

Estudio de procesos de generación, transferencia y
recombinación de carga y transferencia de energía en
polímeros conjugados nano-estructurados.
Modelización, simulación y validación.

Daniel Arnaldo Bellomo

Directora: Dra. Dolores Isabel Rexachs del Rosario

Codirector: Dr. Marcelo Naiouf

Trabajo Final Integrador que será presentado para obtener el grado de
Especialista en Cómputo de Altas Prestaciones y Tecnologías Grid

Facultad de Informática

Universidad Nacional de La Plata

Versión 2015.12.10.00





Índice general

1. Introducción	7
1.1. Motivación	7
1.2. Objetivos	8
1.3. Vinculación institucional	9
1.4. Integrantes	9
1.5. Resumen	10
1.6. Publicaciones relacionadas a este trabajo	10
1.7. Formación de RRHH en programación científica	11
1.8. Estructura de la monografía	11
2. Marco teórico del fenómeno	13
2.1. Antecedentes	13
2.2. Modelo conceptual	13
2.3. Probabilidad de decaimiento en $[0, t]$	13
2.4. Cálculo del Exciton Diffusion Length 1 dimensión	14
2.5. Cálculo del Exciton Diffusion Length 3 dimensiones	15
3. Estado del arte	19
3.1. Modelización	19
3.2. Simulación	19
3.3. Computación de altas prestación	19
3.4. Lenguajes de programación científica	19

4. Propuesta	23
4.1. Modelo conceptual	23
4.2. Modelo computacional	23
4.3. Simulaciones de Monte Carlo	23
4.4. Simulación de altas prestaciones	23
5. Desarrollo computacional	25
5.1. Diseño del simulador	25
5.2. Tipo de simulación	25
5.3. Tipo de escalabilidad	26
5.4. Paralelismo	26
6. Validación experimental	29
6.1. Diseño de los experimentos	29
6.1.1. Puntos aleatorios en la esfera	29
6.2. Entorno experimental	34
6.3. Experimentos y discusión. (precisión y speedup)	34
6.4. Ciencia reproducible	34
7. Conclusiones y trabajos futuros	35
8. Anexos	37
8.1. Licencia del código	37
Bibliografía	39



**KEEP
CALM**

**It's only a
first draft**

Capítulo 1

Introducción

Los polímeros conjugados son materiales orgánicos semiconductores de gran relevancia debido a su actual aplicación en dispositivos orgánicos-electrónicos tales como celdas solares (Lindstrom C. D. 2006), (Oregan B. 1991), (Gratzel M. 2001), (Bach U. 1998), (Asbury J. B. 2001), diodos emisores de luz (OLED) (Kulkarni A. P. 2004), (Chou P-T. 2007), (Gather M. C. 2011), transistores de efecto campo (OFET) (Mabeck J. T. 2006), (Dimitrakopoulos C. D. 2002), (Sirringhaus H. 2005), (Zaumseil J. 2007), memorias moleculares (Scott J. C. 2007), etc. El modelo típico de la estructura electrónica de polímeros conjugados considera que las excitaciones electrónicas en cada cadena polimérica se limitan a segmentos relativamente cortos (de 5 a 15 monómeros) denominados cromóforos (Lupton J. M. 2010), (Lupton J. M. 2012), (Schwartz B. J. 2008). Estos cromóforos actúan en gran medida de forma independiente de manera tal que una cadena de polímero conjugado es vista como un sistema multi-cromofórico en el que la mayoría de la complejidad electrónica es consecuencia de las interacciones entre cromóforos.

Consecuentemente, el funcionamiento y desempeño de dispositivos orgánicos-electrónicos basados en polímeros conjugados depende en gran medida de procesos fotofísicos elementales de transferencia de energía (TE), generación, transporte y recombinación de carga (GTRC) que ocurren entre cromóforos en el seno del polímero y en las interfaces orgánicas/inorgánicas (donde ocurre el contacto eléctrico entre polímero y electrodos) (Zhu X-Y. 2009).

1.1. Motivación

Muchos de estos dispositivos orgánicos-electrónicos se componen de películas ultra delgadas de polímero amorfos con alta a moderada heterogeneidad espacial y temporal. Esta heterogeneidad dificulta un estudio detallado de procesos TE y GTRC en estos materiales por técnicas convencionales que miden promedio de propiedades. El estudio de procesos TE y GTRC a nano-escala en películas poliméricas ultra-delgadas, nanopartículas (NPs) o moléculas poliméricas individuales utilizando técnicas avanzadas de microscopia de fluorescencia (por ejemplo espectroscopia de molécula/partícula individual) permite desentrañar la hete-

roogeneidad mencionada (Lupton J. M. 2010), (Lupton J. M. 2012), (Schwartz B. J. 2008). En particular estas técnicas permiten la determinación de la distribución real de propiedades relevantes mediante su medición de a una partícula por vez en lugar de medir el promedio de un gran conjunto de partículas o moléculas. Debido a que las especies cargadas (polarones, cationes radicales o aniones radicales) en polímeros conjugados resultantes de los procesos GTRC son generalmente no fluorescentes su determinación es indirecta y se basa en:

- a. Desactivación de fluorescencia de estados electrónicos excitados (excitones, cromóforos excitados) mediante especies cargadas (Palacios R. E. 2006), (Palacios R. E. 2007), (Chang Y-L. 2008), (Bolinger J. C. 2008), (R. A. Palacios R. E. Lee K-J. 2009), (Bolinger J. C. 2009), (Fradkin K. 2009), (Chang Y-L. 2009), (G. J. K. Palacios R. E. Chang W. S. 2009) o
- b. Generación de estados electrónicos excitados emisores mediante recombinación de especies cargadas (electroluminiscencia, EL) (Chang Y-L. 2008), (Chang Y-L. 2009) , (Nothaft M. 2012a), (Nothaft M. 2012b)
- c. Así mismo también es posible determinar la TE de excitación entre los cromóforos intrínsecos del polímero y cromóforos dopantes incluidos en la matriz polimérica en pequeña cantidad con el fin de ajustar sus propiedades optoelectrónicas.

Por otro lado para complementar, contrastar y validar los estudios y resultados experimentales es de gran utilidad la creación de modelos computacionales que permitan simular los procesos de TE y GTRC en esos materiales.

1.2. Objetivos

El objetivo general de este trabajo consiste en proponer un modelo (integral) computacional para el estudio de procesos de generación, transferencia y recombinación de carga y transferencia de energía en polímeros conjugados nano-estructurados. Este modelo comprende el proceso de modelización, simulación, validación y técnicas de computación de alto desempeño para calcular la energía de Quenching y poder así continuar las investigaciones del tema que realiza el Laboratorio de Microscopía Óptica Avanzada (LMOA) de la Facultad de Ciencias Exactas Físico-Químicas y Naturales (FCEFQyN) de la Universidad Nacional de Río Cuarto (UNRC).

El LMOA posee un equipamiento de última generación y RRHH de primer nivel. Desarrolla experimentación básica publicando sus trabajos en prestigiosas revistas y congresos. Se presenta el necesidad de comenzar a desarrollar simulaciones computacionales para avanzar en las investigaciones.

Parte de los objetivos es el desarrollo de habilidades y capacidades que permitan optimizar el abordaje multidisciplinar requerido cuando se colabora con profesionales de disciplinas diversas, como es el caso de la investigación básica en el laboratorio.

En este contexto, los temas a abordar comprenden el análisis del sistema, la confección del modelo, validación de los resultados y optimización el código usando técnicas y estrategias de programación paralela y HPC.

1.3. Vinculación institucional

Este proyecto de investigación articula la colaboración entre un grupo de trabajo perteneciente a la UNRC, compuesto por personal técnico, personal docente, alumnos de grado y postgrado a la Facultad de Ciencias Exactas Fisicoquímica y Naturales UNRC; docentes e investigadores del HPC4EAS Research Group perteneciente a la Universidad Autónoma de Barcelona y docentes e investigadores del Instituto de Investigación en Informática LIDI perteneciente a la UNLP. El grupo involucra el trabajo coordinado de 8 personas y cuenta con apoyo económico de SECyT UNRC, ANPCyT-MINCyT Argentina, MINCyT Córdoba y CONICET. Así mismo el proyecto está vinculado mediante la formación de estudiantes a los siguientes programas educativos: Licenciatura en Física y Licenciatura en Química UNRC, Doctorado en Ciencias Químicas UNRC, Doctorado en Física UBA y Maestría en Computación de Alto Desempeño UNLP.

1.4. Integrantes

Alumno

- Daniel Arnoldo Bellomo, alumno de la Maestría en Computación de alto Desempeño (Universidad Nacional de La Plata).

Directores

- Dra. Dolores Isabel Rexachs del Rosario (Universidad Autónoma de Barcelona)
- Dr. Rodrigo Emiliano Palacios (Universidad Nacional de Río Cuarto)
- Dr. Marcelo Naiouf (Universidad Nacional de La Plata)
- Dr. Carlos Alberto Chesta (Universidad Nacional de Río Cuarto)

Colaboradores

Integran el equipo del Laboratorio de Microscopía Óptica Avanzada (LMOA) y participan del presente trabajo: Lic. Rodrigo Ponsio (doctorando), Franco Bellomo y Lucas Bellomo (estudiantes de Lic. en Física).

1.5. Resumen

Se desarrolla una herramienta computacional para calcular la eficiencia de Quenching.

Mediante simulaciones de Montecarlo se pretende estudiar la eficiencia de Quenching para una nanopartícula (NP) determinada. Estas simulaciones van a ser contrastadas con las mediciones experimentales realizadas en el Laboratorio de Microscopia Optica Avanzada (LMOA) de la Universidad Nacional de Río Cuarto (UNRC).

En principio, son tres los experimentos en los que queremos corroborar la eficiencia de Quenching:

1. En el caso de tener los aceptores distribuidos volumetricamente en la NP.
2. Que los aceptores estén sobre la superficie de la NP.
3. En los dos casos anteriores el fotón es generado mediante un laser. Se quiere estudiar que sucede en el caso de que sea generado mediante una electrólisis química.

El LMOA tiene amplia experiencia en la investigación experimental, pero destaco que este es el primer trabajo científico que desarrollan realizando simulaciones computacionales. Para los científicos del laboratorio se crea un área nueva de trabajo, química computacional, por lo que demuestran un especial interes en el trabajo interdisciplinar. La colaboración que brindan para avanzar en este desarrollo es total, dedicando tiempo, atención y respondiendo un sinnumero de preguntas.

1.6. Publicaciones relacionadas a este trabajo

- 11º Encuentro Regional de Probabilidades y Estadística Matemática ERPEN
Departamento de Matemáticas, FCEFQyN, UNRC, 2-3 de diciembre, 2015. ¹
Título: *Transferencia de energía en nanopartículas de polímeros conjugados. Modelización y validación con datos experimentales* (poster) <http://tiny.cc/quc26x>
- XVII Juan José Giambiagi Winter School. Light and light-based technologies.
Departamento de Física J.J. Giambiagi, FCEyN, UBA, 3-7 de agosto, 2015. ²
Título: *Transferencia de energía en nanopartículas de polímeros conjugados. Modelado y contraste con datos experimentales* (poster) <http://tiny.cc/yikz6x>

¹11º Encuentro Regional de Probabilidades y Estadística Matemática (ERPEN), FCEFQyN UNRC <http://intra.exa.unrc.edu.ar/eventos/erpem>

²XVII Giambiagi Winter School, Departamento de Física, FCEyN UBA <http://giambiagi2015.df.uba.ar>

1.7. Formación de RRHH en programación científica

Se trabaja con alumnos de la carrera de Física para desarrollar capacidades de programación científica y HPC aplicadas a física computacional. Esta actividad permite la integración por parte de los alumnos en proyectos reales de la facultad, permitiendo formar parte de grupos de investigación y hacer aportes desde su área específica de formación desarrollando modelos computacionales.

1.8. Estructura de la monografía

El **capítulo 2** desarrolla el marco teórico del fenómeno, resaltando los antecedentes y el modelo conceptual.

El **capítulo 3** muestra los conceptos teóricos de la modelización, simulación, la computación de altas prestaciones y los lenguajes de programación científica que consideramos para este trabajo, justificando nuestra elección.

El **capítulo 4** detalla la propuesta que se lleva a cabo: modelo conceptual, modelo computacional, simulaciones de Monte Carlo y simulaciones de altas prestaciones.

El **capítulo 5** desarrolla el modelo computacional, el diseño del simulador, el tipo de simulación, la escalabilidad y el paralelismo.

El **capítulo 6** desarrolla la validación experimental mostrando el desarrollo de los experimentos y el entorno experimental.

El **capítulo 7** detalla las conclusiones y trabajos futuros.

El **capítulo 8** tiene los anexos (licencia del código).



Capítulo 2

Marco teórico del fenómeno

2.1. Antecedentes

Para realizar el trabajo se toma como antecedente el paper (Jason McNeill 2008)

2.2. Modelo conceptual

2.3. Probabilidad de decaimiento en $[0, t]$

La tasa de cambio de la población de cromóforos excitados N es proporcional a la cantidad de cromóforos excitados, por lo que se puede escribir como

$$\frac{dN}{dt} = -kN \quad (2.1)$$

Donde k es la constante de decaimiento total. La solución de esta ecuación diferencial es

$$N = N_0 e^{-kt} \quad (2.2)$$

Ésta ecuación dice cuantos cromóforos excitados hay en cualquier tiempo t . Queremos determinar cuál es la probabilidad de que un cromóforo decaiga en un intervalo de tiempo $[0, t]$. Para esto, con la ayuda de 2.2, podemos escribir una función que sea el número de decaimientos en el intervalo $[0, t]$ como el número total de cromóforos excitados menos la cantidad de cromóforos existentes en el tiempo t

$$\alpha = N_0 - N_0 e^{-kt} = N_0(1 - e^{-kt}) \quad (2.3)$$

Podemos asignar ahora la probabilidad de un cromóforo de decaer en $[0, t]$ como el cociente de la cantidad de decaimientos en $[0, t]$ y el número total de cromóforos

$$P = \frac{\alpha}{N_0} = 1 - e^{-kt} \quad (2.4)$$

2.4. Cálculo del Exciton Diffusion Length 1 dimensión

Consideremos una caminata aleatoria unidimensional, que en cada paso recorre una distancia ϵ , y que tiene la misma probabilidad de caminar hacia la izquierda que hacia la derecha. La distancia entre el origen y N pasos después es

$$X = \sum_{i=1}^N x_i \quad (2.5)$$

Donde x_i puede ser ϵ con probabilidad p o $-\epsilon$ con probabilidad q , siendo $p = q$. Si queremos ver cuánto vale X promediando en muchas caminatas de este tipo debemos calcular el valor esperado de X . Esto es

$$E(X) = \sum_{i=1}^N E(x_i) = \sum_{i=1}^N (p\epsilon - q\epsilon) = 0 \quad (2.6)$$

Ésto no arroja información respecto al Exciton Diffusion Length. La pregunta que nos hacemos entonces es: ¿cuánto va a diferir típicamente X del su valor esperado? Definimos r como la diferencia entre el valor X de una caminata particular y el valor esperado de X

$$r = X - E(X) = X \quad (2.7)$$

Claramente no tiene sentido preguntarse por el valor esperado de r ya que va a dar cero, de acuerdo con 2.6 y 2.7. Esto es debido a que r puede resultar positivo o negativo con la misma probabilidad, y al sumar estos valores en promedio se compensan, resultando cero su valor esperado. Por esto es útil preguntarse por el valor esperado del valor absoluto (siempre positivo) de r . Para esto calculamos el valor esperado de r al cuadrado como

$$E(r^2) = E((X - E(X))^2) = E(X^2) = E \left[\left(\sum_{i=1}^N x_i \right)^2 \right] = Var(x) \quad (2.8)$$

Esto es por definición la varianza de X . El cuadrado de la sumatoria en 2.8 está compuesto de N términos del tipo x_i^2 , y $N(N-1)$ términos cruzados del tipo $x_i x_j$ con i distinto de j

$$E(r^2) = E \left(\sum_{i=1}^N x_i^2 \right) + E \left(\sum_{i \neq j} x_i x_j \right) \quad (2.9)$$

Notemos que

$$E\left(\sum_{i=1}^N x_i^2\right) = E\left(\sum_{i=1}^N \epsilon^2\right) = E(N\epsilon^2) = N\epsilon^2 \quad (2.10)$$

Y que

$$E\left(\sum_{i \neq j} x_i x_j\right) = 0 \quad (2.11)$$

Debido a que los términos de la sumatoria pueden tener resultados ϵ^2 o $-\epsilon^2$ probabilidad. De (9), (10) y (11) se obtiene que

$$E(r^2) = N\epsilon^2 = Var(X) \quad (2.12)$$

Tomando la raíz cuadrada de (12) podemos calcular la Desviación Standard σ de X

$$\sigma = \sqrt{N}\epsilon \quad (2.13)$$

Ésta es una buena medida de la distancia entre el punto inicial y final de la caminata en N pasos, así que asignamos este valor al Exciton Difussion Length L_D

$$L_D = \sqrt{N}\epsilon \quad (2.14)$$

2.5. Cálculo del Exciton Diffusion Length 3 dimensiones

En tres dimensiones la situación es matemáticamente más compleja pero arroja el mismo resultado. Supongamos una caminata aleatoria tridimensional con la misma probabilidad de caminar hacia todas las direcciones. Supongamos que el largo de cada paso es ϵ , de modo que las componentes cartesianas de cada paso i deben cumplir

$$x_i^2 + y_i^2 + z_i^2 = \epsilon^2 \quad (2.15)$$

El vector distancia entre el origen y N pasos después de la caminata es entonces

$$\vec{R} = \sum_{i=1}^N \vec{r}_i \quad (2.16)$$

Donde $\vec{r}_i = x_i \hat{i} + y_i \hat{j} + z_i \hat{k}$. Al igual que en el caso unidimensional $E(\vec{R}) = 0$, por lo que calculamos

$$E(\vec{R}^2) = E \left[\left(\sum_{i=1}^N \vec{r}_i \right)^2 \right] = E \left(\sum_{i=1}^N \vec{r}_i^2 \right) + E \left(\sum_{i \neq j} \vec{r}_i \vec{r}_j \right) \quad (2.17)$$

El segundo término de la ecuación 2.17 es

$$E \left(\sum_{i \neq j} \vec{r}_i \vec{r}_j \right) = E \left(\sum_{i \neq j} (x_i x_j + y_i y_j + z_i z_j) \right) = 0 \quad (2.18)$$

Este término es cero debido a que los valores de x_i, y_i , y z_i pueden ser tanto negativos como positivos, y tienen la misma probabilidad de ser negativos que positivos. Cuando se computa la esperanza teniendo en cuenta todos los valores posibles de estas variables y su probabilidad, ésta da cero.

Otra forma de visualizar esto es imaginarse una esfera de radio ϵ y fijar arbitrariamente el vector \vec{r}_i apuntando hacia arriba (ver figura 2.1). Luego se observa que el producto punto entre \vec{r}_i y cualquier \vec{r}_j del hemisferio superior de la esfera da un número positivo (debido a la definición de producto punto y que el ángulo entre ellos es menor a 90°), y el producto entre \vec{r}_i y cualquier \vec{r}_j del hemisferio inferior da negativo (debido a que el ángulo es mayor a 90°). Es posible sumar todos los productos punto correspondientes al hemisferio superior y obtener un valor positivo, y por otro lado sumar todos los productos punto correspondientes al hemisferio inferior y obtener un valor negativo. Debido a la simetría esférica de la situación, la magnitud del valor positivo y negativo deben ser iguales, por lo que cuando se sumen se obtendrá cero. Luego se repite este proceso para todo \vec{r}_i (nuevamente dará cero) y finalmente se suman, dando el resultado final cero debido a que todos los términos son cero.

Entonces

$$E(\vec{R}^2) = E \left(\sum_{i=1}^N \vec{r}_i^2 \right) = E \left(\sum_{i=1}^N \epsilon^2 \right) = N\epsilon^2 \quad (2.19)$$

Y el Exciton Diffusion Length es igual que para el caso de una dimensión

$$L_D = \sqrt{N}\epsilon \quad (2.20)$$

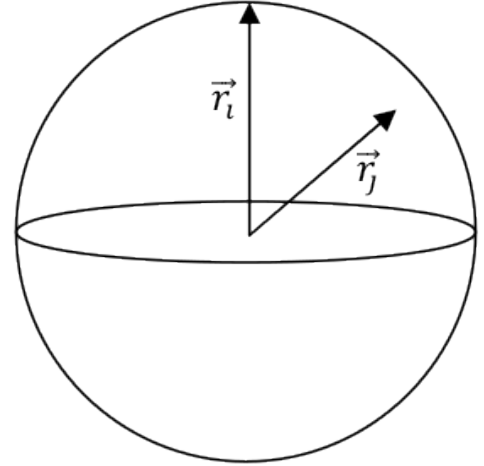


Figura 2.1: esfera de radio ϵ y \vec{r}_i

Capítulo 3

Estado del arte

3.1. Modelización

3.2. Simulación

3.3. Computación de altas prestación

GPU

Las unidades de procesamiento gráfico, GPU, se han convertido en una parte integral de los sistemas actuales de computación. El bajo costo y marcado incremento del rendimiento, permitieron su fácil incorporación al mercado. En los últimos años, su evolución implicó un cambio, dejó de ser un procesador gráfico potente para convertirse en un co-procesador apto para el desarrollo de aplicaciones paralelas de propósito general con demanda de anchos de banda de procesamiento y de memoria sustancialmente superiores a los ofrecidos por la CPU.

La rápida adopción de las GPU como computadora paralela de propósito general se vio favorecida por el incremento tanto de sus capacidades como de las facilidades y herramientas de programación. Actualmente la GPU se ha posicionado como una alternativa atractiva a los sistemas tradicionales de computación paralela (Piccoli 2011).

3.4. Lenguajes de programación científica

Fortran

La comunidad científica ha usado Fortran desde siempre (las primeras versiones del lenguaje son de la década del 50, época de las tarjetas perforadas). Sin dudas, es el lenguaje más usado para resolver problemas científicos. Posee una amplia comunidad de usuarios y programas

desarrollados hace décadas todavía son usados. La bibliografía y documentación es extensa. Existen librerías optimizadas, serial y paralela, y fuertemente probadas para realizar virtualmente todos los cálculos necesarios. Las versiones de OpenMP y MPI soportan nativamente Fortran. Si el factor que va a decidir la elección del lenguaje de programación es el tiempo de ejecución, entendemos que Fortran es la elección correcta.

Python

Lenguaje creado a en el año 1991 (por lo que podemos considerarlo “moderno” comparandolo con Fortran).

Ecosistema científico: IPython/Jupyter (Pérez and Granger 2007), Numpy, Matplotlib, Scipy. Entorno de programación interactivo de vanguardia, simple, libre, gratuito y multiplataforma.

Numpy brinda funcionalidades para trabajar con matrices.

Scipy es la biblioteca científica.

Matplotlib

Julia

Fue diseñado para computación distribuida y en paralelo,

CUDA C

OpenCL

PyCUDA y PyOpenCL (Klöckner et al. 2012)

MPI4Python (L. Dalcin 2005)

Capítulo 4

Propuesta

- 4.1. Modelo conceptual
- 4.2. Modelo computacional
- 4.3. Simulaciones de Monte Carlo
- 4.4. Simulación de altas prestaciones



Capítulo 5

Desarrollo computacional

Se consideran dos desarrollos

- diseño del simulador
- paralelismo

5.1. Diseño del simulador

- mejoras en el diseño original (paper)
- fundamentos teóricos (teoría de simulación).
- en función del funcionamiento se realizan un modelo mejorado (en función de lo que observan los Químicos). El modelo original tienen restricciones teóricas que se van eliminando a medida que se avanza en el trabajo mediante la interacción con los Químicos.
- etapas de construcción (refinación) del modelo.
- validación de la simulación.
- fenómeno físico/químico a modelar desde el punto de vista computacional (detallar, explicar).

5.2. Tipo de simulación

- ¿estática o dinámica? estática
- Montecarlo
- la simulación original es serial
- detalle de la simulación.
- marco teórico computacional.
- como se ha hecho la simulación.

5.3. Tipo de escalabilidad

- fuerte / débil.
- se ejecuta muchas veces para luego calcular promedio y desviaciones.
- calcular tiempo de recursos (profile), ejecución, memoria, etc.

5.4. Paralelismo

- ¿por qué?
- ¿para qué?
- multicore / cluster / GPGPU / cloud
- mejorar tiempos y/o mejorar precisión.
- mayor volumen de datos
- mayor cantidad de “doops”
- mayor cantidad de experimentos.
- speedup

Capítulo 6

Validación experimental

6.1. Diseño de los experimentos

6.1.1. Puntos aleatorios en la esfera

En la superficie

```
n = 3000
R = 5

theta = np.random.uniform(low = 0, high = 2 * np.pi, size = n)
phi = np.random.uniform(low = 0, high = np.pi, size = n)

x = np.sin(phi) * np.cos(theta)
y = np.sin(phi) * np.sin(theta)
z = np.cos(phi)

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z)
```

en la figura 6.1 se muestra que la distribución no es uniforme, se ve claramente la concentración de puntos en los polos.

Entonces, necesitamos mejorar la distribución para hacerla uniforme:

```
n = 3000
theta = 2 * np.pi * np.random.uniform(size=n)
phi = np.arccos(2 * np.random.uniform(size=n) - 1)
```

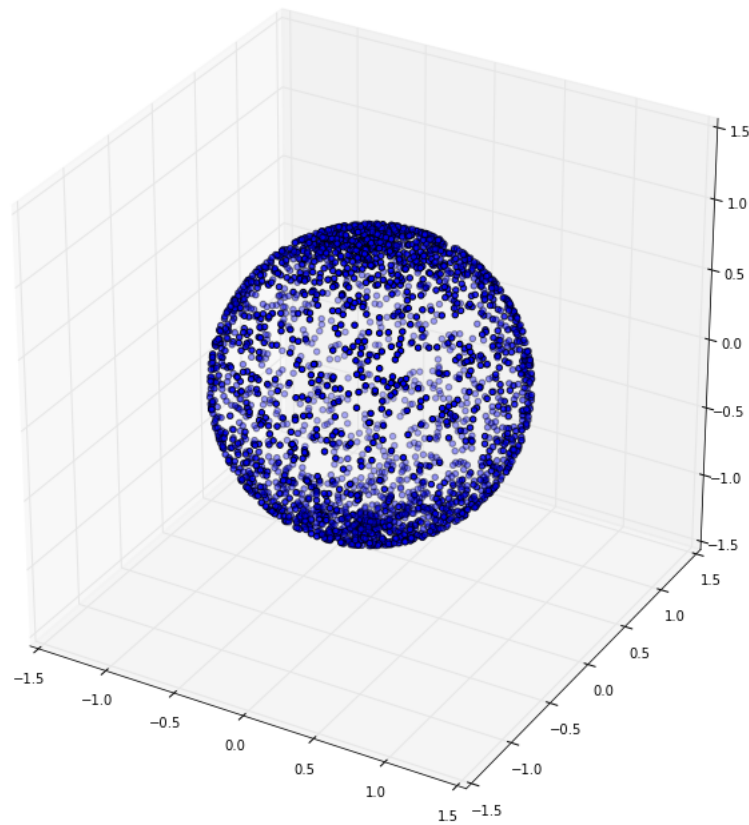


Figura 6.1: distribución no uniforme

```
x = np.sin(phi) * np.cos(theta)
y = np.sin(phi) * np.sin(theta)
z = np.cos(phi)

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z)
```

Ahora la distribución es uniforme en la superficie. Continuamos trabajando en el interior de la superficie.

En el interior

definimos el radio como una variable

```
n = 3000
theta = 2 * np.pi * np.random.uniform(size=n)
```

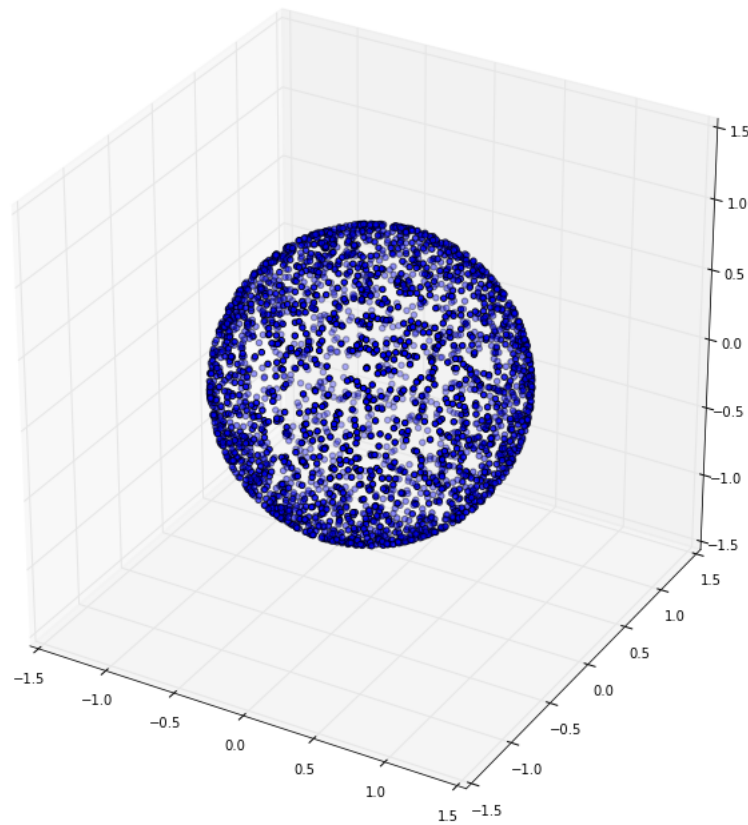


Figura 6.2: distribución uniforme

```

phi = np.arccos(2 * np.random.uniform(size=n) - 1)

#theta = np.random.uniform(low=0, high = 2 * np.pi, size=n)
#phi = np.arcsin(2 * np.random.uniform(low=0, high = 2 * np.pi, size=n) - 1)
u = np.random.uniform(low=0, high=10, size=n)

x = np.sin(phi) * np.cos(theta) * u
y = np.sin(phi) * np.sin(theta) * u
z = np.cos(phi) * u

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z)

se agrega el IF

n = 3000
R = 5

```

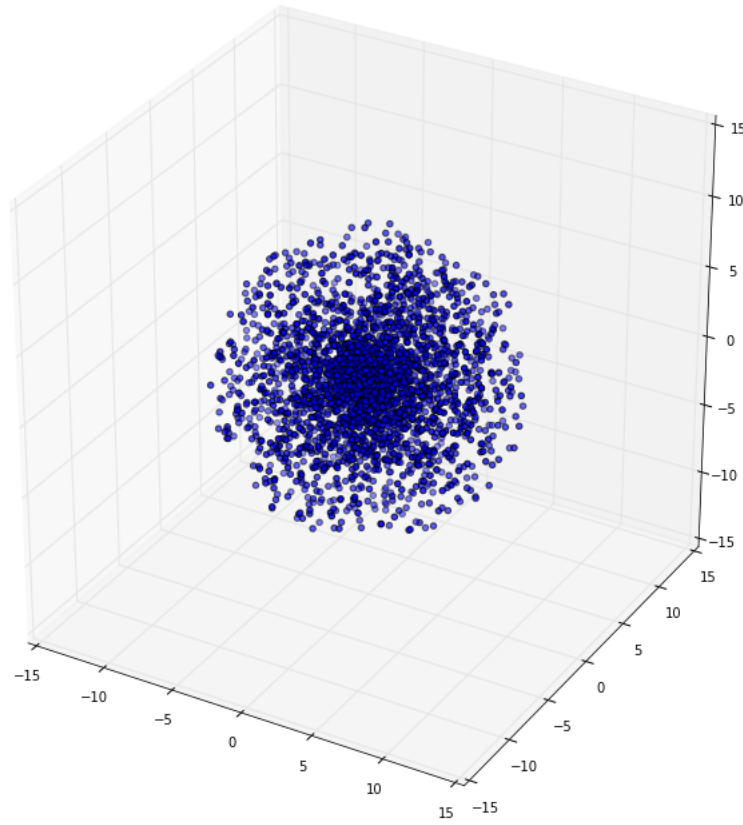


Figura 6.3: distribución no uniforme

```
a = 0

x_l = []
y_l = []
z_l = []

t = time.time()

while a <= n:
    x = np.random.uniform(low = -R, high = R, size=1)
    y = np.random.uniform(low = -R, high = R, size=1)
    z = np.random.uniform(low = -R, high = R, size=1)
    if x * x + y * y + z * z <= R * R:
        a += 1
        x_l.append(x)
        y_l.append(y)
        z_l.append(z)
```



```
print("time =",time.time() - t)

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x_l, y_l, z_l)
```

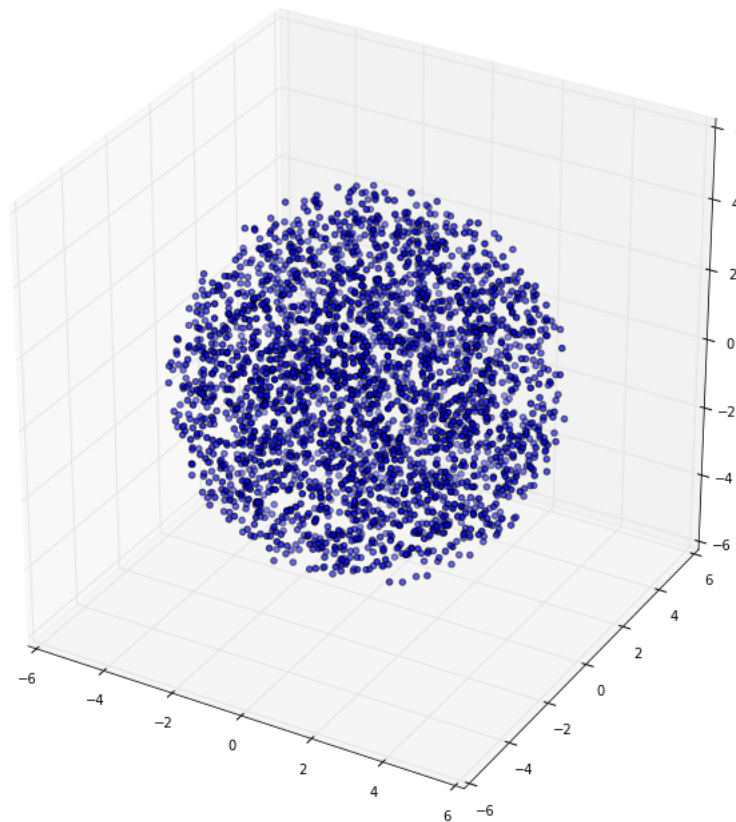


Figura 6.4: distribución uniforme

lo que funciona correctamente pero demasiado lento.

El método eficiente

En el foro de matemática math.stackexchange.com encontramos el método eficiente para obtener puntos al azar en el volumen de esfera con probabilidad uniforme ¹.

Suponemos la esfera centrada en el origen $(0, 0, 0)$.

¹Picking random points in the volume of sphere with uniform probability <http://tiny.cc/nycy6x>

6.2. Entorno experimental

Raspberry Pi para el desarrollo y validación

IPython/Jupyter para mostrar avances

6.3. Experimentos y discusión. (precisión y speedup)

6.4. Ciencia reproducible



Capítulo 7

Conclusiones y trabajos futuros

Fortalecer el trabajo multi/interdisciplinar.

Trabajos Futuros

Analizar performance del “caminante aleatorio tridimensional” usando distintos lenguajes (Julia, Fortran, C) y estrategias de optimización/paralelización, con el objetivo de mejorar los tiempos de ejecución.

Analizar métodos alternativos al que usa Numpy para generar números random, por ej: The Scalable Parallel Random Number Generators Library (SPRNG) <http://www.sprng.org>



Capítulo 8

Anexos

8.1. Licencia del código

The MIT License (MIT)

Copyright © 2015 Laboratorio de Microscopia Óptica Avanzada - UNRC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



Bibliografía

- Asbury J. B., Wang, Hao E. 2001. *Physical Chemistry B*, no. 105: 4545–57.
- Bach U., Comte P., Lupo D. 1998. *Nature*, no. 395: 583–85.
- Bolinger J. C., K-J., Lee. 2008. *Physical Chemistry C*, no. 122(47): 18608–15.
- Bolinger J. C., Lee K-J., Fradkin K. 2009. *Proceedings of the National Academy of Sciences of USA*, no. 106(5): 1342–46.
- Chang Y-L., Chen J. T., Palacios R. E. 2009. *American Chemical Society*, no. 131(40): 14166.
- Chang Y-L., Fan F-R. F., Palacios R. E. 2008. *American Chemical Society*, no. 130(28): 8906–7.
- Chou P-T., Chi Y. 2007. *Chemistry: A European Journal*, no. 13(2): 380–95.
- Dimitrakopoulos C. D., Malenfant P. R. L. 2002. *Advanced Materials*, no. 14(2): 99–117.
- Fradkin K., Bolinger J. C., Palacios R. E. 2009. *Physical Chemistry A*, no. 133(16): 4739–45.
- Gather M. C., Meerholz K., Koehnen A. 2011. *Advanced Materials*, no. 23(2): 233–48.
- Gratzel M. 2001. *Nature*, no. 414: 338–44.
- Jason McNeill, Yueli Zheng, Changfeng Wu. 2008. *American Chemical Society*, no. 112: 1772–81.
- Klöckner, Andreas, Nicolas Pinto, Yunsup Lee, B. Catanzaro, Paul Ivanov, and Ahmed Fasih. 2012. “PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation.” *Parallel Computing* 38 (3): 157–74. doi:[10.1016/j.parco.2011.09.001](https://doi.org/10.1016/j.parco.2011.09.001).
- Kulkarni A. P., Babel A., Tonzola C. J. 2004. *Chemistry of Materials*, no. 16(23): 4556–73.
- L. Dalcin, M. Storti, R. Paz. 2005. “MPI for Python.” *Parallel and Distributed Computing* 65(9): 1108–15. doi:[10.1016/j.jpdc.2005.03.010](https://doi.org/10.1016/j.jpdc.2005.03.010).
- Lindstrom C. D., Zhu X. Y. 2006. *Chemical Reviews*, no. 106: 4281–4300.
- Lupton J. M. 2010. *Advanced Materials*, no. 22: 1689–1721.
- . 2012. *ChemPhysChem*, no. 13(4): 901–7.
- Mabeck J. T., Malliaras G. G. 2006. *Analytical and Bioanalytical Chemistry*, no. 384(2): 343.
- Nothaft M., Jelezko F., Höhla S. 2012a. *Nature Communications*, no. 3: 1–6.

- . 2012b. *Physica Status Solidi B*, no. 249(4): 653–60.
- Oregan B., Gratzel M. 1991. *Nature*, no. 353: 737–40.
- Palacios R. E., Bard A. J., Fan F-R. F. 2006. *American Chemical Society*, no. 128(28): 9028–29.
- Palacios R. E., Grey J. K., Chang W. S. 2009. *Physical Chemistry B*, no. 133(44): 14618–28.
- Palacios R. E., Grey J. K., Fan F-R. F. 2007. *Nature Mater*, no. 6: 680–85.
- Palacios R. E., Rival A., Lee K-J. 2009. *Chemical Physics*, no. 357(1-3): 21–27.
- Pérez, Fernando, and Brian E. Granger. 2007. “IPython: a System for Interactive Scientific Computing.” *Computing in Science and Engineering* 9 (3). IEEE Computer Society: 21–29. doi:[10.1109/MCSE.2007.53](https://doi.org/10.1109/MCSE.2007.53).
- Piccoli, María Fabiana. 2011. *Computación de Alto Desempeño En GPU*. La Plata: Editorial de la Universidad Nacional de La Plata.
- Schwartz B. J. 2008. *Nature Mater*, no. 7: 427–28.
- Scott J. C., Bozano L. D. 2007. *Advanced Materials*, no. 19(11): 1452–63.
- Sirringhaus H. 2005. *Advanced Materials*, no. 17(20): 2411–25.
- Zaumseil J., Sirringhaus H. 2007. *Chemical Reviews*, no. 107(4): 1296–1323.
- Zhu X-Y., Muntwiler M., Yang Q. 2009. *Accounts of Chemical Research*, no. 42(11): 1779–87.