Universidade do Minho
Escola de Ciências

Departamento de Matemática

Sistemas de equações lineares

- Método de Eliminação de Gauss
- Decomposição LU
- Condicionamento de um sistema
- Métodos iterativos

Métodos diretos *vs* iterativos

Muitos problemas de matemática aplicada passam pela resolução de sistemas de equações lineares, geralmente de grande dimensão. Os métodos numéricos para a resolução de sistemas são, geralmente, agrupados em duas grandes classes:

- Métodos Diretos
  - a solução é determinada num número finito de operações aritméticas;
  - na ausência de erros de arredondamento, tais métodos produziriam a solução exata;
  - na prática, os erros de arredondamento são inevitáveis, pelo que os métodos conduzem apenas a soluções aproximadas.
- Métodos Iterativos
  - a partir de uma aproximação inicial para a solução, gera-se uma sequência de aproximações que, sob certas condições, converge para a solução do problema;
  - na prática, o processo será interrompido ao fim de um certo número de iterações.

Neste curso estudaremos apenas os Métodos Diretos

Revisões de Álgebra Linear

Sistema de n equações lineares com n incógnitas:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ \qquad\qquad \cdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$$

x_j - incógnitas

a_{ij} - coeficientes de x_j

b_i - termo independente

Forma matricial:

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \qquad \left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_2 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{array}\right)$$

$$A\,\mathbf{x} = \mathbf{b} \qquad (A \mid \mathbf{b})$$

$A = (a_{ij})$ matriz simples do sistema, $(A \mid \mathbf{b})$ matriz ampliada do sistema,

$\mathbf{x} = (x_i)$ incógnitas, $\mathbf{b} = (b_i)$ termos independentes.

➲ Operações elementares sobre as equações de um sistema (linhas de uma matriz)

OE1: Troca da ordem de duas equações (linhas).

OE2: Multiplicação de uma equação (linha) por um número diferente de zero.

OE3: Soma de uma equação (linha) com um múltiplo de outra equação (linha).

Teorema

Se, sobre as equações de um sistema (linhas de uma matriz), efetuarmos um número finito de operações elementares, obtemos um sistema equivalente ao (uma matriz equivalente por linhas à) inicial.

Nota: No que se segue, supomos que o sistema tem solução única.

Sistemas Triangulares

Sistema Triangular Inferior

A matriz do sistema pode escrever-se como $A = L = (\ell_{ij})$, com $\ell_{ij} = 0,\ j > i$, i.e. a matriz ampliada do sistema é:

$$(L|b) = \left(\begin{array}{ccccc|c} \ell_{11} & 0 & 0 & \dots & 0 & b_1 \\ \ell_{21} & \ell_{22} & 0 & \dots & 0 & b_2 \\ \ell_{31} & \ell_{32} & \ell_{33} & \dots & 0 & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \dots & \ell_{nn} & b_n \end{array}\right), \qquad \ell_{ii} \neq 0$$

Método de Substituição Direta

- $x_1 = \frac{b_1}{\ell_{11}}$;
- $x_2 = \frac{b_2 - \ell_{21} x_1}{\ell_{22}}$;
- $x_3 = \frac{b_3 - \ell_{31} x_1 - \ell_{32} x_2}{\ell_{33}}$,

  $\vdots$
- $x_n = \frac{b_n - \ell_{n1} x_1 - \ell_{n2} x_2 - \cdots - \ell_{n,n-1} x_{n-1}}{\ell_{nn}} = \frac{1}{\ell_{nn}} \left(b_n - \sum_{j=1}^{n-1} \ell_{nj} x_j\right)$.

Substituição direta - algoritmo — subDireta

Dados: L - matriz triangular inferior

b - vetor dos termos independentes

Resultado: b - vetor solução do sistema

$b(1) = \dfrac{b(1)}{L(1,1)}$;

b deve ser um vetor coluna

for $i = 2 : n$

$$b(i) = \frac{b(i) - L(i, 1:i-1)b(1:i-1)}{L(i,i)};$$

end

Nº de operações: $\frac{n^2+n}{2}$ (MD) e $\frac{n^2-n}{2}$ (AS)

Notas:

- Para não aumentar as necessidades de armazenamento, os valores $x(i)$ foram sobrepostos aos valores de $b(i)$.
- $L(i, 1:i-1)b(1:i-1) = \ell_{i1}b_1 - \ell_{i2}b_2 - \cdots - \ell_{i,i-1}b_{i-1}$.

➲ Sistema Triangular Superior

Neste caso, a matriz do sistema pode escrever-se como $A = U = (u_{ij})$, com $u_{ij} = 0,\ j < i$. A resolução do sistema faz-se agora por substituição inversa.

Substituição inversa - algoritmo — subInversa

Dados: U - matriz triangular superior

b - vetor dos termos independentes

Resultado: b - vetor solução do sistema

$$b(n) = \frac{b(n)}{U(n,n)};$$

b deve ser um vetor coluna

for $i = n-1 : -1 : 1$

$$b(i) = \frac{b(i) - U(i, i+1:n)b(i+1:n)}{U(i,i)};$$

end

Nº de operações: $\frac{n^2+n}{2}$ (MD) e $\frac{n^2-n}{2}$ (AS)

Método de Gauss

Relembremos que a ideia do chamado Método de Gauss é fazer uso das operações elementares sobre linhas para converter a matriz ampliada do sistema $(A|b)$ numa matriz da forma $(U|\beta)$, onde U é uma matriz triangular superior - processo de eliminação de Gauss, resolvendo-se posteriormente o sistema correspondente por substituição inversa.

$$(A|b) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right)$$

Método de Gauss

- Eliminação Gaussiana

$$(A|b) \xrightarrow[\text{linhas}]{} (U|\beta), \quad U \text{ triangular superior}$$

- Substituição inversa

  Resolução do sistema triangular superior $Ux = \beta$, por substituição inversa.

Método de eliminação de Gauss - MEG

- **Passo 1** Para $i = 2, \ldots, n$, substituir a linha i pela sua soma com a linha 1 multiplicada por $m_{i1} = -a_{i1}/a_{11}$,

obtendo-se uma nova matriz

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right),$$

onde

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij} - m_{i1} a_{1j};\ i, j = 2, \ldots, n \\ b_i^{(1)} &= b_i - m_{i1} b_1;\ i = 2, \ldots, n. \end{aligned}$$

- **Passo 2** Para $i = 3, \ldots, n$, substituir a linha i pela sua soma com a linha 2 multiplicada por

$$-m_{i2} = -a_{i2}^{(1)} / a_{22}^{(1)};\ i = 3, \ldots, n,$$

vindo, então

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right),$$

onde

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i2} a_{2j}^{(1)};\ i, j = 3, \ldots, n \\ b_i^{(2)} &= b_i^{(1)} - m_{i2} b_2^{(1)};\ i = 3, \ldots, n. \end{aligned}$$

- **Passo k ($k \leq n-1$).** Para $i = k+1, \ldots, n$, substituir a linha i pela sua soma com a linha k multiplicada por

$$-m_{ik} = -a_{ik}^{(k-1)} / a_{kk}^{(k-1)}$$

obtendo-se então uma matriz

$$\left(\begin{array}{cccccccc|c}
a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} & b_1 \\
0 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & a_{2,k+1}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)}
\end{array}\right)$$

com

$$\begin{aligned}
a_{ij}^{(k)} &= a_{ij}^{(k-1)} - m_{ik} a_{kj}^{(k-1)};\ i,j = k+1, \ldots, n \\
b_i^{(k)} &= b_i^{(k-1)} - m_{ik} b_k^{(k-1)};\ i = k+1, \ldots, n.
\end{aligned}$$

Ao fim de $n-1$ passos a matriz do sistema estará reduzida à forma triangular superior, isto é, ter-se-á a seguinte matriz ampliada

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{array}\right)$$

O sistema pode, então, resolver-se por substituição inversa.

Notas:

- Os números $m_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)}$ são chamados multiplicadores.
- Os elementos $a_{kk}^{(k-1)}$ são chamados pivots.
- Nesta fase, estamos a assumir que os *pivots* (que são usados como divisores) são não nulos.

Algoritmo do MEG — MEG

Dados: A - matriz de ordem n

b - vetor dos termos independentes

Resultado: A - matriz triangular superior

b - vetor dos termos independentes modificado

---

for $k = 1 : n - 1$ — contagem do passo

  mult $= A(k+1:n, k)/A(k,k)$; — vetor dos multiplicadores do passo k

  $A(k+1:n, k+1:n) = A(k+1:n, k+1:n) -$ mult $* A(k, k+1:n)$;

  $b(k+1:n) = b(k+1:n) -$ mult $* b(k)$;

end — Nº de operações: $\frac{2n^3+3n^2-5n}{6}$ (MD) e $\frac{n^3-n}{3}$ (AS)

Comentários ao algoritmo:

- Os sucessivos valores $a_{ij}^{(k)}$ e $b_i^{(k)}$ calculados durante o processo são sobrepostos aos valores anteriores.
- A matriz de saída A não é, na realidade, uma matriz triangular superior[1] (os elementos abaixo da diagonal não foram tornados nulos).
- Caso seja necessário, os diversos multiplicadores calculados podem ser armazenados na "parte triangular inferior"de A, fazendo as alterações:

$A(k+1:n,k) = A(k+1:n,k)/A(k,k);$ vetor dos multiplicadores do passo k

$A(k+1:n,k+1:n) = A(k+1:n,k+1:n) - A(k+1:n,k) * A(k,k+1:n);$

$b(k+1:n) = b(k+1:n) - A(k+1:n,k) * b(k)\ ;$

[1] Embora seja muito fácil, usando uma função pré-definida no MATLAB (qual?), obter essa matriz.

Instabilidade do MEG

Se, ao iniciar o passo k do método de eliminação de Gauss, tivermos $a_{kk}^{(k-1)} = 0$ (pivot nulo), o processo falha.

Vamos ver agora que, se $a_{kk}^{(k-1)}$ for muito pequeno (relativamente aos outros elementos), poderá haver problemas de instabilidade.

- O sistema

$$\begin{cases} \varepsilon x_1 + x_2 = 1 \\ x_1 - x_2 = 0 \end{cases}$$

onde $\varepsilon = 10^{-12}$, tem, em aritmética exata, a solução

$$x_2 = = \frac{1}{1+\epsilon}$$

$$x_1 = \frac{1}{1+\epsilon}$$

Note-se que $x_1 = x_2 \approx 1$.

- Considere-se agora a resolução do mesmo sistema numa máquina com sistema de numeração $F(10, 12, -99, 99)$. Como, nesta máquina,

$$fl(-1 - \frac{1}{\varepsilon}) = fl(-(0.1 + 10^{-13}) \times 10^{13}) = -0.1 \times 10^{13} = -\frac{1}{\varepsilon},$$

ao fim do primeiro passo de eliminação ter-se-á a seguinte matriz

$$\left(\begin{array}{cc|c} \varepsilon & 1 & 1 \\ 0 & -1/\varepsilon & -1/\varepsilon \end{array}\right)$$

donde se obtém, de imediato, $\tilde{x_2} = 1$ e $\tilde{x_1} = 0$!!.

Um pivot muito pequeno $\varepsilon = 10^{-12}$ originou um multiplicador muito grande $1/\varepsilon = 10^{12}$ e houve perda de algarismos significativos no cálculo de $a_{22}^{(1)}$.

O coeficiente $-(10^{12} + 1)$ foi substituído por -10^{12}. O erro absoluto de uma unidade no cálculo deste coeficiente (a que corresponde um erro relativamente pequeno) é grande relativamente à grandeza dos outros coeficientes do sistema e à grandeza da solução, vindo, por isso, a afetar grandemente o resultado obtido.

Escolha de pivot

- Consideremos novamente a resolução do sistema anterior em $F(10, 12, -99, 99)$, mas, antes de iniciar a eliminação troquemos a ordem das equações, isto é, consideremos o sistema

$$\left(\begin{array}{cc|c} 1 & -1 & 0 \\ \varepsilon & 1 & 1 \end{array}\right)$$

Após a eliminação, tem-se

$$\left(\begin{array}{cc|c} 1 & -1 & 0 \\ 0 & 1 & 1 \end{array}\right)$$

Com efeito, $fl(1+\varepsilon) = 1$, já que ε é inferior à unidade de erro de arredondamento da máquina. Neste caso, a solução virá

$$\tilde{x_1} = \tilde{x_2} = 1,$$

a qual não difere muito da solução exata.

Seja $a_{kk}^{(k-1)}$ o pivot obtido no passo $k-1$. Se

- $a_{kk}^{(k-1)} = 0 \Longrightarrow$ Método não pode prosseguir (Porquê?).
- $|a_{kk}^{(k-1)}|$ "pequeno" $\Longrightarrow$ Multiplicadores podem ser grandes $\Longrightarrow$ perda de algarismos significativos $\Longrightarrow$ INSTABILIDADE !

Para evitar os problemas acima referidos, o algoritmo deve ser implementado acompanhado da chamada escolha parcial de pivot:

Escolha parcial de pivot

No início do passo k,

- determinar $\max_{k \leq i \leq n} |a_{ik}^{(k-1)}|$;
- sendo $\max_{k \leq i \leq n} |a_{ik}^{(k-1)}| = |a_{\ell,k}^{(k-1)}|$, proceder à troca das linhas ℓ e k.

Note-se que a técnica anterior garante que todos os multiplicadores têm módulo não superior a 1.

Algoritmo do Método de Gauss com escolha parcial de pivot — metGauss

Dados: A - matriz de ordem n

b - vetor dos termos independentes

Resultado: b - solução do sistema

---

for $k = 1 : n-1$ — contagem do passo

Escolha de pivot

Determinar ℓ tal que $|A(\ell, k)| = \max |A(k:n, k)|$.

$A(k, k:n) \leftrightarrow A(\ell, k:n)$; — trocar linhas k e ℓ

$b(k) \leftrightarrow b(\ell)$;

Eliminação de Gauss

mult $= A(k+1:n, k)/A(k, k)$; — vetor dos multiplicadores do passo k

$A(k+1:n, k+1:n) = A(k+1:n, k+1:n) -$ mult $* A(k, k+1:n)$;

$b(k+1:n) = b(k+1:n) -$ mult $* b(k)$;

end

Substituição inversa

b=subInversa($\texttt{triu}(A), b$)

Aplicações

- Cálculo do determinante de A

Se U é uma matriz triangular superior equivalente a A, então

$$\det A = (-1)^{\nu} \det U = (-1)^{\nu} u_{11} u_{22} \dots u_{nn},$$

onde ν designa o número total de trocas de linhas efectuadas no PEG.

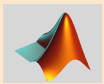
- Cálculo da inversa de A

Se A é invertível e $A^{-1} = (c_1|c_2|\cdots|c_n)$, então

$$A A^{-1} = I \iff A\,c1 = e_1, A\,c_2 = e_2, \dots, A\,c_n = e_n,$$

onde e_i designa a i-ésima coluna da matriz identidade. Para determinar a i-ésima coluna da matriz inversa de A pode resolver-se o sistema $Ac_i = e_i$. Assim, o cálculo de A^{-1} pode fazer-se, resolvendo n sistemas de equações lineares (todos eles com A como matriz simples).

Funções Matlab



Funções pré-definidas

```
    \      diag    det   eye    hilb    inv

linsolve   max   ones   tril   triu   zeros
```

Funções Toolbox AN

```
subDireta   - resolução de um sistema triangular inferior,
              por substituição direta

subInversa  - resolução de um sistema triangular superior,
              por substituição inversa

metGauss    - resolução de um sistema pelo método de Gauss
```

Decomposição LU

Consideremos a aplicação do método de Gauss sem escolha de pivot na resolução do sistema $Ax = b$, com

$$A = \begin{pmatrix} 4 & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$

Se, no processo de eliminação de Gauss, os multiplicadores forem guardados na parte inferior da matriz A, no final deste processo, esta terá a forma (Verifique!)

$$A = \begin{pmatrix} 4 & -9 & 2 \\ 0.5 & 0.5 & 3 \\ -0.25 & -0.5 & 4 \end{pmatrix}$$

Note-se que

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.25 & -0.5 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 4 & -9 & 2 \\ 0 & 0.5 & 3 \\ 0 & 0 & 4 \end{pmatrix}}_{U} = A$$

A eliminação Gaussiana (sem escolha de pivot) aplicada a uma matriz A reduz essa matriz a uma matriz U triangular superior[2] e, se formarmos uma matriz triangular inferior L com diagonal unitária e tendo, na parte estritamente triangular inferior, os multiplicadores usados no processo de redução, então estas matrizes L e U são tais que

$$A = LU.$$

Chama-se **decomposição LU** de uma matriz A à fatorização de A no produto de duas matrizes L e U tais que:

- L é triangular inferior com diagonal unitária;
- U é triangular superior.

[2] Estamos, naturalmente a admitir que a eliminação pode prosseguir até final...

Teorema - Unicidade

Seja A uma matriz quadrada de ordem n invertível. Então, se A admite uma decomposição LU, essa decomposição é única.

Teorema - Existência

Seja A uma matriz quadrada de ordem n. Se todas as submatrizes principais A_k contidas nas primeiras k linhas e k colunas de A; $k = 1, \ldots, n-1$, são invertíveis, então A admite uma decomposição LU. Além disso, se A for invertível, a condição anterior é uma condição necessária para que A tenha uma decomposição LU.

Embora a eliminação Gaussiana e a decomposição LU sejam processos equivalentes, é possível obter a decomposição LU de A, usando um procedimento computacional diferente para a determinação das matrizes L e U.

Algoritmo de Doolitle

Seja

$$A = \underbrace{\begin{pmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ \ell_{k1} & \dots & 1 & & \\ \vdots & & \vdots & \ddots & \\ \ell_{n1} & \dots & \ell_{ni} & \dots & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} u_{11} & \dots & u_{1j} & \dots & u_{1n} \\ & \ddots & \vdots & & \vdots \\ & & u_{jj} & \dots & u_{jn} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}}_{U}.$$

Igualando os elementos correspondentes de ambos os lados da igualdade anterior, tem-se

$$\begin{aligned} a_{kj} &= \ell_{k1}u_{1j} + \ldots + \ell_{k,k-1}u_{k-1,j} + \boxed{u_{kj}}, \quad \text{se } k \leq j, \\ a_{kj} &= \ell_{k1}u_{1j} + \ldots + \boxed{\ell_{kj}}u_{jj}, \quad \text{se } k > j. \end{aligned}$$

As equações anteriores podem ser usadas para obter as incógnitas $\ell_{kj}; k > j$ e $u_{kj}; k \leq j$, desde que se ordenem estas incógnitas convenientemente. Suponhamos que já determinámos as primeiras $k-1$ linhas de U e as primeiras $k-1$ colunas de L; podemos então determinar os elementos assinalados com a caixa, os quais formam a $k-$ésima linha de U e a $k-$ésima coluna de L:

Algoritmo de Doolitle

$$u_{kj} = a_{kj} - \ell_{k1}u_{1j} - \ell_{k2}u_{2j} - \ldots - \ell_{k,k-1}u_{k-1,j}, \quad k \leq j,$$
$$\ell_{kj} = \left(a_{kj} - \ell_{k1}u_{1j} - \ell_{k2}u_{2j} - \ldots - \ell_{k,j-1}u_{j-1,j}\right)/u_{jj}, \quad k > j.$$

Exemplo

$$A = \begin{pmatrix} 4 & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{pmatrix} \qquad \begin{pmatrix} 4 & -9 & 2 \\ 0.5 & 0.5 & 3 \\ -0.25 & -0.5 & 4 \end{pmatrix}$$

- Linha 1 de U:

  $u_{1j} = a_{1j};\ j = 1, 2, 3$ $\qquad$ $u_{11} = 4;\ u_{12} = -9, u_{13} = 2$

- Coluna 1 de L:

  $\ell_{k1} = a_{k1}/u_{11};\ k = 2, 3$ $\qquad$ $\ell_{21} = \frac{1}{2};\ \ell_{31} = -\frac{1}{4}$

- Linha 2 de U:

  $u_{22} = a_{22} - \ell_{21}u_{12}$ e $u_{23} = a_{23} - \ell_{21}u_{13};$ $\qquad$ $u_{22} = \frac{1}{2};\ u_{23} = 3$

- Coluna 2 de L:

  $\ell_{32} = (a_{32} - \ell_{31}u_{12})/u_{22};$ $\qquad$ $\ell_{32} = -\frac{1}{2}$

- Linha 3 de U:

  $u_{33} = a_{33} - \ell_{31}u_{13} - \ell_{32}u_{23};$ $\qquad$ $u_{33} = 4$

Resolução de sistemas usando a decomposição LU

Suponhamos que conhecemos a decomposição LU de uma matriz A e que pretendemos resolver um sistema da forma $Ax = b$.Então:

$$Ax = b \iff (LU)x = b \iff L\underbrace{(Ux)}_{y} = b \iff \begin{cases} Ly = b \\ Ux = y \end{cases}$$

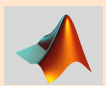
➲ A resolução do sistema $Ax = b$ pode fazer-se começando por resolver o sistema triangular inferior $Ly = b$, por substituição direta, e resolvendo, em seguida, o sistema triangular superior $Ux = y$, por substituição inversa.

➲ Este processo é particularmente útil quando pretendemos resolver vários sistemas de equações que tenham todos a mesma matriz (simples); por exemplo, este será o caso quando procurarmos determinar a inversa de uma matriz A.

Exemplo

Consideremos novamente o sistema $Ax = b$, com

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.25 & -0.5 & 1 \end{pmatrix} \begin{pmatrix} 4 & -9 & 2 \\ 0 & 0.5 & 3 \\ 0 & 0 & 4 \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$



Resolução sistemas

```
>> l=[1 0 0;0.5 1 0;-0.25 -0.5 1];
<< u=[4 -9 2;0 0.5 3;0 0 4];
>> b=[2;3;1];
>> y=subDireta(l,b)

y =
    2.0000
    2.0000
    2.5000
>> x=subInversa(u,y)

x =
    0.7500
    0.2500
    0.6250
```

Escolha de Pivot

Ao descrevermos a equivalência entre a eliminação de Gauss e decomposição LU de uma matriz A, admitimos não haver necessidade de troca de linhas.

Se efetuarmos escolha parcial de pivot no algoritmo de eliminação (o que é necessário, por questões de estabilidade), então ao formarmos as matrizes L e U, no final obter-se-á, não a decomposição LU da matriz A, mas sim da matriz A com as suas linhas trocadas (correspondendo às trocas que foram efectuadas na redução), ou seja, obter-se-á

$$LU = PA,$$

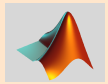
onde P é uma matriz de permutação.

- Uma matriz P é uma matriz de permutação se for obtida da matriz identidade I por troca de linhas; a pré-multiplicação de uma matriz A por uma matriz de permutação P "troca", em A, as mesmas linhas que foram trocadas para obter P a partir de I; uma matriz de permutação é invertível, tendo-se $P^{-1} = P$.
- Sendo $Ax = b$ e tendo-se $PA = LU$, vem

$$\begin{aligned} Ax = b \quad &\iff \quad P(Ax) = Pb \iff (PA)x = Pb \\ &\iff \quad (LU)x = Pb \iff \begin{cases} Ly = Pb \\ Ux = y. \end{cases} \end{aligned}$$

- Assim, se dispusermos da decomposição LU de PA, podemos novamente encontrar a solução do sistema $Ax = b$ resolvendo dois sistemas triangulares, devendo, neste caso, começar por resolver-se o sistema $Ly = Pb$.

Exemplos



FUNÇÕES DO MATLAB

`\` `lu`

➠ Sem pivotagem

```
>> a=[4 -9 2;2 -4 4;-1 2 2];
>> [L U P]=lu(a)
L =
    1.0000         0         0
    0.5000    1.0000         0
   -0.2500   -0.5000    1.0000
U =
    4.0000   -9.0000    2.0000
         0    0.5000    3.0000
         0         0    4.0000
P =
     1     0     0
     0     1     0
     0     0     1
>> b=[2;3;1];y=L\b;x=U\y
x =
    0.7500
    0.2500
    0.6250
```

Com pivotagem

```
>> a=[2 -4 4;4 -9 2;-1 2 2]
a =
     2    -4     4
     4    -9     2
    -1     2     2

>> [L U P]=lu(a)
L =
    1.0000         0         0
    0.5000    1.0000         0
   -0.2500   -0.5000    1.0000

U =
    4.0000   -9.0000    2.0000
         0    0.5000    3.0000
         0         0    4.0000


P =
     0     1     0
     1     0     0
     0     0     1
```

```
>> b=[3;2;1]
b =
     3
     2
     1

>> y=L\ (P*b)

y =
    2.0000
    2.0000
    2.5000

>> x=U\y
x =
    0.7500
    0.2500
    0.6250
```

Condicionamento de um sistema

Normas vetoriais

Seja $X = \mathbb{R}^n$. Uma aplicação $\|\cdot\| : X \longrightarrow \mathbb{R}$ diz-se uma norma vetorial se satisfizer as seguintes propriedades:

N1 $\forall x \in X,\ \|x\| \geq 0$ e $\|x\| = 0$ se e só se $x = 0$.

N2 $\forall x \in X, \forall \alpha \in \mathbb{R},\ \|\alpha x\| = |\alpha|\|x\|$.

N3 $\forall x, y \in X,\ \|x + y\| \leq \|x\| + \|y\|$.

Exemplos: As normas vetoriais mais frequentemente utilizadas são:

$\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (norma 1)

$\|x\|_2 = \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2}$ (norma 2 ou Euclidiana)

$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ (norma ∞ ou do máximo)

Normas matriciais

➲ Seja $X = \mathbb{R}^{n\times n}$. Uma aplicação de X em $\mathbb{R}$ que satisfaça as propriedades *N1–N3* diz-se uma norma matricial.

➲ Uma norma matricial diz-se submultiplicativa se

$$\|AB\| \leq \|A\|\,\|B\|, \quad \forall A, B \in X$$

.

➲ Seja $\|\cdot\|_v$ uma determinada norma vetorial. A função $\|\cdot\|_M : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}$ definida por

$$\|A\|_M = \max_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{\|Ax\|_v}{\|x\|_v}$$

define uma norma matricial, dita induzida pela ou subordinada à norma vetorial considerada.

Pode provar-se que normas matriciais subordinadas às normas vetoriais $\|\cdot\|_1$, $\|\cdot\|_2$ e $\|\cdot\|_\infty$ [3] são dadas por

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|$$

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|$$

$$\|A\|_2 = \left(\rho(A^{\mathrm{T}}A)\right)^{1/2},$$

onde $\rho(A^{\mathrm{T}}A)$ designa o raio espetral da matriz $A^{\mathrm{T}}A$.

Todas estas normas satisfazem, naturalmente, a seguinte condição de compatibilidade com a correspondente norma vetorial

$$\|Ax\|_p \leq \|A\|_p \|x\|_p, \forall A \in \mathbb{R}^{n\times n}, \forall x \in \mathbb{R}^n; p = 1, 2, \infty.$$

[3] Denotamos pelo mesmo símbolo a norma vetorial e a correspondente norma matricial, sendo claro pelo contexto de que norma se trata.

Consideremos um sistema

$$Ax = b$$

e suponhamos que $b \neq 0$ e que A é não singular. Sejam $\delta A \in \mathbb{R}^{n \times n}$ e $\delta b \in \mathbb{R}^n$, respetivamente, uma matriz e um vetor "de perturbação" dos dados do problema, e seja $\tilde{x}$ a solução do problema perturbado, isto é, seja

$$(A + \delta A)\tilde{x} = b + \delta b.$$

Seja δ_x o erro na solução induzido pela perturbação dos dados, isto é, seja

$$\delta_x = \tilde{x} - x.$$

Teorema

Se, para uma qualquer das normas consideradas, for $\|A^{-1}\|\,\|\delta A\| < 1$, *então*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right),$$

onde

$$\mathrm{cond}(A) := \|A\|\,\|A^{-1}\|.$$

A quantidade $\mathrm{cond}(A)$ diz-se número de condição da matriz A (relativamente à norma considerada).

- Se $\|A^{-1}\|\,\|\delta A\| \ll 1$, então

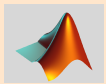
$$1 - \mathrm{cond}(A)\frac{\|\delta A\|}{\|A\|} = 1 - \|A^{-1}\|\,\|\delta A\| \approx 1,$$

pelo que

$$\frac{\|\delta x\|}{\|x\|} \lesssim \mathrm{cond}(A)\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\,|}{\|b\|}\right).$$

Vemos, assim, que:

- um número de condição pequeno é garantia de que o sistema é bem condicionado, isto é, é pouco sensível às perturbações nos dados;
- um número de condição grande é indicador de que o problema poderá ser (e geralmente, é) mal condicionado.



FUNÇÕES DO MATLAB

`cond` `condest` `eig` `max` `norm` `normest`

```
>> a=@(epsilon) [2 -1;-1 0.5+epsilon];b=[1;1];

>> a(-0.001)\b
ans =
   1.0e+03 *
   -0.7495
   -1.5000

>> a(0.001)\b
ans =
   1.0e+03 *
    0.7505
    1.5000

>> cond(a(0.001),inf)
ans =
   4.5000e+03
```

Embora os dois sistemas possam ser considerados como uma “pequena”perturbação um do outro, as suas soluções são muito diferentes. O número de condição elevado explica estes resultados.

Métodos iterativos

A ideia dos métodos iterativos para resolver um sistema $Ax = b$, consiste em, a partir de uma aproximação inicial para a solução, gerar uma sequência de novas aproximações que, sob certas condições, converge para a solução do problema.

Na prática, o processo será interrompido ao fim de um certo número de iterações ou quando uma determinada estimativa para o erro for suficientemente pequena.

Consideremos novamente o problema da resolução de um sistema de n equações lineares a n incógnitas $Ax = b$, onde $A = (a_{ij})$ é não singular e suponhamos que $a_{ii} \neq 0;\ i = 1, \ldots, n$. As equações do sistema podem, naturalmente, ser rearranjadas da seguinte forma

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

Métodos de Jacobi e Gauss-Seidel

Método de Jacobi

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^{(k)}); i = 1, \ldots, n;\ k = 0, 1, 2, \ldots;$$

$x_i^{(0)};\ i = 1, \ldots, n$, dados

No método de Gauss-Seidel, os valores “atualizados”são utilizados, mal estejam disponíveis.

Método de Jacobi

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}); i = 1, \ldots, n;\ k = 0, 1, 2, \ldots$$

$x_i^{(0)};\ i = 1, \ldots, n$, dados

Formulação Matricial

A matriz A do sistema pode decompor-se como

$$A = D + M + N,$$

onde:

- D é uma matriz diagonal (contendo a diagonal de A),
- M é uma matriz estritamente triangular inferior (com a parte triangular inferior de A) e
- N é uma matriz estritamente triangular superior (com a parte triangular superior de A).

Os dois métodos iterativos referidos podem, então, ser descritos matricialmente como

$$x^{(k+1)} = Bx^{(k)} + c,$$

onde:

Método de Jacobi

$$B = B_J := -D^{-1}(M+N) \quad \text{e} \quad c = c_J := D^{-1}b$$

Método de Gauss-Seidel

$$B = B_{GS} := -(D+M)^{-1}N \quad \text{e} \quad c = c_{GS} := (D+M)^{-1}b.$$

As matrizes B_J e B_{GS} são ditas, respetivamente, matriz de iteração de Jacobi e matriz de iteração de Gauss-Seidel.

Convergência

Seja $\epsilon^{(k)}$ o vetor erro na iteração k de um dos métodos considerados, isto é, seja $\epsilon^{(k)} := x - x^{(k)}$. Dizemos que o respetivo método converge se, para qualquer aproximação inicial $x^{(0)}$, isto é, para qualquer erro inicial $\epsilon^{(0)}$, tivermos

$$\lim_{k\to\infty} \epsilon^{(k)} = 0.^{4}$$

O teorema seguinte estabelece uma condição necessária e suficiente de convergência dos métodos iterativos considerados.

Teorema

É condição necessária e suficiente de convergência dos métodos de Jacobi e de Gauss-Seidel que as respetivas matrizes de iteração tenham raio espetral inferior a 1.

Nota: Dada uma matriz quadrada A, chama-se raio espetral de A e denota-se por $\rho(A)$ o número definido por $\rho(A) := \max\{|\lambda_i| : \lambda_i \text{ é valor próprio de } A\}$.

[4] Com o significado de $\lim_{k\to\infty} e_i^{(k)} = 0; i = 1, \ldots, n.$

O teorema seguinte estabelece uma condição **suficiente** de convergência dos métodos iterativos e fornece também majorantes para o erro na iteração k.

Teorema

Seja B a matriz de iteração do método de Jacobi ou de Gauss-Seidel. Se $\|B\| < 1$, então:

1. *O método converge;*
2. $\|e^{(k)}\| \leq \frac{\|B\|}{1-\|B\|}\|x^{(k)} - x^{(k-1)}\|$ *(Estimativa a posteriori para o erro)*
3. $\|e^{(k)}\| \leq \frac{\|B\|^k}{1-\|B\|}\|x^{(1)} - x^{(0)}\|$ *(Estimativa a priori para o erro)*

Nota: As normas usadas são quaisquer normas vetorial e matricial compatíveis e tal que a norma matricial seja submultiplicativa.

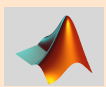
A rapidez de convergência do método depende do "tamanho"do raio espetral da matriz de iteração, sendo, em geral, tanto mais rápida quanto menor for essa quantidade. Pode, além disso, estabelecer-se o seguinte teorema:

Teorema

É condição suficiente de convergência, quer do método de Jacobi, quer do método de Gauss-Seidel, que a matriz do sistema seja de diagonal estritamente dominante.

Nota: Uma matriz quadrada A diz-se de diagonal estritamente dominante se

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|;\ i = 1, \ldots, n.$$



FUNÇÕES TOOLBOX AN

```
metJacobi  - Resolução de um sistema linear pelo método  de Jacobi

metGSeidel - Resolução de um sistema linear pelo método  de Gauss-Seidel

raioJacobi - Matriz de iteração de Jacobi e respetivo raio espetral

raioGSeidel - Matriz de iteração de Gauss-Seidel e  respetivo raio espetral
```