

Introducción

En la ciencia y en la ingeniería de materiales la palabra fractura implica mucho más que una simple separación violenta de un objeto; especialmente a nivel industrial dónde las fracturas son comunes en algunos mecanismos mecánicos como los engranes o ejes están sometidos a una gran fuerza motriz durante el proceso, que los desgasta y puede ser el inicio de una fractura al no disipar de manera adecuada la energía mecánica. Este tema es de vital importancia, tanto que existe una ciencia dedicada exclusivamente a formación y la propagación de grietas y en la cual se invierte una gran cantidad de dinero para comprender este fenómeno y así poder controlarlo.

En esta práctica se simula la propagación de una grieta para matrices con un diferente número de dimensión. Con el fin de recrear una celda inicial con cierta cantidad de núcleos se utilizó el concepto de una matriz de $n \times n$ de dimensión y una celda voronoi. Se asume que las grietas se propagan con mayor facilidad en la frontera que en el interior de núcleo y no existe una dirección de propagación preferencial.

Meta

Examinar de manera sistemática el efecto que el número de semillas y el tamaño de la zona tienen en la distribución de los largos de las grietas que se forman.

Código

```
> library(ggplot2)
> resultados<-data.frame()
> ciclos<-100
> varn<-seq(10,50,10)
>
> for (k in seq(10,50,10) ){
+
+   for (n in varn){
+
+     zona <- matrix(rep(0, n * n), nrow = n, ncol = n)
+     x <- rep(0, k) # ocupamos almacenar las coordenadas x de las semillas
+     y <- rep(0, k) # igual como las coordenadas y de las semillas
+
+     for (semilla in 1:k) {
+       while (TRUE) { # hasta que hallamos una posicion vacia para la semi
+         fila <- sample(1:n, 1)
+         columna <- sample(1:n, 1)
+         if (zona[fila, columna] == 0) {
+           zona[fila, columna] = semilla
+           x[semilla] <- columna
```

```

+         y[semilla] <- fila
+         break
+     }
+ }
+ }
+
+ celda <- function(pos) {
+     fila <- floor((pos - 1) / n) + 1
+     columna <- ((pos - 1) %% n) + 1
+     if (zona[fila, columna] > 0) { # es una semilla
+         return(zona[fila, columna])
+     } else {
+         cercano <- NULL # sin valor por el momento
+         menor <- n * sqrt(2) # mayor posible para comenzar la busqueda
+         for (semilla in 1:k) {
+             dx <- columna - x[semilla]
+             dy <- fila - y[semilla]
+             dist <- sqrt(dx^2 + dy^2)
+             if (dist < menor) {
+                 cercano <- semilla
+                 menor <- dist
+             }
+         }
+         return(cercano)
+     }
+ }
+
+ suppressMessages(library(doParallel))
+ registerDoParallel(makeCluster(detectCores() - 1))
+ celdas <- foreach(p = 1:(n * n), .combine=c) %dopar% celda(p)
+ stopImplicitCluster()
+ voronoi <- matrix(celdas, nrow = n, ncol = n, byrow=TRUE)
+ rotate <- function(x) t(apply(x, 2, rev))
+ png(paste("p4", "d", n, "s", k, "s.png", sep="_"))
+ par(mar = c(0,0,0,0))
+ image(rotate(zona), col=rainbow(k+1), xaxt='n', yaxt='n')
+ graphics.off()
+ png(paste("p4", "d", n, "s", k, "c.png", sep="_"))
+ par(mar = c(0,0,0,0))
+ image(rotate(voronoi), col=rainbow(k+1), xaxt='n', yaxt='n')
+ graphics.off()
+
+ limite <- n # grietas de que largo minimo queremos graficar
+
+ inicio <- function() {
+     direccion <- sample(1:4, 1)
+     xg <- NULL
+     yg <- NULL
+     if (direccion == 1) { # vertical
+         xg <- 1
+         yg <- sample(1:n, 1)
+     } else if (direccion == 2) { # horiz izr -> der
+         xg <- sample(1:n, 1)
+         yg <- 1
+     } else if (direccion == 3) { # horiz der -> izq
+         xg <- n
+         yg <- sample(1:n, 1)
+     }
+ }

```

```

+     } else { # vertical al reves
+       xg <- sample(1:n, 1)
+       yg <- n
+     }
+     return(c(xg, yg))
+   }
+
+   vp <- data.frame(numeric(), numeric()) # posiciones de posibles vecin
os
+   for (dx in -1:1) {
+     for (dy in -1:1) {
+       if (dx != 0 | dy != 0) { # descartar la posicion misma
+         vp <- rbind(vp, c(dx, dy))
+       }
+     }
+   }
+   names(vp) <- c("dx", "dy")
+   vc <- dim(vp)[1]
+
+   propaga <- function(replica) {
+     # probabilidad de propagacion interna
+     prob <- 1
+     dificil <- 0.99
+     grieta <- voronoi # marcamos la grieta en una copia
+     i <- inicio() # posicion inicial al azar
+     xg <- i[1]
+     yg <- i[2]
+     largo <- 0
+     while (TRUE) { # hasta que la propagacion termine
+       grieta[yg, xg] <- 0 # usamos el cero para marcar la grieta
+       largo <- largo + 1
+       frontera <- numeric()
+       interior <- numeric()
+       for (v in 1:vc) {
+         vecino <- vp[v,]
+         xs <- xg + vecino$dx # columna del vecino potencial
+         ys <- yg + vecino$dy # fila del vecino potencial
+         if (xs > 0 & xs <= n & ys > 0 & ys <= n) { # no sale de la zona
+           if (grieta[ys, xs] > 0) { # aun no hay grieta ahi
+             if (voronoi[yg, xg] == voronoi[ys, xs]) {
+               interior <- c(interior, v)
+             } else { # frontera
+               frontera <- c(frontera, v)
+             }
+           }
+         }
+       }
+       elegido <- 0
+       if (length(frontera) > 0) { # siempre tomamos frontera cuando hay
a
+         if (length(frontera) > 1) {
+           elegido <- sample(frontera, 1)
+         } else {
+           elegido <- frontera # sample sirve con un solo elemento
+         }
+         prob <- 1 # estamos nuevamente en la frontera

```

```

+       } else if (length(interior) > 0) { # no hubo frontera para propag
ar
+       if (runif(1) < prob) { # intentamos en el interior
+       if (length(interior) > 1) {
+       elegido <- sample(interior, 1)
+       } else {
+       elegido <- interior
+       }
+       prob <- difiCIL * prob # mas difiCIL a la siguiente
+       }
+     }
+     if (elegido > 0) { # si se va a propagar
+     vecino <- vp[elegido,]
+     xg <- xg + vecino$dx
+     yg <- yg + vecino$dy
+     } else {
+     break # ya no se propaga
+     }
+   }
+
+   if (largo >= limite) {
+     png(paste("p4","d",n,"s",k,replica,".png",sep="_"))
+     par(mar = c(0,0,0,0))
+     image(rotate(grieta), col=rainbow(k+1), xaxt='n', yaxt='n')
+     graphics.off()
+   }
+   return(largo)
+ }
+ #for (r in 1:10) { # para pruebas sin paralelismo
+ #   propaga(r)
+ #}
+ suppressMessages(library(doParallel))
+ registerDoParallel(makeCluster(detectCores() - 1))
+ largos <- foreach(r = 1:ciclos, .combine=c) %dopar% propaga(r)
+ stopImplicitCluster()
+
+ largos<-matrix(largos,byrow=TRUE)
+ largos<-cbind(largos,n,k)
+ resultados<-rbind(resultados,largos)
+ }
+ }
>
>
> colnames(resultados)<-c("Longitud","Dimension","Semillas")
> resultados$Dimension <- as.factor(resultados$Dimension)
> resultados$Semillas <- as.factor(resultados$Semillas)
>
> png("Variacion_dimension.png")
> ggplot(data=resultados, aes(x = Semillas, y= Longitud,fill = Dimension)
) +
+   geom_boxplot(position=position_dodge(1))+
+   facet_grid(.~Dimension)+
+   scale_y_continuous(name="Longitud") +
+   scale_x_discrete(name="Semillas")
>

```

```
> dev.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 azul, fueron realizadas con los siguientes objetivos:

1. Activar el paquete *ggplot2* para crear gráficos atractivos.
2. Crear el *data.frame* “*resultados*” con el fin de guardar los valores obtenidos en cada replica.
3. Modificar el número de réplicas para “*largos*” de manera rápida, evitando buscar dentro del extenso código.
4. Cambiar el vector “*largos*” en una matriz que se llena por filas.
5. Combinar las columnas “*largos*”, “*n*” y “*k*” con el fin de tener la información en una sola matriz.
6. Agregar nuevos valores de “*largos*”, “*n*” y “*k*” para cada valor de las últimas variables.
7. Nombrar las columnas del *data.frame* “*resultados*”.
8. Cambiar una columna de resultado como un factor.
9. Indicar el nombre de salida de la gráfica posterior.
10. Graficar un *boxplot* donde el eje x representa el número de semillas y el eje y el largo de la grieta, dividido en subgrupos acorde a la dimensión de la celda.
11. Separar las cajas del gráfico evitando se grafiquen justo en el borde de la caja anterior.
12. Separar el gráfico por “*dimensión*” de la celda, para mejorar la visualización de los datos.

Nota: Si sólo se tiene interés en los valores de la longitud de la grieta, es posible eliminar las líneas de código en color rojo; las cuales son útiles para tener una imagen de dicha grieta.

Resultados

La figura 1 muestra los valores de longitud de grieta “*largos*” en función al número de semillas “*k*” dentro de la celda y a la dimensión “*n*” de la misma. Primero, se discutirá el caso especial para las celdas de dimensión 10×10, en ellas se observa que el aumento en el número de semillas “*k*” dentro de la celda no repercute de manera dramática en la longitud de la grieta, es decir, tanto la el tamaño de la caja y el valor de la media es similar para los diferentes valores de semillas; esto es

causado por el limitado espacio en la cual la grieta puede propagarse debido al menor número de posiciones posibles (cien) dentro de la celda. Esto incrementa la posibilidad de encontrarse rodeada por posiciones “grieta” o encontrarse en el límite de la matriz.

En el caso de las demás dimensiones, especialmente para la dimensión 30×30 e incluso para la dimensión 20×20, se observa una tendencia al aumento en la media del largo de la grieta en función al número de semillas, en otras palabras, son directamente proporcionales una con respecto a la otra. Este resultado es coherente con lo que se esperaba puesto que un mayor número de semillas significa a su vez una mayor cantidad de posiciones “frontera”, dentro de las cuales se estableció que la grieta se propaga con mayor facilidad que si se encontrara en posiciones “interior”

Para las últimas dos celdas el resultado se encuentra fuera de lo esperado, pero se puede tener una explicación, a menor número de semillas en una matriz más grande, la probabilidad de iniciar en una posición “interior” es mayor a la probabilidad de comenzar en una frontera, lo cual limita la propagación de la misma explicando la disminución en del rango del vector “*largo*”.

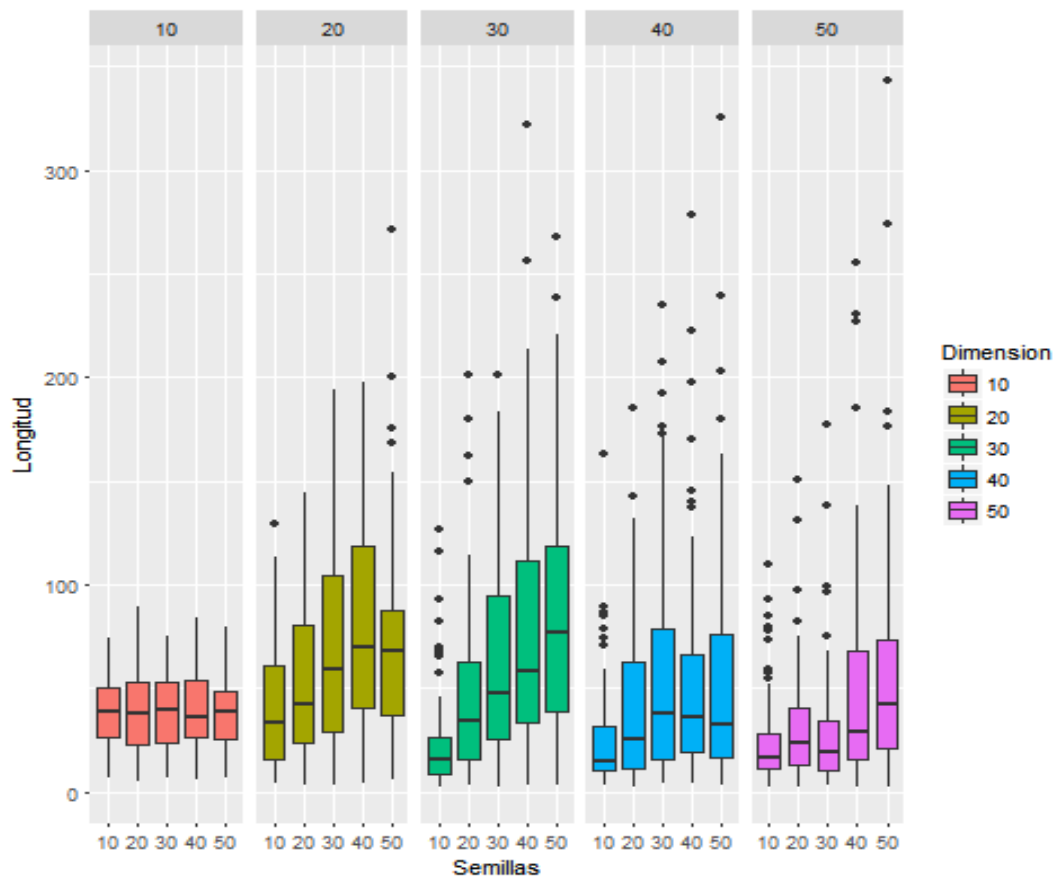


Figura 1 Gráfico caja -bigote de la magnitud del vector grieta con respecto al número de semillas (k) iniciales en la matriz y de las dimensiones (n).

Referencias

[1] Schaeffer, S. (2017). P4 — R paralelo — Schaeffer. [online] Elisa.dyndns-web.com. Available at: <http://elisa.dyndns-web.com/teaching/comp/par/p.html> [Accessed 29 Aug. 2017].