UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA CIVIL

METODOS NUMÉRICOS

PERIODO ACADÉMICO: 2024-2S

DOCENTE: GUILLERMO MACHADO SOTOMAYOR

UNIDAD 1

INTRODUCCIÓN

Conceptos básicos

Análisis de Errores y Convergencia en Técnicas Numéricas: Aplicaciones en Ciencias e Ingeniería

El análisis de errores y la convergencia son aspectos fundamentales en el ***Análisis Numérico***, especialmente cuando se aplican técnicas numéricas para resolver problemas reales en ciencias e ingeniería. Estos conceptos permiten evaluar la precisión de las soluciones numéricas y determinar la eficacia y confiabilidad de los métodos empleados.

1. Análisis de Errores

El análisis de errores se ocupa de estudiar las discrepancias entre la **solución exacta** de un problema matemático y su **aproximación numérica**. Estas discrepancias pueden surgir por diversas razones:

- **Errores de Redondeo**: Debido a la representación finita de los números en las computadoras.
- **Errores de Truncamiento**: Surgen al utilizar una aproximación en lugar de un proceso matemático exacto (por ejemplo, al truncar una serie infinita o al utilizar un método iterativo).
- **Errores de Discretización**: Aparecen cuando se aproximan soluciones continuas mediante discretizaciones finitas (por ejemplo, en métodos de diferencias finitas o elementos

finitos).

Tipos de errores

- **Error Absoluto:** Es la diferencia entre la solución exacta y la solución aproximada:

$$E_a = |x_{exacto} - x_{aproximado}|$$

- **Error Relativo**: Es el error absoluto normalizado con respecto al valor exacto:

$$E_r = \left| \frac{x_{exacto} - x_{aproximado}}{x_{exacto}} \right|$$

- **Error de Redondeo**: Ocurre cuando un número se aproxima en una computadora debido a la limitación de precisión.

$$e_r = Valor_{redondeado} - Valor_{exacto}$$

- **Error de truncamiento:** Ocurre cuando se aproxima un proceso matemático continuo mediante una fórmula discreta o una serie finita.

Ejemplos Prácticos

Recuerde que **el análisis de errores** es fundamental en el Cálculo Numérico, ya que permite evaluar la precisión y exactitud de las soluciones obtenidas mediante técnicas numéricas. Los errores en las soluciones numéricas pueden clasificarse en varios tipos, siendo los más comunes los errores de redondeo y los errores de truncamiento. A continuación, se presentan ejemplos prácticos que ilustran su impacto en problemas reales de ciencias e ingeniería.

Ejemplo de Redondeo:

Se considera la suma de una serie infinita, como

$$S = 0.1 + 0.1 + 0.1 + \ldots$$

que debería ser idual a $S = 0.1 \cdot n$, donde n es el número de términos. En una máquina con precisión limitada, 0.1 no se puede representar exactamente, lo que lleva a errores acumulativos a medida que se realizan muchas operaciones.

Impacto en Problemas Reales:

En cálculos intensivos, especialmente aquellos que involucran operaciones repetitivas como multiplicaciones o sumas en bucles, los errores de redondeo pueden acumularse y conducir a resultados significativamente incorrectos.

Ejemplo Práctico de error de redondeo en Python:

```
import numpy as np

# Suma de 1000 términos de 0.1
exact_sum = 0.1 * 1000
approx_sum = sum([0.1] * 1000)
```

```
print(f"Suma exacta: {exact_sum}")
print(f"Suma aproximada: {approx_sum}")
print(f"Error de redondeo: {exact_sum - approx_sum}")
```

```
Suma exacta: 100.0
Suma aproximada: 99.9999999999986
Error de redondeo: 1.4068746168049984e-12
```

En este código, la diferencia entre la suma exacta y la suma aproximada resulta del error de redondeo al representar 0.1 en el sistema numérico de la computadora.

Resultados y Discusión:

El error de redondeo observado es pequeño, pero en aplicaciones donde se realizan millones de operaciones, este tipo de error puede acumularse y tener un impacto significativo en la precisión final de los resultados.

Ejemplo de Truncamiento:

Este tipo de error es común cuando se usan métodos numéricos que requieren cortar una serie infinita o usar una aproximación para una función continua.

Suponga que utiliza la serie de Taylor para aproximar la función exponencia $f(x) = e^x$:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Si se trunca la serie después de un número finito de términos, se obtiene un error de truncamiento que depende del número de términos utilizados.

Impacto en Problemas Reales:

Los errores de truncamiento son comunes en la resolución numérica de ecuaciones diferenciales, integración numérica, y otros métodos donde se utiliza una serie finita para aproximar una solución.

Ejemplo Práctico en Python:

```
import math

# Aproximación de e^x usando la serie de Taylor truncada
def approx_exp(x, n_terms):
    result = 0
    for n in range(n_terms):
        result += x**n / math.factorial(n)
    return result

x = 1
exact_exp = math.exp(x)
approx_exp_5 = approx_exp(x, 5)
approx_exp_10 = approx_exp(x, 10)
```

```
print(f"Valor exacto de e^{x}: {exact_exp}")
print(f"Valor aproximado con 5 términos: {approx_exp_5}")
print(f"Valor aproximado con 10 términos: {approx_exp_10}")
print(f"Error de truncamiento con 5 términos: {exact_exp - approx_exp_5}")
print(f"Error de truncamiento con 10 términos: {exact_exp - approx_exp_10}")
```

```
Valor exacto de e^1: 2.718281828459045
Valor aproximado con 5 términos: 2.708333333333333
Valor aproximado con 10 términos: 2.7182815255731922
Error de truncamiento con 5 términos: 0.009948495125712054
Error de truncamiento con 10 términos: 3.0288585284310443e-07
```

En este código, se compara la aproximación de $f(x) = e^x$ utilizando 5 y 10 términos de la serie de Taylor. Se observa cómo el error de truncamiento disminuye al aumentar el número de términos.

Resultados y Discusión:

El error de truncamiento disminuye a medida que se incluyen más términos en la serie, pero en la práctica, siempre hay un compromiso entre la precisión y el costo computacional. En problemas reales, se debe encontrar un equilibrio adecuado para minimizar el error sin hacer el cálculo computacionalmente costoso.

Ejemplo Aplicado en Ingeniería: Solución Numérica de Ecuaciones Diferenciales

Dada la ecuación diferencial ordinaria simple:

$$\frac{dy}{dx} = -2y, \ y(0) = 1$$

cuya solución exacta es conocida: $y(x) = e^{-2x}$

Al resolverla utilizando el Método de Euler (se estudiará más a detalle en la Unidad 3), un método numérico simple que introduce tanto errores de redondeo como de truncamiento.

Código en Python:

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

# Parámetros de la ecuación
def dydx(y, x):
    return -2 * y

# Método de Euler para resolver la ODE
def euler_method(y0, x0, h, n):
    y = y0
    x = x0
    ys = [y0]
    xs = [x0]
    for i in range(n):
```

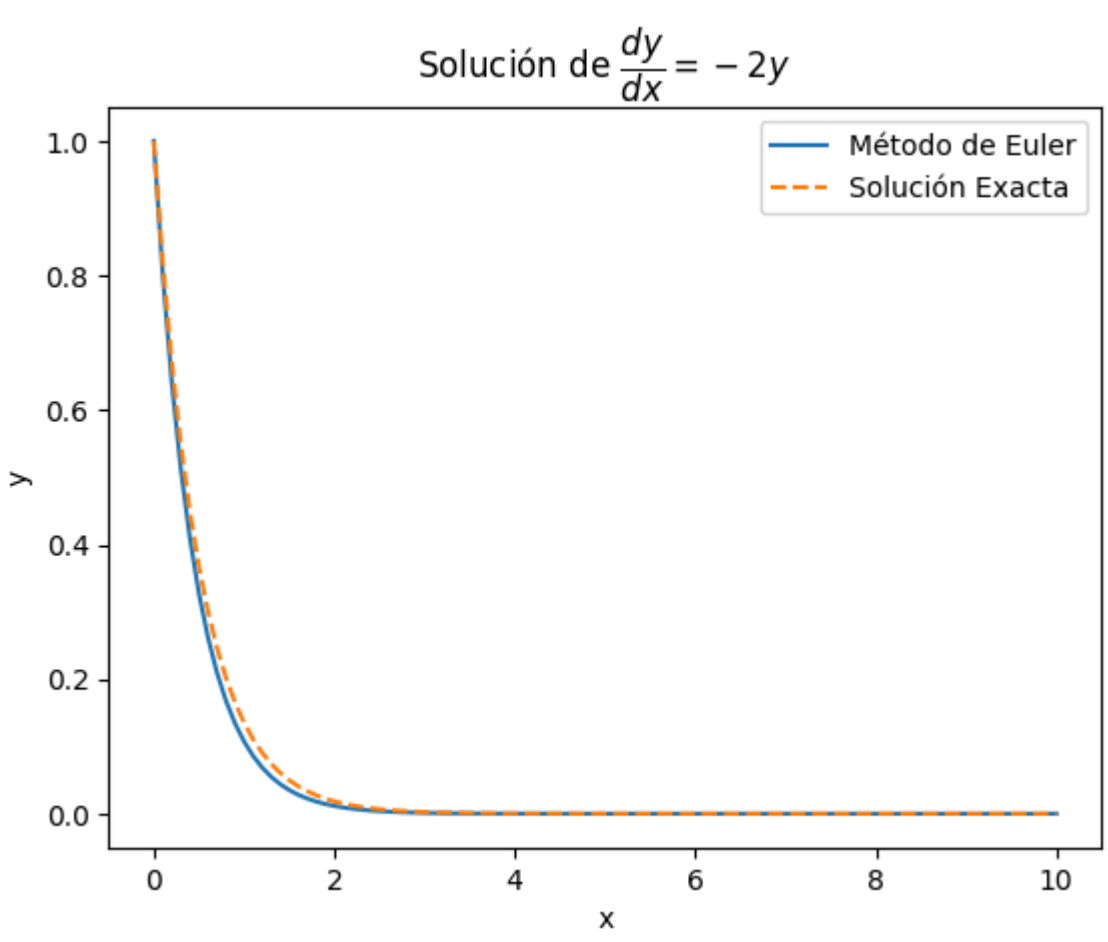
```
            y += h * dydx(y, x)
            x += h
            ys.append(y)
            xs.append(x)
        return xs, ys

    # Condiciones iniciales
    y0 = 1
    x0 = 0
    h = 0.1  # Tamaño de paso
    n = 100  # Número de pasos

    # Solución numérica con el Método de Euler
    xs, ys = euler_method(y0, x0, h, n)

    # Solución exacta
    exact_ys = np.exp(-2 * np.array(xs))

    # Gráfica de los resultados
    plt.plot(xs, ys, label='Método de Euler')
    plt.plot(xs, exact_ys, label='Solución Exacta', linestyle='dashed')
    plt.legend()
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Solución de $\dfrac{dy}{dx} = -2y$')
    plt.show()
```



Resultados y Análisis:

- **Error de Truncamiento**: El método de Euler, al ser un método de primer orden, introduce un error de truncamiento que es proporcional al tamaño del paso h. Este error se observa como una desviación de la solución numérica respecto a la solución exacta.
- **Error de Redondeo**: Aunque menos significativo en este ejemplo, los errores de redondeo también están presentes, especialmente si se reduce el tamaño del paso h y se realizan muchas iteraciones.

Conclusiones

El análisis de los errores de redondeo y de truncamiento es crucial para garantizar la precisión y confiabilidad de las soluciones numéricas en problemas de ciencias e ingeniería. Los errores de redondeo pueden acumularse en cálculos repetitivos, mientras que los errores de truncamiento son el resultado de aproximaciones finitas en procesos continuos.

Estrategias para mitigar los errores:

- **Reducir el Tamaño del Paso:** Disminuir h en métodos como el de Euler reduce el error de truncamiento, aunque esto puede aumentar el error de redondeo si no se maneja adecuadamente.
- **Aumentar la Precisión Numérica**: Utilizar tipos de datos de mayor precisión (como ***float64*** en lugar de ***float32***) puede reducir los errores de redondeo.
- **Métodos de Mayor Orden:** Implementar métodos numéricos de mayor orden, como el método de Runge-Kutta de cuarto orden, puede reducir significativamente el error de truncamiento.

---

2. Convergencia en Métodos Numéricos

La convergencia se refiere al comportamiento de una secuencia de aproximaciones sucesivas al resolver un problema, ya sea iterativamente o mediante refinamiento de mallas (en métodos de discretización). Un método numérico converge si, al incrementar las iteraciones o refinar la discretización, las soluciones aproximadas se acercan a la solución exacta.

Criterios de Convergencia

1. **Convergencia de Orden**: Si un método tiene un orden de convergencia p, esto implica que el error disminuye a una tasa proporcional a h^p, donde h es el tamaño de paso o la malla.

$$E(h) \approx Ch^p$$

2. **Convergencia Lineal**: Un método iterativo tiene convergencia lineal si el error en la n-ésima iteración es proporcional al error en la $n-1$-ésima iteración multiplicado por una constante $0 < \lambda < 1$:

$$E_{n+1} \approx \lambda E_n$$

3. **Convergencia Superlineal**: Ocurre cuando el error decrece más rápido que en el caso lineal, pero no necesariamente cuadráticamente.

4. **Convergencia Cuadrática**: En este caso, el error se reduce cuadráticamente con respecto a las iteraciones:

$$E_{n+1} \approx \lambda (E_n)^2$$

Ejemplos prácticos en Ciencias e Ingeniería

Ejemplo 1: Método de Newton-Raphson

El método de Newton-Raphson es un ejemplo clásico de un método numérico iterativo utilizado para encontrar raíces de ecuaciones no lineales. Su análisis de convergencia es fundamental para determinar su aplicabilidad.

- **Fórmula del método**:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- **Análisis de Errores y Convergencia:**
  - **Error inicial**: Si el error inicial es pequeño, el método generalmente converge cuadráticamente. Sin embargo, si el punto inicial está lejos de la raíz, el método puede divergir o tener una convergencia muy lenta.
  - **Condiciones de convergencia**: El método requiere que la función sea diferenciable y que la derivada no sea cero en la vecindad de la raíz.

Ejemplo práctico:

Supóngase que se quiere encontrar la raíz real positiva de $f(x) = x^2 - 2 = 0$, asumiendo como valor inicial $x_0 = 1.5$ con tolerancia de 1×10^6.

Aquí está la implementación en Python:

```
def newton_raphson(f, df, x0, tol=1e-6, max_iter=100):
    for i in range(max_iter):
        x1 = x0 - f(x0)/df(x0)
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
    return x0

f = lambda x: x**2 - 2
df = lambda x: 2*x
```

```
raiz = newton_raphson(f, df, 1.5)
print(f"Raíz aproximada: {raiz}")
```

```
Raíz aproximada: 1.4142135623730951
```

Ejemplo 2: Método de Diferencias Finitas

El método de diferencias finitas se utiliza ampliamente para resolver ecuaciones diferenciales parciales (EDP). Este método convierte EDPs en un sistema de ecuaciones algebraicas discretas.

- Discretización:

  Para la ecuación de Poisson $\nabla^2 u = f(x, y)$ en una malla uniforme, el método de diferencias finitas se aproxima como:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = f_{i,j}$$

- **Análisis de errores:**
  - Error de Discretización: Depende de la finura de la malla h. Si h es suficientemente pequeño, el error de truncamiento es del orden $O(h^2)$ en el caso de diferencias centradas.
- **Convergencia**:

Si la malla se refina (h disminuye), la solución numérica debe converger hacia la solución exacta de la ecuación diferencial. Este proceso de refinamiento también permite estimar la tasa de convergencia del método.

Ejemplo práctico:

Resolviendo la ecuación de Poisson en un dominio cuadrado con condiciones de frontera dadas.

Código en MATLAB (veáse archivo en: texto del vínculo)

Ejemplo_Práctico_Diferencia_Finitas.mlx

Referencias bibliográficas

- Burden, R. L., & Faires, J. D. (2010). *Numerical Analysis*. Brooks/Cole.
- Chapra, S. C., & Canale, R. P. (2015). *Numerical Methods for Engineers*. McGraw-Hill Education.
- Atkinson, K. E. (1989). *An Introduction to Numerical Analysis*. Wiley.