

Práctica No 9: Interacción entre partículas

Héctor Hernán Montes García

10 de octubre de 2017

Resumen

En la presente práctica estudiamos un problema de interacción entre partículas cargadas con masas no uniformes que se mueven en un cuadrado de lado unitario. El modelo subyacente, aunque simplificado, nos permite entender las bases para diseñar simulaciones en la que interesa saber la dinámica de movimiento de un conjunto de partículas que se afectan entre sí. En primer lugar se estudia el efecto de la masa sobre la velocidad de desplazamiento, en segundo lugar se construye una visualización de la dinámica de movimiento sin considerar colisiones entre partículas o barreras, pero asumiendo que las partículas se deslizan sobre la frontera, una vez que ésta es alcanzada, se asume para esta parte que partículas de masa mayor deberán poseer mayor radio. Finalmente se proponen modelos alternativos para la simulación considerando colisiones pero sin llegar a una implementación práctica.

1. Planteamiento del problema

El problema principal consiste en simular un movimiento de n partículas en un cuadrado unitario, tomando como fuente de movimiento las fuerzas de atracción y repulsión presente entre ellas en razón de una carga eléctrica presente. Los siguientes son los supuestos considerados en la construcción del modelo:

- Las partículas poseerán valores de carga eléctrica que fluctuarán entre $[-1,1]$. La carga eléctrica se construye escalando convenientemente valores aleatorios distribuidos como una normal estándar en forma tal que se restrinjan al rango de variación especificado. Esto permitirá contar con una serie de variaciones de cargas **independientes** que tendrá su utilidad posterior.
- Las partículas poseerán valores de masa que fluctuarán para conveniencia gráfica en el intervalo $[1,5]$. Nuevamente se obtendrán estos valores mediante escalado apropiado de valores aleatorios independientes.¹
- Para inicializar una corrida de la simulación se posicionarán partículas en forma aleatoria en el cuadrado de lado unitario. Las coordenadas de posición nuevamente se asumirán normales estándar con escalamiento apropiado al intervalo $[1,0]$.

¹La distribución de partida se irá modificando conforme se requiera en la experimentación

- La magnitud de las fuerzas de repulsión y de atracción entre dos partículas específicas se asumen proporcionales al producto de sus cargas, e inversamente proporcionales a la distancia entre sus centros de masa.
- Se asume que las partículas son sólidas y de densidad de masa uniforme. Por la naturaleza plana del problema, en realidad cada partícula representa un disco.
- La fuerza total ejercida por todas las partículas sobre una partícula en específico se calcula como la suma vectorial de las fuerzas individuales, usando principio de superposición.
- A efectos de simplificación se asume que la magnitud del desplazamiento por unidad de tiempo será proporcional a la fuerza total actuando sobre la partícula.²
- Se asume que partículas con más masa deberán ser menos propensas a moverse, y partículas con menos masa se moverán con más libertad, más específicamente: se asume que la velocidad de desplazamiento es inversamente proporcional a la masa de la partícula.³

Una vez incorporados los supuestos de alguna forma en la simulación, se solicita verificar mediante análisis gráficos y estadísticos que efectivamente la velocidad de desplazamiento se ve afectada en función de la masa en forma proporcional. En el estudio de la relación masa-velocidad deberá tomarse en cuenta que ésta última no sólo es afectada por la masa, sino también por la carga y las posiciones relativas de las partículas entre sí.

La segunda parte del problema consiste en lograr una representación gráfica realista de la situación usando alguna librería apropiada (se sugirió *ggplot2*, sin embargo veremos que para efectos de mejor control del radio de la partícula en función de su masa la instrucción *symbol* puede ser más efectiva).

La tercera parte del problema consiste en resolver problemas de solapamiento entre partículas, es decir, partículas que tiendan a superponerse una sobre otra deberán ser ajustadas en su desplazamiento, ya sea mediante una estrategia de rebote o de vibración. Para esta parte sin embargo no ofrecemos implementación, sino una serie de pasos de cómo podría lograrse.

A continuación una descripción de la solución dada a los tres problemas.

2. Resultados

Todas las experimentaciones se han realizado en R, y para el efecto se usó una computadora con sistema operativo Windows 10 Pro de 64bits y procesador x64 Intel(R) core(TM) i5-4210U, CPU@ 1.70GHz 2.40GHz y memoria instalada de 6.00GB. Una llamada a la función *detectcores(logical=FALSE)* de R, permitió además constatar que la computadora posee dos núcleos físicos disponibles. A

²Un supuesto más realista sería que la aceleración de la partícula es proporcional a la fuerza total ejercida como lo plantea la segunda ley de Newton, y que dicha aceleración es en esencia constante para una unidad de tiempo suficientemente pequeña, así el desplazamiento sería una función cuadrática del tiempo

³Nuevamente, un supuesto más realista es que la aceleración es inversamente proporcional a la masa

continuación se describe las modificaciones más importantes realizadas al código suministrado, el cual debió ser adaptado para incorporar el componente de masa.

2.1. Algoritmo ajustado para incluir efectos de la masa

Los siguientes son los cambios efectuados sobre el código original:

- El primer cambio fue agregar un parámetro adicional al *data.frame* de partículas, para incluir una variable que contenga su masa. La masa se definió como una variable aleatoria en la que diferentes distribuciones de partida fueron consideradas (distribución normal estándar, distribuciones gamma con parámetros de forma y escala: (1,2), y distribución beta con parámetros de forma y escala: (5,1) que permiten lograr distribuciones de partículas con masas simétricas, mayor densidad de masas pequeñas, o mayor densidad de masas grandes respectivamente. Los valores de masas bajo cada distribución siempre se escalan a valores entre [1,5].
- El segundo cambio fue considerar ocasionalmente una distribución de cargas idénticas en magnitud pero de signos variables. El signo se decide con una distribución binomial de parámetro $p = 0,5$ y n igual al número de partículas. Esto se traduce en definir los valores de carga usando la instrucción `sample(c(-1,1),n,replace = TRUE)`
- Se insertó una pequeña modificación sobre el factor de escala aplicado para lograr desplazamientos no muy grandes. Originalmente se estaba tomando la función *máximo* aplicada a todas las componentes de las fuerzas individuales obtenidas para una iteración dada, lo que lógicamente equivale a extraer *máximo* sobre componentes de fuerza en las direcciones tanto del eje horizontal x como del eje vertical y , y luego *máximo* sobre estos valores a lo largo de todas las partículas. Contrario a esto, se plantea extraer *máximo* a lo largo de las magnitudes de las fuerzas de todas las partículas en una iteración dada. Esto en código se traduce en el par de líneas de la figura 1.
- Se plantea un factor de escala para la determinación de los radios de la partícula en función de la masa, en forma tal que el radio resulte proporcional a la masa pero no supere un valor máximo de radio especificado. Esto se traduce en las líneas de código de la figura 2
- Se modifica la instrucción básica para trazado de gráfica, de acuerdo con las líneas de código de la figura 3. Comentado se ofrece el código equivalente en *ggplot2*, esto con el fin de facilitar la comparación. Note que mientras en *symbol* se pasa el radio en forma explícita al objeto círculo, en *ggplot2* el radio es pasado al parámetro *size*. Este parámetro aunque es usado por *ggplot2* para crear los círculos de un radio proporcional a dicho tamaño, altera la magnitud misma del radio en forma automática para lograr una apariencia más estética del gráfico que sin embargo es inconveniente a efectos de lograr una representación más realista. Este problema no lo presenta la función *symbol*.
- La parte paralelizada también se modificó para usar la librería *parallel* la cual ha ofrecido hasta ahora tiempo de computación más pequeños sobre el

```

1 mag_f<-sqrt(f[1,]^2+f[2,]^2)
2 delta <- 0.05 / max(mag_f)

```

Figura 1: Cambio en factor de escala para desplazamiento máximo

```

mmax <- max(p$m)
mmin <- min(p$m)
p$m <- 1+5*((p$m-mmin)/(mmax-mmin)) # las masas entre 1 y 2
factor_radio<-0.025/max(p$m)
p$r <- factor_radio*p$m

```

Figura 2: Introducción de factor de escala para radio máximo en función de masas

mismo problema de paralelización respecto a los ofrecidos por *DoParallel*. Esto implicó cambiar un poco la referencia al objeto *f* que almacena las fuerzas sobre las partículas en cada iteración, tal como se evidencia en el código de la figura 4. La función *fuerza* se sigue conservando en su versión original. Note que la modificación para incorporar efectos de la masa en el desplazamiento se puede dejar externa a esta función porque depende sólo de la masa de la partícula *i* considerada. Esto se hace explícito en la estructura del *parSapply*. También note que se ha modificado la forma de guardado para conservar los desplazamientos. Sin embargo, esto supone la actualización adicional de las posiciones que en todo caso es paralelizable también. Guardar los desplazamientos sin modificaciones de frontera (sin recortes de las funciones *max(min())*) era importante para no introducir factores extraños en el momento de estudiar el efecto de la masa sobre éstos.

A continuación se presentan los resultados del estudio de relaciones entre la masa y el desplazamiento. Se ha procedido bajo el siguiente esquema de experimentación.

1. Crear masas con distribuciones de partida normales independientes, escalar convenientemente, usarla en conjunto con cargas con distribuciones de partida normales también escaladas convenientemente. Estudiar la media de los desplazamientos logrados por cada partícula a lo largo de una

```

1 png(paste("p9_t", t1, ".png", sep=""), width=700, height=300)
2 print(with(p, symbols(x=x, y=y, circles=r, inches=FALSE,
3                       ann=T, bg=colores[p$g+6], fg=NULL, xlim = c
4                               (-0.1,1),
5                               ylim = c(-0.1,1), main=paste("Paso", t1, sep
6                               = " "))))
7 graphics.off()
8 #print(ggplot(data=p, aes(x=x, y=y, size=r, col=colores[p$g+6]))+
9 #  geom_point(show.legend= FALSE)+xlab("Coordenada x")+
10 #  ylab("Coordenada y")+xlim(c(-0.1,1))+ylim(c(-0.1,1))+
11 #  ggtitle(paste("Paso", t1, sep = " ")))

```

Figura 3: Cambio en función de graficación

experimentación con 100 pasos. Si los desplazamientos dependen de las posiciones de las partículas a lo largo de los 100 pasos, se espera que el hecho de promediarlos a lo largo de los pasos compense el efecto no controlado de la posición sobre el desplazamiento. Por otro lado, si los desplazamientos dependen de la distribución del conjunto total de cargas, también se espera que al ser estas asignadas en forma independiente la operación de promedio también compense sus efectos no controlados a lo largo de la media de los 100 pasos. Es posible que no obstante esto sea insuficiente para lograr una buena explicación de la variabilidad de los desplazamientos, puesto que sólo la masa no está decidiendo dichos valores. Cuando mucho se logrará tener una idea del efecto marginal de la masa sobre el desplazamiento.

2. Modificar la distribución de carga, haciéndola de magnitud igual para todas las partículas, pero de signo variable. El signo se elige como si de un evento Bernoulli de parámetro $p = 0,5$ se tratara. Estudiar nuevamente los desplazamientos medios a lo largo de 100 pasos de la experimentación. Se espera que controlar la carga permita una mejor explicación del desplazamiento promedio en función de la masa de la partícula.
3. Repetir los pasos 1 y 2 para al menos 3 configuraciones diferentes de distribuciones de masas, y extraer conclusiones.

Los resultados de estos análisis se representan en las rectas de regresión de recíprocos de masas vs desplazamientos medios indicadas en las figuras 5,6, y 7 para distribuciones de masa de partida normal, gamma y beta respectivamente. Recuerde que la distribución gamma usada es tal que genera mayor densidad de masas pequeñas, en tanto que la distribución beta genera mayor densidad de masas grandes. La densidad de masas en la normal es pareja. Cada figura establece un comparativo con control y sin control de cargas.

Finalmente la implementación de una visualización donde el tamaño de la partícula es proporcional a su masa en la forma ya con anterioridad detallada se ofrece en el gif animado presente en el repositorio del presente informe. Se puede consultar el archivo *interaccion.R*, para referencia al código usado. Por otro lado, el archivo *experimentos.R*, ofrece una serie de modelos lineales construidos para cada media de desplazamiento vs el tamaño de las partículas, en cada escenario de experimentación. Los resultados no se ponen acá por economía de espacio pero en resumen se tiene que:

- El R^2 ajustado de la regresión para masas normales pasa de ser 0,3233 en el caso del modelo con cargas normales a 0,9565 en el caso del modelo con cargas binarias de idéntica magnitud pero signos variables.
- El R^2 ajustado de la regresión para masas gamma pasa de ser 0,3106 en el caso del modelo con cargas normales a 0,8457 en el caso del modelo con cargas binarias.
- El R^2 ajustado de la regresión para masas gamma pasa de ser 0,3182 en el caso del modelo con cargas normales a 0,8579 en el caso del modelo con cargas binarias.
- Todos los modelos con cargas binarias resultan significativos al nivel de 0,05 o incluso inferior.

```

1  for (iter in 1:tmax) {
2    clusterExport(cluster, "fuerza")
3    f<-parSapply(cluster, 1:n, fuerza)
4    mag_f<-sqrt(f[1,]^2+f[2,]^2)
5    delta <- 0.05 / max(mag_f) # que nadie desplace una paso muy
      largo
6    clusterExport(cluster, "delta")
7    clusterExport(cluster, "f") #Exportamos fuerzas calculadas al
      cluster
8    delta_x[, iter] <- parSapply(cluster, 1:n, function(i) {(1/p[i,] $m)
      *delta * f[1, i]})
9    delta_y[, iter] <- parSapply(cluster, 1:n, function(i) {(1/p[i,] $m)
      *delta * f[2, i]})
10   desplazamientos[, iter] <- sqrt(delta_x[, iter]^2+delta_y[, iter
      ]^2)
11   clusterExport(cluster, "delta_x")
12   clusterExport(cluster, "delta_y")
13   p$x <- parSapply(cluster, 1:n, function(i) {max(min(p[i,] $x+delta_x
      [i], 1), 0)})
14   p$y <- parSapply(cluster, 1:n, function(i) {max(min(p[i,] $y+delta_y
      [i], 1), 0)})
15   clusterExport(cluster, "p") #Actualizamos matriz de posiciones
      para el paso siguiente
16 }

```

Figura 4: Paralelización de cálculos usando librería *parallel*

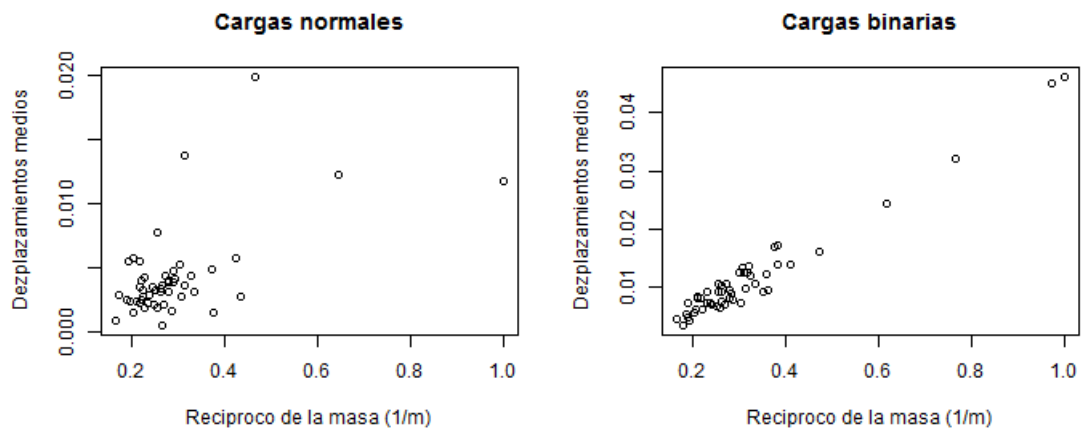


Figura 5: Relación desplazamiento vs recíproco de la carga. Masas normales

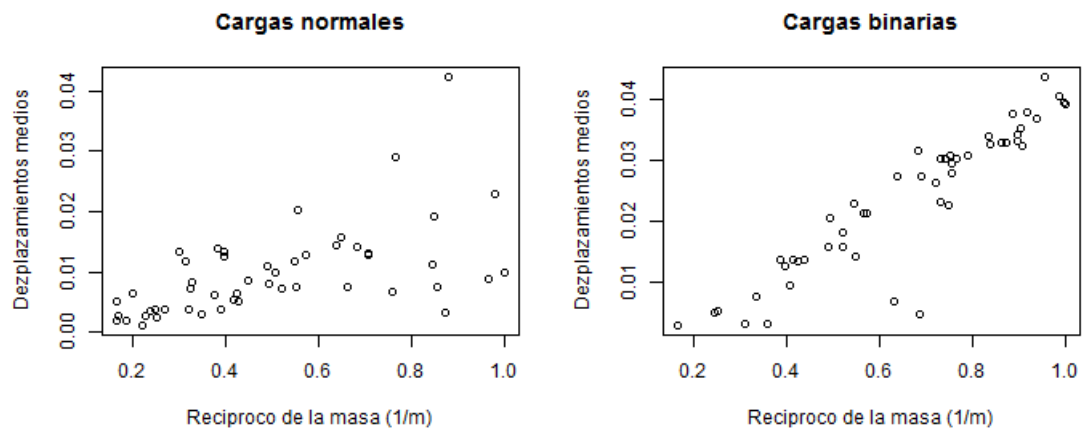


Figura 6: Relación desplazamiento vs reciproco de la carga. Masas gamma

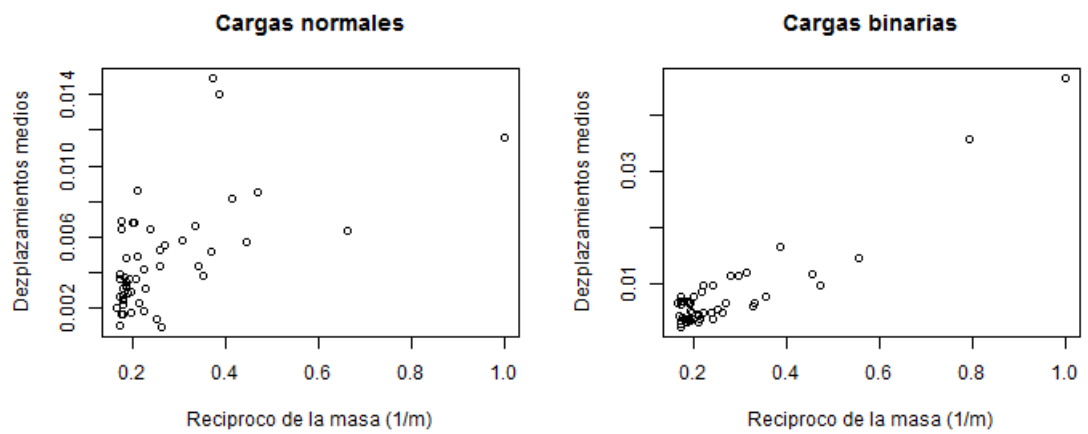


Figura 7: Relación desplazamiento vs reciproco de la carga. Masas beta

2.2. Conclusiones

Controlar la carga de las partículas mejora el nivel de asociación entre las medias de los desplazamientos y los recíprocos de las masas, como era de esperarse de acuerdo con el modelo físico subyacente. Esto siempre se cumple, incluso para sistemas con gran acumulación de cargas positivas o negativas.