Tema 1 - Escalas, condición y estabilidad

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

Facultad de Ciencias - UNAM

14 de septiembre de 2017

Consideraciones

Para dar inicio a esta parte del Tema, debemos de considerar que aunque seamos cuidadosos o no, los errores y las incertidumbres son una parte inherente de los computación científica.

Consideraciones

Algunos de esos errores son debidos a nuestra parte como usuarios, pero otros se deben a la computadora.

Los errores computacionales surgen debido a la limitada precisión con la que los equipos almacenan números o porque los algoritmos o modelos pueden fallar.

Consideraciones

Haremos una revisión sobre los errores e incertidumbres que se pueden presentar en los cálculos computacionales.

Consideraciones

Haremos una revisión sobre los errores e incertidumbres que se pueden presentar en los cálculos computacionales.

Recuerda que siendo este el tema inicial, lo que veamos aquí, se debe de extender en el manejo de todo nuestro curso de Física Computacional.

Hablando de errores

Pensemos que tenemos un programa de alta complejidad.

Para entender el por qué los errores deben ser motivo de preocupación, imaginemos un programa con el siguiente flujo lógico

$$\text{Inicio} \rightarrow U_1 \rightarrow U_2 \rightarrow \ldots \rightarrow U_n \rightarrow \text{Fin} \quad (1)$$

Hablando de errores

$$\text{Inicio} \rightarrow U_1 \rightarrow U_2 \rightarrow \ldots \rightarrow U_n \rightarrow \text{Fin}$$

Donde cada unidad U puede ser una declaración de una tarea o un paso.

Si cada unidad tiene probabilidad p de ser correcta, entonces la probabilidad conjunta P de que todo el programa sea correcto es $P = p^n$.

Hablando de errores

Digamos que tenemos un programa de tamaño mediano con $n = 1000$ pasos y que la probabilidad de que cada paso sea correcto es casi uno, $p \simeq 0.9993$.

Esto significa que terminas con $P \simeq \frac{1}{2}$, es decir, la respuesta final es tan probablemente correcta como equivocada.

Hablando de errores

El problema es que como científicos, queremos un resultado correcto o al menos en el que la incertidumbre sea pequeña y de tamaño conocido.

Fuentes de errores computacionales

Existen al menos cuatro fuentes de errores que se pueden presentar en los cálculos computacionales:

Fuentes de errores computacionales

Existen al menos cuatro fuentes de errores que se pueden presentar en los cálculos computacionales:

1. Un modelo equivocado.

Fuentes de errores computacionales

Existen al menos cuatro fuentes de errores que se pueden presentar en los cálculos computacionales:

1. Un modelo equivocado.
2. Errores aleatorios.

Fuentes de errores computacionales

Existen al menos cuatro fuentes de errores que se pueden presentar en los cálculos computacionales:

1. Un modelo equivocado.
2. Errores aleatorios.
3. Errores por aproximación

Fuentes de errores computacionales

Existen al menos cuatro fuentes de errores que se pueden presentar en los cálculos computacionales:

1. Un modelo equivocado.
2. Errores aleatorios.
3. Errores por aproximación
4. Errores por redondeo.

Un modelo equivocado

Se presentan cuando nuestra propuesta de modelo no es el pertinente, siendo quizá la parte más crítica ya que antes de continuar, debemos de repasar la parte de la física para resolver debidamente nuestro problema

Un modelo equivocado

También se consideran los errores tipográficos, introducir valores que no corresponden, ejecutar el programa incorrecto, usar el archivo de datos equivocado, etc.

Sugerencia:

Un modelo equivocado

También se consideran los errores tipográficos, introducir valores que no corresponden, ejecutar el programa incorrecto, usar el archivo de datos equivocado, etc.

Sugerencia: Si el número de errores comienza a crecer, es momento de ir a casa o tomar un descanso.

Errores aleatorios

Se presenta una imprecisión debida por acontecimientos tales como: fluctuaciones en la electrónica, incidencia de rayos cósmicos, o alguien que se tropieza con un enchufe.

Éstos errores pueden ser raros, pero no tenemos ningún control sobre ellos y su probabilidad aumenta conforme transcurre el tiempo.

Errores por aproximación

La imprecisión surge de la simplificación de las matemáticas para que un problema pueda ser resuelto en la computadora.

Incluyen la sustitución de series infinitas por sumas finitas, intervalos infinitesimales por finitos, y funciones variables por constantes.

Errores por aproximación

Por ejemplo

$$\begin{aligned}\sin(x) &= \sum_{n=1}^{\infty} \frac{(-1)^{n-1}\, x^{2n-1}}{(2n-1)!} \quad \text{(exacta)} \\ &= \simeq \sum_{n=1}^{N} \frac{(-1)^{n-1}\, x^{2n-1}}{(2n-1)!} \quad \text{(algoritmo)} \\ &= \sin(x) + \varepsilon(x, N)\end{aligned} \tag{2}$$

donde $\varepsilon(x, N)$ es el error por la aproximación, en este caso ε corresponde a los términos desde $N + 1$ hasta ∞.

Errores por aproximación

Dado que el error por la aproximación se genera en el algoritmo que usamos para aproximar la matemática, también se le conoce como error del algoritmo.

Errores por aproximación

Para una buena y razonable aproximación, el error debido por la aproximación debería de reducirse cuando el valor de N se incrementa, y debería de anularse en el límite $N \to \infty$.

Errores por aproximación

Para el ejemplo (2), como la escala de N se fija por el valor de x, un pequeño error de aproximación requiere que $N \geqslant x$.

Por lo que si x y N son valores cercanos entre sí, el error de aproximación será grande.

Errores por redondeo

La imprecisión se genera por el número finito de dígitos utilizados para almacenar números de punto flotante.

Estos “errores” son análogos a la incertidumbre en la medición de una cantidad física encontrada en un laboratorio de física.

Errores por redondeo

El error general de redondeo se acumula a medida que el equipo maneja más números, es decir, a medida que aumenta el número de pasos en un cálculo, y puede hacer que algunos algoritmos se vuelvan inestables con un rápido aumento del error.

Errores por redondeo

En algunos casos, el error por redondeo puede convertirse en el componente principal en la respuesta, lo que lleva a lo que los expertos informáticos llaman **basura**.

Errores por redondeo

Por ejemplo, si la computadora mantiene cuatro decimales, entonces almacenará $\frac{1}{3}$ como 0.3333 y $\frac{2}{3} = 0.6667$, donde el equipo ha “redondeado” el último dígito en $\frac{2}{3}$.

Errores por redondeo

Por consiguiente, si hacemos en la computadora un cálculo tan simple como $2\frac{1}{3} - \frac{2}{3}$, produce

$$2\left(\frac{1}{3}\right) - \frac{2}{3} = 0.6666 - 0.6667 = -0.0001 \neq 0 \quad (3)$$

Errores por redondeo

Por consiguiente, si hacemos en la computadora un cálculo tan simple como $2\frac{1}{3} - \frac{2}{3}$, produce

$$2\left(\frac{1}{3}\right) - \frac{2}{3} = 0.6666 - 0.6667 = -0.0001 \neq 0 \quad (3)$$

Aunque el resultado es pequeño, no es 0, y si se repete este tipo millones de veces este cálculo, la respuesta final puede que ni siquiera sea pequeña (la basura genera basura).

Cancelación en la sustracción

Un cálculo que utiliza números que se almacenan de manera aproximada en la computadora, puede devolver una solución aproximada.

Para demostrar este hecho, consideremos que hay un valor de incertidumbre, sea x_c el valor representado en la computadora del valor exacto x, tal que:

$$x_c \simeq x(1 + \varepsilon_x) \tag{4}$$

Cancelación en la sustracción

$$x_c \simeq x(1 + \varepsilon_x)$$

Donde ϵ_x es el error relativo de x_c, el cual se espera que sea similar en magnitud al épsilon de la máquina ϵ_m.

Cancelación en la sustracción

Si usamos ésta notación para una diferencia entre dos valores $a = b - c$, tenemos que:

$$a = b - c \Rightarrow a_c \simeq b_c - c_c \simeq b(1 + \epsilon_b) - c(1 + \epsilon_c)$$
$$\Rightarrow \frac{a_c}{a} \simeq 1 + \epsilon_b \frac{b}{a} - \frac{c}{a}\epsilon_c \tag{5}$$

Cancelación en la sustracción

Vemos que el error resultante en a es un promedio de los errores en b y c, y no hay seguridad en que los dos términos se cancelen.

Cancelación en la sustracción

El error en a_c se incrementa cuando se restan dos valores cercanos $(b \simeq c)$, porque entonces estamos restando las partes más significativas de ambos números y dejando las partes menos significativas propensas a errores:

$$\frac{a_c}{a} \overset{def}{=} 1+\epsilon_a \simeq 1+\frac{b}{a}(\epsilon_b-\epsilon_c) \simeq 1+\frac{b}{a}max(|\epsilon_b|,|\epsilon_c|) \tag{6}$$

Cancelación en la sustracción

Esto muestra que incluso si los errores relativos en b y c pueden cancelar algo, se multiplican por número grande b/a, lo que puede aumentar significativamente el error.

Debido a que no podemos asumir ningún signo de los errores, debemos asumir el peor escenario: el máximo en la expresion (6).

Ejercicio 1

Aprendimos en la secundaria a resolver la ecuación homogénea de segundo grado:

$$a\,x^2 + b\,x + c = 0 \tag{7}$$

que tiene una solución analítica que se puede escribir como

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4\,a\,c}}{2\,a} \quad \text{o}$$

$$x_{1',2'} = \frac{-2\,c}{b \pm \sqrt{b^2 - 4\,a\,c}} \tag{8}$$

Revisando la expresión anterior (8) vemos que la cancelación en la sustracción aumenta (y por tanto, un incremento en el error) cuando $b^2 \gg 4\,a\,c$ debido a que la raíz cuadrada y el siguiente término están muy próximos a cancelarse.

Manos al código

1. Escribe un programa que calcule las cuatro soluciones para valores arbitrarios de a, b y c. Considera las siguientes expresiones:

$$x^2 - 5\,x + 6 = 0$$
$$-x^2 - 7\,x + 3 = 0$$
$$x^2 + 2\,x + 2 = 0$$
$$x^2 + 3\,x + 1 = 0$$

Manos al código

2. Revisa cómo los errores obtenidos en los cálculos, aumentan conforme hay una cancelación en la sustracción de términos y su relación con la precisión de la máquina. Prueba con los siguientes valores $a = 1$, $b = 1, c = 10^{-n}, n = 1, 2, 3, \ldots$

Manos al código

4. Revisa cómo los errores obtenidos en los cálculos, aumentan conforme hay una cancelación en la sustracción de términos y su relación con la precisión de la máquina. Prueba con los siguientes valores $a = 1$, $b = 1, c = 10^{-n}, n = 1, 2, 3, \ldots$
5. Estima el error relativo para los valores obtenidos, considera que la expresión que devuelve x_1, x_2 es la del valor exacto.

Manos al código

6. Con los valores de error y de n, genera una gráfica. Interpreta lo que obtienes.

Manos al código

6. Con los valores de error y de n, genera una gráfica. Interpreta lo que obtienes.
7. Cómo mejorarías el programa para obtener la mayor precisión en tu respuesta?

Ejercicio 2: Suma 1

Del ejercicio anterior, hemos visto que la cancelación en la sustracción de los términos se presentan cuando se suma una serie con signos alternantes.

Ejercicio 2: Suma 1

Del ejercicio anterior, hemos visto que la cancelación en la sustracción de los términos se presentan cuando se suma una serie con signos alternantes.

Consiera ahora la siguiente suma:

$$S_N^{(1)} = \sum_{n=1}^{2N} (-1)^n \frac{n}{n+1} \tag{9}$$

Ejercicio 2: Suma 2

Si sumamos de manera separada los valores impares y los pares de x, tendremos dos sumas:

$$S_N^{(2)} = -\sum_{n=1}^{N} \frac{2\,n-1}{2\,n} + \sum_{n=1}^{N} \frac{2\,n}{2\,n+1} \tag{10}$$

En esta expresión, todos los términos son positivos con sólo una sustracción al final del cálculo.

Ejercicio 2: Suma 3

Podemos eliminar la diferencia mediante una combinación entre las dos sumas anteriores, quedando de la siguiente manera

$$S_N^{(3)} = \sum_{n=1}^{N} \frac{1}{2\,n\,(2\,n+1)} \tag{11}$$

Revisando las tres sumas

Sabemos que aunque el valor de las tres sumas $S_N^{(1)}$, $S_N^{(2)}$, $S_N^{(3)}$, es el mismo, el resultado número puede ser diferente.

1. Escribe un programa que calcule $S_N^{(1)}$, $S_N^{(2)}$, $S_N^{(3)}$.

Revisando las tres sumas

2. Supongamos que $S_N^{(3)}$ es el valor exacto de la suma. Grafica el error relativo contra el número de términos en la suma (tip: usa una escala log-log). Comienza con $N = 1$ hasta $N = 1000000$. Describe la gráfica.

Revisando las tres sumas

2. Supongamos que $S_N^{(3)}$ es el valor exacto de la suma. Grafica el error relativo contra el número de términos en la suma (tip: usa una escala log-log). Comienza con $N = 1$ hasta $N = 1000000$. Describe la gráfica.
3. Identifica en tu gráfica una región en donde la tendencia es casi lineal, ¿qué representa ésta sección con respecto al error?

Error por redondeo en un paso

Comencemos por ver cómo surge el error en una división con las representaciones de computadora de dos números:

$$a = \frac{b}{c} \Rightarrow a_c = \frac{b_c}{c_c} = \frac{b\left(1+\varepsilon_b\right)}{c\left(1+\varepsilon_c\right)}$$

$$\Rightarrow \frac{a_c}{a} = \frac{1+\varepsilon_b}{1+\varepsilon_c} \simeq \left(1+\varepsilon_b\right)\left(1-\varepsilon_c\right) \simeq 1+\varepsilon_b-\varepsilon_c$$

$$\Rightarrow \frac{a_c}{a} \simeq 1+\left|\varepsilon_b\right|+\left|\varepsilon_c\right| \tag{12}$$

Error por redondeo en un paso

Aquí se han ignorado los términos muy pequeños ε^2 y se han añadido errores en valor absoluto ya que no podemos asumir que tenemos la suerte de tener errores desconocidos que se cancelan entre sí.

Error por redondeo en un paso

Debido a que añadimos los errores en valor absoluto, esta misma regla se aplica a la multiplicación.

La ecuación (12) es sólo la regla básica de la propagación de errores a partir del trabajo que se hace en un laboratorio:

Error por redondeo en un paso

Se suman las incertidumbres en cada cantidad involucrada en un análisis para llegar a la incertidumbre general.

Errores por redondeo en varios pasos

Existe un modelo útil para aproximar cómo se acumula el error de redondeo en un cálculo que implica un gran número de pasos.

Errores por redondeo en varios pasos

Consideremos el error en cada paso como un "paso" literal en una *caminata aleatoria*[1], es decir, una caminata para la cual cada paso está en una dirección aleatoria.

[1] Parte del contenido del Tema de Métodos de Montecarlo

Errores por redondeo en varios pasos

La distancia total recorrida en N pasos de longitud r, es en promedio:

$$R \simeq \sqrt{N}\, r \tag{13}$$

Errores por redondeo en varios pasos

La distancia total recorrida en N pasos de longitud r, es en promedio:

$$R \simeq \sqrt{N}\, r \tag{13}$$

Por analogía, el error relativo total ε_0 que se genera luego de N pasos computacionales cada uno, con un error de precisión de la máquina ε_m, es en promedio

$$\varepsilon_{ro} \simeq \sqrt{N}\, \varepsilon_m \tag{14}$$

Errores por redondeo en varios pasos

Si los errores de redondeo en un algoritmo particular no se acumulan de manera aleatoria, entonces se necesita un análisis detallado para predecir la dependencia del error en el número de pasos N.

En algunos casos no puede haber cancelación, y el error puede aumentar como $N\ \varepsilon_m$.

Conclusión sobre los errores por redondeo

Nuestra discusión de los errores tiene una implicación importante que hay que tener presente antes de ser impresionado por un cálculo que requiere horas de ejecución.

Conclusión sobre los errores por redondeo

Nuestra discusión de los errores tiene una implicación importante que hay que tener presente antes de ser impresionado por un cálculo que requiere horas de ejecución.

Un equipo rápido puede completar 10^{10} operaciones de punto flotante por segundo.

Conclusión sobre los errores por redondeo

Esto significa que un programa que se ejecuta durante 3 horas realiza aproximadamente 10^{14} operaciones.

Por lo tanto, si el error de redondeo se acumula aleatoriamente, después de 3 horas esperamos un error relativo de $10^{7}\ \varepsilon_m$.

Conclusión sobre los errores por redondeo

Para que el error sea menor que la respuesta, necesitamos que $\varepsilon_m < 10^{-7}$, lo que requiere doble precisión y un buen algoritmo.

Si queremos una respuesta de mayor precisión, *entonces necesitaremos un algoritmo muy bueno.*

Funciones de Bessel

La acumulación de errores por redondeo a menudo limita la capacidad de un programa para calcular con precisión.

Funciones de Bessel

La acumulación de errores por redondeo a menudo limita la capacidad de un programa para calcular con precisión.

Revisaremos como ejercicio, la manera de calcular las funciones esféricas de Bessel $j_\ell(x)$ y de Neumann $n_\ell(x)$.

Funciones de Bessel

Estas funciones son respectivamente, las soluciones regulares/irregulares (no singulares / singulares en el origen) de la ecuación diferencial

$$x^2\, f''(x) + 2\, x\, f'(x) + \left[x^2 - \ell(\ell+1)\right] f(x) = 0 \quad (15)$$

Funciones de Bessel y esféricas de Bessel

Las funciones esféricas de Bessel se relacionan con las funciones de Bessel de primera clase por

$$j_\ell(x) = \sqrt{\frac{\pi}{2\,x}}\, J_{n+\frac{1}{2}}(x)$$

Funciones de Bessel y esféricas de Bessel

Se encuentran en diversos problemas de la física, coo la expansión de una onda plana en ondas esféricas parciales

$$e^{i\,\mathbf{k}\cdot\mathbf{r}} = \sum_{\ell=0}^{\infty} i^{\ell}\,(2\,\ell+1)\,j_{\ell}(k\,r)\,P_{\ell}(\cos\theta) \qquad (16)$$

Funciones esféricas de Bessel de orden n

A continuación se muestra una gráfica con las funciones esféricas de Bessel de orden $n = 0, \ldots, 5$:

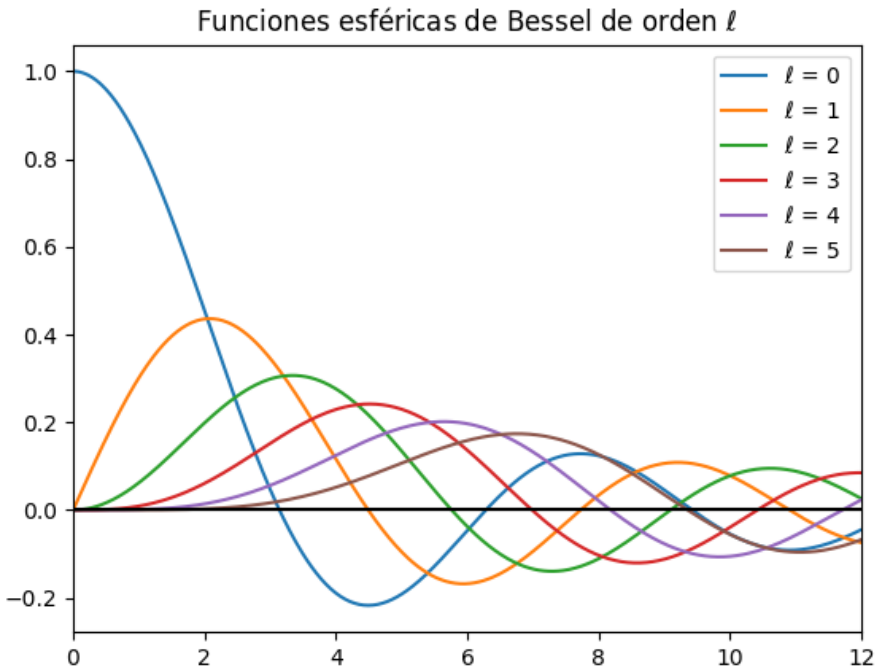


Figura: La gráfica se generó usando la librería de funciones especiales de `python`.

Valores para $\ell = 0, 1$

Los dos primero valores para ℓ son

$$j_0(x) = +\frac{\sin x}{x}, \qquad j_1(x) = +\frac{\sin x}{x^2} - \frac{\cos x}{x} \tag{17}$$

$$n_0(x) = -\frac{\cos x}{x}, \qquad n_1(x) = -\frac{\cos x}{x^2} - \frac{\sin x}{x} \tag{18}$$

Relaciones de recursión

La manera clásica de calcular las funciones de Bessel de orden j: $j_\ell(x)$ es sumar su serie de potencias para valores pequeños de $\frac{x}{\ell}$ y sumando su expansión asintótica para valores de x grandes.

Relaciones de recurrencia

El enfoque que usaremos, se basa en *las relaciones de recurrencia*:

$$j_{\ell+1}(x) = \frac{2\,\ell+1}{x}\, j_\ell(x) - j_{\ell-1}(x) \quad \text{hacia arriba} \tag{19}$$

$$j_{\ell-1}(x) = \frac{2\,\ell+1}{x}\, j_\ell(x) - j_{\ell+1}(x) \quad \text{hacia abajo} \tag{20}$$

Relaciones de recurrencia

Las ecuaciones (19) y (20) son la misma relación:

1. Una escrita para la recurrencia hacia adelante que parte de valores pequeños a valores grandes de ℓ.

Relaciones de recurrencia

Las ecuaciones (19) y (20) son la misma relación:

1. Una escrita para la recurrencia hacia adelante que parte de valores pequeños a valores grandes de ℓ.
2. La otra relación de recurrencia es descendente para valores de ℓ pequeños.

Relaciones de recurrencia

Las ecuaciones (19) y (20) son la misma relación:

1. Una escrita para la recurrencia hacia adelante que parte de valores pequeños a valores grandes de ℓ.
2. La otra relación de recurrencia es descendente para valores de ℓ pequeños.

Relaciones de recurrencia

Las ecuaciones (19) y (20) son la misma relación:

1. Una escrita para la recurrencia hacia adelante que parte de valores pequeños a valores grandes de ℓ.
2. La otra relación de recurrencia es descendente para valores de ℓ pequeños.

Con unas cuantas sumas y multiplicaciones, las relaciones de recurrencia permiten el cálculo rápido y sencillo de todo el conjunto de valores de j_ℓ para x fijo y todo ℓ.

Relaciones de recurrencia

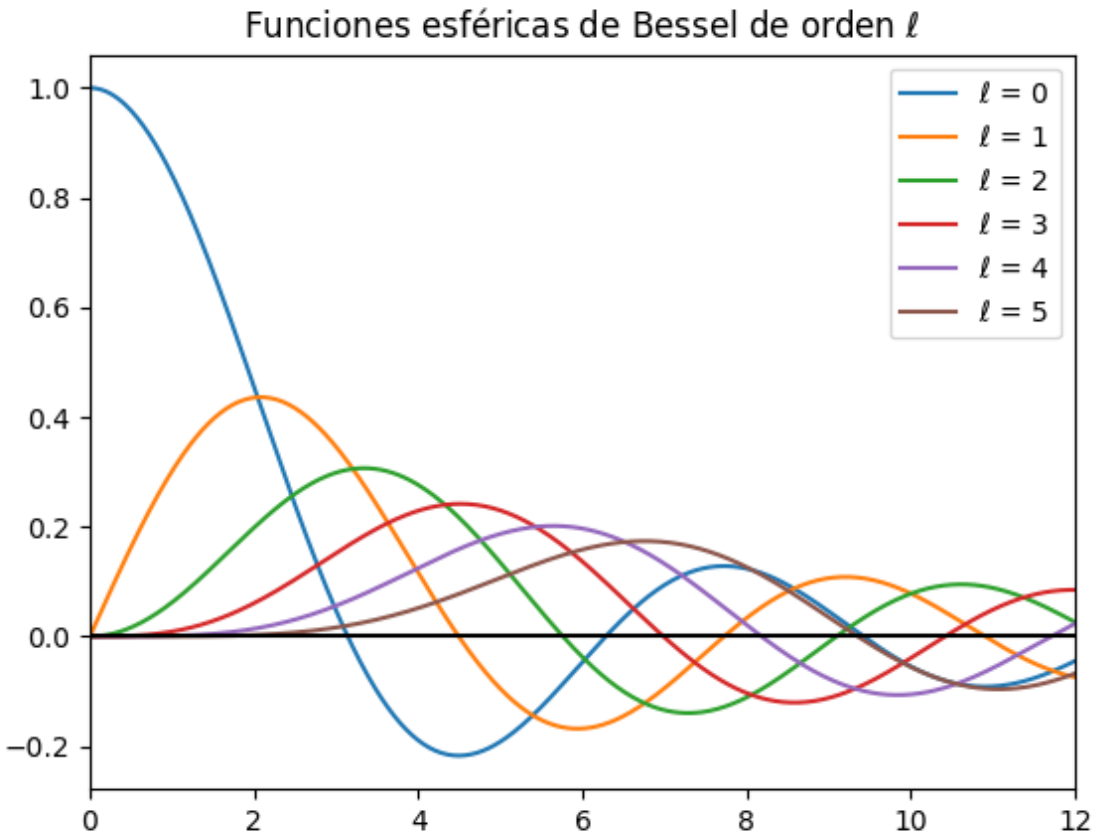
Para la relación de recurrencia hacia adelante de ℓ para un x fijo, comenzamos con las formas conocidas para j_0 y j_1 (17) y usamos (19).

Relaciones de recurrencia

Como veremos, esta recurrencia hacia adelante parece funcionar al principio, pero luego falla.

La razón de la falla se puede ver en las gráficas de $j_\ell(x)$ y $n_\ell(x)$ como función de x.

Relaciones de recurrencia



Relaciones de recurrencia

Si iniciamos con $x \simeq 2$ y $\ell = 0$, veremos que a medida que con la recurrencia hacia adelante para j_ℓ con valores mayores de ℓ con (19), esencialmente tomamos la diferencia de dos funciones “grandes” para producir un valor “pequeño” para j_ℓ.

Relaciones de recurrencia

Este proceso sufre de cancelación sustractiva y siempre reduce la precisión.

A medida que continuamos con la recurrencia, tomamos la diferencia de dos funciones pequeñas con errores grandes y producimos una función más pequeña con un error aún mayor. Después de cierto tiempo, nos quedamos con sólo el error de redondeo (basura).

Estimando el error

Para ser más específicos, llamemos $j_\ell^{(c)}$ al valor numérico que calculamos como nua aproximación para $j_\ell(x)$.

Por lo que si comenzamos con un valor neto j_ℓ, después de un corto tiempo la falta de precisión de la computadora se mezcla efectivamente en un poco de n_ℓ:

$$j_\ell^{(c)} = j_\ell(x) + \varepsilon\, n_\ell(x) \tag{21}$$

Estimando el error

Esto no lo podemos evitar, porque tanto j_ℓ como n_ℓ satisfacen la misma ecuación diferencial y, por esa razón, la misma relación de recurrencia.

La mezcla de n_ℓ se convierte en un problema cuando el valor numérico de $n_\ell(x)$ es mucho mayor que el de $j_\ell(x)$ porque incluso una cantidad minúscula de un número muy grande puede ser grande.

Estimando el error

Por el contrario, si usamos la relación de recurrencia hacia adelante (19) para generar la función esférica de Neumann n_ℓ, no habrá ningún problema porque estamos combinando funciones pequeñas para producir más grandes, ya que es un proceso que no contiene cancelación sustractiva.

Solución al problema

La solución simple a este problema es usar (20) para la recursión descendente de los valores de j_ℓ que comienzan en un valor grande $\ell = L$.

Esto evita la cancelación sustractiva tomando valores pequeños de $j_{\ell+1}(x)$ y $j_\ell(x)$ y generando por la suma un valor mayor en $j_{\ell-1}(x)$.

Solución al problema

Mientras que el error todavía puede mantenerse como una función de Neumann, la magnitud real del error disminuirá rápidamente conforme la recurrencia hacia atrás use valores pequeños de ℓ.

Solución al problema

De hecho, si empezamos iterando hacia atrás con valores arbitrarios para $j_{L+1}^{(x)}$ y $j_L^{(c)}$, después de un corto tiempo llegaremos a la correcta relación de ℓ para este valor de x.

Solución al problema

Aunque el valor numérico de j_0^c así obtenido no será correcto porque depende de los valores arbitrarios asumidos para j_{L+1} y $j_L^{(c)}$, los valores relativos serán precisos.

Solución al problema

Los valores absolutos se fijan a partir del valor conocido (17), $j_0(x) = \sin x/x$.

Debido a que la relación de recurrencia es una relación lineal entre los valores de j_ℓ, sólo necesitamos normalizar todos los valores calculados mediante

Solución al problema

$$j_\ell^{\text{normalizada}} = j_\ell^{(c)}(x) \times \frac{j_0^{\text{analitica}}}{j_0^{(c)}} \tag{22}$$

En consecuencia, después de haber terminado la recurrencia hacia abajo, se obtendrá la respuesta final normalizando todos los valores de $j_\ell^{(c)}$ basados en el valor conocido para j_0.

Propuesta de código

Veamos la manera de implementar un código con `python`para calcular el valor de la función esférica de Bessel con la recurrencia hacia atrás.

Propuesta de código I

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import scipy.special as spl
4
5  Xmax = 40.
6  Xmin = 0.25
7  paso = 0.1
8  orden = 10; inicio = 50
9
10 def abajo (x, n, m):
11     j = np.zeros( (inicio + 2), float)
12     j[m+1] = j[m] = 1.
13
14     for k in range(m, 0, -1):
15         j[k-1] = ((2.*k+1.)/x)*j[k] - j[k+1]
```

Propuesta de código II

```
16
17      escala = (np.sin(x)/x)/j[0]
18
19      return j[n] * escala
20
21 valores_y = []
22 valores_x = []
23
24 for x in np.arange(Xmin, Xmax, paso):
25     valores_y.append(abajo(x, orden, inicio))
26     valores_x.append(x)
27
28 plt.plot(valores_x, valores_y, color='r')
29 plt.axhline(y=0, ls='dashed', color = 'k')
30 plt.xlim([0.25, 40])
31 plt.show()
```

Propuesta de código III

Función aproximada

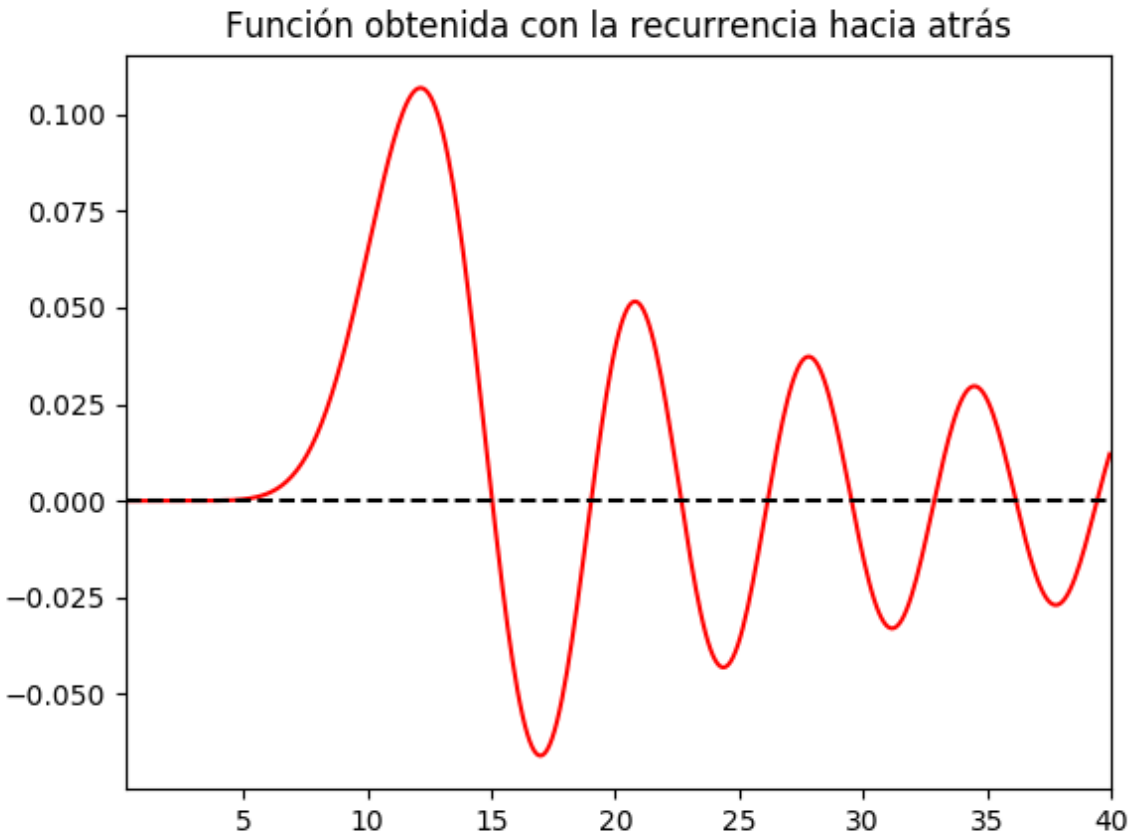


Figura: Gráfica obtenida con el algoritmo propuesto