UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

Universidad Autónoma de Querétaro

Facultad de Ingeniería

Análisis de error

Análisis numérico

Autor:
David Gómez Torres

Agosto 2021

ÍNDICE GENERAL

CAPÍTULO 1

INTRODUCCIÓN

Antes de presentar un serie de ejercicios resueltos y reportar el código con el que fueron resueltos, a continuación se hace un resumen de las páginas 43-61 del libro [**1**].

1.1. Cifras significativas

El concepto de cifras o dígitos significativos se ha desarrollado para designar formalmente la confiabilidad de un valor numérico. Las cifras significativas de un número son aquellas que pueden utilizarse en forma confiable.

El concepto de cifras significativas tiene dos implicaciones importantes en el estudio de los métodos numéricos:

1. Los métodos numéricos dan resultados aproximados. Por lo tanto, se deben desarrollar criterios para especificar qué tan confiables son dichos resultados. Una manera de hacerlo es en términos de cifras significativas.

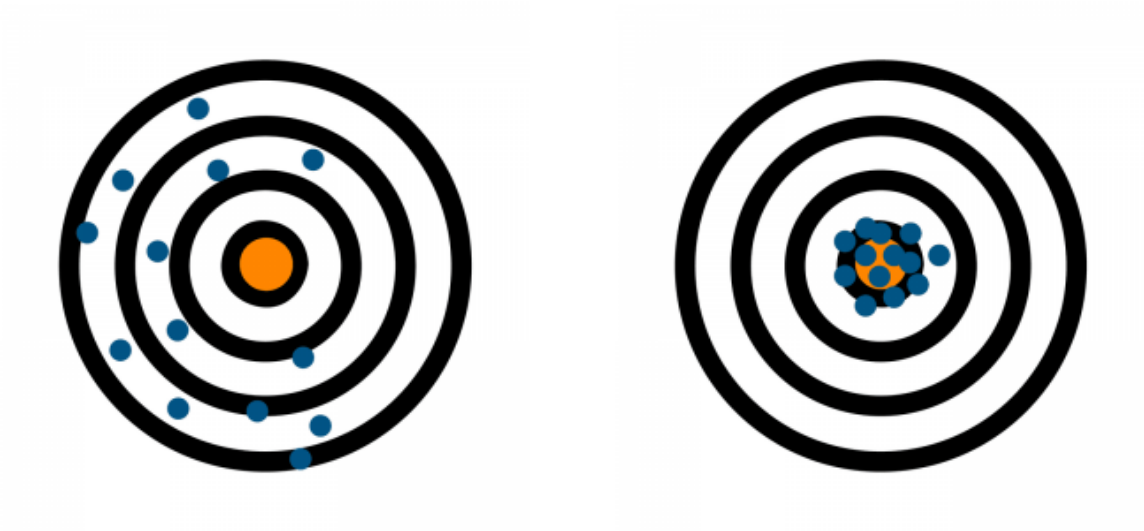
2. Aunque ciertas cantidades como π, e o $\sqrt{7}$ representan cantidades específicas, no se pueden expresar exactamente con un número finito de dígitos. Por ejemplo, $\pi = 3{,}141592653589...$ hasta el infinito. Como las computadoras retienen sólo un número finito de cifras significativas, tales números jamás se podrán representar con exactitud. A la omisión del resto de cifras significativas se le conoce como **error de redondeo**.

1.2. Exactitud y precisión

Los errores asociados con cálculos y medidas se pueden caracterizar con respecto a su exactitud y su precisión. La **exactitud** se refiere a qué tan cercano está el valor calculado o medido del valor verdadero. La **precisión** se refiere a qué tan cercanos se encuentran, unos de otros, diversos valores calculados o medidos.

La **inexactitud** (conocida también como **sesgo**) se define como una desviación sistemática del valor verdadero.
La **imprecisión** (también llamada **incertidumbre**), por otro lado, se refiere a la magnitud en la dispersión de los datos.

Los métodos numéricos deben ser lo suficientemente exactos o sin sesgo para satisfacer los requisitos de un problema particular de ingeniería. También deben ser suficientemente precisos para ser adecuados en el diseño de la ingeniería.



(a) Sin precisión ni exactitud (b) Con precisión y exactitud

Figura 1.1: Diferencia gráfica entre precisión y exactitud

1.3. Definiciones por error

Los errores numéricos surgen del uso de aproximaciones para representar operaciones y cantidades matemáticas exactas. Éstas incluyen los **errores de truncamiento** que resultan del empleo de aproximaciones como un procedimiento matemático exacto, y los **errores de redondeo** que se producen cuando se usan números que tienen un límite de cifras significativas para representar números exactos.

Para ambos tipos de errores, la relación entre el resultado exacto, o verdadero, y el aproximado está dada por: Valor verdadero = valor aproximado + error. Matemáticamente,

$$V_t = V_a + E_t \tag{1.1}$$

donde E_t se usa para denotar el valor exacto del error.

Una desventaja en esta definición es que no toma en consideración el orden de la magnitud del valor que se estima, para eso hay una manera de tomar en cuenta las magnitudes de las cantidades que se evalúan el cual consiste en normalizar el error respecto al valor verdadero, es decir

$$Error\ relativo\ = \frac{E_t}{V_t} \tag{1.2}$$

si quisiéramos representarlo en porcentaje, sólo habría que multiplicar la ecuación (1.2) por 100 %.
Ciertos métodos numéricos usan un método iterativo para calcular los resultados. En tales métodos se hace una aproximación considerando la aproximación anterior, en tales casos, el error a menudo se calcula como la diferencia entre la aproximación previa y la actual.
Por lo tanto, el error relativo porcentual está dado por

$$\epsilon_a = \frac{aproximación\ actual - aproximación\ anterior}{aproximaciónactual} 100\,\% \tag{1.3}$$

A menudo, cuando se realizan cálculos, no importa mucho el signo del error, sino más bien que su valor absoluto porcentual sea menor que una tolerancia porcentual prefijada es, por lo tanto, útil emplear el valor absoluto de las ecuaciones 1.1 a 1.3. En tales casos, los cálculos se repiten hasta que

$$\epsilon_a < \epsilon_s \tag{1.4}$$

Si se cumple la relación anterior, entonces se considera que el resultado obtenido está dentro del nivel aceptable fijado previamente ϵ_s.

Es posible demostrar que si el siguiente criterio se cumple, se tendrá la seguridad que el resultado es correcto en al menos n cifras significativas:

$$\epsilon_s = (0{,}5 \times 10^{2-n})\,\% \tag{1.5}$$

1.4. Errores de redondeo

1.4.1. Representación de números en la computadora

Numéricamente los errores de redondeo se relacionan de manera directa con la forma en que se guardan los números en la memoria de la computadora. La unidad fundamental mediante la cual se representa la información se llama término. Ésta es una entidad que consiste en una cadena de **dígitos binarios** o **bits**.

- **Sistemas numéricos**: Un sistema numérico es simplemente una convención para representar cantidades. Una base es el número que se usa como referencia para construir un sistema. El sistema de base 10 utiliza 10 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) para representar números.

  Para grandes cantidades se usa la combinación de estos dígitos básicos; con la posición o *valor de posición* se especifica su magnitud. El dígito en el extremo derecho de un número entero representa un número del 0 al 9. El segundo dígito a partir de la derecha representa un múltiplo de 10. El tercer dígito a partir de la derecha representa un múltiplo de 100 y así sucesivamente.

- **Representación entera:** El método más sencillo se denomina método de magnitud con signo y emplea el primer bit de un término para indicar el signo: con un 0 para positivo y un 1 para el negativo. Los bits sobrantes se usan para guardar el número.

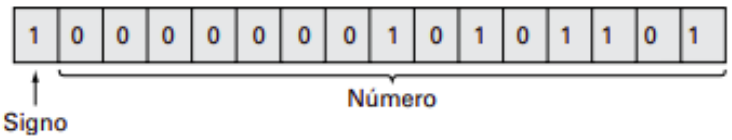


Figura 1.2: Representación entera

- **Representación del punto flotante:** Las cantidades fraccionarias generalmente se representan en la computadora usando la forma de punto flotante. Con este método, el número se expresa como una parte fraccionaria, llamada *mantisa* o significando, y una parte entera, denominada *exponente* o característica, esto es, $m \cdot b^e$ donde m = la mantisa, b = la base del sistema numérico que se va a utilizar y e = el exponente.

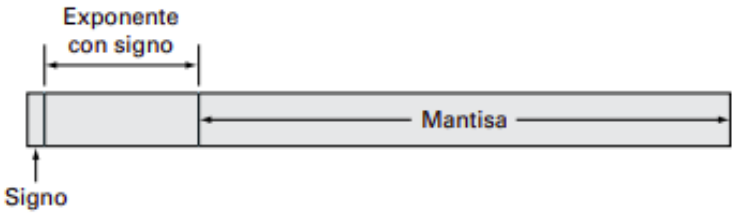


Figura 1.3: Representación del punto flotante

A continuación se enumeran 3 características importantes de la representación de punto flotante:

1. **El rango de cantidades que pueden representarse es limitado.** Intentar emplear números fuera del rango aceptable dará como resultado el llamado *error de desbordamiento (overflow)*. Sin embargo, además de las grandes cantidades, la representación de punto flotante tiene la limitación adicional de que números muy pequeños no pueden representarse.
2. **Existe sólo un número finito de cantidades que puede representarse dentro de un rango.** Es evidente que los números irracionales no pueden representarse de manera exacta. Además, los números racionales que no concuerdan exactamente con uno de los valores en el conjunto tampoco pueden ser representados en forma precisa. A los errores ocasionados por la aproximación en ambos casos se les conoce como *errores de cuantificación*.
3. **El intervalo entre los números aumenta conforme los números crecen en magnitud.** Para normalizar los números de punto flotante, esta proporcionalidad se expresa, para los casos en que se emplea el corte, como

$$\frac{\Delta x}{x} \leq \mathscr{E} \tag{1.6}$$

y, para los casos donde se utiliza el redondeo, como

$$\frac{\Delta x}{x} \leq \frac{\mathscr{E}}{2} \tag{1.7}$$

donde a $\mathscr{E}$ se le denomina épsilon de la máquina, el cual se calcula como

$$\mathscr{E} = b^{1-t} \tag{1.8}$$

donde b es el número base y t es el número de dígitos significativos en la mantisa.

CAPÍTULO 2

METODOLOGÍA

2.1. Cambio de base

Problema 3.1

Convierta los números que se presentan base 2 a base 10: a) 101101; b) 110.011 y c) 0.01101.

```
suma = 0
i = 0

binario = float(input("Ingrese el numero binario: "))

while (binario>= 1):
    d = binario%10
    binario = binario//10
    suma = suma + d*pow(2,i)
    i = i+1

print("Su conversion en decimal es: ", suma)
```

Evidencia del funcionamiento

Figura 2.1: Conversión de números

```
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python3 binario-decimal.py
Ingrese el numero binario: 101101
Su conversion en decimal es:  45.0
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python3 binario-decimal.py
Ingrese el numero binario: 110.011
Su conversion en decimal es:  6.010999999999996
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python3 binario-decimal.py
Ingrese el numero binario: 0.01101
Su conversion en decimal es: 0.40625
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$
```

2.2. Epsilon de la computadora

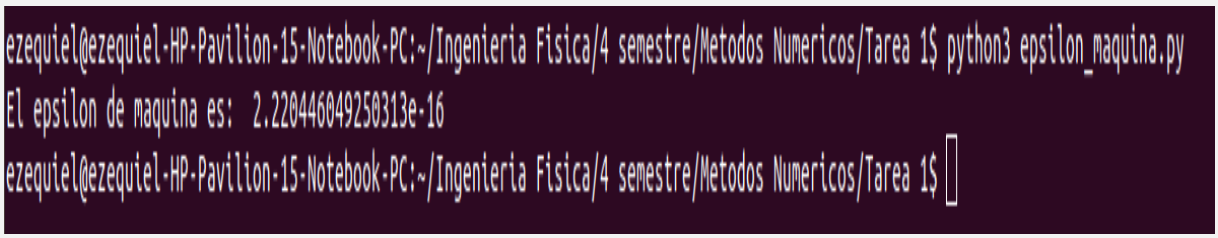
Problema 3.3

Realice su propio programa con base en la figura 3.11 y úselo para determinar el épsilon de máquina de su computadora.

```
epsilon = 1

while(epsilon + 1 > 1):

    epsilon = epsilon/2

epsilon = 2*epsilon

print("El epsilon de maquina es: " ,epsilon)
```

Evidencia del funcionamiento

Figura 2.2: Cálculo del epsilon de máquina



2.3. Serie infinita

Problema 3.5

La serie infinita

$$f(n) = \sum_{i=1}^{n} \frac{1}{i^4}$$

converge a un valor de $f(n) = \pi^4/90$ conforme n se tiende a infinito.

Escriba un programa de precisión sencilla para calcular $f(n)$ para $n = 10000$ por medio de calcular la suma desde $i = 1$ hasta 10000. Después repita el cálculo pero en sentido inverso, es decir, desde $i = 10000$ a 1, con incrementos de –1.
En cada caso, calcule el error relativo porcentual verdadero. Explique los resultados.

```
import numpy as np
import math


#inicializar la serie
serie = 0
serie1 = 0

real = (math.pi**4)/90


#bucle para hacer la sumatoria de 1-10,000
for x in np.arange(1, 10000,1):
    serie += (1/x**4)

ea = ((real-serie)/real)*100

print("\n" + "La aproximacion de 1-10,000 es: " + str(serie))

print("El error relativo porcentual es: " + str(ea) + "%" + "\n\n")

#bucle para hacer la sumatoria de 10,000-1
for x in np.arange(10000,1,-1):
   serie1 += (1/x**4)

ea1 = ((real-serie1)/real)*100

print("La aproximacion de 10,000-1 es: " + str(serie1))
print("El error relativo porcentual es: " + str(ea1) + "%")
```

Evidencia del funcionamiento

Figura 2.3: Serie que converge a $\pi^4/90$

```
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python3 serieinfinita.py

La aproximacion de 1-10,000 es: 1.082323233710861
El error relativo porcentual es: 2.558289554517808e-11%


La aproximacion de 10,000-1 es: 1.082521237710861
El error relativo porcentual es: 2.559239564518810e-11%
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$
```

2.4. Derivada de una función

Problema 3-7

La derivada de $f(x) = 1/(1–3x^2)$ está dada por

$$\frac{6x}{(1-3x^2)^2}$$

¿Esperaría dificultades para evaluar esta función para x = 0.577? Inténtelo con aritmética de 3 y 4 dígitos con corte.

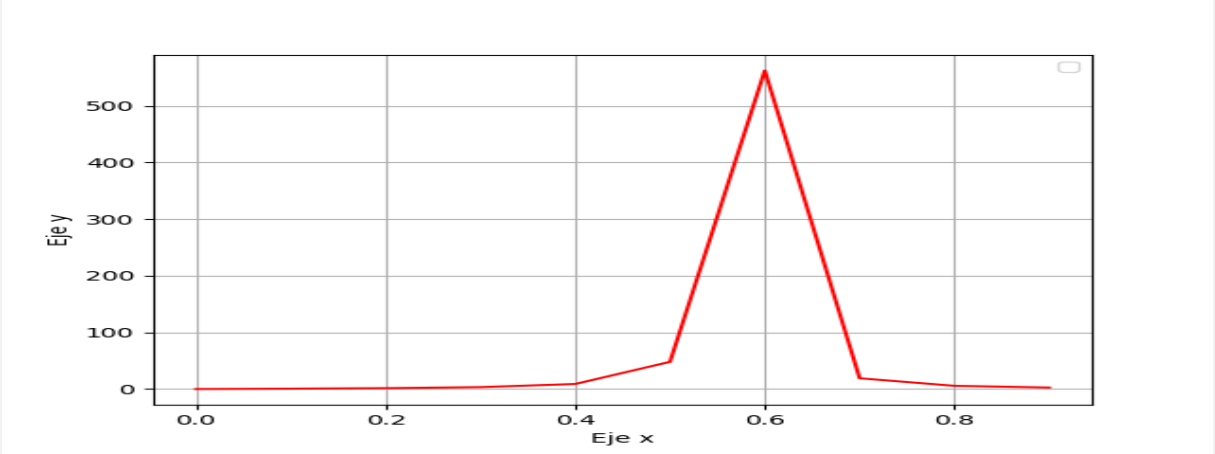
```python
import numpy as np
import matplotlib.pyplot as plt
import math

#imprimir grafica de la derivada
x = np.array(range(10))*0.1
y = np.zeros(len(x))

for i in range(len(x)):
    y[i] = 6*x[i]/(1-3*(x[i])**2)**2

    # y[i] = math.sin(x[i])

plt.plot(x,y,color='r')

plt.xlabel('Eje x')
plt.ylabel('Eje y')

plt.legend()
plt.grid()
plt.show()

#evaluar derivada

#pedir evaluacion
x = float(input("Ingrese el numero a evaluar: "))

#definir funcion
fun = 6*(x)/(1-3*(x)**2)**2

print(round(fun,3)) #3 digitos de corte
print(round(fun,4)) #4 digitos de corte
```

Evidencia del funcionamiento

Figura 2.4: Función evaluada

```
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python3 graficaFuncion.py
Ingrese el numero a evaluar: 0.577
Para 3 digitos de corte  2352910.793
Para 4 digitos de corte  2352910.7926
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$
```

Figura 2.5: Función graficada



2.5. Memoria RAM

Problema 3.9

Calcule la memoria de acceso aleatorio (RAM) en megabytes, que es necesaria para almacenar un arreglo multidimensional de $20 \times 40 \times 120$. Este arreglo es de doble precisión, y cada valor requiere una palabra de 64 bits. Recuerde que una palabra de 64 bits = 8 bytes, y un kilobyte = 2^{10} bytes. Suponga que el índice comienza en 1.

De la multiplicación del arreglo 3-dimensional, tenemos que

$$20 \times 40 \times 120 \;=\; 96,000\ bytes$$

lo que nos queda es convertir esta cantidad en megabytes, siendo

$$1\ megabyte \;=\; 1x10^{6}\ bytes$$

por lo que tenemos la siguiente conversión

$$\frac{96000}{1x10^{6}} = x$$

Por lo que la conversión nos queda como **0.096 megabytes**

2.6. Dividir y promediar

Problema 3.13

El método "dividir y promediar", un antiguo método para aproximar la raíz cuadrada de cualquier número positivo a se puede formular como

$$x = \frac{x + a/x}{2}$$

Escriba una función bien estructurada para implementar este algoritmo

```
error = 0
e = 0.00001
i = 0
y = 0

a = float(input("Introduce un numero: "))
x = float(input("Introduce una aproximacion: "))

while(error <= e):
    y = (x + (a/x))/2
    i += 1
    error = abs(y-x)
    x = y

print("La raiz es " + str(y))
```

Evidencia del funcionamiento

Figura 2.6: Raíz cuadrada a través del método iterativo

```
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$ python
Introduce un numero: 16
Introduce una aproximacion: 3
La raiz es 4.0000
ezequiel@ezequiel-HP-Pavilion-15-Notebook-PC:~/Ingenieria Fisica/4 semestre/Metodos Numericos/Tarea 1$
```

CAPÍTULO 3

CONCLUSIONES

En esta tarea, se revisaron los conceptos fundamentales que sirven como introducción para los métodos matemáticos. Esto resulta importante porque se estudiaron los errores asociados en las aproximaciones de los métodos, asunto que resultará relevante en lo que resta del curso, pues, así podemos tener una medida cuantitativa de cuán bueno esta siendo nuestra aproximación.

Incluso, esto cobra relevancia en la programación de los métodos que se estudiarán en el curso, porque la cuantificación de los errores con respecto al valor real sirve como parámetro para parar las iteraciones, de lo contrario habría un problema para saber cómo interrumpir bloque.

CAPÍTULO 4

BIBLIOGRAFÍA

1. Chapra, S. C., Canale, R. P. (2011). *Numerical methods for engineers* (Vol. 1221). New York: Mcgraw-hill.