

2018-A : Análisis y métodos numéricos.

Chapter One

Preliminaries Mathematical

Teorema de Taylor con restos de Lagrange:

$f \in C^n[a, b]$ y $f^{(n+1)}$ existe en (a, b) entonces para cualquier punto x y c en $[a, b]$:

$$f(x) = \sum_{k=0}^n \frac{1}{(k+1)!} f^{(k)}(c)(x-c)^k + E_n(x)$$

Donde:

$$E_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-c)^{n+1}$$

Serie de Maclaurin

$$f(x) = \sum_{k=0}^n \frac{1}{(k+1)!} f^{(k)}(0)(x-0)^k + E(x)$$

Donde:

$$E(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-0)^{n+1}$$

Redondeo:

Es la acción de aproximar la penúltima cifra decimal respecto a la información de la última cifra, o hasta la cifra a la que se quiere llegar:

Sí la última cifra o la cifra siguiente a la que se quiere tomar en cuenta es:

0, 1, 2, 3, 4, no hay cambios

5, 6, 7, 8, 9 se aproxima a el siguiente dígito

Se cumple que:

$$|x - \bar{x}| \leq \frac{1}{2} 10^{-n}$$

Donde n es el dígito que espero aproximar.

x es el número originar.

\bar{x} es el número aproximado.

Convergencia o límite de una sucesión:

Sea $\{s_n\}$, la convergencia es el valor al que tienden los valores de una sucesión, cuando ésta toma valores muy grandes.

$$\lim_{n \rightarrow \infty} s_n = L$$

Big O y Little o

Big-O es la cota superior de una función y se denota $f(x) = O(c * g(x))$ si:

$$\lim_{n \rightarrow \infty} \frac{f(x)}{g(x)} = L$$

Little-o es la cota superior de una función y se denota $f(x) = o(c * g(x))$ si:

$$\lim_{n \rightarrow \infty} \frac{f(x)}{g(x)} = 0$$

Ordenes de convergencia:

Nos permiten indicar la velocidad de convergencia de una secuencia.

Sea $\{s_n\}$ una sucesión y q un orden de convergencia, entonces decimos que la sucesión converge con orden q si:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x|}{|x_n - x|^q} = \mu, \mu > 0, \mu < \infty$$

Casos con $q = 1$:

- Sí $q = 1$ y $\mu < 1$, converge de manera **lineal**.
- Sí $q = 1$ y $\mu \rightarrow 0$, converge de manera **superlineal**.

Chapter Two
Computer Arithmetic

Fuentes básicas de error [7][m²riros]

1. **Error del modelo:** Condiciones aproximadas.
2. **Error del método:** Problema preciso no puede ser resuelto de forma exacta.
3. **Error residual:** Originados por una serie infinita al tomar solo una parte finita.
4. **Error inicial:** Parámetros aproximados.

5. **Error de redondeo:** Originados por la representación finita de los números.
6. **Error de operación:**
 - a. Operación **exacta** de **números aproximados**.
 - b. Operación **aproximada** de **números exactos**.
7. **Error sistemático:**
 - a. Defecto del instrumento de medición.
 - b. Condiciones del ambiente.
 - c. Metodología de la medición.
 - d. Precisión limitada del instrumento.

Error absoluto, Error relativo y pérdida de precisión

Sea x un número real y \bar{x} una aproximación a ese número real:

Error absoluto:

$$E_a = |x - \bar{x}|$$

Error relativo:

$$E_r = \frac{|x - \bar{x}|}{|x|}, \quad |x| \neq 0$$

Error relativo porcentual:

$$E_r = \frac{|x - \bar{x}|}{|x|} * 100, \quad |x| \neq 0$$

Precisión de la aproximación:

$$|x - \bar{x}| \leq \varepsilon, \quad \varepsilon \text{ es el umbral}$$

Teorema de pérdida de precisión:

Sí x y y son número positivos binarios normalizados, tal que $x > y$.

$$2^{-q} \leq 1 - \frac{y}{x} \leq 2^{-p}$$

Donde q es la mayor y p la menor pérdida de bits significativos en la resta de $x - y$.

Inestabilidad del problema o acondicionamiento

Pequeños cambios en los datos de entrada, producen grandes cambios en la salida.

Inestabilidad numérica

Se dice que hay inestabilidad numérica, cuando pequeños errores en una etapa del proceso, son magnificados en las etapas siguientes, haciendo que se degrade seriamente la exactitud del resultado.

Chapter Three

Solution of Non-Linear Equations

Teorema de Bolzano: Sea $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$, si la función toma valores opuestos en los extremos del intervalo, es decir $f(a)f(b) < 0$, entonces en el intervalo hay una raíz de la función, $f(x) = 0$, $x \in [a, b]$.

Teorema de Rolle: Sea $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$, una función continua y derivable, tal que $f(a) = f(b)$, entonces existe un punto intermedio $c \in [a, b]$, tal que $f'(c) = 0$.

Métodos para el cálculo de raíces

- **Métodos cerrados:** Se basan en que la **función cambia de signo en la vecindad de la raíz**, por lo que **requieren un intervalo $[a, b]$** para empezar a buscar.
- **Métodos abiertos:** Se basan en formular que solo **requieren un valor inicial x_0** .

1. Bisection [Método cerrado] [Versión del libro: 22] [Versión simplificada: 12] [Orden de convergencia: Linealmente, a razón de 1/2]

El método de la bisección está soportado por el teorema de Bolzano.

INPUT $a, b, M, \delta, \varepsilon$

$u \leftarrow f(a)$

$v \leftarrow f(b)$

$e \leftarrow b - a$

OUTPUT a, b, u, v

if $\text{sign}(u) == \text{sign}(v)$ **then**

stop

for $k = 1, 2, \dots, M$ **do**

$e \leftarrow e/2$

```

c ← a + e
w ← f(c)
OUTPUT k, c, w, e
if |e| < δ or |w| < ε then
    stop
if sign( w ) ≠ sign( u ) then
    b ← c
    v ← w
else
    a ← c
    u ← w
end if
end

```

Método simplificado

```

INPUT a, b, M, δ, ε
if f(a)f(b) < 0 then
    for k = 1,2,...,M do
        c = a + (b-a)/2
        if f(c) < ε or ε < δ then
            return c
        if f(c)f(a) < 0 then
            b=c
        else
            a=c
    end for
return c

```

2. Newton-Raphson [12] [Método abierto][**Orden de convergencia:** Al menos cuadrático]

Convergencia: Ya que el número de cifras con este método suele duplicarse en cada iteración, esto es conocido como convergencia cuadrática.

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

Algoritmo

```

INPUT  $x_0$ , M, ε, δ
v ← f(x0)
OUTPUT 0, x0, v

```

```

if  $|v| < \varepsilon$  then
    stop
for  $k = 1, 2, \dots, M$  do
     $x_1 \leftarrow x_0 - v/f'(x_0)$ 
     $v \leftarrow f(x_1)$ 
    OUTPUT  $k, x_1, v$ 
    if  $|x_1 - x_0| < \delta$  or  $|v| < \varepsilon$  then
        stop
     $x_0 \leftarrow x_1$ 

```

3. Secant [Método abierto][Orden de convergencia: Superlineal]

Convergencia: Superlineal, más lento que Newton Raphson pero más rápido que bisección.

Para este método, se necesitan conocer dos valores: x_0 y x_1
 Converge si x_0 y x_1 están cercanos a la raíz de f .

$$x_{n+1} = x_n - f(x_n) \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right], \quad n \geq 1$$

```

INPUT  $x_0, x_1, M, \varepsilon, \delta$ 
 $v \leftarrow f(x_1)$ 
OUTPUT  $0, x_1, v$ 
if  $|v| < \varepsilon$  then
    stop
for  $k = 1, 2, \dots, M$  do
     $dividendo \leftarrow f(x_1) * (x_1 - x_0)$ 
     $divisor \leftarrow f(x_1) - f(x_0)$ 
     $cociente \leftarrow dividendo/divisor$ 
     $s = x_1 - cociente$ 
     $v \leftarrow f(s)$ 
    OUTPUT  $k, s, v$ 
    if  $|s - x_1| < \delta$  or  $|v| < \varepsilon$  then
        stop
     $x_0 \leftarrow x_1$ 
     $x_1 \leftarrow s$ 

```

4. Fixed points and functional iteration

Sea $f(x)$, una función, para resolver $f(x) = 0$, se ordenan los términos de una manera equivalente:

$$f(x) = 0$$

$$x - F(x) = 0$$

$$x = F(x)$$

$$x_{n+1} = F(x_n)$$

Mapeo contractivo: Se dice que F es contractivo si existe un λ tal que $0 < \lambda < 1$

$$|F(x) - F(y)| \leq \lambda |x - y|$$

Solution of non-linear equations
Computing roots of polynomial

5. Algoritmo de horner [Polinomios]:

$$p(z) = (z - z_0)q(z) + p(z_0)$$

En el que tenemos que $p(z_0)$ es el residuo.

Algoritmo

```

INPUT  $n, (a_i : 0 \leq i \leq n), z_0$ 
for  $k = 0, 1, \dots, n-1$  do
    for  $j = n-1, n-2, \dots, k$  do
         $a_j \leftarrow a_j - z_0 a_{j+1}$ 
    end
end
OUTPUT  $(a_i : 0 \leq i < n)$ 

```

Matrices positivas definidas:

$$x^T Ax > 0$$

- Si una matriz es positiva definida, asumimos que es simétrica, o sea que $A^T = A$.
- Sus valores propios son mayores que cero, $\lambda_i > 0$ y reales.

Factorización LU [Se calcula sin pivoteo]:

- **A** debe ser **no-singular**.

$$Ax = b$$

$$A = LU$$

$$LUx = b$$

$$Lz = b$$

$$Ux = z$$

- **Factorización Doolittle:** Sí se fija que para cada elemento de la diagonal de L el valor de 1.
- **Factorización Crout:** Sí para cada elemento de la diagonal de U el valor es 1.
- **Factorización de Cholesky:** Sí **A** es simétrica definida positiva, entonces tiene una única factorización tal que $U = L^T$, donde **L** es la matriz triangular inferior y todos los elementos de su diagonal son positivos.

Factorización PA=LU [Se calcula con pivoteo]: Cuando se hace una factorización LU con pivoteo, los cambios en el orden de las filas deben almacenarse en la matriz de pivoteo.

$$PA = LU$$

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Matriz diagonalmente dominante: Son las matrices en cuya diagonal se encuentra un valor mayor o igual a la suma del valor absoluto del resto de valores de la fila.

Cuando una matriz es diagonalmente dominante, no es necesario hacer pivotaje para proceder con la eliminación Gaussiana.

Normas vectoriales:

- **Norma-0:** Número de elementos diferentes de 0 en el vector.
- **Norma-1:** Suma de el valor absoluto de todos los elementos.
- **Norma-2:** Raíz cuadrada de la suma del cuadrado de cada elemento del vector.
- **Norma-∞:** $\max(|x_i|)$

Normas matriciales:

- **Norma-1:** $\max(\text{Suma de cada columna})$
- **Norma-2:** $\max(\sigma_i)$, donde σ es un valor singular de la matriz.
- **Norma-∞:** $\max(\text{Suma de cada fila})$

Número de condición de una matriz: Este número nos permite conocer qué tan cerca está una matriz de la singularidad.

$$k = \|A\| \|A^{-1}\|$$

$$k = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Si la matriz está cerca a la singularidad, el valor de k se va a 0.

Refinamiento iterativo: Si $x^{(0)}$ es la solución aproximada a el sistema $Ax = b$, entonces la solución exacta está dada por:

Solución exacta:

$$x = x^{(0)} + e^{(0)}$$
$$x = x^{(0)} + A^{-1}(b - Ax^{(0)})$$

Vector de error:

$$e^{(0)} = A^{-1}(b - Ax^{(0)})$$

Residuo:

$$r^{(0)} = b - Ax^{(0)}$$

Métodos de solución de sistemas de ecuaciones:

- **Directos:** Son los que llegan a la solución exacta, salvo a errores de redondeo, tras un número finito de pasos.
- **Indirectos:** Son los que permiten llegar a unos resultados que idealmente convergen a la solución. La naturaleza de estos métodos es generalmente iterativa, y suelen detenerse cuando se alcanza una exactitud definida, ya que la solución requiere de infinitos pasos para ser alcanzada..

Solución numérica de sistemas de ecuaciones lineales usando métodos iterativos:

Para resolver el sistema $Ax = b$, se escribe el sistema de la forma:

$$Qx = (Q - A)x + b$$

Tenemos que:

$$Qx^{(k+1)} = (Q - A)x^k + b$$

La elección de **Q** implica dos condiciones:

1. la secuencia $\{x^k\}$ es fácil de calcular.
2. La secuencia $\{x^k\}$ converge rápidamente a la solución.

Suponiendo que **Q** y **A** son no-singulares:

$$\begin{aligned} x &= (I - Q^{-1}A)x + Q^{-1}b \\ x^k &= (I - Q^{-1}A)x^{k-1} + Q^{-1}b \end{aligned}$$

Teorema de convergencia:

Si $\|I - Q^{-1}A\| < 1$, entonces la secuencia $\{x^k\}$ converge a la solución $Ax = b$.

Método de Richardson [Usado principalmente en matrices dispersas]:

Con el método de richardson, definimos que la matriz **Q** será la matriz **I**.

$$\begin{aligned} x^{k+1} &= (I - A)x^k + b \\ x^{k+1} &= x^k + r^k \\ r^k &= b - Ax^k \end{aligned}$$

Convergencia:

El método converge si:

$$\|I - A\| < 1$$

Algoritmo

```
INPUT  $n, (a_{ij}), (b_i), (x_i), M$ 
for  $k=1,2,\dots,M$  do
  for  $i=1,2,\dots,M$  do
    
$$r_i \leftarrow b_i - \sum_{j=0}^n a_{ij}x_j$$

  end
  for  $i=1,2,\dots,M$  do
    
$$x_i = x_i + r_i$$

  end
end
```

Método de Jacobi [Matrices dispersas]:

Define la matriz **Q** como la **matriz diagonal**, donde los elementos de la diagonal, son los a_{ii} .

Condición: **A** tiene que ser **diagonalmente dominante**.

Convergencia: Si **A** es diagonalmente dominante, el método converge a la solución de $Ax = b$.

Algoritmo:

```
INPUT  $n, (a_{ij}), (b_i), (x_i), M$ 
for  $k = 1, 2, \dots, M$  do
  for  $i = 1, 2, \dots, n$  do
    
$$u_i \leftarrow \left( b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j \right) / a_{ii}$$

  end
  for  $i = 1, 2, \dots, n$  do
    
$$x_i \leftarrow u_i$$

  end
end
```

Sistemas equivalentes para Jacobi:

Sistema de la forma $Ax = b$:

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

Sistema equivalente [Recordar que se despeja por la diagonal]:

$$x_1 = b_1/a_{11} - a_{12}x_2/a_{11}$$

$$x_2 = b_2/a_{22} - a_{21}x_1/a_{22}$$

Análisis general:

Sea $G = I - Q^{-1}A$ y $c = Q^{-1}b$, el radio espectral de A es:

$$\rho(A) = \max\{|\lambda| : \det(A - \lambda) = 0\}$$

Teorema de convergencia:

El método de Jacobi, $x^{(k)} = Gx^{(k-1)} + c$ converge a $(I - G)^{-1}c$, si $\rho(G) < 1$, o sea $\rho(I - Q^{-1}A) < 1$.

Método de Gauss-Seidel [Converge más rápido] [Matrices dispersas]:

Define a **Q** como la matriz **triangular inferior de A**.

Restricciones:

- A debe ser definida positiva.

Convergencia: Converge si **A** es **diagonalmente dominante**, y si:

$$\|I - Q^{-1}A\| < 1$$

Algoritmo:

INPUT $n, (a_{ii}), (b_i), (x_i), M$

for $k = 1, 2, \dots, M$ **do**

for $i = 1, 2, \dots, n$ **do**

$$x_i \leftarrow \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j \right) / a_{ii}$$

end

end

Sistemas equivalente para Gauss-Seidel:

Sistema de la forma $Ax = b$:

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

Sistema equivalente:

$$x_1^k = b_1/a_{11} - a_{12}x_2^{k-1}/a_{11}$$

$$x_2^k = b_2/a_{22} - a_{21}x_1^k/a_{22}$$

Método del gradiente [Método tipo krylov]:

La solución del sistema $Ax = b$ cae dentro de un espacio de krylov, **cuyas dimensiones** son del **grado** del **polinomio mínimo** de **A**.

*Se llama **polinomio mínimo** de A, al polinomio de menor grado y mónico que anula a A.*

Condiciones:

1. A es real, simétrica y definida positiva, esto incluye no-singular.

El problema de solucionar $Ax = b$ es **EQUIVALENTE** a solucionar la forma cuadrática:

$$q(x) = \frac{1}{2} \langle x, Ax \rangle - \langle x, b \rangle$$

Entonces la motivación es tomar un problema lineal de la forma $Ax = b$, y convertirlo en una forma cuadrática con estructura convexa:

$$q(x) = \frac{1}{2}x^T Ax - bx + c$$

Con la condición de minimizar el vector x en la forma cuadrática.

El gradiente está dado por:

$$\nabla q(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b$$

$$\nabla q(x) = \frac{1}{2}(A^T + A)x - b$$

Minimizar x dada $q(x)$ implica encontrar los mínimos de la función, o sea: $\nabla q(x) = 0$, y verificar mínimos.

Método Steepest Descent [Método del gran descenso/pendiente descendente]:

Condiciones[Gradiente general]: A es real, simétrica y definida positiva.

Dado un $\nabla q(x)$, en la dirección en la que $q(x)$ incrementa rápidamente, buscamos una $-\nabla q(x)$.

$$\begin{aligned}\nabla q(x) &= Ax - b \\ -\nabla q(x_{i+1}) &= b - Ax_{i+1} = r_{i+1}\end{aligned}$$

En este caso α_i (llamado paso) es el gradiente ortogonal a la recta.:

$$\begin{aligned}x_{i+1} &= x_i + \alpha_i(-\nabla q(x_i)) \\ x_{i+1} &= x_i + \alpha_i r_i\end{aligned}$$

Luego tenemos que el residuo anterior y el actuar son ortogonales:

$$r_{i+1}^T r_i = 0$$

Encontrando el paso:

$$\begin{aligned}r_i^T r_{i+1} &= r_i^T r_i - \alpha_i r_i^T A r_i \\ \alpha_i &= \frac{r_i^T r_i}{r_i^T A r_i}\end{aligned}$$

De manera resumida:

$$\begin{aligned}r_i &= b - Ar_i \\a_i &= \frac{r_i^T r_i}{r_i^T A r_i} \\x_{i+1} &= x_i + \alpha_i r_i\end{aligned}$$

Gradiente conjugado [Ideal para matrices dispersas]:

Condiciones: A simétrica y positiva definida.

Consideraciones: Decimos que r_i es conjugado al vector p_i con respecto a la matriz A , si:

$$\langle r_i, p_i \rangle_A = \langle r_i, A p_i \rangle = 0$$

Condición de parada: $\|r_k\| > \varepsilon$

Algoritmo para $k = 0$:

$$\begin{aligned}r_0 &= b - Ax_0 \\p_0 &= r_0 \\x_0 &= x_0\end{aligned}$$

Algoritmo para $k > 0$:

$$\begin{aligned}\alpha_k &= \frac{r_k^T r_k}{p_k^T A p_k} \\x_{k+1} &= x_k + \alpha_k p_k \\r_{k+1} &= r_k + \alpha_k A p_k \\s_k &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \\p_{k+1} &= r_{k+1} + s_k p_k\end{aligned}$$

