Resumen Cálculo Numérico

basado en el resumen de Fernando Nellmeldín

Cristian Escudero

September 30, 2012

@TODO: Agregar criterio de convergencias a todos.

1 Métodos Directos

1.1 Eliminación de Gauss

Los elementos de la matriz $A^{(k+1)}$ se calculan:

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} & \text{si } i \leq k, \\ a_{ij}^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}\right) & \text{si } i \geq k+1, \text{ y } j \geq k+1, \\ 0 & \text{si } i \geq k+1, \text{ y } j \leq k. \end{cases}$$

Si tenemos **pivoteo parcial**, trabajamos con el **vector de permutación**:

```
# Se resuelve en n - 1 pasos
for i = 1 : n - 1
    if (parcial)
        # Ponemos la fila con maximo valor
        [tr, p] = max(abs(Ab(idx(i:n), i)));
        p = p + i - 1;

        if (idx(i) != idx(p))
            temp = idx(p);
            idx(p) = idx(i);
            idx(i) = temp;
        end
    end
    # ... sigue el metodo...
end
```

Y trabajamos usando el `idx(i)` en vez de `i` para los subíndices.

<u>Nota</u>: Si encontramos una columna de ceros, el método no determina una solución **única**.

1.2 Factorización LU

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix} = LU$$

Ejemplo 3x3:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = LU$$
$$= \begin{bmatrix} l_{11}\,u_{11} & l_{11}\,u_{12} & l_{11}\,u_{13} \\ l_{21}\,u_{11} & l_{21}\,u_{12} + l_{22}\,u_{22} & l_{21}\,u_{13} + l_{22}\,u_{23} \\ l_{31}\,u_{11} & l_{31}\,u_{12} + l_{32}\,u_{22} & l_{31}\,u_{13} + l_{32}\,u_{23} + l_{33}\,u_{33} \end{bmatrix}$$

1.2.1 Factorización de Cholesky

Nota: En a_{ij}^{upper}: $min(i,j) - 1 = i - 1$; y en a_{ij}^{lower}: $min(i,j) - 1 = j - 1$.

$$a_{ij} = \sum_{k=1}^{min(i,j)} l_{ik}\,u_{kj},$$
$$a_{ij}^{upper} = \sum_{k=1}^{i-1} l_{ik}\,u_{kj} + l_{ii}\,u_{ij} \qquad \Rightarrow\; u_{ij} = \frac{1}{l_{ii}}\left[a_{ij} - \sum_{k=1}^{i-1} l_{ik}\,u_{kj}\right]$$
$$a_{ij}^{lower} = \sum_{k=1}^{j-1} l_{ik}\,u_{kj} + u_{jj}\,l_{ij} \qquad \Rightarrow\; l_{ij} = \frac{1}{u_{jj}}\left[a_{ij} - \sum_{k=1}^{j-1} l_{ik}\,u_{kj}\right]$$

Factorización de **Doolittle** ($l_{ii} = 1$) $\qquad u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}\,u_{kj}$

Factorización de **Crout** ($u_{ii} = 1$) $\qquad l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}\,u_{kj}$

2 Métodos Iterativos

Quiero llevar el SEAL a una forma iterativa ($\mathbf{x} = T\mathbf{x} + C$) para poder resolverla:

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ A &= D - L - U \\ (D - L - U)\mathbf{x} &= \mathbf{b} \end{aligned}$$

Jacobi:

$$\begin{aligned} D\mathbf{x} - (L + U)\mathbf{x} &= \mathbf{b} \\ D\mathbf{x} &= (L + U)\mathbf{x} + \mathbf{b} \\ \mathbf{x} &= D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b} \\ \mathbf{x} &= T\mathbf{x} + C \end{aligned}$$

Gauss-Seidel:

$$\begin{aligned} (D - L)\mathbf{x} - U\mathbf{x} &= \mathbf{b} \\ (D - L)\mathbf{x} &= U\mathbf{x} + \mathbf{b} \\ \mathbf{x} &= (D - L)^{-1}U\mathbf{x} + (D - L)^{-1}\mathbf{b} \\ \mathbf{x} &= T\mathbf{x} + C \end{aligned}$$

SOR:

$$\begin{aligned} (D - wL)\mathbf{x} &= [(1 - w)D + wU]\mathbf{x} + w\mathbf{b} \\ \mathbf{x} &= (D - wL)^{-1}[(1 - w)D + wU]\mathbf{x} + (D - wL)^{-1}w\mathbf{b} \\ \mathbf{x} &= T\mathbf{x} + C \end{aligned}$$

Nociones básicas de convergencia:

- Si $||T|| < 1 \;\; \forall || \cdot || \Rightarrow$ convergen todos.
- Si A es e.d.d $\Rightarrow$ converge Jacobi y Gauss-Seidel.
- Si A es d.p & $0 < w < 2 \Rightarrow$ SOR converge.
- $\rho(T) < 1 \iff T\mathbf{x} + C$ converge.

2.1 Jacobi

Despejamos x_i del SEAL:

$$
\begin{aligned}
a_{11}\,x_1 + a_{12}\,x_2 + a_{13}\,x_3 = b_1 \qquad &\Rightarrow \qquad x_1 = \frac{1}{a_{11}}\,[b_1 - (a_{12}\,x_2 + a_{13}\,x_3)] \\
a_{21}\,x_1 + a_{22}\,x_2 + a_{23}\,x_3 = b_2 \qquad &\Rightarrow \qquad x_2 = \frac{1}{a_{22}}\,[b_2 - (a_{21}\,x_1 + a_{23}\,x_3)] \\
a_{31}\,x_1 + a_{32}\,x_2 + a_{33}\,x_3 = b_3 \qquad &\Rightarrow \qquad x_3 = \frac{1}{a_{33}}\,[b_3 - (a_{31}\,x_1 + a_{32}\,x_2)]
\end{aligned}
$$

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

2.2 Gauss-Seidel

$$
x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k-1)} \right)
$$

2.3 SOR (Succesive Over-Relaxation)

$$
x_i^{(k)} = (1-w)\, x_i^{(k-1)} + \frac{w}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k-1)} \right)
$$

2.4 Gradiente Conjugado

Elige las direcciones de búsqueda ($\mathbf{v}^{(k)}$) durante el proceso iterativo de modo que los $\mathbf{r}^{(k)}$ sean mutuamente ortogonales.

1. Partimos de un $\mathbf{x}^{(0)}$ y usamos la **dirección de máximo descenso** $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^0$ como $\mathbf{v}^{(1)}$.
2. Calculamos el paso de avance t en la dirección $\mathbf{v}$ y la solución aproximada $\mathbf{x}^{(k)}$ (iniciamos con $k = 1$):

$$
t = \frac{< \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} >}{< \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} >}, \qquad \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k \mathbf{v}^{(k)}.
$$

3. Si $\mathbf{x}^{(k)}$ es la solución de $A\mathbf{x} = \mathbf{b}$ terminamos. Sino calculamos: $\mathbf{r}^{(k)} = \mathbf{b} - t_k A\mathbf{v}^{(k)}$, actualizamos el vector de búsqueda:

$$
\mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + s_k \mathbf{v}^{(k)}, \qquad s_k = \frac{< \mathbf{r}^{(k)}, \mathbf{r}^{(k)} >}{< \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} >},
$$

   y volvemos al paso *2*.

2.4.1 Precondicionadores

Para solventar los casos en el que la matriz A a resolver esté mal condicionada, se propone un *precondionamiento*. Es decir, encontrar una matriz P tal que: $\kappa(P^{-1}A) \approx 1$.

Jacobi. El precondicionamiento más simple es: $P = D_s$, con D_s diagonal de A. Es efectivo si la matriz es **diagonal dominante**.

Block-Jacobi. Una matriz con bloques de submatrices sobre la diagonal de A.

SOR: $P = (D_s + w\ L_s)\ D_s^{-1}\ (D_s + w\ U_s)$.

Cholesky incompleta. Si A es **simétrica, definida positiva**, puede descomponerse en: $A = C^T C = C^{*T} C^* + R$, con C^* una descomposición de *Cholesky* restringida a la estructura rala de A. Tomando $P = C^{*T} C^*$, se espera que el κ sea pequeño.

2.4.2 Criterios de Corte

Hay varias formas de decidir cuando detener el proceso iterativo:

- **Error absoluto:** $||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}|| \leq$ tolerancia$_1$. La tolerancia$_1$ depende del significado físico de la variable $\mathbf{x}$.
- **Error relativo** $\frac{||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}||}{||\mathbf{x}^{(k)}||} \leq$ tolerancia$_2$. Aquí se es independiente la tolerancia$_2$ de las unidades.
- **Error en la aproximación**: $||\mathbf{r}^{(k)}|| \leq$ tolerancia$_3 \cdot ||\mathbf{b}||$. Es independiente de las unidades debido a que se refiere a la norma del vector de términos independientes.

@@TODO: Hay un apéndice en la diapositiva de iterativos. Revisar.

3 Solución de Ecuaciones No Lineales de una Variable

3.1 Método de la Bisección

El *método de la Bisección* procede buscando una raíz propuesta en la mitad del intervalo (a, b), repitiendo iterativamente el proceso.

Es un método lento, de convergencia *lineal*, pero **siempre** *converge*. Es *robusto* (siempre encuentra solución), por lo que por esa razón es usado para iniciar otros métodos más eficientes.

Sea $f(x)$ **contínua** en $[a, b]$ y $f(a) \cdot f(b) < 0$. Entonces, por el **teorema del valor medio**, $\exists\ a < p < b\ /\ f(p) = 0$.

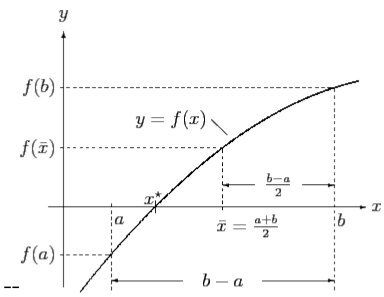

Figure 1: *Método de la Bisección*, representación gráfica.

3.1.1 Algoritmo

Supongamos $a_1 = a$, $b_1 = b$, $i = 1$.

1. Calculamos el punto medio $p_i = a_i + \frac{b_i - a_i}{2} = \frac{a_i + b_i}{2}$.
2. Si $f(p_i) = 0 \Rightarrow p_i$, encontramos la raíz, y terminamos.
3. Si $f(p_i) \cdot f(a_i) < 0$:
   - Entonces, $b_{i+1} = p_i$, y $a_{i+1} = b_i$.
   - De lo contrario, $a_{i+1} = p_i$, y $b_{i+1} = b_i$.
4. Incrementamos i, y volvemos al paso *1*.

3.1.2 Criterios de corte

Siendo una tolerancia $\mathcal{E} > 0$:

- **Error absoluto:** $|p_i - p_{i-1}| \leq \mathcal{E}$.
- **Error relativo:** $\frac{|p_i - p_{i-1}|}{|p_i|} \leq \mathcal{E}$. Es independiente de las unidades y del significado físico de estas.
- **Error en la aproximación:** $|f(p)| \leq \mathcal{E}$.

3.2 Iteración de Punto Fijo

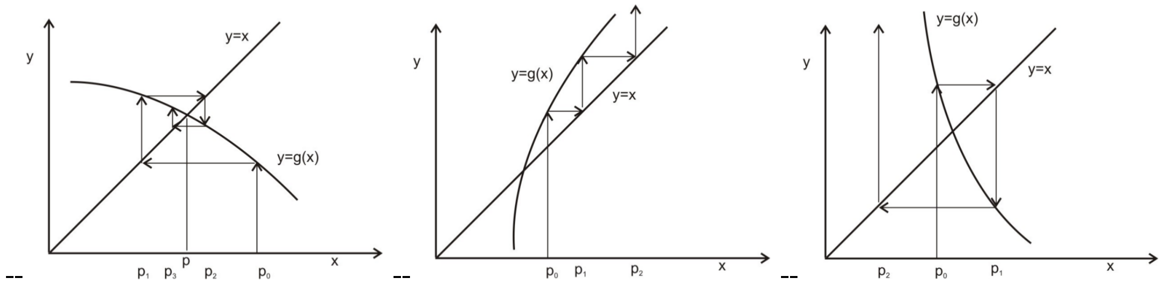

Figure 2: *Iteración de Punto Fijo*, **convergencia** por $|g'(x)| < 1$ (izquierda), **divergencia** por $g'(x) < -1$ (medio), y por $|g'(x)| > 1$(derecha).

3.2.1 Algoritmo

1. Se escoge una aproximación inicial p_0.
2. Se genera la sucesión $\{p_n\}_{n=1}^{\infty}$, con $p_n = g(p_{n-1})$.
3. Si la sucesión converge en p y si g es continua, entonces:

$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g(\lim_{n\to\infty} p_{n-1}) = g(p),$$

   y obtenemos una solución con $p = g(p)$.

3.3 Método de Newton-Raphson

Sea $f \in C^2[a,b]$, el método construye una sucesión $\{p_n\}$ con la siguiente fórmula de recurrencia:

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f(p_{n-1})}, \quad n \geq 1.$$

Convergencia: está dada por el **teorema** 2.5.

4 Interpolación y aproximación de funciones