gen();
sleep(swait);
}
if (ostatf)
    gen_freq();
fexit(0);
}
void
fexit(int code)
{
    int i;
    if (ffnamep != NULL)
        fclose(ffnamep);
    exit(1);
}
```

```

File options.c
.....

void
read_options(int argc, char **argv)
{
    DIR *dp;
    int fn = FALSE;
    char c;
    char *tail;
    opterr = 0;
    errno = 0;
    /*-
    Options:
h: Help
    C<int>: Number of cell columns
    R<int>: Number of cell rows
    S<int>: Scale (pixels per cell)
    F<int>: Separation between cells (in pixels)
    Q<int>: Window border width (in pixels)
    T<char>: Topology (b=bounded, h=horizontal-cylinder,
                v=vertical-cylinder, t=torus)
    d<int>: Dead color
    a<int>: Alive color
    l<int>: Background color
    p<str>: Path to games files
    f<str>: Games filename to load
        n: Print a list of games defined in the input file and exit
    r<int>: Initial game number to use
        s: Switch game each generation
    t<int>: If switch game, do it each <int> generations
    w<int>: Wait <int> seconds each generation
        o: Output statistics (deads and alives)
O: Output statistics (frequency of rules used)
    i<int>: Random initialization (alives = <int>%)
    g<int>: Initial configuration number <int> (0 by default)
    G<int>: Run up to generation #G (always G > g)
        b: Run in batch mode
        v: Verbose
    */-
while ((c = getopt(argc, argv, "hC:R:S:P:Q:T:d:a:l:p:f:nr:st:w:oG:i:g:G:bv")) != -1)
    switch (c) {
case 'h': { /* Help ----- */
    printf("This is xesca V%s, %s\n",VERSION,DATE);
    printf("The XWindow Extended Simulator of Cellular Automata\n");
    printf("(AKA tugli, The Unleashed Game of Life)\n");
    printf("Usage: xesca [OPTION]\n");
    printf("OPTIONS:\n");
    printf("  -C COLS\tNumber of cell columns\n");
    printf("  -R ROWS\tNumber of cell rows\n");
    printf("  -S SCALE\tScale (pixels per cell)\n");

```

```

printf(" -P SEP\tSeparation between cells (in pixels)\n");
printf(" -Q WIDTH\tBorder WIDTH of window\n");
printf(" -T [bhvt]\tThe topology to use\n");
printf("      \tb=bounded, h=horizontal cylinder.\n");
printf("      \tv=vertical cylinder, t=torus\n");
printf(" -d COLOR\tCOLOR for dead cells\n");
printf(" -a COLOR\tCOLOR for alive cells\n");
printf(" -l COLOR\tBackground COLOR\n");
printf("      \t\tPossible COLORS are\n");
printf("      \t\t0:black, 1:white, 2:grey.\n");
printf("      \t\t3:brown, 4:blue, 5:green.\n");
printf("      \t\t6:red, 7:yellow, 8:purple\n");
printf(" -p PATH\tPath to input files\n");
printf(" -f FILE\tInput file\n");
printf(" -r GAME\tInitial game to use\n");
printf(" -s \tSwitch game each TIME times\n");
printf(" -t TIME\tSet TIME (see -s option)\n");
printf(" -w SEC\tWait SEC seconds after a new generation\n");
printf(" -o \tOutput statistics (deads and alives)\n");
printf(" -O \tOutput statistics (frequency of used rules)\n");
printf(" -i PERCENT\tRandom initialization, PERCENT alives\n");
printf(" -g NUMBER\tSet the number of the first generation\n");
printf(" -G NUMBER\tRun up to generation #NUMBER\n");
printf(" -b \tRun in \"batch\" mode (without visualization)\n");
printf(" -v \tVerbose\n");
fexit(0);
}
break;
case 'C': { /* Columns ----- */
    cols = strtoul(optarg,&tail,0);
    if (errno) {
        fprintf(stderr,"Error: 'C' option, invalid argument \"%s\"\n",optarg);
        fexit(1);
    }
    if (cols < 1) {
        fprintf(stderr,"Error: 'C' option, at least 1 column is required\n");
        fexit(1);
    }
}
break;
case 'R': { /* Rows ----- */
    rows = strtoul(optarg,&tail,0);
    if (errno) {
        fprintf(stderr,"Error: 'R' option, invalid argument \"%s\"\n",optarg);
        fexit(1);
    }
    if (rows < 1) {
        fprintf(stderr,"Error: 'R' option, at least 1 row is required\n");
        fexit(1);
    }
}
break;

```

```

case 'S': { /* Scale ----- */
    scale = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'S' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if ((scale < 1) || (scale > 10)) {
fprintf(stderr,"Error: 'S' option, scale must be between 1 and 10\n");
fexit(1);
    }
}
break;

case 'P': { /* Cell separation ----- */
    cellsep = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'P' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if (cellsep < 0) {
fprintf(stderr,"Error: 'P' option, separation must greater than zero\n");
fexit(1);
    }
}
break;

case 'Q': { /* Window border width ----- */
    border = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'Q' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if (border < 0) {
fprintf(stderr,"Error: 'Q' option, border must greater than zero\n");
fexit(1);
    }
}
break;

case 'T': { /* Topology ----- */
    if (optarg == NULL) {
fprintf(stderr,"Error: 'T' option, topology [bhvt] expected\n");
fexit(1);
    }
    topo = optarg[0];
    if ((topo != 'b') && (topo != 'h') && (topo != 'v') && (topo != 't')) {
fprintf(stderr,"Error: 'T' option, topology [bhvt] expected\n");
fexit(1);
    }
}
break;

case 'd': { /* Dead color ----- */
    deadc = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'd' option, invalid argument \"%s\"\n",optarg);

```

```

fexit(1);
}
if ((deadc < 0) || (deadc > NUM_COLORS-1)) {
fprintf(stderr, "Error: 'd' option, value must be between 0 and %d\n", NUM_COLORS-1);
fexit(1);
}
}
break;
case 'a': { /* Alive color ----- */
alivec = strtoul(optarg, &tail, 0);
if (errno) {
fprintf(stderr, "Error: 'a' option, invalid argument \"%s\"\n", optarg);
fexit(1);
}
if ((alivec < 0) || (alivec > NUM_COLORS-1)) {
fprintf(stderr, "Error: 'a' option, value must be between 0 and %d\n", NUM_COLORS-1);
fexit(1);
}
}
break;
case 'l': { /* Line color ----- */
linec = strtoul(optarg, &tail, 0);
if (errno) {
fprintf(stderr, "Error: 'l' option, invalid argument \"%s\"\n", optarg);
fexit(1);
}
if ((linec < 0) || (linec > NUM_COLORS-1)) {
fprintf(stderr, "Error: 'l' option, value must be between 0 and %d\n", NUM_COLORS-1);
fexit(1);
}
}
break;
case 'p': { /* Path to input files ----- */
if (optarg == NULL) {
fprintf(stderr, "Error: 'p' option, directory not specified\n");
fexit(1);
}
dp = opendir(optarg);
if (dp == NULL) {
fprintf(stderr, "Error: 'p' option, cannot open the directory \"%s\"\n", optarg);
fexit(1);
}
closedir(dp);
path = (char *) strdup(optarg);
if (path == NULL) {
fprintf(stderr, "Error: cannot allocate memory\n");
fexit(1);
}
}
break;
case 'f': { /* Games filename ----- */
fname = (char *) strdup(optarg);

```

```

    if (fname == NULL) {
fprintf(stderr,"Error: 'f' option, cannot allocate memory\n");
fexit(1);
    }
    fn = TRUE;
}
break;
case 'n': /* Print a list of games ----- */
    listg = TRUE;
    break;
case 'r': { /* Initial game to use ----- */
    igr = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'r' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if (igr < 1) {
fprintf(stderr,"Error: 'r' option, value must be greather than 1\n");
fexit(1);
    }
}
break;
case 's': /* Switch game ----- */
    sgame = TRUE;
    break;
case 't': { /* Switch game each 't' times ----- */
    tgame = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 't' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if (tgame < 1) {
fprintf(stderr,"Error: 't' option, argument must be greather than zero\n");
fexit(1);
    }
}
break;
case 'w': { /* Wait some seconds each generation ----- */
    swait = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'w' option, invalid argument \"%s\"\n",optarg);
fexit(1);
    }
    if (swait < 1) {
fprintf(stderr,"Error: 'w' option, argument must be greather than zero\n");
fexit(1);
    }
}
break;
case 'o': /* Output statistics (deads and alive) ----- */
    ostat = TRUE;
    break;

```

```

case 'O': /* Output statistics (frequency of used rules) ----- */
    ostats = TRUE;
    break;
case 'i': { /* Random initialization (%lives) ----- */
    inirand = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'i' option, invalid argument \"%s\\\"n\",optarg);
fexit(1);
    }
    if ((inirand < 0) || (inirand > 100)) {
fprintf(stderr,"Error: 'i' option, argument must be between 0 and 100 \\n");
fexit(1);
    }
    }
break;
case 'g': { /* First generation number ----- */
    inigen = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'g' option, invalid argument \"%s\\\"n\",optarg);
fexit(1);
    }
    if (inigen < 0) {
fprintf(stderr,"Error: 'g' option, argument cannot be negative\\n");
fexit(1);
    }
    }
break;
case 'G': { /* Last generation number ----- */
    endgen = strtoul(optarg,&tail,0);
    if (errno) {
fprintf(stderr,"Error: 'G' option, invalid argument \"%s\\\"n\",optarg);
fexit(1);
    }
    if (endgen < 0) {
fprintf(stderr,"Error: 'G' option, argument cannot be negative\\n");
fexit(1);
    }
    }
break;
case 'b': /* Run in batch mode ----- */
    batch =TRUE;
    break;
case 'v': /* Verbose ----- */
    verbose = TRUE;
    break;
case '?': { /* Others ----- */
    if (isprint (optopt)) {
fprintf(stderr,"Error: unknown option '%c'\\n",optopt);
fexit(1);
    }
    else {
fprintf(stderr,"Error: unknown option character '\\x%x'\\n", optopt);

```

```

fexit(1);
    }
}
break;
}

/* Some checks */
if ((endgen < inigen) && (endgen != -1)) {
    fprintf(stderr, "Error: initial generation number (%d) less than final one (%d)\n", endgen, inigen);
    fexit(1);
}

generation = inigen;

/* Check the (path +)filename */
if (path == NULL) path = DEF_PATH;
if (path == NULL) {
    fprintf(stderr, "(Big) Error: no default path defined in Imakefile\n");
    fexit(1);
}

if (fname == NULL) fname = DEF_FNAME;
if (fname == NULL) {
    fprintf(stderr, "(Big) Error: no default games filename defined in Imakefile\n");
    fexit(1);
}

if (path[strlen(path)-1] != '/') {
    path = (char *) realloc(path, sizeof(char)*(strlen(path)+2));
    if (path == NULL) {
        fprintf(stderr, "Error: cannot allocate memory\n");
        fexit(1);
    }
    path = (char *) strcat(path, "/");
}

fname = (char *) malloc(sizeof(char)*(strlen(fname)+strlen(path)+1));
if (fname == NULL) {
    fprintf(stderr, "Error: cannot allocate memory\n");
    fexit(1);
}

fname = (char *) strcat(fname, path);
fname = (char *) strcat(fname, fname);
fnamep = fopen(fname, "r");
if (fnamep == NULL) {
    fprintf(stderr, "Error: cannot open file \"%s\"\n", fname);
    fexit(1);
}
}
}

void
verify_local_option(unsigned int game, char option, char *value, int nline)
{
    char *tail;
    if (value == NULL) {
        fprintf(stderr, "Error: %s [%d], invalid local option\n", fname, nline);
        fexit(1);
    }
    if (option == 'a') {

```

```

errno = 0;
lgame[game]->alivec = strtoul(value,&tail,0);
if (errno) {
    fprintf(stderr,"Error: %s [%d], invalid argument in '-a' option\n",fname,nline);
    fexit(1);
}
if ((lgame[game]->alivec < 0) || (lgame[game]->alivec > NUM_COLORS-1)) {
    fprintf(stderr,"Error: %s [%d], color value must be between 0 and %d\n",fname,nline,NUM_COLORS-1);
    fexit(1);
}
}
else {
    errno = 0;
    lgame[game]->tgame = strtoul(value,&tail,0);
    if (errno) {
        fprintf(stderr,"Error: %s [%d], invalid argument in '-t' option\n",fname,nline);
        fexit(1);
    }
    if (lgame[game]->tgame < 0) {
        fprintf(stderr,"Error: %s [%d], argument must be greather than zero in 't' option\n",fname,nline);
        fexit(1);
    }
}
}

void
local_options(unsigned int game, char *options, int nline)
{
    char *opt;
    const char delim[] = "-at";
    char *token;
    char s;
    if ((options[1] != 'a') && (options[1] != 't')) {
        fprintf(stderr,"Error: %s [%d], invalid local option '%c'\n",fname,nline,options[1]);
        fexit(1);
    }
    opt = strdup(options);
    token = strtok(opt,delim);
    if (token == NULL) {
        fprintf(stderr,"Error: %s [%d], invalid argument\n",fname,nline);
        fexit(1);
    }
    verify_local_option(game,options[1],token,nline);
    token = strtok(NULL,delim);
    if (options[1] == 'a')
        s = 't';
    else
        s = 'a';
    if (token != NULL)
        verify_local_option(game,s,token,nline);
}

```

**ESTA TESIS NO SALE  
DE LA BIBLIOTECA**

This file is part of  
 xesca-1.0, XWindow Extended Simulator of Cellular Automata  
 (AKA tugli, The Unleashed Game of Life)  
 Copyright (C) 1997 Francesco Rocher

.....

.TH XESCA 1 "2 Sept 1997"

.SH NAME

xesca \- XWindow Extended Simulator of Cellular Automata  
 (AKA tugli, The Unleashed Game of Life)

.SH SYNOPSIS

.B xesca

.RB [ \-Onsobvh ]

.RB [ \-C

.IR column \- ]

.RB [ \-R

.IR row \- ]

.RB [ \-S

.IR scale \- ]

.RB [ \-P

.IR separation \- ]

.RB [ \-Q

.IR borderwidth \- ]

.RB [ \-T

.IR topology \- ]

.RB [ \-d

.IR color \- ]

.RB [ \-a

.IR color \- ]

.RB [ \-l

.IR color \- ]

.RB [ \-p

.IR path \- ]

.RB [ \-f

.IR filename \- ]

.RB [ \-r

.IR game \- ]

.RB [ \-t

.IR times \- ]

.RB [ \-w

.IR seconds \- ]

.RB [ \-i

.IR percent \- ]

.RB [ \-g

.IR begin \- ]

.RB [ \-G

.IR end \- ]

.SH DESCRIPTION

.I xesca

is an extended cellular automata simulator for XWindow. It is able to run cellular automata similar to the John H. Conway's Game of Life. The extension added is related with the transition function: for each cell of the automata, its state at time

```
.I t+1
```

not only depends on the state of the cell and the number of neighbors at time

```
.I t,
```

but also depends on the state of the cell and the number of neighbors at time

```
.I t-1.
```

.SH "OPTIONS"

.TP

.BI \-C\ columns

Set the number of columns (in number of cells).

.TP

.BI \-R\ rows

Set the number of rows (in number of cells).

.TP

.BI \-S\ scale

Set the scale of the cells (in pixels).

.TP

.BI \-P\ separation

Set the separation between cells (in pixels).

.TP

.BI \-Q\ borderwidth

Set the border width of the window.

.TP

.BI \-T\ topology

Set the topology to use.

.B topology

must be one of the letters

```
.I b, h, v, t,
```

where

```
.I b
```

means

```
.I bounded
```

(the cells of the right, left, top and bottom have less neighbors than the others).

```
.I h
```

means

```
.I horizontal cylinder
```

(the cells of the right and the left have less neighbors, and the cells of the top and bottom are connected),

```
.I v
```

means

```
.I vertical cylinder
```

(rotated horizontal cylinder) and

```
.I t
```

means

```
.I torus
```

(both horizontal and vertical conditions).

```

.TP
.BI \-d\ color
Set the 'dead' color (for drawing dead cells). Run 'xesca -h' to see
which codes are allowed.
.TP
.BI \-a\ color
Set the 'alive' color (for drawing alive cells).
.TP
.BI \-l\ color
Set the color of lines between calls. The width of lines is selected with
the
.I
\-P
option.
.TP
.BI \-p\ path
Set the path where to find the input file. By default it is set to the
current directory.
.TP
.BI \-f\ filename
If no file is given, the default input file is 'default.games'. See this
file to learn more about the file format and how to define new games.
.TP
.BI \-n
Do nothing. Print a list with the games encountered in the input file and
exit. This options is useful if you want to see which games are defined in
a file or for verify the correctness of a file of games.
.TP
.BI \-r\ game
Set the initial game to use. If you put more than one game in an input
file, with this option you can select a game.
.B game
must be an integer, not the name of the game.
.TP
.BI \-s
Switch the game each
.I n
generations.
.I n
can be set with the
.I \-t
option. With this option the program will use a different game each
.I n
generations to evolve the configuration of the automata. The initial game
to use is selected with the
.I \-r
option. While switching, when the last game
is reached then the first one is used (and go cycling).
.TP
.BI \-t\ times
Set the number of times that a game is used in the
.I \-s

```

option. By default it is set to 1. This option overrides any local option of the games. See the 'default.games' to learn more about local options for games.

.TP

.BI \-v\ seconds

Wait

.I seconds

seconds between generations.

.TP

.BI \-o

Output statistics to stdout. The output is a three column formatted list of integers, where the first one is the generation number, the second is the number of dead cells and the last the number of alives cells.

.TP

.BI \-D

Output the frequency of rules used per game. This option is useful in the design phase of a new game.

.TP

.BI \-i\ percent

Set the initial configuration of alives cells randomly. It puts the state at time

.I t

and the state and the number of neighbors at time

.I t-1.

By default the initial configuration is created at 50% of alives cells.

.TP

.BI \-g\ begin

Set the number of the initial configuration. By default is 1.

.TP

.BI \-G\ end

Set the number of the last configuration. If this option is given, the program exits when the last configuration is reached. Else, it loops forever.

.TP

.BI \-b

Run in batch mode (without visualization).

.TP

.BI \-v

Verbose. Explain something.

.SH AUTHORS

Copyright (C) 1997 Francesc Rocher.

.LP

e-mail: rocher@gnomics.udg.es

WWW: <http://gnomics.udg.es/~rocher/>

xesca homepage: <http://gnomics.udg.es/~rocher/xesca/>

.LP

Please send comments, suggestions, improvements or bugs by e-mail to the author. Visit the xesca homepage, where you will find the latest release and more information.

.SH LICENSE

.LP

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

.LP

This file is part of

xesca-1.0, XWindow Extended Simulator of 2D Cellular Automata

(AKA tugli, The Unleashed Game of Life) -

Copyright (C) 1997 Franceac Rocher

Before reading this file I recommend you having a look to the README file.

This file is the default file loaded by xesca. Only special lines are understood by the program, the rest are ignored.

Here starts a new game definition. First at all, its name (more than 0 characters). Note the special character '\*' at the beginning. After the name, in a new line, two characters: the 'dead' and the 'alive' characters (the special mean of these characters is obvious ;-). By default, dead is '.' and alive is 'o' (you can change these defaults in the Imakefile). To indicate you are using the default characters, put an '--' at the first column. If you use different ones, put it in the first and in the second column (see the examples below). Then, immediately, the rules of the game. Only 'dead' and 'alive' characters are understood in each line, the rest are ignored. Don't put any different line between rules, the program expects to find eight consecutive lines with rules.

Here is the first example, the John H. Conway's Game of Life --as a particular case of an extended cellular automaton. It is defined using a simple definition form, where the transition function of one cell is independent of the state of the cell at time t-1 (i.e., it depends on the state of the cell at time t and the number of neighbors both at time t and t-1). '\*' before the name means 'simple definition form'.

\*Conway's Game of Life (simple form)

```
.o ----- You cant put '.o', or '*' to
8 | . . . o . . . . | . . o o . . . . | indicate you are using the
7 | . . . o . . . . | . . o o . . . . | default 'dead' and 'alive'
6 | . . . o . . . . | . . o o . . . . | characters.
5 | . . . o . . . . | . . o o . . . . |
4 | . . . o . . . . | . . o o . . . . |
3 | . . . o . . . . | . . o o . . . . |
2 | . . . o . . . . | . . o o . . . . |
1 | . . . o . . . . | . . o o . . . . |
0 | . . . o . . . . | . . o o . . . . |
| |-----|-----|
| 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 --> Neighbors at time t
|      DEAD           ALIVE -----> State at time t
|-----> Neighbors at time t-1
```

Before starting with rules, you can put 'local' parameters to the game, one for the 'alive color' and another for the number of times to play the game

when the '-s' option is given (see the man page). The way to specify these parameters is very simple, because you must use the options of the command line (but only '-a<color>' and '-t<times>' are understood). Local options MUST begin at the first column with '-', and can only be specified in one line. The '-t' option is overridden by the global '-t' option. If you don't provide any parameter, defaults values are taken. See an example below.

The same game as above, using a 'complete definition form': now the transition function of one cell depends on the state of the cell at time t and t-1, and on the number of neighbors at time t and t-1.

\*\*Conway's Game of life (complete form)

-a 2 -t 2

```

* -----
  8 | . . . o . . . . | . . o . . . . |
  7 | . . . o . . . . | . . o . . . . |
A 6 | . . . o . . . . | . . o . . . . |
L 5 | . . . o . . . . | . . o . . . . |
I 4 | . . . o . . . . | . . o . . . . |
V 3 | . . . o . . . . | . . o . . . . | Note: In a complete
E 2 | . . . o . . . . | . . o . . . . | definition form you
  1 | . . . o . . . . | . . o . . . . | MUST put a separation line
  0 | . . . o . . . . | . . o . . . . |
    |-----|-----| <--
  8 | . . . o . . . . | . . o . . . . |
  7 | . . . o . . . . | . . o . . . . |
D 6 | . . . o . . . . | . . o . . . . |
E 5 | . . . o . . . . | . . o . . . . |
A 4 | . . . o . . . . | . . o . . . . |
D 3 | . . . o . . . . | . . o . . . . |
  2 | . . . o . . . . | . . o . . . . |
  1 | . . . o . . . . | . . o . . . . |
  0 | . . . o . . . . | . . o . . . . |
  | |-----|-----|
  | | 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 --> Neighbors at time t
  | |      DEAD          ALIVE -----> State at time t
  | |-----> Neighbors at time t-1
+-----> State at time t-1

```

At the end of this file you will find two templates for your new games. Fill in the rules, put '+' or '=' before the name (depending on the form you choose) and play!. You can define more than one game per file. If you do so then you can switch from one game to another while running the program (see the man page for how to do this). The game used by default is the first one loaded. Moreover, you can do switching between games each 'n' times.

```

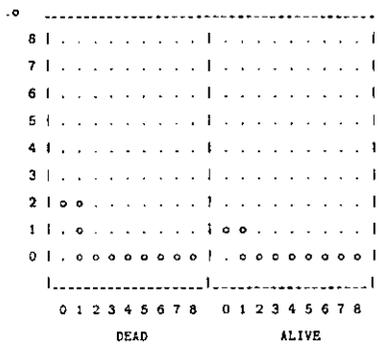
=====
IMPORTANT NOTE
=====

```

If you find new, impressive, curious, surprising games, please send me the file "your\_impressing\_games" (with the definition games plus comments and/or parameters for running them) and I'll make it public in the xesca homepage. Send me also, if you think it is convenient, screen shots in any graphic format. I hope people will waste some time trying to find new games, in simple or complete form (even combinations of switching games) to make a big collection of them. Also you can contribute with new ideas, suggestions, improvements or comments about the program. Acid flames are not welcome.

-----

\*Ondas espirales



File rgames.c

```

.....

void
read_games()
{
    char *line = NULL;
    char **linep = &line;
    size_t n = 0;
    size_t *N = &n;
    ssize_t Rline;
    int nline = 0;
    char startch = START_CHAR;
    char deadch = DEAD_CHAR;
    char alivech = ALIVE_CHAR;
    int pos, validc;
    TGame *gamep = NULL;
    TGame **rgame = NULL;
    char simple = FALSE;
    char *s = " ";

    if (verbose || listg)
        printf("Reading file \"%s\"\n", filename);
    while (!feof(filename)) {
        Rline = getline(linep, N, filename);
        nline++;
        if ((Rline == -1) && (!feof(filename))) {
            fprintf(stderr, "Unknown error: %s [%d]\n", filename, nline);
            fexit(1);
        }
        if (line[0] == startch) {
            /* Read the definition ----- */
            /* 1. Allocate memory ----- */
            gamep = (TGame *) malloc(sizeof(TGame));
            if (gamep == NULL) {
                fprintf(stderr, "Error: cannot allocate memory\n");
                fexit(1);
            }
            rgame = (TGame **) realloc(lgame, sizeof(TGame *) * (gamec + 1));
            if (rgame == NULL) {
                fprintf(stderr, "Error: cannot allocate memory\n");
                fexit(1);
            }
            lgame = rgame;
            lgame[gamec] = gamep;
            /* 2. Read the name and type of the game ----- */
            if (line[1] == startch)
                simple = FALSE;
            else
                simple = TRUE;
            lgame[gamec] -> simple = simple;
            lgame[gamec] -> name = (char *) malloc(sizeof(char) * (Rline - 2 + simple));

```

```

    if (lgame[gamec]->name == NULL) {
fprintf(stderr,"Error: %s [%d], cannot allocate memory\n",fname,nline);
fclose(1);
    }
    strncpy(lgame[gamec]->name,&line[2-simple],Rline-3+simple);
    lgame[gamec]->name[Rline-3+simple] = '\0';
    if (verbose || listg) {
if (simple)
    s = " *";
else
    s = " **";
printf("%3d %s%s",gamec+1,s,lgame[gamec]->name);
    }

/* 2.5 See if any option is given ----- */
lgame[gamec]->alivec = alivec;
lgame[gamec]->tgame = 1;
Rline = getline(linep,N,ffnamep);
if (line[0] == '-') {
local_options(gamec,line,nline);
if (verbose || listg)
    printf("%s",line);
Rline = getline(linep,N,ffnamep);
    }
else
if (verbose || listg) printf("\n");

/* 3. Read the dead and alive characters ----- */
nline++;
if (feof(ffnamep)) {
fprintf(stderr,"Error: %s [%d], eof encountered\n",fname,nline);
fclose(1);
    }
if (Rline == 1) {
fprintf(stderr,"Error: %s [%d], at least 'C' is required\n",fname,nline,startch);
fclose(1);
    }
if ((line[0] != startch) && (Rline < 3)) {
fprintf(stderr,"Error: %s[%d], at least two characters are required\n",fname,nline);
fclose(1);
    }
if (line[0] != startch) {
deadch = line[0];
alivech = line[1];
    }

/* 4. Read the rules of the game ----- */
for (row = 18-(10*(int)simple); row >= 0; row--) {
if ((row != 9) && (!simple) || (simple)) {
Rline = getline(linep,N,ffnamep);
nline++;
if (feof(ffnamep)) {
fprintf(stderr,"Error: %s [%d], eof encountered\n",fname,nline);
fclose(1);
    }
}
}
}

```

```

if (Rline < 19) {
    fprintf(stderr, "Error: %s [%d], at least 18 characters are required\n", fname, nline);
    fexit(1);
}
validc = 0;
pos = 0;
col = 0;
while ((validc < 19) && (pos < Rline-1)) {
    if (line[pos] == deadch) {
        lgame[gamec]->rule[col][row] = 0;
        if (simple)
            lgame[gamec]->rule[col][row*10] = 0;
        validc++;
        col++;
    }
    if (line[pos] == alivech) {
        lgame[gamec]->rule[col][row] = 1;
        if (simple)
            lgame[gamec]->rule[col][row*10] = 1;
        validc++;
        col++;
    }
    pos++;
    if (col == 9) col = 10;
}
if (validc < 18) {
    fprintf(stderr, "Error: %s [%d], not enough valid characters\n", fname, nline);
    fexit(1);
}
}
else
    Rline = getline(linep, N, ffnamep); /* Separation line */
    }
    gamec++;
}
/* ----- */
}
fclose(ffnamep);
if (gamec == 0) {
    fprintf(stderr, "Error: there are no games sets in %s\n", fname);
    fexit(1);
}
if (igame > gamec) {
    fprintf(stderr, "Error: 'r' option, initial game %d non existent\n", igame);
    fexit(1);
}
if (verbose || listg)
    printf("%s, %3d games read\n", fname, gamec);
}

```

```

File genera.c
*****
static int cml, cpl, rml, rpl;
static void
next_gen()
{
    generation++;
    lgen++;
    deads = 0;
    alives = 0;
    if (sgame) {
        if (tgame > 1) {
            if (generation % tgame == 0) {
                ngame = (ngame+1) % ggamec;
                game = lgame[ngame];
            }
        }
        else {
            if (lgen % tgame -> tgame == 0) {
                ngame = (ngame+1) % ggamec;
                game = lgame[ngame];
            }
        }
    }
    for (col = 1; col <= cols; col++) {
        for (row = 1; row <= rows; row++) {
            if ((topo == 'v') || (topo == 't')) {
                if (col == 1) cml = cols; else cml = col - 1;
                if (col == cols) cpl = 1; else cpl = col + 1;
            }
            else {
                cml = col - 1; cpl = col + 1;
            }
            if ((topo == 'h') || (topo == 't')) {
                if (row == 1) rml = rows; else rml = row - 1;
                if (row == rows) rpl = 1; else rpl = row + 1;
            }
            else {
                rml = row - 1; rpl = row + 1;
            }
            nei = cell[cml][rml].state + cell[col][rml].state + cell[cpl][rml].state
+ cell[cml][row].state + cell[cpl][row].state
+ cell[cml][rpl].state + cell[col][rpl].state + cell[cpl][rpl].state;
            nstate[col][row] =
game->rule[cell[col][row].state*10+nei][cell[col][row].pstate*10+
cell[col][row].pnei];
            if (ostatsf)
game->rulestat[cell[col][row].state*10+nei][cell[col][row].pstate*10+
cell[col][row].pnei]++;
            cell[col][row].pnei = nei;

```

```

    cell[col][row].pstate = cell[col][row].state;
}
}
for (col = 1; col <= cols; col++) {
    for (row = 1; row <= rows; row++) {
        cell[col][row].state = nstate[col][row];
        if (ostat)
            if (cell[col][row].state == 0)
                deads++;
            else
                alives++;
                x_draw_cell();
        }
    }
    if (ostat)
        printf("\X%d \Y%d \X%d$\backslash$n", generation, deads, alives);
}

void
initialize()
{
    int g,i,j;
    float r;
    /* Memory allocation ----- */
    rroup = (TCell **) malloc(sizeof(TCell *)*(cols+2));
    if (rroup == NULL) {
        fprintf(stderr, "Error: cannot allocate memory$\backslash$n");
        fxexit(1);
    }
    for (i = 0; i < cols+2; i++) {
        rroup[i] = (TCell *) malloc(sizeof(TCell)*(rows+2));
        if (rroup[i] == NULL) {
            fprintf(stderr, "Error: cannot allocate memory$\backslash$n");
            fxexit(1);
        }
    }
    cell = &rroup[0];
    nroup = (char **) malloc(sizeof(char *)*(cols+2));
    if (nroup == NULL) {
        fprintf(stderr, "Error: cannot allocate memory$\backslash$n");
        fxexit(1);
    }
    for (i = 0; i < cols+2; i++) {
        nroup[i] = (char *) malloc(sizeof(char)*(rows+2));
        if (nroup[i] == NULL) {
            fprintf(stderr, "Error: cannot allocate memory$\backslash$n");
            fxexit(1);
        }
    }
    nstate = &nroup[0];
    /* Clean ----- */
    for (col = 0; col <= cols+1; col++)
        for (row = 0; row <= rows+1; row++) {

```

```

    cell[col][row].state = 0;
    cell[col][row].pstate = 0;
    cell[col][row].pnei = 0;
    nstate[col][row] = 0;
}

/* Random initialization ----- */
if (inirand > 0) {
    r = (cols*rows*inirand/100);
    srand(time(0));
    deads = cols*rows;
    for (i = 0; i < r; i++) {
        col = 1+(rand()%cols);
        row = 1+(rand()%rows);
        cell[col][row].state = 1;
        cell[col][row].pstate = rand() % 2;
        cell[col][row].pnei = rand() % 9;
        x_draw_cell();
        alives++;
        deads--;
    }
    sleep(swait);
}

if (ostat)
    printf("%4d %8d %8d\\backslash$n",generation,deads,alives);
if (ostatf)
    for (g = 0; g < gamec; g++)
        for (j = 0; j <= 8; j++) {
            for (i = 0; i <= 8; i++)
                lgame[g]->rulestat[i][j] = 0;
            for (i = 10; i <= 18; i++)
                lgame[g]->rulestat[i][j] = 0;
        }
}

void
game_freq(unsigned int game)
{
    int i,j;
    unsigned long int ngen;
    char s;
    int r;
    unsigned long int total = 0;
    float p;
    ngen = endgen - inigen;
    printf("%$\\backslash$n$\\backslash$n$\\Xs$\\backslash$n$\\backslash$n$.lgame[game]->name);
    printf("------$\\backslash$n");
    for (row = 18-10*lgame[game]->simple; row >= 0; row--) {
        if (row != 9) {
            s = ' ';
            if (!lgame[game]->simple) {
                if (row == 16) s = 'A';
                if (row == 15) s = 'L';
                if (row == 14) s = 'I';

```

```

if (row == 13) s = 'V';
if (row == 12) s = 'E';
if (row == 6) s = 'D';
if (row == 5) s = 'E';
if (row == 4) s = 'A';
if (row == 3) s = 'D';
    }
    if (row > 9)
r = row-10;
    else
r = row;
    printf("%c %ld! ",s,r);
    for (col = 0; col < 19; col++) {
if (col != 9) {
p = (float) 1000*lgame[game]->rulestat[col][row]/(ngen*cols*rows);
if (p == 0.0)
printf(" . ");
else
printf("%3.0f ",p);
}
else
printf("| ");
    }
    printf("|$\backslashbackslash$");
}
printf(" -----$\backslashbackslash$");
printf(" 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8$\backslashbackslash$");
printf("          DEAD                          ALIVE$\backslashbackslash$");
}
void
gen_freq()
{
int g,i,j;
if (agame)
for (g = 0; g < gamec; g++)
game_freq(g);
else
game_freq(igame-1);
}

```

File xfuncs.c

```

.....
static void
x_init()
{
    char *dname;
    int black, white;
    int i;
    int nscreen;
    int width, height;
    XColor exact;
    if (batch)
        return;
    dname = (char *) getenv("DISPLAY");
    if (dname == NULL) {
        fprintf(stderr, "Error: variable DISPLAY not set\\backslashn");
        fexit(1);
    }
    display = XOpenDisplay(dname);
    if (display == NULL) {
        fprintf(stderr, "Error: cannot open display \\backslash$\\backslash$\\backslash$\\backslashn", dname);
        fexit(1);
    }
    screen = ScreenOfDisplay(display, DefaultScreen(display));
    black = BlackPixelOfScreen(screen);
    white = WhitePixelOfScreen(screen);
    width = 2*border + cols*scale + (cols-1)*cellsep;
    height = 2*border + rows*scale + (rows-1)*cellsep;
    window = XCreateSimpleWindow(display, RootWindowOfScreen(screen),
        0, 0, width, height, 0, white, black);
    gc = XCreateGC(display, window, 0, &xgcv);
    XSelectInput(display, window, ExposureMask);
    XMapWindow(display, window);
    nscreen = DefaultScreen(display);
    for (i = 0; i < NUM_COLORS; i++) {
        if (!XAllocNamedColor(display, DefaultColormap(display, nscreen),
            fixedcolors[i], &xcolor[i], &exact)) {
            fprintf(stderr, "Error: color allocation failed\\backslashn");
            fprintf(stderr, "Probably you should close any color hungry application\\backslashn");
            fexit(1);
        }
        xgcv.foreground = exact.pixel;
        if ((cellgc[i] = XCreateGC(display, window, GCForeground |
            GCBackground, &xgcv)) == 0) {
            fprintf(stderr, "Error: color creation failed\\backslashn");
            fexit(1);
        }
    }
    XFillRectangle(display, window, cellgc[linec],
    0, 0, width, height);
}
static void

```

```
x_draw_cell()
{
    static int    ri, ci;
    static char value;
    static int pcol, prow;
    if (batch)
        return;
    pcol = border+(col-1)*(scale+cellsep);
    prow = border+(row-1)*(scale+cellsep);
    if (cell[col][row].state == 1)
        value = game->alivec;
    else
        value = deadc;
    if (scale == 1) {
        XDrawPoint(display, window, cellgc[value], pcol, prow);
    }
    else {
        XFillRectangle(display, window, cellgc[value],
            pcol, prow, scale, scale);
    }
}
```

```

File data.h
.....
#include <dirent.h>
#include <errno.h>
#include <getopt.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <X11/Xlib.h>
#define TRUE 1
#define FALSE 0
#ifndef DEF_COLS /* Default number of columns (in cells) */
#define DEF_COLS 100
#endif
#ifndef DEF_ROWS /* Default number of rows (in cells) */
#define DEF_ROWS 100
#endif
#ifndef DEF_SCALE /* Default number of pixels per cell */
#define DEF_SCALE 4
#endif
#ifndef DEF_CELL_SEP /* Default separation between cells (in pixels) */
#define DEF_CELL_SEP 1
#endif
#ifndef DEF_TOPOLOGY
#define DEF_TOPOLOGY 't'
#endif
#ifndef ALIVE_COLOR
#define ALIVE_COLOR 6
#endif
#ifndef DEAD_COLOR
#define DEAD_COLOR 0
#endif
#ifndef LINE_COLOR
#define LINE_COLOR 0
#endif
#ifndef BORDER
#define BORDER 1
#endif
#ifndef STATES
#define STATES 18
#endif
#ifndef DEF_INITIALIZATION
#define DEF_INITIALIZATION 50
#endif
#ifndef START_CHAR
#define START_CHAR '*'
#endif
#ifndef DEAD_CHAR
#define DEAD_CHAR '.'
#endif
#ifndef ALIVE_CHAR
#define ALIVE_CHAR 'o'
#endif
#endif

```

```

#ifndef DEF_PATH
#define DEF_PATH "./"
#endif

#ifndef DEF_FNAME
#define DEF_FNAME "default.gam"
#endif

#define NUM_COLORS 9

static Display *display;
static Screen *screen;
static Window window;
static GC gc, cellgc[NUM_COLORS];
static XGCValues xgcv;
static XColor xcolor[NUM_COLORS];
static char *fixedcolors[NUM_COLORS] = {"Black", "White", "Gray",
    "Brown", "Blue", "Green",
    "Red", "Yellow", "Purple"};

static int cols = DEF_COLS; /* Number of cell columns */
static int rows = DEF_ROWS; /* Number of cell rows */
static int scale = DEF_SCALE; /* Number of pixels per cell */
static int cellsep = DEF_CELL_SEP; /* Number of pixels between cells */
static char topo = DEF_TOPOLOGY; /* Topology used */

typedef struct
{
    char state; /* Current state (at time t) */
    char pstate; /* Past state (at time t-1) */
    char pnei; /* Past neighbours (at time t-1) */
} TCell;

static TCell **rovp;
static TCell **cell = NULL; /* The cells array */
static char **nrovp;
static char **nstate = NULL;

static char deadc = DEAD_COLOR; /* Dead color */
static char alivec = ALIVE_COLOR; /* Alive color */
static char linec = LINE_COLOR; /* Line color (background) */
static char border = BORDER; /* Border width */
static int col, row; /* Current [col][row] of the processed cell */
static int nei; /* Number of Neighbours */
static int inirand = DEF_INITIALIZATION;

static unsigned long int generation = 0; /* Generation counter */
static unsigned long int lgen = 0; /* 'Local' generation counter */
static unsigned long int inigen = 1; /* Initial generation number */
static unsigned long int endgen = -1; /* Final generation number */
static unsigned long int alives = 0; /* Alives counter */
static unsigned long int deads = 0; /* Deads counter */

typedef struct
{
    char simple; /* (TRUE/FALSE) Definition form used */
    char *name; /* The name of the game */
    char rule[19][19]; /* Game rules */
    unsigned long int rulestat[19][19]; /* Rules statistics */
    char alivec; /* Alive color for 'this' game */
    unsigned int tgame; /* Times to play 'this' game when

```

```

'-s' option is given. It is
overridden by the global option
'-t<int>' when <int> > 1. These
parameters (alivec and tgame) are
set to the global defaults.      */
} TGame;

static unsigned int gamec = 0; /* Games counter */
static TGame *game = NULL; /* Current game used */
static char sgame = FALSE; /* Switch games each 'tgame' times */
static unsigned int tgame = 1;
static unsigned int swait = 0; /* Wait some seconds */
static unsigned int igame = 1; /* Initial game to use */
static unsigned int ngame = 0; /* Next game to use */
static TGame *lgame = NULL; /* Loaded games */
static char verbose = FALSE; /* Explain something */
static char batch = FALSE; /* Run in batch mode
/* (without visual output) */

char *path = NULL; /* Path to games files */
char *fname = NULL; /* Games file to open */
char *ffname = NULL; /* Full games (path +)filename */
FILE *ffnamep = NULL; /* The 'ffname' FILE */
char listg = FALSE; /* Make a list of games */
char ostat = FALSE; /* Output statistics */
char ostatt = FALSE; /* Output statistics (freq.) */

```

## PROGRAM ENTROPY

```

.....
INTEGER i, cont, j, r, rr
PARAMETER (n=3000, nmin=1000, sat=2500)
INTEGER x(n), y(n), z(n), w(n*5), t(sat)
INTEGER nmax, d, k, l, zzz, yyy, cero
DOUBLE PRECISION ss(sat), s(sat), zz, tam, tt
LOGICAL orden
.....
c   Aquí se inicia la lectura del archivo de datos

ss(1)=DBLE(0.0)
d=n-nmin
do i=1,n
  READ(*,*) x(i), y(i), z(i)
enddo

c   Definición del número de células en el arreglo

nmax=y(1)+z(1)
zzz=z(1)
yyy=y(1)
.....
c   Aquí se redefine el arreglo a manipularse

do k=1, n-nmin
v(k)=z(k+nmin)
enddo
.....
c   ordenamiento burbuja para el próximo conteo del número total
c   de aparición de cada una de la células en el periodo de medición

orden=.FALSE.
50 if(orden)go to 19
orden=.TRUE.
do 34 j=1, d-1
  if(v(j) .GT. v(j+1)) then
    temp=w(j)
    v(j)=w(j+1)
    w(j+1)=temp
    orden=.FALSE.
  endif
34 continue
go to 50
19 continue
.....
c   definición inicial del arreglo que determinará en cada
c   una de sus entradas, el número de apariciones de un
c   ensamble de células vivas. la longitud de este arreglo
c   es igual al número de celdas.

do r=1, nmax

```

```

      t(r)=0
    enddo
.....
c   conteo del número de apariciones de cada ensamble

      cont=1
      v(d+1)=nmax+1
      do i=1,d
if(v(i) .EQ. v(i+1))then
      cont=cont+1
else
      if(v(i) .EQ. 0)then
        cero=cont
      else
        t(v(i))=cont
      endif
      cont=1
    endif
  enddo
.....
c   cálculo de la entropía

      tt=DBLE(1.0)
      if(cero .NE. 0)then
        ce=DBLE(cero)/DBLE(d)
      ce=ce*DLOG(tt/ce)
      endif
      do l=1,nmax
        if(t(l) .GT. 0.0 ) then
          zz=DBLE(t(l))/DBLE(d)
          s(l)=DBLE(-1.0)* zz * DLOG(zz)
        else
          s(l)=DBLE(0.0)
        endif
        if(l .EQ. 1)then
          ss(1)=s(l)
        else
          ss(1)=ss(1-1)+s(l)
        endif
      enddo
.....
      ncu=DBLE(nmax)
      tam=Sqrt(ncu)
      zzzz=DBLE(zzz)

c   impresión de tama\`nc, porcentaje, entropía total

      WRITE(*,*) tam, ss(nmax)
.....
      STOP
      END

```

Este proceso se ejecuta para obtener los archivos de datos que serán analizados

```
\#!/bin/csh
set limj = 100
set j = 1
while ( "$j" <= "$limj" )
./xesca -o -b -R 50 -C 50 -r 1 -G 3000 > ../xxr/xxx+"$j"
d j++
end
exit
```

.....

En este proceso los datos, obtenidos anteriormente, son analizados por el ejecutable h del programa ENTROPY. Los archivos de salida son graficados.\

```
#!/bin/csh
set limj = 100
set j = 1
while ( "$j" <= "$limj" )
./h < xxx+"$j" >> ../entropias/entropias50
rm xxx+"$j"
d j++
end
exit
```

# Referencias

- [1]. Exploring Complexity : An Introduction. by Gregoire Nicolis, Ilya Prigogine (Contributor), G. Nicolis. 1989
- [2]. Introduction to Nonlinear Science. by G. Nicolis. 1995
- [3]. Synergetics : an introduction : nonequilibrium phase transitions and self-organization in physics, chemistry, and biology. by H. Haken. 1983 Springer-Verlag
- [4]. Cellular Automata and Complexity. Stephen Wolfram. 1994 Addison-Wesley Publishing Company (1994)
- [5]. From Clocks to Chaos; The Rhythms of Life by Leon Glass, Michael C. Mackey. 1988
- [6]. Y.B.Chernyak, A.B.Feldman, R.J.Cohen. Correspondence between discrete and continuous models of excitable media: Trigger waves. Phys. Rev. E **55** 3 (1997)
- [7]. André Longtin, Dante R. Chialvo. Stochastic and Deterministic Resonances for Excitable Systems. Phys. Rev. Lett. **81** 18 (1998)
- [8]. Kyoung J. Lee. Wave Pattern Selection in an Excitable System. Phys. Rev. Lett. **79** 15 (1997)
- [9]. Yuri B. Chernyak. Steady state plane wave propagation speed in excitable media. Phys. Rev. E **56** 2 (1997)
- [10]. Bakhtier Vasiev, Florian Siegert, Cornelis Weijer. Multiarmed spirals in excitable media. Phys. Rev. Lett. **78** 12 (1997)
- [11]. A. Hagberg, E. Meron. Propagation failure in excitable media. Phys. Rev. E **57** 1 (1998)
- [12]. M. Andrecut. A simple three-states cellular automaton for modelling excitable

media. *I. J. Mod. Phys. B.* **12** 5 (1998)

[13]. Mircea Andreicut. Spiral waves in a cyclic cellular automaton model for excitable media. *Mod. Phys. Lett. B.* **11** 30 (1997)

[14]. A. B. Feldman, Y. B. Chernyak, R. J. Cohen. Wave-front propagation in a discrete model of excitable media. *Phys. Rev. E* **57** 6 (1998)

[15]. Richard A. Gray, Arkady M. Pertsov, José Jalife. Spatial and Temporal organization during cardiac fibrillation. *Nature* **392** 75-78 (1998)

[16]. Bayly, P. V.; KenKnight, B. H.; W. M. Smith. Spatial organization, predictability and determinism in Ventricular Fibrillation. *Chaos* **8**, 1, 103-115 (1998)

[17]. Maria G. Signorini; Sergio Cerruti; Diego di Bernardo. Simulation of Heartbeat Dynamics: A Nonlinear Model. *Int. J. bifurcation and Chaos*, **8**(8), 1725-1731 (1998)

[18]. José Jalife, Richard A Gray, Gregory E Morley, Jorge M Davidenko. Self-organization and the dynamical nature of ventricular fibrillation. *Chaos* **8**(1), 79-93 (1998)

[19]. V Krinsky, A Pumir. Models of defibrillation of cardiac tissue. *Chaos* **8**(1), 188-203 (1998)

[20]. Peng-Sheng Chen, Alan Garfinkel, James N Weiss. Computerized mapping of fibrillation in normal ventricular myocardium. *Chaos* **8**(1), 127-136 (1998)

[21]. Madison S Spach. The stochastic nature of cardiac propagation due to the discrete cellular structure of the myocardium. *Int. J. Bifurcation and Chaos* **6**(9), 1637-1656 (1996)

[22]. B H Mitchell, A H Bailey, John Anderson. Cellular Automaton Model of Ventricular Fibrillation. *IEEE Trans Bio Eng* **39**(3) 253-259 (1992)

[23]. Evans S. J., Hastings H. M., Nangia S. Ventricular Fibrillation: one spiral or many. *Proc r Soc Lond B Biol Sci*, 1998 Nov; **265**(1411): 2167-70

[24]. Alain Karma, Herbert Levine, Xiaoqin Zou. Theory of pulse instabilities in electrophysiological models of excitable tissues. *Physica D* **73**, 113-127 (1994)

[25]. D. T. Kaplan, J. M. Smith, B Saxberg, R. J. Cohen. Nonlinear Dynamics in Cardiac Conduction. *Math Biosci* **90**, 19-48 (1988)

[26]. A. Pumir, V. Krinsky. Unpinning of a rotating wave in cardiac muscle by an electric field. *J Theor Biol* **199**, 311-319 (1999)

[27]. L. Glass, P. Hunter, A McCulloch. *Theory of Heart*. Springer-Verlag (1991)

- [28] Leon Glass, "Nonlinear dynamics of physiological function and control", *Chaos* 1 (3)247 (1991)
- [29] Martin Gerhardt, Heike Schuster, "A cellular automata model of excitable media", *Physica D* 46 (1990)392-415
- [30] Francesc Rocher, XESCA-1.0 XWindow Extended Simulator of 2D Cellular Automata. <http://gnomics.udg/rocher/xesca>