

# Generación Aleatoria de partículas en un cubo mediante el Algoritmo Random Sequential Adsorption (RSA)

Julio Enrique Santana Lora ([julio.santana@umcc.cu](mailto:julio.santana@umcc.cu)) UMCC  
Reyder Cruz De la Osa ([reyder@uci.cu](mailto:reyder@uci.cu)) UCI

## RESUMEN

En el presente trabajo desarrollamos el estudio y la implementación de dos variantes del algoritmo Random Sequential Adsorption para la generación de partículas en una celda cúbica unitaria, de gran utilidad en la modelación de materiales compuestos, en la rama de la mecánica.

Luego importamos la geometría generada por nuestra aplicación para el software de Elemento Finito ANSYS para poder realizar un mallado de la misma lo cual permitirá el estudio de las propiedades efectivas del material representado por nuestra geometría

Palabras clave: *random sequential adsorption, elemento finito, volumen representativo, fracción volumétrica*

## **INTRODUCCIÓN**

La disciplina de Mecánica de los Sólidos tiene un problema clásico, el cual es la determinación de las diferentes propiedades efectivas de un material compuesto. Esto consiste en el estudio de la reacción del mismo ante acciones realizadas sobre él, tales como la aplicación de fuerzas o cargas, el sometimiento a cambios de temperaturas, entre otras. Cuando solo se requiere el estudio en una dimensión o dos, varios métodos analíticos bien difundidos pueden ser utilizados para el cálculo de dichas propiedades. Bastante distinta es la tarea cuando se requieren tres dimensiones. Para lograr llevar a cabo el estudio en este caso, se necesita contar con una descripción de la geometría del material en cuestión, lo más cercana posible a la realidad, para posteriormente, realizar los cálculos necesarios para llevar a cabo la simulación sobre dicha geometría, utilizando métodos numéricos. Como su nombre lo indica un material compuesto está formado por dos o más constituyentes, en los que existe uno que contiene a los demás el cual se conocerá como medio y los restantes serán vistos en formas de partículas situadas de acuerdo a alguna función de distribución aleatoria dentro del mencionado substrato o medio.

Nuestra tarea en concreto consiste en crear la aplicación que se encargue de generar dicha geometría. A su vez es de gran importancia, lograr de alguna manera que el resultado producido por nuestro programa pueda ser usado como entrada en este caso, del software profesional de Elemento Finito Ansys que se encargará de llevar a cabo la resolución numérica del problema, con toda comodidad pues sería automático el proceso de carga de la geometría del material por parte de dicho software.

Estudios preliminares a este trabajo, han demostrado la eficacia del algoritmo Random Sequential Adsorption (RSA) para la generación aleatoria de partículas inmersas dentro de otro material, así como la posibilidad de obtener una configuración de las mismas que sea adaptable para realizar una discretización de una geometría mediante la conversión de la misma a mallas, que es un requisito que impone Ansys para el trabajo con las geometrías importadas hacia él.

La idea de realizar este trabajo tiene como antecedente el cálculo de propiedades efectivas para materiales compuestos 3D mediante la combinación del Método de Homogeneización Asintótica y el Método de Elemento Finito realizado en [3], donde se consideró como geometría del compuesto un cubo con una sola inclusión esférica. En [3] la geometría se genera haciendo uso de la interfaz gráfica, de elementos geométricos y de operaciones booleanas del programa Ansys. Esto, como es de suponer, constituye una tarea larga, tediosa y repetitiva, debido a que se realiza manualmente; además de que por mucho esfuerzo que se le dedicara por esta vía no se pueden obtener configuraciones complejas de la geometría del material, ya que más que una inclusión esférica se

busca tener varias partículas con posiciones aleatorias. Actualmente valiéndonos de nuestra aplicación este trabajo es llevado a cabo con sencillez y de manera automática se puede pasar a trabajar en Ansys con una geometría generada solo pidiéndole al usuario algunos parámetros de la aplicación.

Aunque en la bibliografía consultada se encontró la utilización del RSA para la generación de este tipo de geometrías, las herramientas de software desarrolladas por otros desarrolladores con este fin, están ubicadas más bien en el marco académico y no han sido liberadas para el uso de ellas por terceras personas, existiendo en sus artículos solo una breve mención al funcionamiento de las mismas. De aquí la importancia de contar con una implementación del RSA en nuestro propio entorno.

### **RSA Clásico.**

El primer acercamiento al problema fue mediante la implementación del algoritmo RSA en su versión clásica, ampliamente documentada en diferentes artículos sobre el tema.

Como ya se ha adelantado de alguna forma el RSA de manera general puede ser visto como la generación y localización aleatoria de un número de partículas no solapadas o superpuestas unas con otras, dentro de un substrato o medio  $d$ -dimensional. Por partículas podemos entender cualquier tipo de objeto geométrico con una dimensión similar a la del substrato en el que será absorbida la misma. De esta forma en la versión en la que el medio es unidimensional descrita por primera vez por Renyi (1954) [5] y más conocida como parqueo aleatorio de carros, por la similitud con este problema, se toma como medio una recta y se tratan de ubicar segmentos aleatoriamente sobre la misma.

En dos y tres dimensiones la mayoría de los estudios teóricos del algoritmo, contemplan las partículas como esferas en el caso tridimensional y círculos o discos en el caso bidimensional.

### **Planteamiento del Algoritmo Random Sequential Adsorption(RSA)**

1. *Se quieren ubicar  $n$  partículas en un medio  $d$ -dimensional:*

*Para cada uno de los valores  $i$  entre 1 y  $n$*

- a) *Seleccionar un punto  $\mathbf{o}_i$  en la superficie de manera aleatoria. La distribución que debe seguir la selección de estos puntos debe ser especificada por el usuario. La uniforme es la tomada por defecto en la aplicación desarrollada por nosotros aunque se da al usuario la posibilidad de definir nuevas funciones de distribución.*

- b) *Situar el centro de la partícula  $P_i$  en el punto  $o_i$  y comprobar que ninguna de las otras partículas previamente aceptadas  $P_1...P_{i-1}$  intercepte ninguna fracción de la región que le pertenece con  $P_i$ . La formulación de esta comparación es típica de cada geometría, para una mejor comprensión véase el ejemplo de abajo. Hay que resaltar aquí que si  $o_i$  esta muy cercano al límite del medio parte de la región ocupada por la partícula pudiera ubicarse fuera del substrato contenedor, lo que sería un problema completamente normal, que sería resuelto con las Condiciones de Límite Periódico que se verán más adelante.*
- c) *Si no ocurre solapamiento entre  $P_i$  y ninguna de las partículas comprobadas, se dice que esta partícula actual es aceptada y es adsorbida por el medio. De ocurrir sobreposición se desecha la partícula  $P_i$  y el valor de su punto  $o_i$  y se regresa al paso 1.a para realizar un nuevo intento de deposición.*

*Por poner un ejemplo sobre cómo se realiza la comprobación de solapamiento, en el caso en que el medio fuera un volumen cúbico y las partículas fueran esferas, se debería chequear en la esfera  $P_i$  con cada una de las otras esferas adsorbidas la siguiente condición:*

*Llamando  $P_j$  a la esfera que se chequea con  $P_i$  actualmente y  $o_j$  y  $o_i$  a sus centros.*

$$\|o_i - o_j\| \geq P_j.\text{radio} + P_i.\text{radio}$$

El caso particular de nuestra aplicación se considera al medio como tridimensional, representado por un cubo. Las partículas por su parte se toman predefinidamente como esferas, aunque se dan facilidades para si en un futuro se programa la lógica de nuevas geometrías, poderlas cargar dinámicamente solamente especificando algunos datos de las mismas, tales como la ubicación física, o sea la librería de enlace dinámico que la contiene.

Es necesario añadir dadas las condiciones del problema que nos compete que a la hora de realizar una modelación matemática de materiales compuestos, es de vital importancia tener en cuenta las características de este como microestructura. Sin embargo preservar la información concerniente a la distribución de cada una de las porciones microscópicas individuales de un material determinado conlleva a un grado de exhaustividad que se torna impracticable debido a la cantidad de cálculos numéricos que son exigidos. Es necesario entonces definir el concepto de Volumen Representativo del Elemento (RVE) que no es más que la mínima porción de la micro-estructura del elemento

que es estadísticamente representativa del compuesto como un todo, para que los elementos de la teoría macroscópica puedan ser aplicados a la muestra.

Drugan y Willis[1] demostraron que el tamaño del RVE para el problema de nuestra consideración es bastante pequeño en el sentido estadístico. Esto significa que tomando un pequeño fragmento cúbico del material, representado por unas pocas esferas, se aproximara bastante bien el valor real de la solución del material visto macroscópicamente.

Dado lo dicho anteriormente y dando por hecho que el cubo que tomaremos por medio es una porción microscópica que se halla inmersa dentro del material, se debe garantizar que la misma cumpla con las Condiciones de Límites Periódicos( Periodic Boundary Conditions PBC en inglés). Esto significa que se quiere exigir que el cubo contenga cierta periodicidad con respecto a sus caras opuestas para que pueda ser replicado hasta el infinito mediante traslaciones en cada una de las direcciones de los ejes cartesianos llenando todo el espacio, quitando además el problema de tratar con el borde del material, debido a que ahora este puede ser visto como un medio sin fin. En otras palabras, el material visto macroscópicamente estaría formado por infinitas repeticiones del pequeño fragmento cúbico tomado.

Descrito matemáticamente, si una de nuestras partículas estuviera localizada en el punto  $o_1 = (x, y, z)$ , podemos asumir ahora que existen infinitas partículas localizadas en los puntos:

$(x + L \cdot i, y + L \cdot j, z + L \cdot k)$  donde  $L$  es la longitud de los lados de la celda y  $i, j, k \in \mathbb{Z}$ .

Cuando en realidad solo guardamos los datos y realizamos los cálculos sobre  $o_1$  con lo que evidentemente se gana en sencillez en la representación.

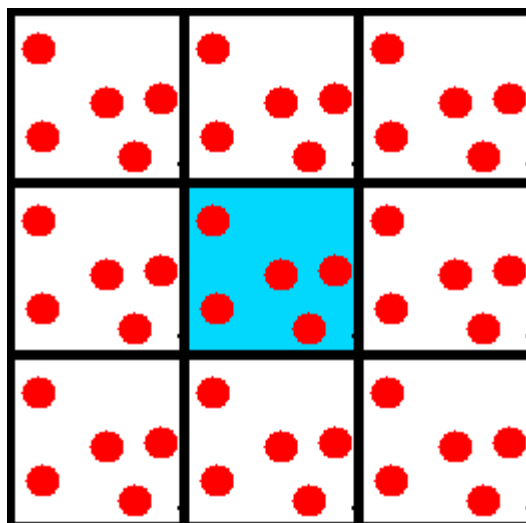


Figura #1: Ejemplo de Aplicación de Condiciones de Límite Periódico

Estas cuestiones introducen un aspecto muy interesante y es que a partir de ahora el concepto de cercanía entre las partículas, no solo debe ser aplicado a las que están en una misma unidad de la celda, sino que debe ser extendido a las celdas vecinas también. De esta forma la definición de Solapamiento de partículas dada en **1.b** debe ser ampliada para incluir el chequeo no solo de las partículas de la celda en cuestión, sino también de las celdas contiguas que por estar muy cercanas al límite con esta pueden presentar solapamientos entre sus partículas. En base a esto un algoritmo fue desarrollado para realizar la extensión al chequeo de solapamiento ya visto, el cual es bastante complejo, pero realiza la comprobación de manera muy eficiente, mejorando el costo algorítmico del planteamiento clásico expuesto en **1.b**.

Hablando de otro aspecto muy importante para todo algoritmo como la condición de parada, le hemos impuesto al nuestro que se alcance una fracción volumétrica específica. El concepto de fracción volumétrica ( $\xi$ ) podría verse como la parte del total del medio que esta cubierta por partículas. O sea el algoritmo se detiene cuando por ejemplo la 0,4 parte del medio este cubierto de partículas. El límite para la mayor fracción volumétrica que se pueda alcanzar en el caso de las esferas esta situado cerca de 0,53. Es conveniente aclarar que dicho limite ha sido demostrado teóricamente, pero puede ser difícil alcanzarlo durante cada simulación que se realice debido a las características aleatorias de la misma, así como a muchos otros factores que entran en juego. Es por esto que nuestro algoritmo también establece un numero máximo de intentos fallidos que de alcanzarse daría por fallido el intento de generación actual, culminando el mismo.

Una vez llevada a cabo la implementación y habiéndose corrido con variaciones en sus parámetros como por ejemplo diferentes fracciones de volumen, se comprobó que el presente algoritmo es muy bueno para obtener fracciones de volumen de hasta 0,35. Una vez por encima de esta, las corridas del mismo se vuelven lentas e inestables. La principal carencia o debilidad que podría ser la causante de este comportamiento sería la manera en que se selecciona la posición para cada nueva absorción de una geometría. Una vez que gran cantidad de partículas han sido adsorbidas, la cantidad de puntos libres en el medio van a pasar a ser una minoría. Esto trae consigo que gran cantidad de generaciones aleatorias del punto  $o_i$  no logren encontrar un espacio libre para realizar una absorción, por lo que en algunas corridas del algoritmo se alcance la cantidad de intentos de localización máxima definida sin que se pueda realizar esta acción, cuando aún, aunque escasos, existen espacios libres en el medio.

### **RSA mejorado**

La siguiente variante que hemos denominado en nuestro trabajo como RSA mejorado, fue desarrollada a partir de las ideas brindadas por Segurado y

Llorca[7]. Más que una implementación completamente nueva se puede decir que es una extensión del método clásico que ya fue enunciado, pues necesita en varias etapas de su funcionamiento o bien el uso de la totalidad de los pasos definidos anteriormente o una parte de estos. El funcionamiento de la nueva modificación del RSA podría quedar de la siguiente manera.

#### Algoritmo RSA Mejorado

1. Leer  $\xi$ . Según el valor de la misma se pueden dar dos casos:

- Si  $\xi \leq 0,3$  se aplica el RSA clásico, el cual como es conocido resuelve este problema de manera muy eficiente. Luego se toma como resultado de la corrida la respuesta obtenida y se termina.
- Si  $\xi \geq 0,3$  se expanden los lados del cubo, manteniendo el volumen de las partículas a ser generadas constantes, de manera que se logre que  $\xi$  alcance un valor menor que 0,3 y se pueda aplicar la versión RSA Clásico sin complicaciones.
- Aplicar RSA Clásico
- **⊗ Mientras  $\xi$  no alcance el valor inicial que tenía antes de ser alargado el cubo, se realiza lo siguiente:**
- Para  $i$  desde 1 hasta  $N$  (Número de Partículas)
 
$$o_i = o_i^{old} \cdot C_f \text{ (donde } C_f \text{ es un factor de compresión } < 1 \text{ definido por el usuario. Esto significa que se comprimen las coordenadas del centro de cada partícula)}$$
- Luego de la compresión se chequea el posible solapamiento entre las partículas que puede haber aparecido como consecuencia. Si se halla que dos partículas cualquiera  $i$  y  $j$  se solapan, se calcula un vector  $\vec{u}$  que tendrá la forma que es respuesta a la inecuación

$$\vec{u} \cdot (\vec{o}_i \cdot \vec{o}_j) \leq 0$$

La idea geométrica de este vector es que tendrá un sentido opuesto al lugar donde ocurre el solapamiento. Luego el centro de la partícula  $i$  se reubica según la expresión

$$\vec{o}_i = \vec{o}_i^{old} + \vec{u} \cdot \text{radio} \cdot s1_c$$

Donde  $s1_c$  es una nueva constante definida por el usuario. Este proceso se repite mientras halla sobreposición entre dos pares cualesquiera. Cuando ninguno de los volúmenes de las partículas está

intersectado con otro se puede decir que la fracción de volumen a variado de la siguiente manera:

$$\xi = \frac{\xi_{old}}{C_f^3}$$

Esto la acercará más al valor inicial que tenía antes de expandirse el cubo. Llegado a este punto la iteración actual culmina y se vuelve a  $\otimes$ .

Si en la eliminación de solapamiento, se realizarán un número grande de intentos sin éxito, es necesario realizar una vuelta atrás o *backtracking* al algoritmo hasta la ultima  $\xi$  en que no hubo solapamiento y realizar algunos cambios en parámetros del algoritmo, tales como las disminución del factor de compresión  $C_f$ . Todo esto garantizará la convergencia del algoritmo hacia el resultado que deseamos.

De este nueva versión del RSA, puede ser dicho a manera de resumen que sirvió para generar configuraciones de partículas con una  $\xi$  de hasta 0.45 y mayor en ocasiones, lo que mejora lo obtenido por el RSA clásico.

No obstante, se le puede criticar que más lento en su funcionamiento, pues la vuelta atrás incluida en él trae consigo que el ciclo 1) deba ser repetido en una gran cantidad de ocasiones.

### **Aspectos de Implementación**

La implementación del software fue llevada a cabo aprovechando las potencialidades brindadas por la plataforma .NET, por intermedio del lenguaje insignia de la misma, C#. Se implementó una aplicación con interfaz Windows, así como una biblioteca de vinculos dinámicos (.dll) que contiene la lógica del algoritmo, para que de esta forma sea reusable.

Gran parte de las potencialidades brindadas por el framework de .NET fueron usadas. Un ejemplo fehaciente fue el extensivo aprovechamiento de las estructuras de datos definidas en *System.Collection* para la programación de las dos variantes del algoritmo RSA.

Uno de los propósitos iniciales fue que la aplicación permitiera la incorporación dinámica de tipos de partículas programados según su necesidad después de que el programa ya se encontrara en uso. Para esto fue necesario el diseño riguroso de una jerarquía de partículas que describa con claridad las



propiedades geométricas de las mismas y que además sirva como la base para efectuar la herencia de las nuevas geometrías a incorporar. De esta forma, si en un futuro se quisiera que el algoritmo generara elipsoides en vez de esferas, bastaría con definir una clase que describa a esta geometría y que a su vez herede de partícula, empaquetarla en nuevo ensamblado(.dll), y luego especificársela a la aplicación por medio de la interfaz grafica que se encuentra preparada para tal fin. Para lograr esta funcionalidad fue de vital importancia el uso de las clases relacionadas con el mecanismo de Reflection, que nos permiten instanciar objetos, cuando solo se llega a conocer su tipo solo en tiempo de ejecución.

Otra herramienta presente en .NET que fue de gran ayuda fue el conjunto de clases que se encarga de manipular ficheros XML. Algunos ficheros de este tipo fueron definidos para asegurar la persistencia en el disco de los parámetros iniciales de las corridas del algoritmo, así como varios otros datos.

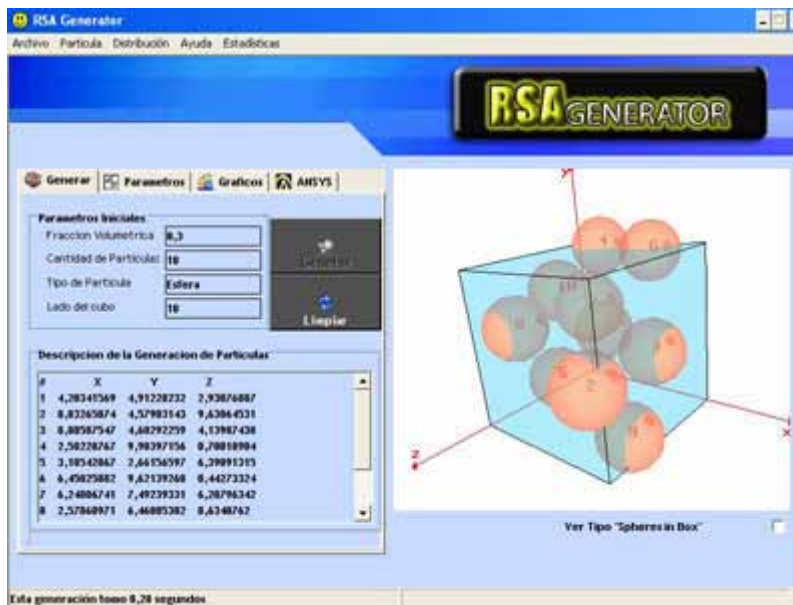


Figura #2: Vista de la aplicación

Por otra parte el API de funciones gráficas OpenGL fue extensamente usado en la programación de la interfaz visual. Se hizo necesario el uso de wrappers o “envoltorios” de este API para portarlo a .NET que fueron programados por un tercer fabricante llamado TAO.

Por último para cumplir con el objetivo de que la generación de las partículas, efectuada en C# fuera utilizable en el software profesional ANSYS, fue necesario hacer un extensivo estudio del funcionamiento del mismo. Ansys es un paquete de software de análisis por elementos finitos de propósito general ampliamente usado en la industria y la investigación. Es propiedad de la empresa Ansys .Inc. Es usado para simular la respuesta de sistemas físicos a diferentes

acciones como sobrecarga estructural, efectos termales y electromagnéticos entre otros, ya que el mismo tiene como fin predecir como será el comportamiento de los sistemas así como determinar los posibles riesgos que el mismo pueda tener. Está estructurado en varios módulos que a su vez pueden ser vistos como un todo a través de la interfaz gráfica, pero pueden ser usados independientemente uno del otro. Entre ellos están los de resolución, los de post- procesamiento y pre-procesamiento. Precisamente este último es el que resulta de especial interés para el desarrollo de nuestra aplicación. El mismo presenta dos maneras de introducir la geometría del material a modelar: una mediante interfaz gráfica y otra mediante comandos batch. La opción de los comandos a todas luces, es lo que se necesitaba para lograr nuestro objetivo. Concretamente nuestra aplicación se encarga de crear un fichero con una secuencia batch que describe las características geométricas brindadas resultado de la salida del algoritmo RSA. Este fichero puede ser usado sin ninguna dificultad en ANSYS, lo cual constituye uno de los objetivos primordiales que nos planteamos lograr desde un principio.

## **CONCLUSIONES**

Se logró obtener una aplicación que posee implementadas dos variantes del algoritmo Random Sequential Adsorption. De la misma forma, aunque sólo se implementó el caso de las esferas como geometría, se da la posibilidad de extender el programa con otras geometrías para las partículas.

Se logró importar satisfactoriamente la geometría generada por nuestra aplicación hacia el software de Elemento Finito ANSYS. De esta forma se pudo realizar un mallado eficiente de la geometría obtenida.

Se exponen las ventajas y desventajas de cada uno de los métodos implementados. Conociendo esto y la configuración de partículas que se debe generar, se deberá utilizar uno u otro en dependencia del que mejor se adapte a la situación particular a resolver en cada momento.

Algunas ideas tales como la generación de otras geometrías, o la inclusión de otras variantes del RSA que sean capaces de garantizar una mayor fracción de volumen serían líneas interesantes para darle continuidad al trabajo actual.

## **REFERENCIAS**

[1] Drugan. W.J, Willis.J.R. .*A Micromechanics-Based Nonlocal Constitutive Equation and Estimates for Representative Volume Element Size for Elastic Composites*.1996.*J.MechPhys Solids*.497-524

[2] Evans J.W. *Random and Cooperative Sequential Adsorption*. 1993. *Review of Modern Physics*. 1281-1324

[3] León, Ángela *Cálculo de propiedades efectivas para materiales compuestos en 3D: Método de Elemento Finito combinado con el Método de Homogeneización Asintótica*. Tesis de Doctorado de la Universidad de La Habana, Enero 2007

[4] Penrose M. , Yukich J.E. . *Limit Theory for Random Sequential Packing and Deposition*. 2002. *The annals of applied probability*. 272-301

[5] Renyi A.. *On a One-Dimensional Random Space Filling Problem*. 1958

[6] Rintoul M., Torquato S.. *Reconstruction of Structures of Dispersions*. 1997. *Journal of colloidal and interface science*. 467-476

[7] Segurado J., Llorca J.. *A Numerical Aproximation to Th Elastic Properties of Sphere-Reinforced Composites*. 2002. *Journal of mechanics and Physics of solids*. 2108-2121

[8] Wang J.. *A Fast Algorithm for Random Sequential Adsorption of Discs*. 1991

[9] Widam B. *Random Sequential Addition of Spheres to a Volume*. 1966. *The journal of chemical physics*. 3888-3894