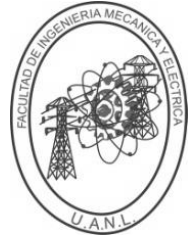


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



Simulación Computacional de Nanomateriales

Dr. Satu Elisa Schaeffer

Práctica 2

Autómata celular

María de los Angeles Martínez Rodríguez

1490139

Ciudad Universitaria, N. L., 21 de agosto del 2017

Introducción

El “juego de la vida” es una representación de un autómata celular en dos dimensiones, este autómata se representa por una matriz booleana, cuyo valor cero “0” representa una célula muerta (celda blanca), mientras que el valor uno “1” representa una célula viva (celda negra). La supervivencia de cada celda es determinada por las 8 celdas que se encuentran alrededor (vecindad), y la condición de supervivencia es para cada celda el número de celdas vivas en su vecindad debe ser igual a 3.

```
sum(vecindad) - actual[fila, columna] == 3)
```

Meta

Determinar el número de iteraciones en función de la probabilidad inicial de celda viva que dura la simulación conocida como “autómata celular” sin que se mueran todas las celdas

Código

```
> library(parallel)
> cluster<-makeCluster(detectCores()-1)
> resultados<-data.frame()
>
> for(p in 1:9){
+   vivir<-parSapply(cluster,1:9,function(x){x/10})
+   clusterExport(cluster,"vivor")
+   dim<-10
+   num<-dim**2
+   v<-c()
+
+   for (corrida in 1:100){
+
+     actual <-matrix(sample(c(0,1),num ,prob=c(1-vivor[p],vivor[p])),rep1
ace=TRUE), nrow=dim, ncol=dim)
+     suppressMessages(library("sna"))
+     png(paste("P",vivor[p],"C",corrida,"Paso_0",".png",sep="_"))
+     plot.sociomatrix(actual, diaglab=FALSE, main="Inicio")
+     graphics.off()
+
+     paso <- function(pos) {
+       fila <- floor((pos - 1) / dim) + 1
+       columna <- ((pos - 1) %% dim) + 1
+       vecindad <- actual[max(fila - 1, 1) : min(fila + 1, dim),
+                         max(columna - 1, 1): min(columna + 1, dim)]
+       return(1 * ((sum(vecindad) - actual[fila, columna]) == 3))
+     }
+   }
```

```

+ clusterExport(cluster, "dim")
+ clusterExport(cluster, "paso")
+
+
+ for (iteracion in 1:15) {
+   clusterExport(cluster, "actual")
+   siguiente <- parSapply(cluster, 1:num, paso)
+   if (sum(siguiente) == 0) {
+     break;
+   }
+   actual <- matrix(siguiente, nrow=dim, ncol=dim, byrow=TRUE)
+   salida = paste("P",vivir[p],"C",corrida,"Paso",iteracion,".png", se
p="_")
+   tiempo = paste("Paso", iteracion,sep=" ")
+   png(salida)
+   plot.sociomatrix(actual, diaglab=FALSE, main=tiempo)
+   graphics.off()
+ }
+
+ v<-c(v,iteracion-1)
+ }
+
+ resultados<-rbind(resultados,v)
+ }
>
> rownames(resultados)<-vivir
>
> stopCluster(cluster)
>
> png("Probabilidad.png")
> boxplot(data.matrix(resultados), use.cols=FALSE,
+         xlab="Probabilidad", ylab="Paso",
+         main="Iteraciones en f(probabilidad de vida)")
> graphics.off()

```

Desarrollo del código

El código R presentado en la sección anterior es resultado de modificaciones al código R utilizado como ejemplo en clase [1]. Estas modificaciones, presentadas en color morado, fueron realizadas con los siguientes objetivos:

1. Declarar la tabla "resultados" para guardar el número de iteraciones durante las cuales la matriz "actual" cuenta con al menos una celda viva.
2. Crear un elemento cíclico "p" de 9 repeticiones, el cual permitirá cambiar el valor de la probabilidad inicial de vida.
3. Crear el vector "vivir" para obtener los valores porcentuales posibles para la probabilidad inicial de celdas vivas.
4. Crear un elemento cíclico "corrida" de 100 repeticiones, con el fin de obtener valores
5. Modificar el valor de la probabilidad inicial de celdas vivas dentro de la matriz "actual" utilizando el comando *sample* y como argumentos: los valores de 0 y 1, el

- número de valores requeridos (num), la probabilidad de vida en función a la posición del vector “vivir” y el comando *replace* para repetir dichos valores más de una vez.
6. Declarar el vector “v” para guardar la cantidad de pasos o iteraciones posibles para cada matriz, para después poder unirlos en la tabla “resultados”.
 7. Evitar la sobre escritura de archivos “.png”.
 8. Agregar nuevos elementos al vector “v” después de cada “corrida”.
 9. Agregar filas nuevas en la tabla de datos “resultado”, cada fila representa el cambio en la probabilidad inicial de vida “vivir”.
 10. Cambiar el valor por default en el eje x de la gráfica caja-bigote y asociarlo con la probabilidad inicial de vida “vivir”.

Las líneas de código en color rojo pueden omitirse para la comodidad y optimización del código.

Resultados

Dentro de este experimento se omitieron las probabilidades de 0% y 100% de celdas vivas a causa de la obviedad del resultado, es decir, con una probabilidad “0%” el número de pasos posibles es cero, mientras que en una probabilidad 100%, el número de pasos posibles es cero puesto que sobrepoblación de celdas vivas impide el nacimiento de nuevas celdas. La *figura 1* muestra el número de pasos posibles en función a la probabilidad inicial de celdas vivas dentro de la matriz, dicha probabilidad está en el rango de 10% a 90%.

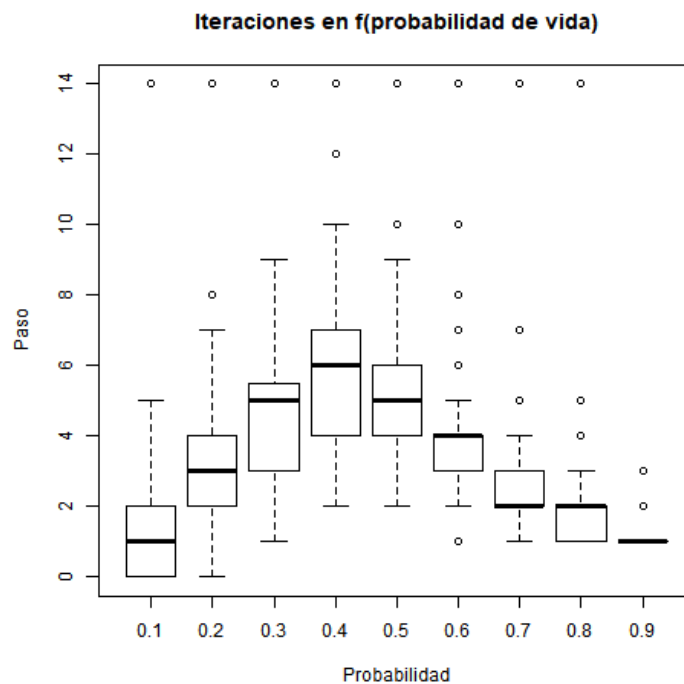


Figura 1 Gráfico caja-bigote que muestra el número de iteración en función a la probabilidad inicial de células vivas para una matriz de 100 celdas

En ambos extremos de dicho rango se observa que el número de pasos posibles sin que se mueran todas las celdas no es mayor a 1, en la mayoría de los casos; esto se explica de la siguiente manera: en probabilidades iniciales de vida bajas, las celdas vivas se encuentran demasiado alejadas unas de otras impidiendo cumplir la condición de 3 celdas vivas vecinas para continuar subsistiendo, mientras que en probabilidades de vida muy altas no existe espacio vacío suficiente para dispersar las celdas vivas causando vecindades de más de 3 celdas vivas lo cual limita la existencia de dicha celda en la iteración. En esta simulación, específicamente en la probabilidad 10%, se encuentran algunos arreglos posibles que permiten el aumento en la cantidad de pasos antes de que mueran las celdas a pesar de la baja cantidad de celdas vivas. Esto indica que para ese escenario las celdas vivas iniciales, las cuales son colocadas al azar, se encontraban muy cerca una de la otra.

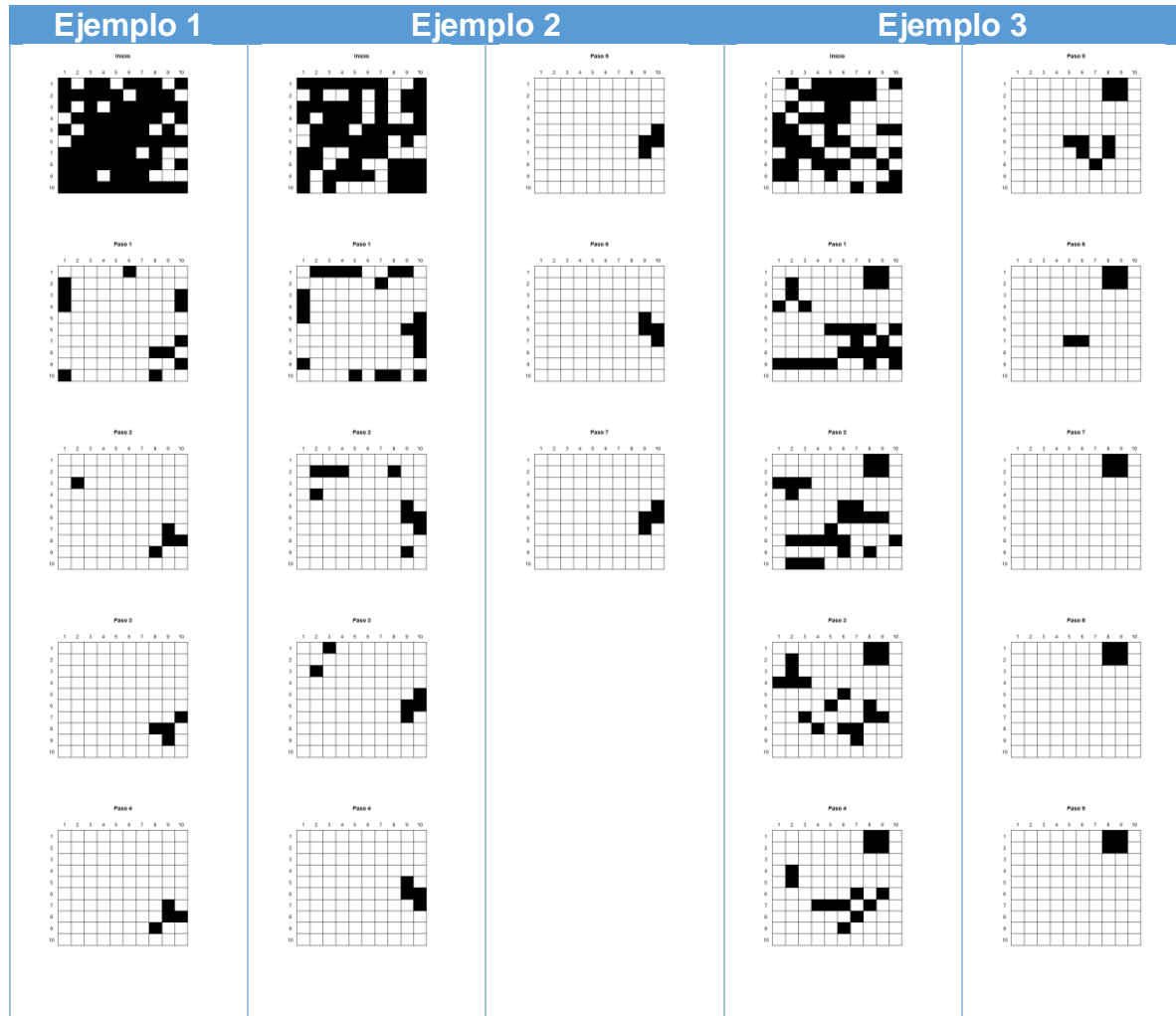
El aumento en la probabilidad inicial de vida conlleva un aumento en la cantidad de celdas vivas dentro de la matriz, creciendo la posibilidad de que dichas celdas se encuentren dentro de una misma vecindad. Esto permite que más celdas cumplan con la condición establecida, permitiendo así el incremento en el número de pasos posibles para cada probabilidad. Esto se puede observar en el rango de 20% a 40%. En este último, se encuentra el valor máximo de pasos posibles para una matriz de 10x10, el cual es 10 pasos.

A partir del 50% se observa una disminución en el número de pasos, como se explicó en el párrafo anterior a mayor probabilidad inicial de vida, más celdas vivas se encuentran dentro de la matriz, provocando una mayor concentración de celdas vivas lo cual incumple la condición de sólo 3 celdas vivas vecinas, probando la “muerte” de dichas celdas en la siguiente iteración. En el [anexo A](#) se pueden encontrar ejemplos de la evolución de la matriz “actual” para probabilidades de vida inicial de 10% a 90%. Además, se pueden observar dos ejemplos animados de dicha evolución para cada probabilidad en la carpeta “[GIF](#)” de la subcarpeta “P2” de este repositorio.

A pesar de que existió un aumento significativo en el número máximo de pasos antes de que todas las celdas mueran, se observa que no existe una gran diferencia en la media de pasos para cada la probabilidad ya que los valores son muy parecidos. Para aumentar el número de pasos antes de que las celdas mueran se recomienda aumentar la dimensión de la matriz. Aquí se puede encontrar la gráfica para una matriz de 20x20.

A partir de una probabilidad de 20% y hasta 80% se encuentran valores atípicos en nuestro experimento, esto puede explicarlo observando las gráficas producidas por código. En la *tabla 1* se muestran algunos ejemplos de arreglos de figuras geométricas cíclicas dentro de la matriz, estos arreglos cumplen con la condición previamente establecida y generan una geometría simétrica en el siguiente paso de la simulación, desencadenando un ciclo “infinito” el cual sólo se interrumpe cuando se termina el número de iteraciones preestablecidas en el código. En la carpeta “[GIF](#)”, pueden encontrarse animaciones en formato “.gif” para visualizar de manera dinámica estas anomalías.

Tabla 1 Crecimiento celular cíclico.



Referencias

[1] Schaeffer, S. (2017). P2 — R paralelo — Schaeffer. [online] Elisa.dyndns-web.com. Available at: <http://elisa.dyndns-web.com/teaching/comp/par/p2.html> [Accessed 22 Aug. 2017].

Anexo A

Tabla 1. Crecimiento celular en función a la probabilidad inicial de células vivas.

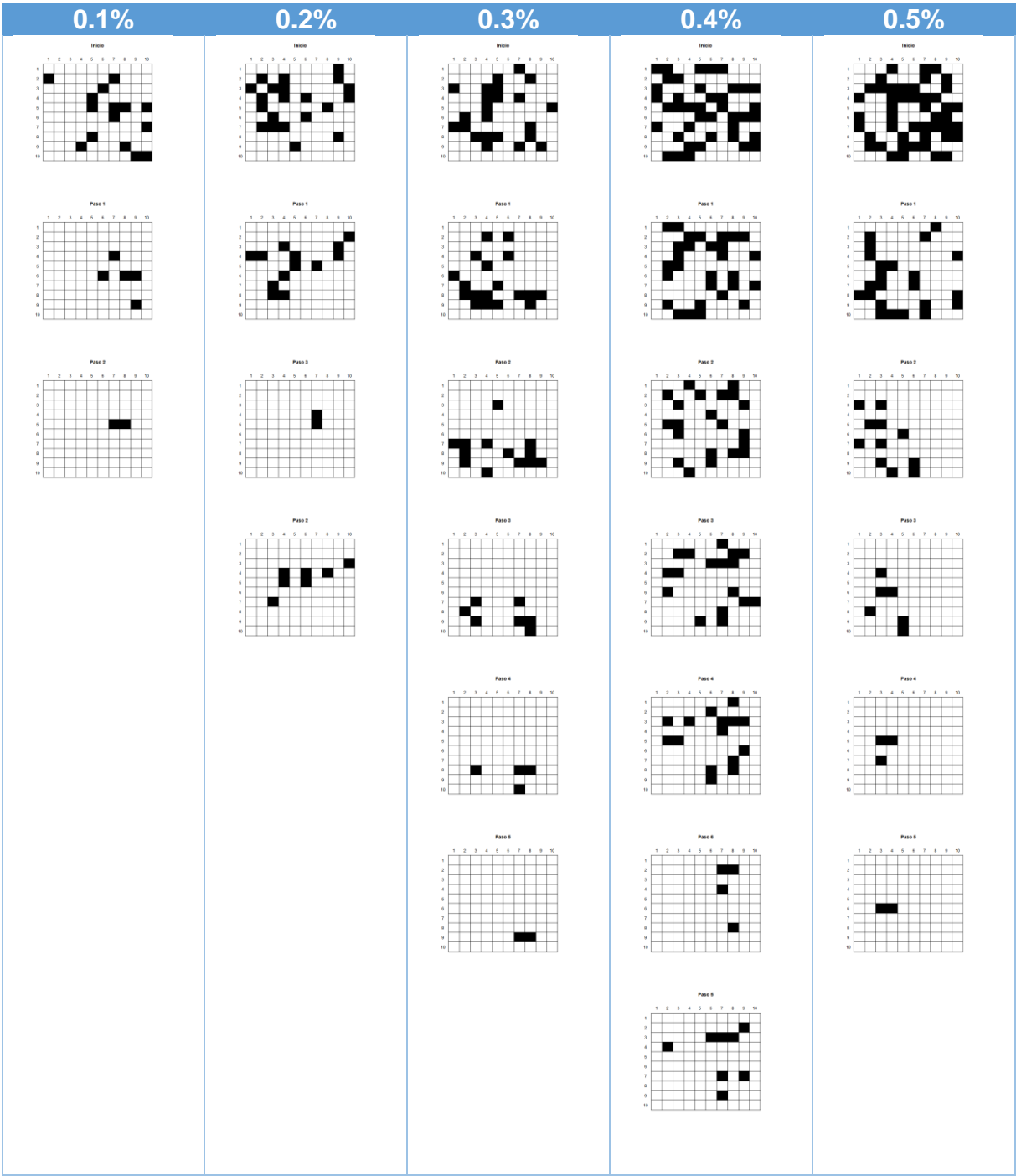


Tabla 1 Crecimiento celular en función a la probabilidad inicial de células vivas (continuación).

