SME0305 - 2016

Roberto F. Ausas / Gustavo C. Buscaglia

ICMC - Ramal 736628, rfausas@gmail.com
ICMC - Ramal 738176, gustavo.buscaglia@gmail.com

O que há por trás da barra $\backslash$

Já vimos que existem problemas na engenharia em que precisamos resolver grandes sistemas de equações lineares:

- Cálculo de redes elétricas e hidráulicas;
- Cálculo de estruturas $\rightarrow$ Mecânica dos solidos;
- Cálculo de escoamentos $\rightarrow$ Mecânica dos fluidos;

Vamos estudar métodos para resolver sistemas de equações lineares da forma:

$$A\mathbf{x} = \mathbf{b}$$

em que

- A é uma matriz de $n \times n$
- $\mathbf{x}$ e $\mathbf{b}$ são vetores de dimensão n
  - $\mathbf{x}$ vetor de incógnitas
  - $\mathbf{b}$ vetor conhecido de lado direito ou RHS

Ex. $n = 2$, $A\mathbf{x} = \mathbf{b}$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

ou

$$\begin{array}{ccccc} a_{11}x_1 & + & a_{12}x_2 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & = & b_2 \end{array}$$

Em geral vamos escrever:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ . & & . & & & & . & & . \\ . & & . & & & & . & & . \\ . & & . & & & & . & & . \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{array}$$

Revisão de Algebra linear

Antes de começar aplicar métodos, lembremos o seguinte teorema importante:

Para um sistema quadrado de $n \times n$, a solução é única se uma das seguintes proposições equivalentes se cumpre

- A é não singular;
- $\det(A) \neq 0$;
- As filas de A são linearmente independentes;
- Existe a matriz inversa A^{-1};
- imagem$(A) = \mathbb{R}^n$ (São todos os vetores que podem-se escrever como combinação linear das colunas de A);
- nucleo$(A) = \{\mathbf{0}\}$ (São todos os vetores $\mathbf{z}$ para os que $A\,\mathbf{z} = 0$);

Métodos de resolução

Há dois tipos de métodos para resolver sistemas de equações:

1. **Métodos diretos**: Dão a solução exata a menos dos erros de arredondamento.

   – Eliminação de Gauss - Escalonamento

   – Métodos baseados em decomposições: LU, Cholesky, QR

2. **Métodos iterativos**: Iteramos até atingir uma solução aproximada.

   – Jacobi e Gauss-Seidel

   – Métodos dos Gradientes, Gradientes conjugados e outros mais sofisticados.

Que sabemos resolver?

Matriz triangular superior: $a_{ij} = 0$ sempre que $i > j$.

Matriz triangular inferior: $a_{ij} = 0$ sempre que $i < j$.

São matrizes da forma: $n = 4$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}, \quad L = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix}$$

Sistema triangular superior: $U\,x = y$:

Substitução regressiva (Backward substitution)

$$
\begin{array}{ccccccccc}
u_{11}x_1 & + & u_{12}x_2 & + & \dots & + & u_{1n}x_n & = & y_1 \\
 & & u_{22}x_2 & + & \dots & + & u_{2n}x_n & = & y_2 \\
 & & & & & & . & & . \\
 & & & & & & . & & . \\
 & & & & & & . & & . \\
 & & & & & & u_{nn}x_n & = & y_n
\end{array}
$$

$$
\begin{aligned}
x_n &= \frac{1}{u_{nn}}\, y_n \\
x_i &= \frac{1}{u_{ii}} \left(y_i - \sum_{j=i+1}^{n} u_{ij}x_j \right), \quad i = n-1, \dots, 1
\end{aligned}
$$

Sistema triangular inferior: $L\,y = b$:

Substitução progressiva (Forward substitution)

$$\begin{array}{ccccccccc}
l_{11} y_1 & & & & & & & = & b_1 \\
l_{21} y_1 & + & l_{22} y_2 & & & & & = & b_2 \\
\cdot & & & & & & & & \cdot \\
\cdot & & & & & & & & \cdot \\
\cdot & & & & & & & & \cdot \\
l_{n1} y_1 & + & l_{n2} y_2 & + & \dots & + & l_{nn} y_n & = & b_n
\end{array}$$

$$\begin{aligned}
y_1 &= \frac{1}{l_{11}}\, b_1 \\
y_i &= \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right), \quad i = 2, \dots, n
\end{aligned}$$

Escalonamento - Decomposição LU

Primeiro precisamos lembrar as operações elementares:

As seguintes operações aplicadas a um sistema linear geram um sistema linear que é equivalente (i.e., os sistemas têm a mesma solução):

(i) Multiplicar uma equação por um escalar;

(ii) Mudar uma equação pela soma dela mesma e de um multiplo não zero de qualquer outra equação;

(iii) Trocar a ordem de duas equações;

Cada uma dessas operações pode ser representada por uma matriz que, multiplicada a esquerda da matriz A do sistema, produz a operação desejada. Vejamos por exemplo (ii):

Utilizando matrizes triangulares inferiores da forma (para $n = 4$ por exemplo):

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & 0 & 1 & 0 \\ l_{41} & 0 & 0 & 1 \end{pmatrix}$$

cuja inversa é:

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{41} & 0 & 0 & 1 \end{pmatrix}$$

que também é triangular inferior, temos

$$L^{-1} \cdot A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{41} & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \dots$$

(fazer na lousa)

i.e., pegamos a primeira linha, a multiplicamos pelo fator l_{i1} e a substraimos da linha i para $i = 2, 3, 4$.

Similarmente, se tivermos os fatores l's em otra coluna, p.e.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -l_{32} & 1 & 0 \\ 0 & -l_{42} & 0 & 1 \end{pmatrix}.$$

Verficar-lo.

Escalonamento - Decomposição LU

Vamos resolver o sistema $A\mathbf{x} = \mathbf{b}$ (Ex. para $n = 4$)

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
a_{21}	a_{22}	a_{23}	a_{24}	b_2
a_{31}	a_{32}	a_{33}	a_{34}	b_3
a_{41}	a_{42}	a_{43}	a_{44}	b_4

1. Substrair a fila 1 multiplicada por $l_{i1} = a_{i1}/a_{11}$ da fila i, $i = 2, \dots, 4$
2. Definir $a'_{ik} = a_{ik} - l_{i1}a_{1k}$, $i, k = 2, \dots, 4$
3. Definir $b'_i = b_i - l_{i1}b_1$, $i = 2, \dots, 4$

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	a'_{32}	a'_{33}	a'_{34}	b'_3
0	a'_{42}	a'_{43}	a'_{44}	b'_4

1. Substrair a fila 2 multiplicada por $l'_{i2} = a'_{i2}/a'_{22}$ da fila i, $i = 3, \ldots, 4$
2. Definir $a''_{ik} = a'_{ik} - l'_{i2}a'_{2k}$, $i, k = 3, \ldots, 4$
3. Definir $b''_i = b'_i - l'_{i2}b'_2$, $i = 3, \ldots, 4$

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	0	a''_{33}	a''_{34}	b''_3
0	0	a''_{43}	a''_{44}	b''_4

1. Substrair a fila 3 multiplicada por $l''_{43} = a''_{43}/a''_{33}$ da fila i, $i = 3, \ldots, 4$
2. Definir $a'''_{44} = a''_{44} - l''_{43}a''_{34}$
3. Definir $b'''_4 = b''_4 - l''_{43}b''_3$.

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	0	a''_{33}	a''_{34}	b''_3
0	0	0	a'''_{44}	b'''_4

É uma matriz triangular superior!

Agora, vamos supor que guardamos os fatores $l's$ e vamos armar umas matrizes:

$$L_1 = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & & 1 & \\ l_{41} & & & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & l'_{32} & 1 & \\ & l'_{42} & & 1 \end{pmatrix}, \quad L_3 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & l''_{43} & 1 \end{pmatrix}$$

Cujas inversas erãm

$$L_1^{-1} = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ -l_{31} & & 1 & \\ -l_{41} & & & 1 \end{pmatrix}, \quad L_2^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & -l'_{32} & 1 & \\ & -l'_{42} & & 1 \end{pmatrix}, \quad L_3^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -l''_{43} & 1 \end{pmatrix}$$

Então, tudo esse processo que temos feito para achar uma matriz triangular superior, pode-se ver que de fato foi:

$$U = L_3^{-1}\, L_2^{-1}\, L_1^{-1}\, A$$

Já que o produto de matrizes triangulares inferiores também é triangular inferior, e a inversa também é, resulta:

$$U = L^{-1}\, A \Rightarrow A = L\, U$$

O que acabamos de fazer se chama fatoração LU, em que as matrizes L e U são:

$$U = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l'_{32} & 1 & 0 \\ l_{41} & l'_{42} & l''_{43} & 1 \end{pmatrix}$$

A matriz L só tem 1's na diagonal, de fato, poderiamos ter aproveitado a matriz A original para guardar os l's, ou seja, não precisamos criar matrices adicionais.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ l_{21} & a'_{22} & a'_{23} & a'_{24} \\ l_{31} & l'_{32} & a''_{33} & a''_{34} \\ l_{41} & l'_{42} & l''_{43} & a'''_{44} \end{pmatrix}$$

$\Rightarrow$ Os passos para resolver $A\mathbf{x} = \mathbf{b}$ são:

- Achar os fatores L e U

$$L\,U = A$$

- Notando que $A\mathbf{x} = (LU)\mathbf{x} = L(U\mathbf{x}) = \mathbf{b}$, resolver com **forward substitution**:

$$L\,\mathbf{y} = \mathbf{b}$$

- Resolver com **backward substitution**:

$$U\,\mathbf{x} = \mathbf{y}$$

Decomposição LU: Algoritmo básico

A implementação seria:

```
for k=1:n-1
  if(A(k,k) == 0)
    error('Elemento nulo na diagonal');
  end
  for i=k+1:n
    A(i,k) = A(i,k)/A(k,k);
    for j=k+1:n
      A(i,j) = A(i,j) - A(i,k)*A(k,j);
    end
  end
end
```

Decomposição LU: Teorema

A matriz $A \in \mathbb{R}^{n\times n}$ tem uma única decomposição $A = LU$ se e só se todas as submatrices principais $A_k = A(1:k, 1:k),\ k = 1, \ldots, n-1$ são não singulares.

Matrizes comuns na engenharia

Em algumas matrizes é fácil saber que se verificam as condições do teorema:

- Matrizes simetricas e definidas positivas:

  Uma matriz $A \in \mathbb{R}^{n\times n}$ diz-se definida positiva, se $\forall\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{x} \neq \mathbf{0},\ \ \mathbf{x}^T A\mathbf{x} > 0$.

- Matrizes com diagonal estritamente dominante:

  Uma matriz é de diagonal estritamente dominante por linha se:

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad i = 1, \dots, n$$

e por columas se

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ji}|, \quad i = 1, \dots, n$$

Eliminação de Gauss com Pivoting

Para outras matrizes em geral, o processo de eliminação de Gauss não pode ser completado sem recorrer a troca de linhas da matriz.

Vejamos um exemplo (na lousa)

Inclusive, em alguns casos, da para fazer, mas para **evitar** fatores a_{ii} muito pequenos e para minimizar **erros de arredondamento** se faz troca de linhas, para que o a_{ii} seja o maior possível. Aqui que aparecem as matrizes de permutação (Lembrar das operações elementares)

Isto se chama pivoting!

Matrizes de permutação

Uma matriz de permutação $P \in \mathbb{R}^{n\times n}$ é uma matriz em que cada fila e coluna tem apenas uma entrada igual a 1 e o resto das entradas são 0. As filas de P são permutações de filas da matriz identidade.

- A inversa de uma matriz de permutação é a sua trasposta, i.e., $P^{-1} = P^T$.
- O produto de uma matriz de permutação por uma matriz A, da como resultado a matriz A com suas filas permutadas do mesmo jeito.

Ex. $n = 4$

$$PA = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Eliminação de Gauss com Pivoting

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
a_{21}	a_{22}	a_{23}	a_{24}	b_2
a_{31}	a_{32}	a_{33}	a_{34}	b_3
a_{41}	a_{42}	a_{43}	a_{44}	b_4

1. Permutar filas $i = 1, \dots, 4$ (se necessário) para que $a_{11} \neq 0$ e o maior possivel. Esse elemento será o Pivot (guardar P_1)
2. Substrair a fila 1 multiplicada por $l_{i1} = a_{i1}/a_{11}$ da fila i, $i = 2, \dots, 4$
3. Definir $a'_{ik} = a_{ik} - l_{i1}a_{1k}$, $i, k = 2, \dots, 4$
4. Definir $b'_i = b_i - l_{i1}b_1$, $i = 2, \dots, 4$

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	a'_{32}	a'_{33}	a'_{34}	b'_3
0	a'_{42}	a'_{43}	a'_{44}	b'_4

1. Permutar filas $i = 2, \ldots, 4$ (se necessário) para que $a'_{22} \neq 0$ e o maior possivel. Esse elemento será o próximo Pivot (guardar P_2)
2. Substrair a fila 2 multiplicada por $l'_{i2} = a'_{i2}/a'_{22}$ da fila i, $i = 3, \ldots, 4$
3. Definir $a''_{ik} = a'_{ik} - l'_{i2}a'_{2k}$, $i, k = 3, \ldots, 4$
4. Definir $b''_i = b'_i - l'_{i2}b'_2$, $i = 3, \ldots, 4$

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	0	a''_{33}	a''_{34}	b''_3
0	0	a''_{43}	a''_{44}	b''_4

1. Permutar filas $i = 3, \dots, 4$ (se necessário) para que $a''_{33} \neq 0$ e o maior possivel. Esse elemento será o próximo Pivot (guardar P_3)
2. Substrair a fila 3 multiplicada por $l''_{43} = a''_{43}/a''_{33}$ da fila i, $i = 3, \dots, 4$
3. Definir $a'''_{44} = a''_{44} - l''_{43} a''_{34}$
4. Definir $b'''_4 = b''_4 - l''_{43} b''_3$.

x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	b_1
0	a'_{22}	a'_{23}	a'_{24}	b'_2
0	0	a''_{33}	a''_{34}	b''_3
0	0	0	a'''_{44}	b'''_4

É uma matriz triangular superior!

Agora, vamos supor que guardamos duas coisas

- Os fatores $l's$
- E as permutações, que poderiam se pensar como matrizes de permutação P_1, P_2, P_3

Como antes, vamos pegar os fatores $l's$ e vamos armar umas matrizes:

$$L_1 = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & & 1 & \\ l_{41} & & & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & l'_{32} & 1 & \\ & l'_{42} & & 1 \end{pmatrix}, \quad L_3 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & l''_{43} & 1 \end{pmatrix}$$

Cujas inversas já sabiamos calcular.
Então, tudo esse processo que temos feito para achar uma matriz triangular superior, pode-se ver que de fato foi:

$$U = L_3^{-1}\, P_3\, L_2^{-1}\, P_2\, L_1^{-1}\, P_1\, A$$

ou, fazendo um reacomodamento:

$$U = \underbrace{L_3^{-1}(P_3\, L_2^{-1}\, P_3^{-1})\,(P_3\, P_2\, L_1^{-1}\, P_2^{-1}\, P_3^{-1})}_{L^{-1}}\ \underbrace{(P_3 P_2 P_1)}_{P}\ A$$

Finalmente

$$U = L^{-1}\,P\,A \Rightarrow P\,A = L\,U$$

O que acabamos de fazer se chama fatoração LU com pivoting parcial, em que as matrizes L e U são:

$$U = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l'_{32} & 1 & 0 \\ l_{41} & l'_{42} & l''_{43} & 1 \end{pmatrix}$$

Igual que antes, veja que na matriz L só tem 1's na diagonal.

Resolução de sistemas

$\Rightarrow$ Os pasos para resolver $A\mathbf{x} = \mathbf{b}$ são:

- Achar os fatores L e U e guardar as permutações P tal que

$$L\,U = P\,A$$

- Notando que $(LU)\mathbf{x} = (PA)\mathbf{x} = P(A\mathbf{x}) = P\mathbf{b}$, resolver com **forward substitution**:

$$L\,\mathbf{y} = P\,\mathbf{b}$$

- Resolver com **backward substitution**:

$$U\,\mathbf{x} = \mathbf{y}$$

Este processo está garantido para matrizes não singulares pelo teorema:

Teorema 2

Se $A \in \mathbb{R}^{n \times n}$ é não singular, então podemos achar uma matriz de permutação P tal que PA satisfaz as condições do **Teorema 1**, i.e., $PA = LU$ existe e é unica.

Complexidade computacional

Quantas operações temos que fazer (#flops)

- Decomposição LU:

$$\#\text{flops}_{LU} = \frac{2}{3}n^3 + \mathcal{O}(n^2)$$

- Forward substitution:

$$\#\text{flops}_{F} = n^2 + \mathcal{O}(n)$$

- Backward substitution:

$$\#\text{flops}_{B} = n^2 + \mathcal{O}(n)$$

- Em total:

$$\#\text{flops} = \frac{2}{3}n^3 + \mathcal{O}(n^2) + 2n^2 + \mathcal{O}(n) \sim \mathcal{O}(n^3)$$

i.e., se para resolver um sistema de 100 equações preciso 1 milisegundo $\rightarrow$ para resolver um sistema com 1000, vou precisar 1 segundo!

Decomposição de Cholesky

Se A é simétrica e definida positiva, podemos construir uma fatoração melhor:

$$A = H\,H^T$$

Em que H é uma matriz triangular inferior que tem elementos positivos na diagonal.

Decomposição de Cholesky: Algoritmo

$$h_{11} = \sqrt{a_{11}}$$

para $i = 2, \ldots, n$

$$h_{ij} = \frac{1}{h_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik}\, h_{jk} \right), \quad j = 1, \ldots, i-1$$

$$h_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2}$$

Resolução via Cholesky: Complexidade computacional

- Em total:

$$\#\text{flops} = \frac{1}{3}n^3 + \mathcal{O}(n^2)$$

i.e., que demora a mitade do tempo que a decomposição LU!

Funções de Octave/Matlab

Temos várias funções para resolver sistemas por método directos:

- `lu(A)`
- `chol(A)`
- `inv(A)` (Tentar não usar em geral)
- Temos a barra: “\”:

```
x = A \ b
```

A barra “\”

A “\” é bastante sofisticada. Dependendo da estrutura e tipo de matriz, vai escolher o mais apropriado.

Em Octave: Para matrizes cheias, sempre se usa LU ou QR. Mas, para matrizes em formato esparso:

- Se A é triangular superior ou inferior, ou inclusive uma permutação de uma matriz triangular, os algoritmos de backward ou forward substitution são usados.
- Se A é simetrica e com elementos positivos na diagonal a decomposição de Cholesky é testada.
- Se nada do anterior serve, a decomposição LU com pivoting é realizada.
- Se A não é quadrada se procura uma solução no sentido dos quadrados mínimos.

Cuidado: Há algumas diferenças entre Octave e Matlab, no funcionamento da $\backslash$. Ver o livro de Quarteroni, na pagina 144 para mais detalhes.

Matrizes Esparsas

De maneira informal, uma matriz esparsa é uma matriz com uma quantidade suficiente de zeros, para valer a pena tomar vantagem disso.

A ideia básica, é so guardar os elementos diferentes de zero da matriz, e as posições em que eles se encontram. Desta forma vamos **economizar**:

- Memoria;
- Número de operações;
- Tempo de cálculo;

Em Octave temos varias funções para trabalhar com matrizes esparsas:

- `Asp = sparse(Af)` $\rightarrow$ Para convertir de formato cheio a formato esparso;
- `Af = full(Asp)` $\rightarrow$ Para convertir de formato esoarsi a formato cheio;
- `nnz(A)` $\rightarrow$ Para contar o número de não zeros da matriz;
- `spy(A)` $\rightarrow$ para visualizar a estrutura da matriz;
- `lu(A)` (decomposição LU), `chol(A)` (decomposição de Cholesky), a barra $\backslash$, e outras funções e operadores podem trabalhar sem problemas tanto com matrizes cheias como esparsas.