



Métodos espectrales: algunas aplicaciones en física

Spectral methods: some applications in physics

M. REYES¹

Recibido: 8 de octubre de 2021 / Aceptado: 20 de noviembre de 2021

RESUMEN

Los métodos espectrales son herramientas poderosas en las ciencias naturales y en diversas áreas de la ingeniería, el propósito de este trabajo es mostrar como aplicar los métodos espectrales a tres problemas de física, solución de la ecuación de difusión, cálculo del tercer coeficiente del virial y su primera y segunda derivada para el potencial de Lennard-Jones modificado de un fluido simple y detección de periodicidad de un sistema mecánico. El enfoque es computacional, considerando que la transformadas requeridas de Fourier, transformada seno y coseno de Fourier, se pueden implementar fácilmente en su versión rápida en cualquier lenguaje de programación.

ABSTRACT

Spectral methods are powerful tools in natural sciences and in various areas of engineering, the purpose of this work is to show how to apply spectral methods to three physics problems, solution of the diffusion equation, calculation of the third coefficient of the virial and its first and second derivatives for the modified Lennard-Jones potential of a simple fluid and periodicity detection of a mechanical system. The approach is computational, considering that the required Fourier transforms, sine and cosine Fourier transform can be easily implemented in their fast version in any programming language.

PALABRAS CLAVES

Transformada coseno de Fourier, transformada seno de Fourier, convoluciones, coeficiente del virial, ecuación de difusión, periodicidad.

KEYWORDS

Cosine Fourier transform, sine Fourier transform, convolutions, diffusion equation, virial coefficient, periodicity

¹Departamento de Física. Universidad Nacional Autónoma de Honduras en el Valle de Sula.
marco.reyes@unah.edu.hn

* Esta obra está bajo una licencia Creative Commons Reconocimiento - NoComercial 4.0 Internacional

* This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

I | INTRODUCCIÓN

Los métodos espectrales o análisis de Fourier son herramientas poderosas para físicos teóricos y experimentales. Al físico experimental le ayuda a establecer una imagen clara de datos obtenidos de un experimento, extraer información de señales complicadas, a reconstruir la estructura de un cristal a partir del patrón de difracción de rayos x, para producir filtros en circuitos electrónicos y la lista puede seguir (Walker, 2013). Un físico teórico utiliza la transformada de Fourier para resolver ecuaciones diferenciales, ecuaciones integrales y evaluar integrales mediante convoluciones.

Fourier fue el primero en representar una función periódica $f(t)$ con un periodo T , como una combinación de funciones senos y cosenos. Revisemos algunos resultados importantes, comenzando con la representación en serie de Fourier de una función periódica

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)] \quad (1)$$

donde a_n y b_n son los llamados coeficientes de Fourier y $\omega = \frac{2\pi}{T}$ es la frecuencia natural de la función. El análisis de Fourier no es sólo para funciones periódicas, se puede aplicar a funciones no periódicas. Para lograrlo la serie de Fourier, en la ecuación 1 las sumas se convierten en integrales de Fourier, la función no periódica puede ser pensada como una función periódica con un periodo extendido a infinito, con lo cual se obtiene.

$$g(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{ikt} dt \quad (2)$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(k) e^{-ikt} dk \quad (3)$$

Las ecuaciones 2 y 3 definen la transformada de Fourier y su respectiva inversa.

En nuestros cálculos vamos a requerir la transformada de Fourier de derivadas, tomando la derivada de f y sustituyendo en la ecuación 2 se tiene

$$g_1(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{df(t)}{dt} e^{ikt} dt \quad (4)$$

integrando por partes se llega a

$$g_1 = -ikg(k) \quad (5)$$

esto puede ser generalizado para cualquier derivada

$$g_n = (-ik)^n g(k) \quad (6)$$

La transformada rápida de Fourier aprovecha los resultados de cálculos previos para reducir el número de operaciones, en particular aprovecha la periodicidad y simetría de las funciones trigonométricas para calcular la transformada con aproximadamente $N \log_2 N$ operaciones Brigham (1988).

Para funciones o datos que tienen simetría par o impar con respecto al punto medio, aparecen las transformadas seno y coseno de Fourier. Podemos calcular dichas transformadas, primero reflejando las datos para crear la simetría requerida y luego usar la transformada discreta de Fourier en su versión rápida. Si $f(t)$ esta definida en $0 < t < \infty$, $f(t)$ se puede representar por

$$f(t) = \frac{2}{\pi} \int_0^{\infty} F_s(k) \sin(kt) dk \quad (7)$$

donde $F_s(k)$ está dado por

$$F_s(k) = \int_0^{\infty} f(t) \text{sen}(kt) dt \tag{8}$$

$F_c(k)$ es la transformada seno de Fourier. Se tiene expresiones similares para la transformada coseno de Fourier,

$$f(t) = \frac{2}{\pi} \int_0^{\infty} F_c(k) \text{cos}(kt) dk \tag{9}$$

donde $F_c(k)$ está dado por

$$F_c(k) = \int_0^{\infty} f(t) \text{cos}(kt) dt \tag{10}$$

Las transformadas de Fourier se puede utilizar para evaluar integrales tipo convolución, consideremos dos funciones $f(x)$ y $g(x)$ con transformadas de Fourier F y G respectivamente. Definimos la operación

$$f * g = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(y) f(x-y) dy \tag{11}$$

como la convolución de las dos funciones f y g sobre el intervalo $(-\infty, \infty)$. Ahora transformamos la integral en 11 introduciendo la transformada de Fourier

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(y) f(x-y) dy = \int_{-\infty}^{\infty} F(t) G(t) e^{-itx} dt \tag{12}$$

Este resultado puede ser interpretado como sigue: la transformadas de Fourier inversa de un producto de transformadas de Fourier es la convolución de las funciones originales $f * g$, esto de acuerdo a Arfken y Hans J (1995), Brigham (1988)

La transformada de Fourier se puede utilizar para evaluar cierto de tipo de integrales, consideremos la integral

$$\int_0^{\infty} \frac{\text{cos}\omega t dt}{t^4 + \beta^4} = \frac{\pi}{2\beta^3} e^{-\beta\omega/\sqrt{2}} \text{sen}\left(\frac{\pi}{4} + \frac{\beta\omega}{\sqrt{2}}\right)$$

La integral es equivalente a

$$Re \int_0^{\infty} \frac{e^{-\omega t} dt}{t^4 + \beta^4}$$

La integral tiene solución exacta mediante el método de residuos, la siguiente tabla muestra la integral evaluada numéricamente con la transformada rápida de Fourier Coseno y de forma exacta así como la diferencia entre ellas. En la sección apéndice, se encuentra el programa 1, con el cual se genero la tabla.

wn	valor-num	valor-exacto	error
0.00000000	8.88576508	8.88576603	9.53674316E-07
0.502654850	8.63690376	8.63690472	9.53674316E-07
1.00530970	8.00575447	8.00575447	0.00000000
1.50796449	7.14004850	7.14004850	0.00000000
2.01061940	6.15572548	6.15572500	4.76837158E-07
2.51327419	5.14032078	5.14031982	9.53674316E-07
3.01592898	4.15682554	4.15682554	0.00000000
3.51858401	3.24770331	3.24770474	1.43051147E-06

4.02123880	2.43878579	2.43878603	2.38418579E-07
4.52389336	1.74287105	1.74287295	1.90734863E-06
5.02654839	1.16296029	1.16296136	1.07288361E-06

=====

el error es muy bajo, como muestra la tabla.

II | ECUACIÓN DE DIFUSIÓN

Consideremos la ecuación de difusión en una dimensión

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \tag{13}$$

donde κ es la constante de difusión. La ecuación de difusión es usada para calcular el movimiento de gases y líquidos, así como el flujo de calor en conductores térmicos, la función $T(x, t)$ depende de la posición y el tiempo.

Consideremos $T(x, t)$ a lo largo de una varilla. Comenzamos en el tiempo $t = 0$ conociendo la distribución inicial de temperatura, $T(x, 0)$, el objetivo es encontrar la temperatura a lo largo de una varilla en cualquier tiempo.

sí $\tau(k, t) = \mathcal{F}(T(x, t))$ entonces

$$\tau(k, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} T(x, t) e^{-ikx} dx \tag{14}$$

y podemos escribir la temperatura como

$$T(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tau(k, t) e^{ikx} dk \tag{15}$$

sustituyendo 15 en 13

$$\frac{\partial}{\partial t} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tau(k, t) e^{ikx} dk = \kappa \frac{\partial^2}{\partial x^2} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tau(k, t) e^{ikx} dk$$

en el lado izquierdo la derivada parcial opera $\tau(k, t)$, pero $\tau(k, t)$ no es una función de x , así sobre el lado derecho la derivada opera sólo en el factor exponencial

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{\partial \tau}{\partial t} e^{ikx} dk = \frac{\kappa}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (-k^2 \tau) e^{ikx} dk$$

ahora multiplicamos ambos lados por $e^{-ik'x}$, e integramos sobre x

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ik'x} \int_{-\infty}^{\infty} \frac{\partial \tau}{\partial x} e^{ikx} dk dx = \frac{\kappa}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ik'x} \int_{-\infty}^{\infty} \tau(k, t) (-k^2) e^{ikx} dk dx$$

intercambiando el orden de integración y usando

$$\int_{-\infty}^{\infty} e^{i(k-k')x} dx = 2\pi \delta(k - k')$$

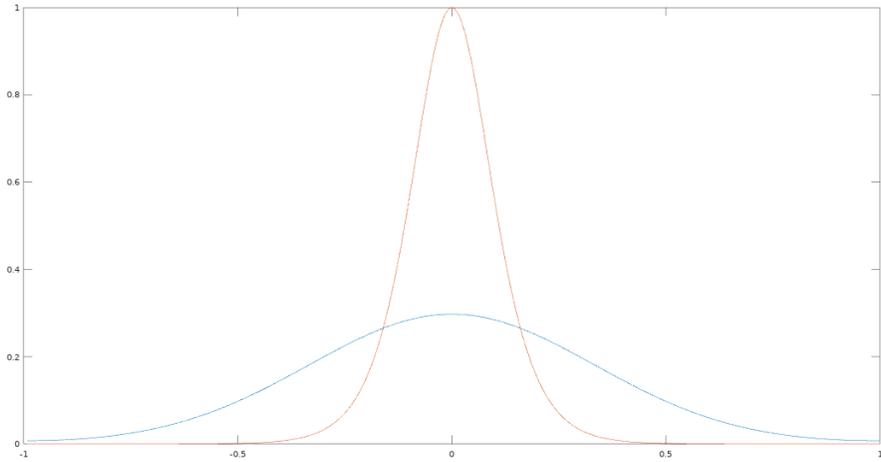


Figura 1: Distribución de temperaturas: gráfico rojo $T(x,0)$, gráfico azul $T(x,5)$

se obtiene

$$\frac{\partial \tau}{\partial t} = -k^2 \kappa \tau(k, t) \quad (16)$$

resolviendo 16 por separación de variables, se obtiene

$$\tau(k, t) = e^{-k^2 \kappa t} \tau(k, 0) \quad (17)$$

donde $\tau(k, 0)$ se obtiene tomando la transformada de la distribución inicial de temperatura.

$$\tau(k, 0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} T(x, 0) e^{-ikx} dx$$

reemplazando los resultados anteriores en 15 se obtiene la temperatura en el tiempo t

$$T(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-k^2 \kappa t} \tau(k, 0) e^{ikx} dx \quad (18)$$

Consideremos una varilla de longitud $L = 1$ con una distribución de temperatura inicial dada por

$$T(x, 0) = \text{sech}(8x)$$

Al aplicar el método descrito obtenemos la configuración de temperatura en $t = 5$ segundos, lo cual se muestra en la figura 1. El gráfico muestra también superpuesta la configuración inicial de temperatura.

El gráfico muestra como evoluciono la temperatura en la barra desde el estado inicial al tiempo de interés. La implementación numérica se realizo con Octave, programa 2, aprovechando que este software libre tiene librerías de la transformada rápida de Fourier y su inversa, su programación se simplifica bastante.

III | MECÁNICA ESTADÍSTICA

La ecuación de estado del virial, nos permite estudiar gases que tienen comportamiento no ideal, gases que tienen un coeficiente de compresión mayor de uno, es una ecuación tipo serie de potencias, en cada término de la serie podemos identificar un coeficiente del virial, el cual toma en cuenta las interacciones entre los átomos.

$$\frac{P}{kT} = \rho + B_2(T)\rho^2 + B_3(T)\rho^3 + B_4(T)\rho^4 + \dots \quad (19)$$

En la ecuación del virial 19, P es presión, k es la constante de Boltzmann, T temperatura, ρ es la densidad numérica, B_2 el segundo coeficiente del virial, B_3 el tercer coeficiente del virial, y la ecuación sigue con los términos de orden superior.

B_2 toma en cuenta las interacciones por pares de átomos, B_3 toma en cuenta interacciones de tres átomos y así sucesivamente. Se puede deducir la ecuación del virial a partir de la función de partición macrocanónica, los detalles se pueden revisar en McQuarrie (s.f.).

Para moléculas lineales, B_2 se puede escribir así

$$B_2(T) = -\frac{1}{2V} \int d\vec{r}_1 \int [e^{-\beta u(r_{12})} - 1] d\vec{r}_{12} \quad (20)$$

$$B_2(T) = -2\pi \int_0^\infty [e^{-\beta u(r)} - 1] r^2 dr \quad (21)$$

La expresión entre corchetes es la función de Mayer, y u es la energía potencial entre los átomos o moléculas y $\beta = 1/kT$. B_2 se puede calcular con los métodos de cuadratura de forma relativamente fácil, pero los coeficientes superiores B_3, B_4, \dots son integrales multidimensionales muy complejas, generalmente se utilizan métodos de Monte Carlo para evaluarlas.

La función de Mayer $f = e^{-\beta u} - 1$ requiere la energía potencial entre las moléculas, vamos a utilizar el potencial de Lennard-Jones modificado. El potencial de Lennard Jones entra dentro de la categoría de potenciales realistas.

$$u(r) = 4\epsilon \left\{ \left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right\} \quad (22)$$

ϵ es la altura del pozo de potencial (donde está el mínimo de energía) y σ es el diámetro de colisión es decir, la distancia cuando $u = 0$. El potencial de Lennard Jones funciona muy bien para moléculas pequeñas y no polares. En la fórmula de Lennard-Jones, la pared repulsiva no es vertical sino que tiene pendiente finita; esto implica que si dos moléculas tienen energías cinéticas altas, pueden traslaparse, los átomos o moléculas se modelan como esferas blandas.

Broughton y Gilmer propusieron un potencial de Lennard-Jones modificado, que básicamente adopta la forma de un potencial truncado con lo cual se evita cualquier discontinuidad y presenta algunas ventajas para predecir propiedades termodinámicas por métodos perturbativos Fuchizaki y Asano (2013), Ushcats (2014).

$$u(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] + c_1 & r \leq 2.3\sigma \\ c_2 \left(\frac{\sigma}{r} \right)^{12} + c_3 \left(\frac{\sigma}{r} \right)^6 + c_4 \left(\frac{\sigma}{r} \right)^2 + c_5 & 2.3\sigma < r < 2.5\sigma \\ 0 & r \geq 2.5\sigma \end{cases}$$

donde $c_1 = 0.0163169237\epsilon$, $c_2 = 3136.5686\epsilon$, $c_3 = -68.069\epsilon$, $c_4 = -0.0833111261\epsilon$ $c_5 = 0.746882273\epsilon$

Nuestro objetivo es ilustrar el calculo de B_3 con transformada rápida de Fourier, ya que B_3 tiene la forma característica de las integrales de convolución.

$$B_3 = -\frac{1}{3V} \int \int \int f_{12} f_{13} f_{23} d\vec{r}_1 d\vec{r}_2 d\vec{r}_3 \quad (23)$$

usando $\rho_j = \vec{r}_j - \vec{r}_i$, la ecuación 23 se convierte en

$$B_3 = -\frac{1}{3} \int \int f(|\rho_2|) f(|\rho_3|) f(|\rho_3 - \rho_2|) d\vec{\rho}_2 d\vec{\rho}_3 \quad (24)$$

En su forma general el tercer coeficiente del virial considera la energía de interacción de tres moléculas lo cual se puede escribir como

$$U_{123} = \sum u_{ij} + u_{123} + w_{123} \quad (25)$$

El primer término en la ecuación 25 es la suma de los tres pares de interacción; este es el término aditivo

$$\sum u_{ij} = u_{12} + u_{13} + u_{23} \quad (26)$$

El segundo término en 25 u_{123} representa la energía de traslape no aditiva y w_{123} es la energía de dispersión triple dipolo. Escribiendo las interacciones de esta forma, el tercer coeficiente es

$$B_3 = B_3(\text{aditivo}) + \Delta B_3(\text{no-aditivo}) + \Delta B_3(\text{dispersión}) \quad (27)$$

Nos vamos a centrar en el término aditivo el cual tiene la estructura de convolución de acuerdo a la ecuación 12. Necesitamos la transformada de Fourier en tres dimensiones

$$f(k) = \int f(\vec{r}) e^{-i\vec{k}\cdot\vec{r}} d^3r \quad (28)$$

Estas tres integrales pueden reducirse a una sola integral si $f(\vec{r})$ depende sólo de $r = |\vec{r}|$, usando coordenadas esféricas, podemos escribir $d\vec{r} = dx dy dz \rightarrow r^2 dr \sin\theta d\theta d\phi$ y $\vec{k}\cdot\vec{r} = kr \cos\theta$, con lo cual la ecuación 28 se escribe ahora

$$f(k) = \int_0^\infty \int_0^\pi \int_0^{2\pi} f(r) e^{-ikr \cos\theta} r^2 dr \sin\theta d\theta d\phi \quad (29)$$

haciendo las integrales con respecto a θ y ϕ la ecuación 29 queda así

$$f(k) = 4\pi \int_0^\infty f(r) r \frac{\sin kr}{k} dr \quad (30)$$

De acuerdo a la ecuación 30 necesitamos la transformada seno de Fourier, Gould y Tobochnik (2010), la cual la vamos a implementar en su versión rápida y la vamos a denotar como TRFS.

Con los coeficientes del virial y sus derivadas podemos calcular propiedades termodinámicas de gases reales, por ejemplo el factor de compresión Z , la energía interna ΔU , entalpía ΔH , entropía ΔS , energía de Helmholtz ΔA , energía de Gibbs ΔG y fugacidad f , Bruce E. Poling y O'Connell (2003). Estas son algunas ecuaciones de esas propiedades.

$$Z = 1 + \frac{B_2}{V} + \frac{B_3}{V^3} \quad (31)$$

$$\frac{\Delta U}{RT} = \frac{T}{V} \frac{dB_2}{dT} + \frac{T}{V^2} \frac{dB_3}{dT} \quad (32)$$

$$\frac{\Delta H}{RT} = \frac{1}{V} \left(T \frac{dB_2}{dT} - B_2 \right) + \frac{1}{2V^2} \left(T \frac{dB_3}{dT} - 2B_3 \right) \quad (33)$$

Para calcular B_3 nos basamos en el trabajo de Shaul, Schultz y Kofke Katherine R. S. Shaul y Kofke (2010), este es el algoritmo:

1. Discretizar f , $\frac{df}{d\beta}$ y $\frac{d^2f}{d\beta^2}$.
2. Calcular la TRFS de la función de Mayer, $f(k) = TRFS(fr)$
3. Elevar al cuadrado la TRFS, $f^2(k) = fk2$
4. Se calcula la transformada inversa de $fk2$, $fk2^{-1} = TRFS^{-1}(fk2)$
5. Se integra el producto $fk2^{-1}f(r)r^2$. Así se obtiene B_3
6. Se integra el producto $3fk2^{-1}\frac{df}{d\beta}r^2$. Así se obtiene $\frac{B_3}{d\beta}$
7. Se calcula $dsuma2 = dsuma2 + 3fk2^{-1}\frac{d^2f}{d\beta^2}r^2$ como una primera aproximación de $\frac{d^2B_3}{d\beta^2}$
8. Calcular TRFS de la primera derivada de f , $TRFS\left(\frac{df}{d\beta}\right) = dfk1$
9. Se eleva al cuadrado $dfk1^2 = dfk2$
10. Se calcula la transformada inversa seno del cuadro del paso anterior $TRFS^{-1}(dfk2) = dfkk$
11. Se integra el producto $6(dfkk)(fr)r^2$. Así se obtiene $\frac{d^2B_3}{d\beta^2}$

La transformada rápida de Fourier se baso en el algoritmo explicado por Pang (s.f.), la transformada rápida seno de Fourier y su inversa, se implemento de acuerdo a las recetas numéricas de William H. Press (s.f.). El calculo de B_3 , $\frac{dB_3}{d\beta}$ y $\frac{d^2B_3}{d\beta^2}$ se implemento en gfortran, el programa 3, que esta en la sección apéndice, genera un archivo externo que en combinación con gnuplot se obtienen las gráficas de interés.

Las figuras (2), (3) y (4) muestran B_3 , $\frac{B_3}{d\beta}$ y $\frac{d^2B_3}{d\beta^2}$ en función de la temperatura reducida.

B_3 para el potencial de Lennard-Jones modificado fue calculado por Ushcat Ushcats (2014) utilizando cuadratura Gaussiana, la tabla (1) muestra los resultados obtenidos por cuadratura y con transformada rápida de Fourier.

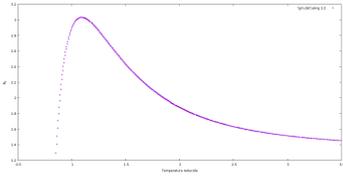


Figura 2: B_3 versus temperatura reducida

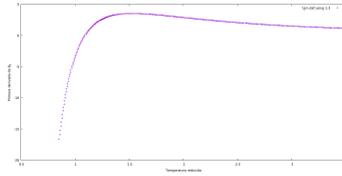


Figura 3: $B_3/d\beta$ versus temperatura reducida

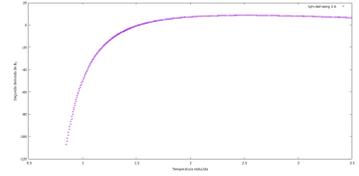


Figura 4: $d^2B_3/d\beta^2$ versus temperatura reducida

T_r	B_3 Cuadratura	B_3 TRF	$(\Delta B_3/B_3)100$
0.85	1.2890	1.2955	0.5000
0.90	2.1690	2.1737	0.2500
1.00	2.9031	2.8890	0.4800
1.20	2.9326	2.9342	0.0005
1.50	2.4275	2.4359	0.0035

Tabla 1: B_3 calculado con cuadratura Ushcats (2014) y con la Transformada rápida de Fourier

IV | DETECCIÓN DE PERIODICIDAD

A los físicos experimentales les interesa detectar tendencias en las mediciones que obtienen de los experimentos. Vamos a simular los datos obtenidos en un laboratorio de un sistema mecánico, para ello consideremos el sistema mecánico compuesto de tres masas, tres resortes que obedecen la ley de Hooke, dos amortiguadores y dos fuerzas externas.

En principio podemos encontrar la solución analítica de este sistema mecánico, sin embargo la idea es simular un experimento donde medimos las posiciones y velocidades de las masas, para ello planteamos las ecuaciones dinámicas para este sistema y resolvemos numéricamente utilizando el método de Runge Kutta de cuarto orden en versión vectorizada, ya que obtenemos un sistema de seis ecuaciones diferenciales de primer orden.

El espectro de potencia muestra la distribución de potencia de una señal versus la frecuencia

$$P_n = \frac{(A_n^2 + B_n^2)}{4N} \quad (34)$$

El espectro de potencia nos ayuda a detectar periodicidad, los picos en el espectro de potencia nos dice que hay frecuencias en la serie de Fourier que tienen amplitudes altas. Donde A_n y B_n son la parte real y compleja de la transformada de Fourier de la posición de una de las masas. Las ecuaciones dinámicas y condiciones iniciales escogidas son las siguientes

$$\begin{aligned} M_1 y_1'' + B_1 y_1' + K_1 y_1 - B_1 y_2' - K_2 y_2 &= F_1(t) \\ -K_2 y_2 + M_3 y_3'' + B_2 y_3' + (K_3 + K_2) y_3 - K_2 y_3 &= F_3(t) \\ -B_1 y_1' - K_1 y_1 + M_2 y_2'' + B_1 y_2' + (K_1 + K_2) y_2 - K_2 y_3 &= 0 \end{aligned}$$

Las constantes y las condiciones iniciales son:

$$K_1 = 5, K_2 = 6, K_3 = 7, M_1 = 1.0, M_2 = 1.5, M_3 = 2, B_1 = 0.1, B_2 = 0.1, F_1 = 1, F_3 = 0$$

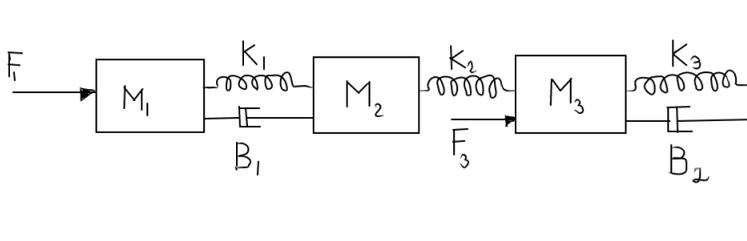


Figura 5: Sistema mecánico de interés

$$y_1(0) = 0, y_2 = 0, y_3 = 0, y_1'(0) = 0, y_2'(0) = 0, y_3'(0) = 0$$

Aquí tenemos un sistema de tres ecuaciones diferenciales de segundo orden, que se pueden escribir como un sistema de seis ecuaciones diferenciales de primer orden.

$$x_1' = x_4 \quad (35)$$

$$x_2' = x_5 \quad (36)$$

$$x_3' = x_6 \quad (37)$$

$$x_4' = \frac{1}{M_1} [-B_1 x_4 - K_1 x_1 + B_1 x_5 + K_2 x_2 + F_1] \quad (38)$$

$$x_5' = \frac{1}{M_2} [B_1 x_4 + K_1 x_1 - B_1 x_5 - (K_2 + K_1) x_2 + K_2 x_3] \quad (39)$$

$$x_6' = \frac{1}{M_3} [K_2 x_2 - B_2 x_6 - (K_2 + K_3) x_3 + F_3] \quad (40)$$

Al resolver numéricamente las ecuaciones 35 a 40 y calcular el espectro de potencia de la masa m_1 se obtienen las siguientes gráficas

Al calcular el espectro de potencia se obtienen algunos picos, que corresponden a las frecuencias dominantes García (s.f.).

V | CONCLUSIONES

1. La transformada rápida de Fourier, nos permite evaluar fácilmente integrales complejas que tienen en su integrando las funciones seno o coseno, utilizando la transformada seno o coseno de Fourier en su versión rápida para lograr más eficiencia computacional.
2. Se resolvió la ecuación de difusión mediante la transformada rápida de Fourier.
3. Se calculó el tercer coeficiente del virial y su primera y segunda derivada con respecto a β , con transformada rápida de Fourier, aprovechando que B_3 son integrales que tienen la forma de típica de una convolución
4. Se simuló el movimiento de un sistema de masas acopladas con resortes y amortiguadores, para detectar periodicidad en los datos, utilizando el espectro de potencia.

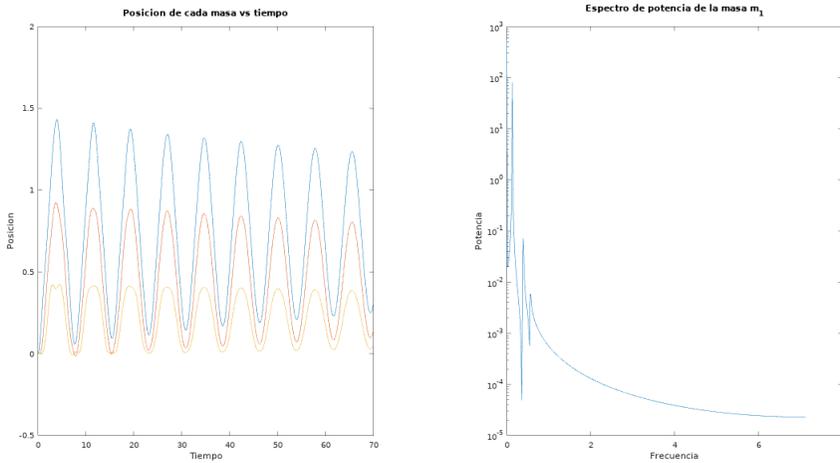


Figura 6: Posición versus tiempo de las masas y el espectro de potencia versus frecuencia

VI | APÉNDICE: PROGRAMAS DE LOS PROBLEMAS ABORDADOS

Estimado lector, cualquier duda sobre la implementación de los programas, puede contactarme a mi correo, el cual aparece al inicio del artículo.

Programa 1: este programa calcula la integral planteada en el artículo. Este programa se compiló y corrió en el IDE Geany en Linux y en el IDE CodeBlocks en Window, la subrutina TRF de acuerdo a Pang (s.f.), la transformada rápida de Fourier coseno en base William H. Press (s.f.), los resultados son idénticos en ambos entornos de desarrollo.

```

program trfcos
implicit none
integer,parameter::m=10,n=2*m
integer::i,k,j,np
real::fr(n+1),fi(n+1),dt,t,f,wn,Ie,tol,wnn,pi
pi=4.0*atan(1.0)
dt=100/float(n)
np=10
print*,'=====/'
print*,'Evaluacion de la integral de cero a infinito de la'
print*,'funcion cos(wt)/(t^2+b^2), donde b=0.5, calculada con'
print*,'Tranformada rapida Coseno de Fourier'
print*,'=====/'
print*,'   wn           valor-num           valor-exacto           error'
print*,'=====/'
do i=1,n
  t=(i-1)*dt
  fr(i)=f(t)
  fi(i)=0.0

```

```

enddo
call trcos(fr,fi,n,m)
do j=0,np
  wn=0.5*j
  do k=1,n
    wnn=2.0*pi*(k-1)/(dt*n)
    tol=0.03
    if (abs(wnn-wn)<tol) then
      print*,wnn,dt*fr(k),Ie(wnn),abs(dt*fr(k)-Ie(wnn))
    endif
  enddo
enddo
print*,'=====/'
end program trfcos

function Ie(w)
implicit none
real::Ie,pi,w,b
b=0.5
pi=4.0*atan(1.0)
Ie=(pi/(2.0*b**3))*exp(-b*w/sqrt(2.))*sin(pi/4+(b*w/sqrt(2.)))
end function Ie

function f(t)
real::t,f,b
b=0.5
f=1.0/(t**4+b**4)
end function f

subroutine trcos(y,yi,n,m)
implicit none
integer::n,j,m
real::y(n+1),suma,y1,y2,theta,wi,wpi,wpr,wtemp,pi,yi(n+1),wr
pi=4.0*atan(1.0)
wr=1.0
wi=0.0
theta=pi/n
wpr=-2.0*sin(0.5*theta)**2
wpi=sin(theta)
suma=0.5*(y(1)-y(n+1))
y(1)=0.5*(y(1)+y(n+1))
do j=1,n/2-1 ! j=n/2 es innecesario ya que y(n/2+1) no cambia
  wtemp=wr
  wr=wr*wpr-wi*wpi*wr ! se lleva a cabo la recurrencia
  wi=wi*wpr+wtemp*wpi+wi
  y1=0.5*(y(j+1)+y(n-j+1))
  y2=(y(j+1)-y(n-j+1))

```

```

    y(j+1)=y1-wi*y2
    y(n-j+1)=y1+wi*y2
    suma=suma+wr*y2
    yi(j)=0.0
enddo
call TRF(y,yi,n,m)
y(n+1)=y(2)
y(2)=suma
do j=4,n,2
    suma=suma+y(j)
    y(j)=suma
enddo
return
end subroutine trcos

```

```

SUBROUTINE TRF(AR,AI,N,M)
!
! AR: es la parte real de los datos
! AI: es la parte imaginaria
! se debe cumplir N=2**M
IMPLICIT NONE
INTEGER, INTENT (IN) :: N,M
INTEGER :: N1,N2,I,J,K,L,L1,L2
real:: PI,A1,A2,Q,U,V
real,INTENT (INOUT), DIMENSION (N) :: AR,AI
!
PI = 4.0*ATAN(1.0)
N2 = N/2
!
N1 = 2**M
IF(N1.NE.N) STOP 'Indices no coinciden'
!
L = 1
DO K = 1, N-1
    IF (K.LT.L) THEN
        A1 = AR(L)
        A2 = AI(L)
        AR(L) = AR(K)
        AR(K) = A1
        AI(L) = AI(K)
        AI(K) = A2
    END IF
    J = N2
    DO WHILE (J.LT.L)
        L = L-J
        J = J/2
    END DO

```

```

    END DO
    L = L+J
END DO

!
L2 = 1
DO L = 1, M
    Q = 0.0
    L1 = L2
    L2 = 2*L1
    DO K = 1, L1
        U = COS(Q)
        V = -SIN(Q)
        Q = Q + PI/L1
        DO J = K, N, L2
            I = J + L1
            A1 = AR(I)*U-AI(I)*V
            A2 = AR(I)*V+AI(I)*U
            AR(I) = AR(J)-A1
            AR(J) = AR(J)+A1
            AI(I) = AI(J)-A2
            AI(J) = AI(J)+A2
        END DO
    END DO
END DO
END SUBROUTINE TRF

```

Programa 2: este programa calcula la difusión en una varilla. La implementación se realizó con Octave, recomiendo utilizar la versión gráfica, la cual tiene un editor incluido, esto nos da un ambiente muy parecido a Matlab, el programa también funciona en Matlab.

```

% este programa resuelve la ecuacion de difusion
% con metodos espectrales
L=1.0;
N=256;
dx=(2*L)/N;
x=(1-N/2:N/2)*dx; % discretizacion de la mitad del espacio
nu=0.01; % parametro de difusion
T=5.0; % tiempo donde se calcula la solucion
dk=pi/L; % discretizacion en el espacio de Fourier
k=[0:N/2 1-N/2:-1]*dk; %vector de longitudes de onda
k2=k.^2;
uo=sech(8.0*x).^2; % condiciones iniciales
trf_uo=fft(uo); % TRF de uo
uT=real(ifft(exp(-nu*k2*T).*trf_uo)); % solucion en el tiempo t=T
plot(x,uT,x,uo);

```

Programa 3: este programa calcula el tercer coeficiente del virial del potencial de Lennard-Jones

modificado, así como la primera y segunda derivada de tercer coeficiente del virial con respecto a beta, se implementa en gfortran, el programa funciona muy bien en los entornos de desarrollo Geany en Linux y Codeblocks en Windows. La subrutina TRF se baso en Pang (s.f.), la transformada rapida de Fourier seno y seno inverso se baso en William H. Press (s.f.)

```

program B3_LJm
integer,parameter::n=4096,m=12
integer::i
double precision,dimension(0:n):: fr, fk, ifk2, x, fk2
double precision,dimension(0:n):: dfr1, dfr2, dfk1, dfk2
double precision::L, dr, f, df1, df2, r, pi, Tr, B3, suma
double precision::d1B3, d2B3, dsuma1, dsuma2
open(unit=6, file='LJm.dat')
L=50
Tr=0.85
pi=4.0*atan(1.0)
do while (Tr<=3.5)
fr(0)=0.0
dfr1(0)=0.0
dfr2(0)=0.0
do i=1,n-1
dr=L/float(n)
x(i)=dr*i
r=x(i)
fr(i)=f(r,Tr)
dfr1(i)=df1(r,Tr)
dfr2(i)=df2(r,Tr)
enddo
call seno_TRF(fr, fk, n, dr)
do i=0,n-1
fk(i)=(1.0/float(n))*fk(i)
enddo
do i=0,n-1
fk2(i)=fk(i)*fk(i)
enddo
call senoi_TRF(fk2, ifk2, n, dr)
do i=0,n-1
ifk2(i)=(.25/float(n))*ifk2(i)
enddo
suma=0.0
dsuma1=0.0
dsuma2=0.0
do i=0,n-1
suma=suma+(ifk2(i)*fr(i)*x(i)*x(i))
dsuma1=dsuma1+(3*ifk2(i)*dfr1(i)*x(i)*x(i))
dsuma2=dsuma2+(3*ifk2(i)*dfr2(i)*x(i)*x(i))
enddo

```

```

call seno_TRF(dfr1,dfk1,n,dr)
do i=0,n-1
dfk1(i)=(1./float(n))*dfk1(i)
enddo
do i=0,n-1
dfk1(i)=dfk1(i)*dfk1(i)
enddo
call senoi_TRF(dfk1,dfkk2,n,dr)
do i=0,n-1
dfkk2(i)=(0.25/float(n))*dfkk2(i)
enddo
do i=0,n-1
dsuma2=dsuma2+(6*dfkk2(i)*fr(i)*x(i)*x(i))
enddo
B3--(2./3.)*pi*dr*suma
d1B3--(2./3.)*pi*dr*dsuma1
d2B3--(2./3.)*pi*dr*dsuma2
write(6,100)Tr,B3,d1B3,d2B3
100 format(f6.2,f15.4,f17.4,f19.5)
Tr=Tr+0.005
enddo
close(6)
end program B3_LJm

```

```

function f(r,Tr)
implicit none
double precision::r,Tr,sigma,epsil,c5
double precision::beta,x,u,f,c1,c2,c3,c4
beta=1./Tr
c1=0.0163169237
c2=3136.5686
c3=-68.069
c4=-0.0833111261
c5=0.746882273
sigma=1.0
epsil=1.0
if (r<=2.3*sigma) then
u=4*epsil*((sigma/r)**12-(sigma/r)**6)+c1
elseif ((r>2.3*sigma).and. (r<2.5*sigma)) then
u=c2*(sigma/r)**12+c3*(sigma/r)**6+c4*(r/sigma)**2+c5
else
u=0
endif
x=-beta*u
f=exp(x)-1
end function f

```

```
function df1(r,Tr)
implicit none
double precision::r,Tr,sigma,epsil,c1,c2,c3
double precision::beta,u,x,df1,alfa,c4,c5
beta=1./Tr
c1=0.0163169237
c2=3136.5686
c3=-68.069
c4=-0.0833111261
c5=0.746882273
sigma=1.0
epsil=1.0
if (r<=2.3*sigma) then
u=4*epsil*((sigma/r)**12-(sigma/r)**6)+c1
elseif ((r>2.3*sigma).and. (r<2.5*sigma)) then
u=c2*(sigma/r)**12+c3*(sigma/r)**6+c4*(r/sigma)**2+c5
else
u=0
endif
x=-beta*u
alfa=exp(x)
if (alfa>0) then
df1=-u*alfa
else
df1=0
endif
end function df1
```

```
function df2(r,Tr)
implicit none
double precision::r,Tr,sigma,epsil,c1,c2,c3
double precision::u,x,df2,beta,alfa,c4,c5
beta=1./Tr
epsil=1.0
c1=0.0163169237
c2=3136.5686
c3=-68.069
c4=-0.0833111261
c5=0.746882273
sigma=1.0
epsil=1.0
if (r<=2.3*sigma) then
u=4*epsil*((sigma/r)**12-(sigma/r)**6)+c1
elseif ((r>2.3*sigma).and. (r<2.5*sigma)) then
u=c2*(sigma/r)**12+c3*(sigma/r)**6+c4*(r/sigma)**2+c5
else
u=0
```

```

endif
x=-beta*u
alfa=exp(x)
if (alfa>0) then
df2=u*u*alfa
else
df2=0
endif
end function df2

```

```

SUBROUTINE TRF (AR,AI,N,M)
! AR: es la parte real de los datos
!AI: es la parte imaginaria
! se debe cumplir N=2**M
IMPLICIT NONE
INTEGER, INTENT (IN) :: N,M
INTEGER :: N1,N2,I,J,K,L,L1,L2
double precision :: PI,A1,A2,Q,U,V
double precision, INTENT (INOUT), DIMENSION (N) :: AR,AI
PI = 4.0*ATAN(1.0)
N2 = N/2
N1 = 2**M
IF(N1.NE.N) STOP 'Indices no coinciden'
L = 1
DO K = 1, N-1
  IF (K.LT.L) THEN
    A1 = AR(L)
    A2 = AI(L)
    AR(L) = AR(K)
    AR(K) = A1
    AI(L) = AI(K)
    AI(K) = A2
  END IF
  J = N2
  DO WHILE (J.LT.L)
    L = L-J
    J = J/2
  END DO
  L = L+J
END DO
L2 = 1
DO L = 1, M
  Q = 0.0
  L1 = L2
  L2 = 2*L1
  DO K = 1, L1
    U = COS(Q)

```

```

V      = -SIN(Q)
Q      = Q + PI/L1
DO J = K, N, L2
  I      = J + L1
  A1     = AR(I)*U-AI(I)*V
  A2     = AR(I)*V+AI(I)*U
  AR(I)  = AR(J)-A1
  AR(J)  = AR(J)+A1
  AI(I)  = AI(J)-A2
  AI(J)  = AI(J)+A2
END DO
END DO
END DO
END SUBROUTINE TRF

subroutine seno_TRF(fr, fk, nr, dr)
implicit none
integer::i, nr, m
double precision::fr(0:nr), fk(0:nr)
double precision::fr2(0:2*nr), fk2(0:2*nr)
double precision::dr, dk, pi, rmax, r
m=12
pi=4.0*atan(1.0)
fk2(0)=0
fk2(nr)=0
fr2(0)=0
fr2(nr)=0
do i=1, nr-1
  fr2(i)=i*dr*fr(i)
  fr2(nr+i)=- (nr-i)*dr*fr(nr-i)
  fk2(i)=0
  fk2(nr+i)=0
enddo
call TRF(fr2, fk2, 2*nr, m+1)
dk=pi/(nr*dr)
fk(0)=0
rmax=nr*dr
do i=1, nr-1
  r=i*dr
  fk(0)=fk(0)+4*pi*(r*fr(i)*r)*dr
  fk(i)=-4*pi*rmax*fk2(i)/(i*dk)
end do
return
end subroutine seno_TRF

subroutine senoi_TRF(fk, fki, nr, dr)
implicit none

```

```

!integer,parameter::nr=1024,m=10
integer::i,nr,m
double precision::fk2(0:2*nr),fr2(0:2*nr)
double precision::fk(0:nr),fki(0:nr)
double precision::dk,pi,dr
fk2(0)=0
m=12
fk2(nr)=0
fr2(0)=0
fr2(nr)=0
pi=4*atan(1.0)
dk=pi/(nr*dr)
do i=1,nr-1
    fk2(i)=i*dk*fk(i)
    fk2(2*nr-i)=-fk2(i)
    fr2(i)=0
    fr2(2*nr-i)=0
enddo

call TRF(fk2,fr2,2*nr,m+1)
    fki(0)=0
    do i=1,nr-1
        fki(0)=fki(0)+0.5/(pi*pi)*(fk(i)*i*dk*i*dk)*dk
        fki(i)=0.5/(pi*pi)*(-nr*fr2(i)/(i*dr))*dk
    enddo
    return
end subroutine senoi_TRF

```

Programa 4: el siguiente programa resuelve el sistema de ecuaciones, dadas por las ecuaciones 35 a la 40, utilizando Runge Kutta de cuarto orden, a continuación calcula la transformada de Fourier de la posición de la masa m_1 , así como el espectro de potencia, se procede a graficar, las posiciones de cada masa en función del tiempo y el espectro de potencia versus la frecuencia en una escala semilogarítmica, la implementación se realizó en Octave, el programa llama a las funciones rk4paso.m y dy.m, esta última define el sistema de ecuaciones diferenciales. Es importante que el programa principal y las funciones estén en la misma carpeta, para que funcione.

```

% El siguiente es el programa principal, primero resuelve un sistema de
% ecuaciones diferenciales con el metodo de Runge Kutta de orden 4
% y luego calcula el espectro de potencia de la masa 1
clc,clear;
n=1000;
a=0; % tiempo inicial
b=70.0; % tiempo final
t(1)=a;
y0=[0,0,0,0,0,0]; % condiciones iniciales
y(1,:)=y0;
h=(b-a)/n; % incremento del tiempo
for i=1:n-1

```

```

    t(i+1)=t(i)+h;
    y(i+1,:)=rk4paso(t(i),y(i,:),h);
endfor
fs=1/h; % frecuencia de muestreo
fk(1:n)=(0:n-1)*fs/n; % vector frecuencia
F1=fft(y(:,1));
potencia(1:n)=F1.*conj(F1)/n; % espectro de potencia de masa 1
subplot(1,2,1)
plot(t,y(:,1),t,y(:,2),t,y(:,3)) % se grafica posiciones vs tiempo
xlabel('Tiempo');
ylabel('Posicion');
title('Posicion de cada masa vs tiempo');
subplot(1,2,2)
semilogy(fk(1:n/2),potencia(1:n/2)); % se grafica el espectro de potencia
xlabel('Frecuencia')
ylabel('Potencia');
title('Espectro de potencia de la masa m_{1}');

```

La siguiente función define el sistema de ecuaciones diferenciales.

```

function z=dy(t,y)
% sistema de ecuaciones diferenciales
m1=1.0;
m2=1.5;
m3=2.0;
k1=5;
k2=6;
k3=7;
b1=0.1;
b2=0.1;
F1=1;
F3=0;
z(1)=y(4);
z(2)=y(5);
z(3)=y(6);
z(4)=(1./m1)*(-b1*y(4)-k1*y(1)+b1*y(5)+k2*y(2)+F1);
z(5)=(1./m2)*(b1*y(4)+k1*y(1)-b1*y(5)-(k1+k2)*y(2)+k2*y(3));
z(6)=(1./m3)*(k2*y(2)-b2*y(6)-(k2+k3)*y(3)+F3);
endfunction

```

La siguiente función implementa un paso del método de Runge Kutta de cuarto orden.

```

function y=rk4paso(t,w,h)
%un paso del metodo Runge-Kutta de orden 4
k1=dy(t,w);
k2=dy(t+h/2,w+h*k1/2);
k3=dy(t+h/2,w+h*k2/2);
k4=dy(t+h,w+h*k3);

```

```
y=w+h*(k1+2*k2+2*k3+k4)/6;
endfunction
```

I REFERENCIAS

- Arfken, G. B., y Hans J. W. (1995). *Mathematical methods for physicists*. Academic Press. Descargado de www.academicpress
- Brigham, E. O. (1988). *The fast fourier transform and its applications*. Prentice Hall. Descargado de www.prenticehall.com
- Bruce E. Poling, J. M. P., y O'Connell, J. P. (2003). *The properties of gases and liquids*. McGraw-Hill. Descargado de www.mcgrawhill.com
- Fuchizaki, K., y Asano, Y. (2013). Predicting the thermodynamics properties of the modified lennard-jones fluid from the lennard-jones equation of state. *Journal of the physical society of Japan*, 82.
- Garcia, A. L. (s.f.). *Numerical methods for physics*.
- Gould, H., y Tobochnik, J. (2010). *Statistical and thermal physics: with computer applications*. Princeton University Press. Descargado de www.princeton.com
- Katherine R. S. Shaul, A. J. S., y Kofke, D. A. (2010). The effect of truncation and shift on virial coefficients of lennard-jones potentials. *Collect. Czech. Commun*, 75, 447-462.
- McQuarrie, D. A. (s.f.). *Statistical mechanics*.
- Pang, T. (s.f.). *An introduction to computational physics*.
- Ushcats, M. V. (2014). Modified lennard-jones model: virial coefficients to 7th order. *The Journal of Chemical Physics*, 140.
- Walker, D. (2013). *Computational physics*. MedTec.
- William H. Press, W. T. V. B. F., Saul A. Teukolsky. (s.f.). *Numerical recipes in fortran 90*.