Algoritmia Aplicada

Ano lectivo 2011-2012

Aula 10 – Algoritmos Matemáticos: Resolução de Sistemas de Equações

Sumário

- Resolução de sistemas de equações
  - Introdução. Número de soluções. Operações possíveis em sistemas
  - Métodos directos
  - Métodos iterativos
  - Métodos de factorização

Resolução de Sistemas de Equações Lineares

➢ Introdução

- Sistema de N equações lineares a N incógnitas (variáveis)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1N}x_N = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2N}x_N = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{N1}x_1 + a_{N2}x_2 + \ldots + a_{NN}x_N = b_N \end{cases}$$

Ou sob a forma matricial:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_N \end{bmatrix}$$

Resolução de Sistemas de Equações Lineares

- **Ou, sob forma matricial compacta:**

$$Ax = b$$ **com:**

$A_{(NxN)}$: matriz dos coeficientes

$b_{(Nx1)}$: vector dos termos independentes

$x_{(Nx1)}$: vector das incógnitas (ou variáveis

- **Resolver o sistema de equações consiste em:**

Dados A e b, determinar x que satisfaça todas as equações

Resolução de Sistemas de Equações Lineares

- **Nº de soluções**
  - **Três casos podem ocorrer:**

a) Nenhuma solução, como no exemplo seguinte em que as duas equações são contraditórias:

$$\begin{cases} 2x_1 + 2x_2 = 1 \\ 2x_1 + 2x_2 = 3 \end{cases} \quad \text{ou} \quad \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Resolução de Sistemas de Equações Lineares

b) Muitas soluções, como no caso seguinte em que as duas equações são equivalentes:

$$\begin{cases} 2x_1 + 2x_2 = 1 \\ 4x_1 + 4x_2 = 2 \end{cases} \quad \text{ou} \quad \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

De entre as muitas soluções, é fácil chegar às três seguintes:

$$\begin{matrix} x_1 = 0 \\ x_2 = 0.5 \end{matrix} \; ; \quad \begin{matrix} x_1 = 0.5 \\ x_2 = 0 \end{matrix} \; ; \quad \begin{matrix} x_1 = 0.25 \\ x_2 = 0.25 \end{matrix}$$

Resolução de Sistemas de Equações Lineares

c) Uma solução única, como no caso seguinte em que essa solução é (x_1=0.5 ; x_2 = 0.5):

$$\begin{cases} 2x_1 + 2x_2 = 2 \\ 2x_1 + 4x_2 = 3 \end{cases} \quad \text{ou} \quad \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

- **Assumiremos, no que se segue, que o sistema de equações a resolver terá uma solução única**

Resolução de Sistemas de Equações Lineares

➢ Operações possíveis em sistemas de equações (sem alterar a solução)

i. **Trocar a ordem das equações (matricialmente corresponderá a trocar linhas de A e de b)**

ii. **Alterar os nomes das variáveis (matricialmente corresponderá a trocar colunas de A e a ordem das variáveis)**

iii. **Multiplicar equações por uma constante (matricialmente corresponderá a multiplicar linhas de A e b por uma constante**

iv. **Substituir uma equação pela soma dela com uma outra equação (matricialmente corresponderá a substituir uma linha de A e b pela soma dessa linha com uma outra)**

Resolução de Sistemas de Equações Lineares

- **Ilustração das operações (i) e (ii) com o exemplo:**

$$\begin{cases} 2x_1 + 3x_2 + 4x_3 = 5 \\ 3x_1 + 4x_2 + 5x_3 = 6 \\ 4x_1 + 5x_2 + 6x_3 = 7 \end{cases} \quad \text{ou} \quad \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 7 \end{bmatrix}$$

- **Algoritmos correspondentes, para o caso geral, admitindo a seguinte representação computacional para as matrizes:**

$A_{(NxN)}$

$b_{(Nx1)}$

Representados por um array bidimensional A, sendo:

$$A[i,j] = a_{ij} \quad ; j \neq N+1$$

$$A[i,N+1] = b_i$$

$x_{(Nx1)}$

Representados por um array X

Resolução de Sistemas de Equações Lineares

(i) Trocar a ordem das equações

- **No exemplo, trocando a 1ª com a 2ª equação, teremos:**

$$\begin{cases} 3x_1 + 4x_2 + 5x_3 = 6 \\ 2x_1 + 3x_2 + 4x_3 = 5 \\ 4x_1 + 5x_2 + 6x_3 = 7 \end{cases} \quad \text{ou} \quad \begin{bmatrix} 3 & 4 & 5 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 7 \end{bmatrix}$$

- **No caso geral**

(troca das linhas I e J)

```
Procedure TROCAMAT(A,X,N,I,J)
        N1 ← N+1
        DO FOR COL=1 TO N1
                TEMP ← A[I,COL]
                A[I,COL] ← A[J,COL]
                A[J,COL] ← TEMP
```

Resolução de Sistemas de Equações Lineares

(ii) Alterar os nomes das variáveis

- **Para o exemplo apresentado, vamos escrever as equações, considerando a seguinte tabela de conversão:**

Designação Original	Nova Designação
x_1	x_2'
x_2	x_3'
x_3	x_1'

Resolução de Sistemas de Equações Lineares

- **Teremos então:**

Sistema Original	Novo Sistema
$2x_1 + 3x_2 + 4x_3 = 5$	$4x_3 + 2x_1 + 3x_2 = 5$
$3x_1 + 4x_2 + 5x_3 = 6$	$5x_3 + 3x_1 + 4x_2 = 6$
$4x_1 + 5x_2 + 6x_3 = 7$	$6x_3 + 4x_1 + 5x_2 = 7$

Passará a designar-se por x_1' x_2' x_3'

Note-se que, se fosse resolvido o "Novo Sistema", era necessário apresentar a solução segundo as designações originais (por exemplo, o valor de x_2', nesse "Novo Sistema", deveria ser atribuído à variável original x_1)

Resolução de Sistemas de Equações Lineares

- **Em termos computacionais, procede-se do seguinte modo:**
  - ⇒ **a tabela de conversão é lida para um array NOVO, através das seguintes instruções:**

```
DO FOR K=1 TO N
    READ (ORIGINAL, NOVO [ORIGINAL])
```

sendo, para o exemplo: NOVO[1]=2; NOVO[2]=3; NOVO[3]=1

  - ⇒ **a tabela inversa, pode ser obtida do seguinte modo(preenchimento do array ORIG):**

```
DO FOR ORIGINAL=1 TO N
    ORIG[NOVO [ORIGINAL]] ← ORIGINAL
```

sendo, para o exemplo: ORIG[1]=3; ORIG[2]=1; ORIG[3]=2

Resolução de Sistemas de Equações Lineares

- **Vamos admitir que o array A contém a matriz original A, tendo então a seguinte constituição para o exemplo:**

	1	2	3
1	A[1,1]=**2**	A[1,2]=**3**	A[1,3]=**4**
2	A[2,1]=**3**	A[2,2]=**4**	A[2,3]=**5**
3	A[3,1]=**4**	A[3,2]=**5**	A[3,3]=**6**

Colunas da matriz original A

Linhas da matriz original A

- **Para processar a nova matriz A, podemos ainda utilizar o mesmo array A.**
  Para o efeito, bastará recorrer ao array ORIG

Resolução de Sistemas de Equações Lineares

- Por exemplo, para escrever por ordem os coeficientes da nova matriz A, utilizaremos o seguinte algoritmo:

```
ESCREVE_NOVAMAT(A,ORIG,N)
    DO FOR I=1 TO N
        DO FOR J=1 TO N
                WRITE(A[I,ORIG[J]])
        WRITE
```

I, J são linha e coluna da nova matriz A

- De facto, tendo em conta que ORIG=

3	1	2
1	2	3

  virá para o exemplo:

	1	2	3
1	A[1,3]=**4**	A[1,1]=**2**	A[1,2]=**3**
2	A[2,3]=**5**	A[2,1]=**3**	A[2,2]=**4**
3	A[3,3]=**6**	A[3,1]=**4**	A[3,2]=**5**

Colunas da nova matriz A

Linhas da nova matriz A

que corresponde à nova matriz A

Resolução de Sistemas de Equações Lineares

- **De modo semelhante, vamos admitir que o array X contém as variáveis pela ordem original (x_1, x_2, ...,x_n), vindo para o novo exemplo:**

1	2	3
X[1]=$\mathbf{x_1}$	X[2]=$\mathbf{x_2}$	X[3]=$\mathbf{x_3}$

Posição no vector original x

- **O algoritmo seguinte escreve as varáveis pela nova ordem:**

```
ESCREVE_NOVAX(X,ORIG,N)
    DO FOR I=1 TO N
        WRITE(X[ORIG[I]])
```

I é a posição no novo vector x

Resolução de Sistemas de Equações Lineares

- De facto, tendo em conta que ORIG= | 3 | 1 | 2 | virá para o exemplo:

3	1	2
1	2	3

Posição no novo vector x

1	2	3
X[3]=$\mathbf{x_3}$	X[1]=$\mathbf{x_1}$	X[2]=$\mathbf{x_2}$

Resolução de Sistemas de Equações Lineares

- **Métodos de resolução de sistemas de equações**
  - **A – Métodos Directos**
    - **Eliminação de Gauss (triangularização)**
    - **Inversão explícita (Método de Gauss-Jordan)**
  - **B – Métodos Iterativos**
    - **Método de Gauss-Seidel**
  - **C – Métodos de Factorização**
    - **Factorização LU**

Resolução de Sistemas de Equações Lineares

- **Os métodos serão ilustrados com o seguinte sistema de quatro equações lineares a quatro incógnitas (sendo A simétrica):**

$$\underbrace{\begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{b}}$$

RSEqsLineares - Métodos Directos

Eliminação de Gauss

- **Exemplo:**

$$\begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & -1 & -1 & 0 \\ 0 & 5/3 & -1/3 & 0 \\ 0 & 0 & 8/5 & -1 \\ 0 & 0 & 0 & 3/8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4/3 \\ 8/5 \\ 2 \end{bmatrix}$$

- **Resolução:** • Triangularização **(Matriz triangular superior)**

  \+

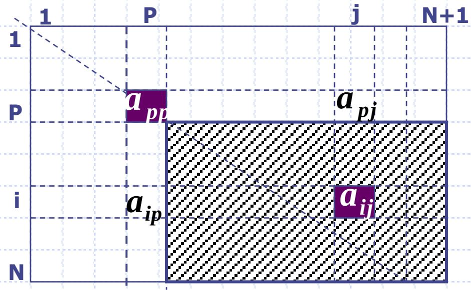
  • Substituição **("Back substitution")**

RSEqsLineares - Métodos Directos

- **Triangularização:**
  - **Incluir vector b na coluna N+1 da matriz A**
  - **(N-1) pivots a considerar na diagonal principal**

$$\left(a_{11}, a_{22}, \ldots, a_{N-1N-1}\right)$$

  - **Para cada pivot (a_{pp}) processar os elementos sombreados, utilizando a expressão indicada:**



$$a_{ij} \leftarrow a_{ij} - \frac{a_{ip}a_{pj}}{a_{pp}}$$

Elementos fora da linha e da coluna do pivot

RSEqsLineares - Métodos Directos

- **Substituição**
  - **Caso de N=4**

$$a_{44}x_4 = b_4 \rightarrow x_4$$
$$a_{33}x_3 + a_{34}x_4 = b_3 \rightarrow x_3$$
$$a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \rightarrow x_2$$
$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \rightarrow x_1$$

  - **Caso geral**

$$a_{ii}x_i \sum_{k=i+1}^{N} a_{ik}x_k = b_i$$

$$x_i = \frac{b_i - \sum_{k=i+1}^{N} a_{ik}x_k}{a_{ii}}$$

Nota: Nas expressões anteriores, os coeficientes já não são os originais visto que já houve a fase da triangularização

RSEqsLineares - Métodos Directos

- **Complexidade do algoritmo:**
  - **Para resolver um sistema de N equações:**
    - **São necessários cerca de $\frac{N^3}{3}$ operações aritméticas; há 3 ciclos imbricados no algoritmo (pivots; linhas; colunas) fase de triangularização**
    - **São necessários cerca de $\frac{N^2}{2}$ operações aritméticas; há 2 ciclos imbricados no algoritmo (há várias variáveis; cada variável é calculada por um somatório) fase de substituição**

RSEqsLineares - Métodos Directos

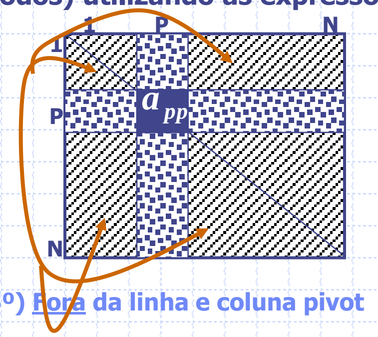
- **Método de Gauss-Jordan (Inversão explícita)**
  - **Exemplo:**

$$\begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 & 2/3 & 2/3 \\ 1/3 & 2/3 & 1/3 & 1/3 \\ 2/3 & 1/3 & 5/3 & 5/3 \\ 2/3 & 1/3 & 5/3 & 8/3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- **Resolução:** • **Inversão de A(A^{-1})**
  +
  • **Cálculo de X**

RSEqsLineares - Métodos Directos

- **Inversão**
  - **N pivots a considerar** $\left(a_{11}, a_{22}, \ldots, a_{NN}\right)$
  - **Para cada pivot (a_{pp}) processar os elementos sombreados (todos) utilizando as expressões seguintes:**



1º) Pivot $a_{pp} \leftarrow \frac{1}{a_{pp}}$

2º) Coluna pivot $a_{ip} \leftarrow a_{ip} \cdot a_{pp}$

3º) Fora da linha e coluna pivot $a_{ij} \leftarrow a_{ij} - a_{ip} \cdot a_{pj}$

4º) Linha pivot $a_{pj} \leftarrow -a_{pj} \cdot a_{pp}$

RSEqsLineares - Métodos Directos

- **Cálculo de X**

$$x_i = \sum_{k=1}^{N} a_{ik} b_k$$

Nota: Nesta expressão os coeficientes a_{ik} já não são os originais!

- **Notas importantes:**
  - **É preservada a simetria da matriz original**
  - **A matriz A^{-1} é cheia, mesmo que A seja esparsa**
- **Complexidade do algoritmo:**

  Para resolver um sistema de N equações:
  - **Cerca de N^3 operações aritméticas na fase de inversão**
  - **Cerca de N^2 operações aritméticas na 2ª fase (cálculo de X)**

RSEqsLineares - Métodos Iterativos

- **Método de Gauss-Seidel**
  - **Apenas terá algum interesse para sistemas de equações de pequena dimensão (N pequeno)**
  - **Formulação:**

$$\begin{cases} x_1 = \left(b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1N}x_N\right) \Big/ a_{11} \\ x_2 = \left(b_2 - a_{21}x_1 - a_{23}x_3 - \ldots - a_{2N}x_N\right) \Big/ a_{22} \\ \ldots \\ x_i = \left(b_i - a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{ii-1}x_{i-1} - a_{ii+1}x_{i+1} - \ldots - a_{iN}x_N\right) \Big/ a_{ii} \\ \ldots \\ x_N = \left(b_N - a_{N1}x_1 - a_{N2}x_2 - \ldots - a_{NN-1}x_{N-1}\right) \Big/ a_{NN} \end{cases}$$

RSEqsLineares - Métodos Iterativos

- **Estas expressões resultam directamente da forma canónica de um sistema de equações:**

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1N}x_N = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2N}x_N = b_2 \\ \ldots \\ a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii-1}x_{i-1} + a_{ii}x_i + a_{ii+1}x_{i+1} + \ldots + a_{iN}x_N = b_i \\ \ldots \\ a_{N1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{NN}x_N = b_N \end{cases}$$

RSEqsLineares - Métodos Iterativos

- **Algoritmo:**

  1- Arbitrar solução inicial (Por Ex.: $x_i = 0; \forall_i$)

  2- Actualizar $x_1, x_2 \ldots x_i \ldots x_N$ pelas expressões do slide anterior, entretanto sempre com os valores de x_i mais actuais (Por Ex.:, na 2ª expressão já entramos com o valor de x_1 calculado pela 1ª expressão)

  3- Testar convergência:

$$\left| \frac{x_i^{actual} - x_i^{anterior}}{x_i^{actual}} \right| < Tolerância; \forall_i$$

RSEqsLineares - Métodos Iterativos

4- Terminar se o teste de convergência for satisfatório ou regressar ao passo 2 em caso contrário

- **Erro: Com a aplicação deste método, não determinamos uma solução exacta mas antes uma solução aproximada. Tal não constitui problema visto que o erro (tolerância) pré-definido não terá sido excedido**

RSEqsLineares -Métodos Factorização

Nestes métodos, a matriz A (ou a sua inversa A^{-1}) é decomposta num conjunto de factores. É com base nesta factorização que, posteriormente, se procede ao cálculo de X

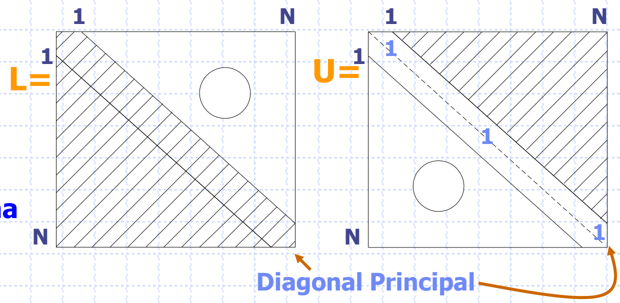
Factorização LU

- **A matriz A é decomposta no produto: A = LU**

em que

L – matriz triangular superior

U – matriz triangular superior com elementos unitários na diagonal principal



Deste modo, é possível memorizar L e U em apenas N x N posições

RSEqsLineares -Métodos Factorização

- **Exemplo:**

A=

3	-1	-1	0
-1	2	0	0
- 1	0	2	-1
0	0	-1	1

L=

3			
-1	5/3		
- 1	-1/3	8/5	
0	0	-1	3/8

U=

1	-1/3	-1/3	0
	1	-1/5	0
		1	-5/8
			1

L **U**

3	-1/3	-1/3	0
-1	5/3	-1/5	0
- 1	-1/3	8/5	-5/8
0	0	-1	3/8

L e U memorizadas em 4x4 posições

É não simétrica

RSEqsLineares -Métodos Factorização

- **Resolução:**
  - **Factorização propriamente dita**

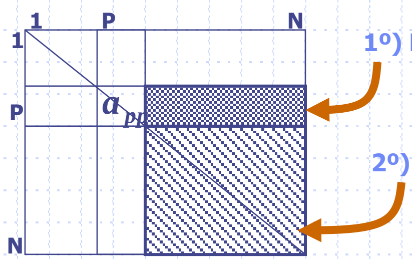
    +
  - **Determinação de X**
- **Factorização:**

  As matrizes L e U não são obtidas individualmente e depois colocadas em N x N posições!

  Em vez disso, as matrizes L e U são obtidas simultaneamente, como se fosse uma matriz única (N x N), mediante um conjunto de operações de pivotagem efectuadas sobre a matriz original A

RSEqsLineares -Métodos Factorização

- **(N –1) pivots a considerar na diagonal principal ($a_{11}, a_{22}, \ldots, a_{N-1\ N-1}$)**
- **Para cada pivot (a_{pp}), processar os elementos sombreados, utilizando as expressões seguintes:**



1º) Linha pivot

$$a_{pj} \leftarrow a_{pj} / a_{pp}$$

2º) Fora da linha e coluna pivot

$$a_{ij} \leftarrow a_{ij} - a_{ip} \cdot a_{pj}$$

RSEqsLineares -Métodos Factorização

- **Como resultado destas operações, obteremos uma matriz que contém, implicitamente, L (ocupando a parte triangular inferior e a diagonal principal) e U (ocupando a parte triangular superior – pelo que a diagonal principal, apenas constituída por 1's, não é memorizada)**

- **Determinação de X:**
  - $$\underbrace{LU}_{A}\,X=b \iff \begin{cases} L\,Y = b & (1) \\ U\,X = Y & (2) \end{cases}$$

RSEqsLineares -Métodos Factorização

- **Determinação em duas fases:**

  1ª) Determinar Y, mediante resolução de (1) por substituição ("forward substitution")

  ⬇

  Determina sucessivamente $y_1, y_2, \ldots, y_N$

  2ª) Determinar X, mediante resolução de (2) por substituição ("backward substitution")

  ⬇

  Determina sucessivamente $x_N, \ldots, x_2, x_1$

RSEqsLineares -Métodos Factorização

- **PIVOTAGEM**
  - **Os métodos anteriores falham se o pivot a seleccionar numa dada iteração for zero. Do mesmo modo, pode haver problemas numéricos (erros apreciáveis) se os pivots tiverem valores muito baixos**
  - **Nestes casos sugere-se a utilização de um dos seguintes métodos de selecção de elementos pivot:**

M1 Procurar na coluna pivot (em todas as linhas que ainda não tenham sido usadas como linhas pivot) o elemento de maior valor absoluto. Trocar a linha que contém este elemento com a linha pivot original

RSEqsLineares -Métodos Factorização

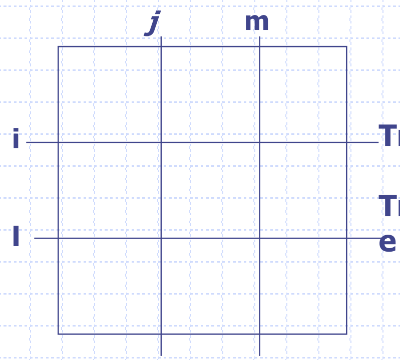
M2 Procurar em todas as linhas e colunas que ainda não tenham sido usadas como pivot; seleccionar o maior dos elementos; trazer este elemento para a posição desejada

M3 Procurar de entre todos os elementos da diagonal principal aquele que tiver o maior valor; trocar linhas e colunas por forma a trazer este elemento para a posição desejada

M2 é o método que permite obter melhor precisão, sendo, no entanto, aquele que mais tempo consome

RSEqsLineares -Métodos Factorização

Por exemplo, colocar a_{ij} na posição (l,m):



Trocar linha i com linha l

Trocar coluna j com coluna m
e as respectivas varáveis

Nota: qualquer elemento de uma matriz pode ser colocado em qualquer posição mediante troca adequada de linhas e de colunas

Próxima aula

- Ordenação