Practica 7: Ejercicios de diferenciación numérica

Matematicas por Computadora
Alumno: Itzel Estrada Arcos (368980)
Docente: Ing. Jesus Padron

Sabado 28 de marzo 2025

0.1 Introduction.

La diferenciación numérica es una técnica importante en el análisis numérico que permite estimar la derivada de una función cuando no existe una expresión analítica, sino un valor discreto. En el Capítulo 4, Sección 4.1 del libro Análisis Numérico de J. El método utiliza interpolación polinomial para producir una estimación precisa de la derivada, reduciendo el error asociado.
Además, la fórmula de diferencias de orden superior se presenta como un método de cinco puntos, lo que mejora la precisión en comparación con el método básico. La sección también analiza el comportamiento de error, que también depende del tamaño del paso.
`h` es el número de puntos utilizados para la aproximación. Este enfoque es particularmente útil en situaciones donde las derivadas exactas son difíciles o imposibles de calcular y se usa ampliamente en ingeniería, física e informática. El propósito de esta sección es proporcionar una comprensión clara de los diferentes enfoques de la diferenciación numérica para evaluar la precisión y efectividad de estos métodos en diferentes escenarios.

0.2 Desarrollo de la practica.

0.2.1 Proceso paso a paso.

Como nota principal, utilice cuatro codigos diferentes para resolver los cuatro ejercicios en programacion, ya que si los juntaba el codigo no me ejecutaba correctamente, ademas, por suerte no se especifico como debia de ser el codigo en parte.

Primer codigo.
En este codigo realice el ejercicio no. 17

Para definir los valores de la tabla.
Cree un diccionario llamado `f_x` para almacenar los valores conocidos de la función `f(x)` en puntos específicos. Aquí, las claves son los valores de `x` y los valores son los resultados de la función evaluada en esos puntos.
- f(0.2)=0.9798652
- f(0.4)=0.9177710
- f(0.6)=0.8080348
- f(0.8)=0.6386093
- f(1.0)=0.3843735

Para definir el valor de h.
Asigne a la variable h el valor `0.2`, que representa la distancia o intervalo entre los valores consecutivos de `x`.

$$h = 0.2 \tag{1}$$

Esto me indica que la tabla está espaciada uniformemente, lo que es necesario para aplicar correctamente la fórmula de cinco puntos.

Para aplicar la fórmula de cinco puntos para `f(0.4)`.
Utilice la fórmula de cinco puntos hacia adelante para calcular la derivada en `x)0.4`:

$$f(x) = \frac{-25f(x-h) + 48f(x) - 36f(x+h) + 16f(x+2h) - 3f(x+3h)}{12h} \tag{2}$$

Para calcular el resultado para `f´(0.4)`.
Se realizan las multiplicaciones y se suman los resultados. La suma se divide entre `12×0.2 = 2.4`, lo que da el valor aproximado de `f´(0.4)`. Por ultimo, el resultado se almacena en la variable `f_prime_04`.

Para aplicar la fórmula de cinco puntos para `f´(0.8)`.
Realice las operaciones correspondientes. El resultado lo divido entre `2.4`. y el valor calculado se almacena en la variable `f_prime_08`.

Por ulrimo, para mostrar los resultados.
Finalmente, los valores de `f´(0.4)` y `f´(0.8)` se muestran en pantalla utilizando la función print e imprimo el valor aproximado de la derivada en ambos puntos.

Segundo codigo.
En este codigo realice el ejercicio no. 19

Primero, para importar librería necesaria
Para el programa utilice la librería NumPy para manejar operaciones matemáticas y trabajar con arreglos de datos. Esta librería facilita el cálculo de diferencias y manipulaciones numéricas.

Para definir los datos de entrada.
Cree dos arreglos: uno para los valores de tiempo y otro para la distancia recorrida en esos momentos. Estos valores representan datos discretos de un objeto en movimiento, donde se conoce la posición en instantes específicos.

Para inicializar un arreglo para la velocidad.
Cree un arreglo para almacenar las velocidades calculadas. Este arreglo tiene la misma longitud que el de tiempo y distancia, pero está inicialmente lleno de ceros. La velocidad se calculará en cada punto de tiempo usando métodos de diferenciación numérica.

Para calcular velocidades con diferencias centradas.
Para los puntos intermedios, utilice el método de diferencias centradas. Esta

técnica calcula la velocidad promedio entre dos puntos alrededor del tiempo actual, aplicando la fórmula:

$$vi = \frac{d_{i+1} - d_{i-1}}{t_{i+1} - t_{i-1}} \tag{3}$$

Esto nos ofrece una mejor aproximación porque considera cambios antes y después del punto.

Para calcular velocidades en los extremos.
Para los extremos (el primer y el último punto), no se pueden aplicar diferencias centradas, por lo que se utilizan:
Diferencias hacia adelante para el primer punto:

$$vo = \frac{d_1 - d_0}{t_1 - t_0} \tag{4}$$

Diferencias hacia atrás para el último punto:

$$v_{final} = \frac{d_{final} - d_{final-1}}{t_{final} - t_{final-1}} \tag{5}$$

Estos métodos solo consideran un valor posterior o anterior, por lo que son menos precisos que las diferencias centradas.

Y por ultimo, para mostrar los resultados.
Se muestra un resumen con los tiempos y sus velocidades aproximadas. Y utilizo un bucle para recorrer y mostrar cada tiempo junto con su velocidad correspondiente.

Tercer codigo.
En este codigo realice el ejercicio no. 20

Para importar la librería necesaria.
Utilizo la librería NumPy para manejar operaciones numéricas y trabajar con arreglos. Esto facilita la manipulación de datos y los cálculos de diferencias.

Para definir los datos de entrada.
Decidi crear dos arreglos:

Uno para los valores de tiempo `(t)` medidos en segundos.
Otro para los valores de corriente `(i)` medidos en amperios.
Estos datos representan mediciones de corriente a distintos instantes de tiempo en un circuito eléctrico.

Para definir parámetros del circuito.
Asigné valores constantes a:
L: Inductancia del circuito en henrios `(H)`.

R: Resistencia del circuito en ohmios.
Estos valores se utilizan para calcular el voltaje usando la ley de Kirchhoff para circuitos RL.

Para inicializar un arreglo para la derivada.
Cree un arreglo para almacenar los valores aproximados de la derivada:

$$\frac{di}{dt} \tag{6}$$

Este arreglo tiene la misma longitud que el de tiempos y corrientes, pero inicialmente está lleno de ceros.

Para calcular la derivada con diferencias centradas.
Para los puntos intermedios (del segundo al penúltimo), utilice el método de diferencias centradas para aproximar a la `derivada no. 4`:

$$\frac{di}{dt} = \frac{i_{i+1} - i_{i-1}}{t_{i+1} - t_{i-1}} \tag{7}$$

Esta técnica nos ofrece una mejor aproximación porque considera valores antes y después del punto actual.

Para calcular la derivada en los extremos.
Para los extremos (primer y último punto), no se pueden usar diferencias centradas, por lo que se emplean:
Diferencias hacia adelante para el primer punto:

$$\frac{di}{dt} = \frac{i_1 - i_0}{t_1 - t_0} \tag{8}$$

Diferencias hacia atrás para el último punto:

$$\frac{di}{dt} = \frac{i_{final} - i_{final-1}}{t_{final} - t_{final-1}} \tag{9}$$

Estas fórmulas solo utilizan valores cercanos, por lo que son menos precisas que las diferencias centradas.

Para calcular el voltaje usando la ley de Kirchhoff.
Una vez obtenidas las derivadas `no.4` , se aplica la fórmula para calcular el voltaje `e(t)`:

$$e(t) = L\frac{di}{dt} + Ri(t) \tag{10}$$

Esta fórmula proviene de la ecuación del circuito RL, donde el voltaje es la suma del voltaje inducido por la inductancia y la caída de voltaje en la resistencia.

Por ultimo, para mostrar los resultados.

Finalmente, se imprimen los valores aproximados del voltaje e(t) `formula no.8` para cada instante de tiempo y se recorre el arreglo para mostrar el tiempo y su correspondiente voltaje calculado.

Cuarto codigo
En este codigo realice el ejercicio no. 21

Para importar la librería necesaria.
Utilice NumPy para:
Definir funciones matemáticas (como el seno).
Realizar cálculos numéricos y manipular valores de manera eficiente.

Para definir la función a derivar.
Definí una función `f(x)`, que en este caso es el seno `(sin(x))`, pero puede ser cambiada por cualquier otra función si es necesario.
Esta función será evaluada en diferentes puntos para calcular su derivada.

Para definir el punto donde se evaluará la derivada.
Elegí el punto:

$$x_0 = \frac{pi}{4}(45^o \tag{11}$$

Donde se calculará la derivada.
Este valor está expresado en radianes, y como:

$$sin(\frac{pi}{4}) = \frac{\sqrt{2}}{2} \tag{12}$$

Su derivada en ese punto es conocida y se puede comparar con el resultado aproximado.

Para hacer un bucle para calcular aproximaciones.
Se inicia un bucle que recorre valores de `n` desde 1 hasta 20.
Para cada valor de `n`, se calcula un valor `h`, definido como:

$$h = 10^{-n} \tag{13}$$

Esto genera valores cada vez más pequeños de:

$$h = x_0 + h \tag{14}$$

Para mejorar la precisión de la aproximación.

Para aproximarnos a la derivada con diferencias hacia adelante.
Se utiliza la fórmula de diferencias hacia adelante para aproximar la derivada:

$$f(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} \tag{15}$$

Esta fórmula evalúa la pendiente de la recta secante entre `X0 y X0+h`, lo que da una aproximación de la derivada.

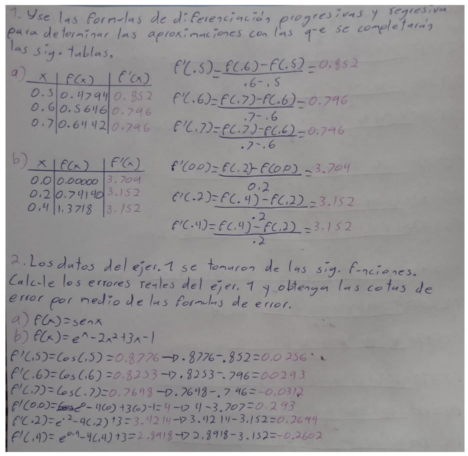
Para mostrar los resultados.

Para considerar sobre errores numéricos.
A medida que `h` se vuelve muy pequeño `formula no. 11`, el valor de la aproximación puede verse afectado por:
Errores de redondeo: Cuando `h` es extremadamente pequeño, los cálculos pierden precisión debido a las limitaciones de la aritmética de punto flotante.
Errores de truncamiento: Aunque disminuir `h` reduce el error de truncamiento, llega un punto donde los errores de redondeo dominan el resultado.

0.3 Conclusion.

La práctica me permitió explorar diferentes métodos de diferenciación numérica para estimar derivadas de funciones cuando solo se disponen de valores discretos. Logre implementar varios códigos para resolver distintos ejercicios, aplicando diferencias centradas para puntos intermedios y diferencias hacia adelante y atrás para los extremos, logrando aproximaciones precisas. Además, calcule derivadas en problemas relacionados con movimiento, circuitos eléctricos y funciones matemáticas, utilizando NumPy para optimizar los cálculos. A medida que se redujo el valor de `h`, observe que la precisión mejora, aunque también surgen errores de redondeo. Esta práctica me brindó una comprensión sólida de los enfoques utilizados en la diferenciación numérica y su aplicación en diversos escenarios.

0.4 Ejercicios hechos a mano



1. Use las formulas de diferenciación progresivas y regresiva para determinar las aproximaciones con las que se completarán las sig. tablas.

a)

x	$f(x)$	$f'(x)$
0.5	0.4794	0.852
0.6	0.5646	0.796
0.7	0.6442	0.796

$f'(.5)=\frac{f(.6)-f(.5)}{.6-.5}=0.852$

$f'(.6)=\frac{f(.7)-f(.6)}{.7-.6}=0.796$

$f'(.7)=\frac{f(.7)-f(.6)}{.7-.6}=0.796$

b)

x	$f(x)$	$f'(x)$
0.0	0.00000	3.704
0.2	0.74140	3.152
0.4	1.3718	3.152

$f'(0.0)=\frac{f(.2)-f(0.0)}{0.2}=3.704$

$f'(.2)=\frac{f(.4)-f(.2)}{.2}=3.152$

$f'(.4)=\frac{f(.4)-f(.2)}{.2}=3.152$

2. Los datos del ejer. 1 se tomaron de las sig. funciones. Calcule los errores reales del ejer. 1 y obtenga las cotas de error por medio de las formulas de error.

a) $f(x)=\operatorname{sen} x$

b) $f(x)=e^x-2x^2+3x-1$

$f'(.5)=\cos(.5)=0.8776 \rightarrow .8776-.852=0.0256$

$f'(.6)=\cos(.6)=0.8253 \rightarrow .8253-.796=0.0293$

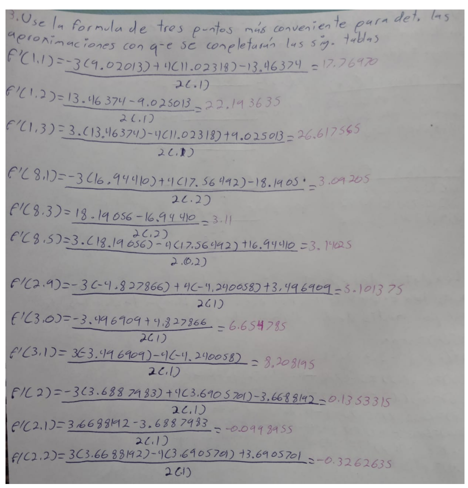
$f'(.7)=\cos(.7)=0.7648 \rightarrow .7648-.796=-0.0312$

$f'(0.0)=e^0-4(0)+3(0)-1=4 \rightarrow 4-3.707=0.293$

$f'(.2)=e^{.2}-4(.2)+3=3.4214 \rightarrow 3.4214-3.152=0.2694$

$f'(.4)=e^{0.4}-4(.4)+3=2.8918 \rightarrow 2.8918-3.152=-0.2602$

Figure 1: Ejercicios 1 y 2



3. Use la formula de tres puntos más conveniente para det. las aproximaciones con que se completarán las sig. tablas

$f'(1.1)=\frac{-3(9.02013)+4(11.02318)-13.46374}{2(.1)}=17.76970$

$f'(1.2)=\frac{13.46374-9.025013}{2(.1)}=22.193635$

$f'(1.3)=\frac{3.(13.46374)-4(11.02318)+9.025013}{2(.1)}=26.617565$

$f'(8.1)=\frac{-3(16.94410)+4(17.56492)-18.1905}{2(.2)}=3.09205$

$f'(8.3)=\frac{18.19056-16.94410}{2(.2)}=3.11$

$f'(8.5)=\frac{3.(18.19056)-4(17.56492)+16.94410}{2.(0.2)}=3.14025$

$f'(2.9)=\frac{-3(-4.827866)+4(-4.240058)+3.496909}{2(1)}=5.101375$

$f'(3.0)=\frac{-3.496909+4.827866}{2(1)}=6.654785$

$f'(3.1)=\frac{3(-3.496909)-4(-4.240058)}{2(.1)}=8.208195$

$f'(2)=\frac{-3(3.6887983)+4(3.6905701)-3.6688192}{2(.1)}=0.1353315$

$f'(2.1)=\frac{3.6688192-3.6887983}{2(.1)}=-0.0998955$

$f'(2.2)=\frac{3(3.6688192)-4(3.6905701)+3.6905701}{2(1)}=-0.3262635$

Figure 2: Ejercicio 3

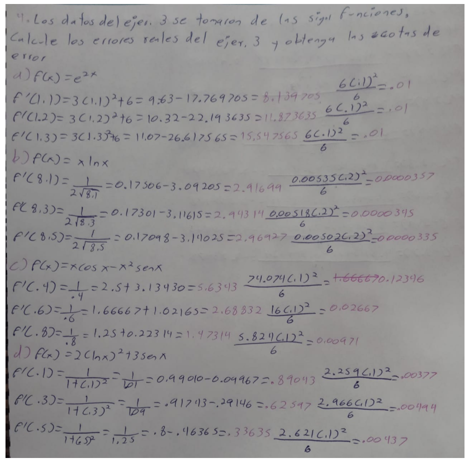


Figure 3: Ejercicio 4

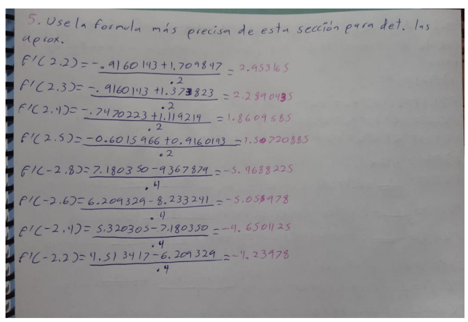


Figure 4: Ejercicio 5

0.5 Resultados

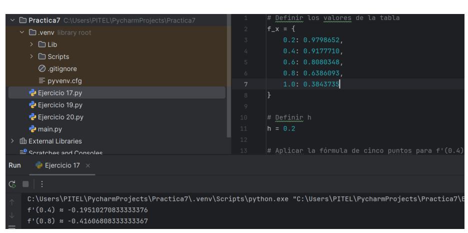


Figure 5: Resultado del ejercicio 17

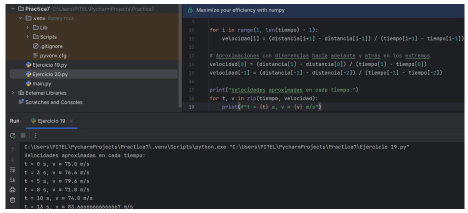


Figure 6: Resultado del ejercicio 19

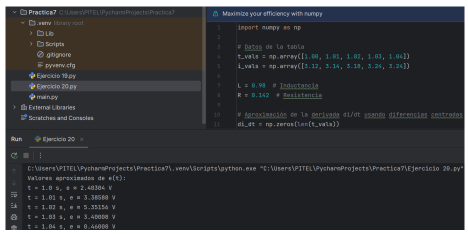


Figure 7: Resultado del ejercicio 20

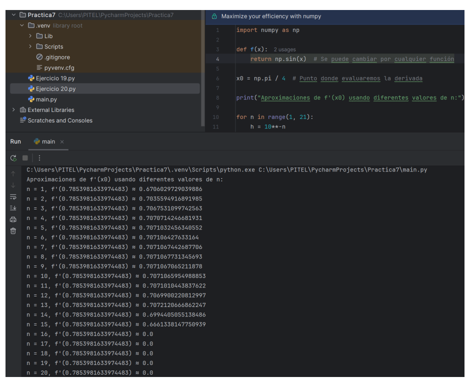


Figure 8: Resultado del ejercicio 21

0.6 Anexos

```python
# Definir los valores de la tabla
f_x = {
    0.2: 0.9798652,
    0.4: 0.9177710,
    0.6: 0.8080348,
    0.8: 0.6386093,
    1.0: 0.3843735
}

# Definir h
h = 0.2

# Aplicar la fórmula de cinco puntos para f'(0.4)
f_prime_04 = (-25*f_x[0.2] + 48*f_x[0.4] - 36*f_x[0.6] + 16*f_x[0.8] - 3*f_x[1.0]) / (12*h)

# Aplicar la fórmula de cinco puntos para f'(0.8)
f_prime_08 = (-25*f_x[0.4] + 48*f_x[0.6] - 36*f_x[0.8] + 16*f_x[1.0]) / (12*h)

# Mostrar resultados
print(f"f'(0.4) ≈ {f_prime_04}")
print(f"f'(0.8) ≈ {f_prime_08}")
```

Figure 9: Primer Codigo

```python
import numpy as np

# Datos de la tabla
tiempo = np.array([0, 3, 5, 8, 10, 13])
distancia = np.array([0, 225, 383, 623, 742, 993])

# Aproximación de la velocidad con diferencias centradas
velocidad = np.zeros(len(tiempo))

for i in range(1, len(tiempo) - 1):
    velocidad[i] = (distancia[i+1] - distancia[i-1]) / (tiempo[i+1] - tiempo[i-1])

# Aproximaciones con diferencias hacia adelante y atrás en los extremos
velocidad[0] = (distancia[1] - distancia[0]) / (tiempo[1] - tiempo[0])
velocidad[-1] = (distancia[-1] - distancia[-2]) / (tiempo[-1] - tiempo[-2])

print("Velocidades aproximadas en cada tiempo:")
for t, v in zip(tiempo, velocidad):
    print(f"t = {t} s, v ≈ {v} m/s")
```

Figure 10: Segundo Codigo.

```python
import numpy as np

# Datos de la tabla
t_vals = np.array([1.00, 1.01, 1.02, 1.03, 1.04])
i_vals = np.array([3.12, 3.14, 3.18, 3.24, 3.24])

L = 0.98  # Inductancia
R = 0.142  # Resistencia

# Aproximación de la derivada di/dt usando diferencias centradas
di_dt = np.zeros(len(t_vals))

for i in range(1, len(t_vals) - 1):
    di_dt[i] = (i_vals[i+1] - i_vals[i-1]) / (t_vals[i+1] - t_vals[i-1])

# Aproximaciones con diferencias hacia adelante y atrás en los extremos
di_dt[0] = (i_vals[1] - i_vals[0]) / (t_vals[1] - t_vals[0])
di_dt[-1] = (i_vals[-1] - i_vals[-2]) / (t_vals[-1] - t_vals[-2])

# Cálculo del voltaje e(t) = L * di/dt + R * i
e_vals = L * di_dt + R * i_vals

print("Valores aproximados de e(t):")
for t, e in zip(t_vals, e_vals):
    print(f"t = {t} s, e ≈ {e} V")
```

Figure 11: Tercer Codigo.

```python
import numpy as np

def f(x):  2 usages
    return np.sin(x)  # Se puede cambiar por cualquier función

x0 = np.pi / 4  # Punto donde evaluaremos la derivada

print("Aproximaciones de f'(x0) usando diferentes valores de n:")

for n in range(1, 21):
    h = 10**-n
    derivada = (f(x0 + h) - f(x0)) / h
    print(f"n = {n}, f'({x0}) ≈ {derivada}")
```

Figure 12: Cuarto Codigo.