Computación II:

Bloque 2: Resolución de sistemas de ecuaciones.

Búsqueda de ceros de una función.

* Búsqueda de ceros de una función.

(1.) Método de la Bisección.

Se realiza a partir del Teorema de Bolzano:
- $f(x)$ continua en $[a,b]$.
- $f(a) \cdot f(b) < 0 \Rightarrow$ cambio de signo. $\Rightarrow$ raíz r en el intervalo $[a,b]$.

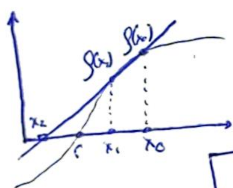
Si $x_1 = a$ y $x_2 = b \Rightarrow$ el nuevo punto es:

$$\boxed{x_3 = \frac{x_1 + x_2}{2}} \longrightarrow \text{Si } f(x_3) \simeq 0 \text{ terminamos.}$$

Sino, comparamos $f(x_1)$ y $f(x_3)$ o $f(x_2)$ y $f(x_3)$ a ver cuál cumple el Teorema de Bolzano y seguimos iterando.

Método no válido si $f(x)$ tiene raíces múltiples.

(2.) Método de la secante.



$\Rightarrow$ x_0 y x_1 a la izquierda o a la derecha de la raíz r. Por si acaso, x_0 más lejos que x_1. $f(x)$ continua.

$$\boxed{x_2 = x_1 - f(x_1) \cdot \frac{(x_0 - x_1)}{(f(x_0) - f(x_1))}} \Rightarrow \text{Si } f(x_2) \simeq 0 \text{ terminamos.}$$

Sino, seguimos iterando con x_1 y x_2.

(3.) Método de la falsa posición.

x_1 y x_0 en la vecindad de una raíz r t.q. $f(x_1) \cdot f(x_0) < 0$
↳ Teorema Bolzano.

$f(x)$ continua.

$$\boxed{x_2 = x_1 - f(x_1) \cdot \frac{(x_0 - x_1)}{(f(x_0) - f(x_1))}} \Rightarrow \text{Si } f(x_2) \simeq 0 \text{ terminamos.}$$

Sino, comparamos $f(x_2)$ con $f(x_1)$ y $f(x_0)$ y seguimos iterando con los puntos que cumplan el Teorema de Bolzano.

(4.) Método de Newton

x_0 próximo a la raíz r. Usamos $f(x)$ y $f'(x)$.
↳ tiene raíz r.

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}} \Rightarrow \text{Si } f(x_{n+1}) \simeq 0 \text{ terminamos.}$$

Sino seguimos iterando (calculamos x_2 con $f(x_1)$ y $f'(x_1)$).

5. Método de iteración del punto fijo.

Dado $f(x)$, hacemos la reordenación $x = g(x)$.

Ejemplo: $f(x) = x^2 - 2x - 3 \Rightarrow x = g(x) = \sqrt{2x+3}$ o $x = g(x) = \frac{(x^2-3)}{2}$ o $x = g(x) = \frac{3}{x-2}$

En función de la reordenación puede converger a r o no.

Iteramos de la siguiente forma:

$$\boxed{\begin{array}{l} x_1 = g(x_0) \\ x_2 = g(x_1) \\ \vdots \\ x_{n+1} = g(x_n) \end{array}}$$

$\Rightarrow x_0$ próximo a r.

$\Rightarrow$ pruebas las $f(x_i)$

Aceleración de la convergencia.

Estimación mejorada de la raíz, $\boxed{r \approx x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}}$ con $\Delta x_n = x_{n+1} - x_n$, $\Delta^2 x_n = \Delta x_{n+1} - \Delta x_n$

$\Rightarrow$ Útil cuando la función es difícil de evaluar.

* Sistemas lineales: Métodos directos.

1. Método L.U.

Dado $\boxed{A \cdot x = b} \Rightarrow \boxed{LU \cdot x = b}$, $A = LU$

$$\begin{cases} Lz = b & (1) \\ Ux = z & (2) \end{cases} \Rightarrow \text{Sacamos } x$$

L y U triangulares:

$$A = \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} = L \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ L_{10} & 1 & 0 \\ L_{20} & L_{21} & 1 \end{bmatrix} \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ 0 & U_{11} & U_{12} \\ 0 & 0 & U_{22} \end{bmatrix} \rightarrow 1^{\circ}$$

$\downarrow 2^{\circ}$

$$\boxed{\begin{array}{l} U_{ij} = A_{ij} - \sum_{k=1}^{i-1} L_{ik} U_{kj} \quad (i \le j) \; ; \; U_{ij} = 0 \quad (i > j) \\ L_{ij} = \frac{1}{U_{jj}} \cdot \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} \cdot U_{kj}\right) \quad (i > j) \; ; \; L_{ij} = 0 \; (i < j) \; ; \; L_{ii} = 1 \end{array}}$$

Luego $Lz = b \Rightarrow$ tenemos z. Luego $Ux = z \Rightarrow$ tenemos x.

2. Método LU tridiagonales

Dado $\boxed{Ax = f}$ con A tridiagonal: $\Rightarrow$ Cuadrada no singular

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & 0 & a_n & b_n \end{bmatrix} ; \; f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix} \Rightarrow L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \alpha_2 & 1 & & & \vdots \\ 0 & \alpha_3 & \ddots & & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha_n & 1 \end{bmatrix} ; \; U = \begin{bmatrix} \beta_1 & \gamma_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & \gamma_2 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \gamma_{n-1} \\ 0 & \cdots & \cdots & 0 & \beta_n \end{bmatrix}$$

$\boxed{a_1, c_n, \alpha_1 \text{ y } \gamma_n \text{ no existen.}}$

Fórmulas:

$$\boxed{\begin{array}{l} \gamma_i = c_i \, , \; i = 1 \ldots n. \\ \beta_1 = b_1 \longrightarrow \alpha_i = \frac{a_i}{\beta_{i-1}} \\ \hookrightarrow \beta_i = b_i - \alpha_i c_{i-1} \end{array}}$$

CÁLCULO L Y U.

$$\boxed{\begin{array}{l} z_1 = f_1 \; ; \; z_i = f_i - \alpha_i z_{i-1} \quad i = 2, \ldots n \\ x_n = \frac{z_n}{\beta_n} \; ; \; x_i = \frac{z_i - c_i \cdot x_{i+1}}{\beta_i} \quad i = n-1, \ldots, 1 \end{array}}$$

$\hookrightarrow$ Regresiva

CÁLCULO DE Z Y X $\rightarrow$ Solución.

③ Eliminación Gauss

Algoritmo: $\boxed{F_i = F_i - \left(\frac{A_{ij}}{A_{jj}}\right) \cdot F_j}$ ⟹ Elegimos F_i y F_j } Durante el proceso elemento mayor magnitud en la diagonal

⟹ No podemos tener ceros en la diagonal. Aplicamos pivoteo (cambio de filas) para evitarlo.

⟹ Proporciona una descomposición LU de la matriz A. Para ello:

- Sustituimos los ceros por debajo de la diagonal por la razón de coeficientes (A_{ij}/A_{jj}) ⟹ Hecha durante el proceso.

- L = unos diagonal y lo que queda por debajo.
- U = A escalonada
- $\det(A) = (-1)^{\text{intercambios filas}} \det(L) \cdot \det(U)$

④ Eliminación Gauss - Jordan

Variante del esquema de eliminación Gaussiano:

- Se hacen ceros al mismo tiempo los elementos por encima y por debajo de la diagonal.
- Los elementos de la diagonal se hacen igual a 1.
- Matriz coeficientes ⟶ matriz identidad
- $A|b \Rightarrow I|x$ ⟶ Solución
- Se aplica la técnica del pivote (ceros en la diagonal no permitidos).

* Sistemas lineales: Métodos iterativos

① Método de Jacobi

$$\boxed{x_i^{(k+1)} = \frac{b_i}{A_{ii}} - \sum_{j=1}^{n} \frac{A_{ij}}{A_{ii}} x_j^{(k)}}$$

- Asumimos valores iniciales $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$
- Siguiente iteración: $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ (de golpe).
- Tablita

② Método de Gauss - Seidel * Para ① y ② matriz dominante diagonal ⟶ + fácil reordenamos para facilitar convergencia.

$$\boxed{x_i^{(k+1)} = \frac{b_i}{A_{ii}} - \sum_{j=1}^{i-1} \frac{A_{ij}}{A_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^{n} A_{ij}/A_{ii}\, x_j^{(k)}}$$

- Podemos asumir solo $x_1^{(0)}$
- Vas necesitando los valores anteriores de lo que sería la misma iteración en Jacobi.

③ Método de sobrerrelajación

Acelerar la convergencia del método de Gauss - Seidel.

$$\boxed{x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{A_{ii}} \left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{(k+1)} - \sum_{j=1}^{n} A_{ij} x_j^{(k)} \right) = x_i^{(k)} + \omega \cdot \text{residuo}}$$

ω ⟶ factor de relajación.

- $0 < \omega < 1$ ⟶ Sistemas que no convergen en Gauss - Seidel.
- $1 < \omega < 2$ ⟶ Acelerar la convergencia en sistemas que son convergentes.
- Nunca mayor que 2 para evitar divergencias. Técnica útil para sistemas de ecuaciones derivados del análisis de ecuaciones diferenciales en derivadas parciales.

* Sistemas no lineales.

1. Método de sustituciones sucesivas.

Dado:

$$\left.\begin{array}{l} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{array}\right\} \xrightarrow[\text{Despejando las variables}]{\text{Reordenamos}} \begin{array}{l} x_1 = F_1(x_1, x_2, \dots, x_n) \\ x_2 = F_2(x_1, x_2, \dots, x_n) \\ \vdots \\ x_n = F_n(x_1, x_2, \dots, x_n) \end{array}$$

Podemos resolver de dos formas:

- Análogamente a Jacobi:

$$\boxed{x_i^{(k+1)} = F_i\left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right), \quad i = 1, 2, \dots, n}$$

- Análogamente a Gauss-Seidel:

$$\boxed{x_i^{(k+1)} = F_i\left(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k)}, \dots, x_n^{(k)}\right), \quad i = 1, 2, \dots, n}$$

2. Método de Newton-Raphson

Variable x y función f son vectores de M elementos, donde M es el número de ecuaciones:

$$\boxed{x^{(n+1)} = x^{(n)} - \left(f'(x^{(n)})\right)^{-1} \cdot f(x^{(n)})} \Rightarrow \text{Tomamos valores iniciales}$$

Donde:

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_M \end{pmatrix}; \quad f(x) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_M) \\ \vdots \\ f_M(x_1, x_2, \dots, x_M) \end{pmatrix}; \quad f'(x) = \underbrace{\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_M} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_M} \end{pmatrix}}_{\text{Matriz Jacobiana}}$$

Si el cálculo de la inversa de la matriz de las derivadas es problemático, se puede determinar $x^{(n+1)}$ resolviendo el siguiente sistema de ecuaciones lineal (por ejemplo por el método LU):

$$\boxed{\left(f'(x^{(n)})\right)\left(x^{(n+1)} - x^{(n)}\right) = -f(x^{(n)})}$$

Bloque 3: Diagonalización de matrices

* El problema de los autovalores

El problema planteado por $Ax = \lambda x$ con $x \neq 0$ tiene solución si:

$$\boxed{p(\lambda) = \det(A - \lambda I)x = 0 \Rightarrow \det(A - \lambda I) = 0}$$

1º Se encuentran los autovalores λ.
2º Se sustituyen estos valores en $(A - \lambda I) \cdot v = 0$
3º De aquí sacamos los v (autovectores) asociados a cada λ.

* Diagonalización de matrices simétricas y cuadradas: El método de Jacobi.

El método sirve para matrices simétricas y $n \times n$ (hasta 5×5). Nos permite calcular todos los autovalores y autovectores. La idea del método es realizar rotaciones, de forma que se anule el elemento más grande fuera de la diagonal.

Tenemos una matriz A_0:

$$A_0 = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \Rightarrow \text{Cuadrada y simétrica.}$$

1º Elegimos el a_{ij} de fuera de la diagonal (→ mejor encima.) cuyo valor absoluto sea el mayor.

2º Calculamos θ_1: $\theta_1 = \begin{cases} \frac{1}{2} \operatorname{arctg}\left(\frac{2a_{ij}}{a_{ii} - a_{jj}}\right) & \text{si } a_{ii} \neq a_{jj} \\ \frac{\pi}{4} & \text{si } a_{ii} = a_{jj} \end{cases}$

3º Calculamos la matriz de rotación: $R_1(\theta) = \begin{cases} \text{"1" en la diagonal} \\ R_{ii}, R_{jj} = \cos(\theta_1) \\ R_{ij} = -\operatorname{sen}(\theta_1) \\ R_{ji} = \operatorname{sen}(\theta_1) \\ \text{"0" en el resto.} \end{cases}$

4º La nueva matriz $\boxed{A_1 = R_1^t \cdot A_0 \cdot R_1}$

5º La matriz U se calcula en cada iteración (matriz autovectores):

$\boxed{U_k = U_{k-1} \cdot R_k} \Rightarrow$ En la primera iteración $\boxed{U_1 = R_1}$

6º El proceso termina cuando el elemento a_{ij} de fuera de la diagonal con mayor valor absoluto es menor que una tolerancia. Es decir, muy próximo a 0.