**Notas de Aulas de Algoritmos Numéricos**
**A eliminação de Gauss**

# 1 Introdução

Um caminho possível para se resolver um sistema linear $Ax = b$, quadrado, não singular (det A $\neq 0$), é empregar o método de eliminação de Gauss. O método consiste em realizar operações elementares nas equações para que se tenha um sistema equivalente com as equações reescritas de uma forma mais conveniente Conveniente, neste contexto, singinfica ter as $n$ equações de forma que seja simples obter de sua solução. A ideia é manipular as equações mantendo a solução do sistema.

É possível mostrar que, ao efetuar operações elementares nas equações, a solução do sistema se mantém inalterada

O processo empregado na eliminação de Gauss consiste em reescrever o sistema quadrado $Ax = b$ em um sistema equivalente triangular $Ux = c$ pois é de fácil solução.

Assim, para se resolver o sistema, via eliminação de Gauss há duas fases: a fase de eliminação progressiva, (também chamada de triangularização) e a fase de retrosubstituição (também conhecida por substituição regressiva). *Obs: chama-se de eliminação de Gauss ingênua aquela realizada sem pivoteamento.*

Estas notas de aulas têm por objetivo fazer, sem muita formalidade, uma análise do comportamento dos erros de arredondamento durante as operações realizadas na triangularização na eliminação de Gauss. São apenas complementares ao livro.

# 2 Sistemas lineares

Um sistema linear de dimensão $n$

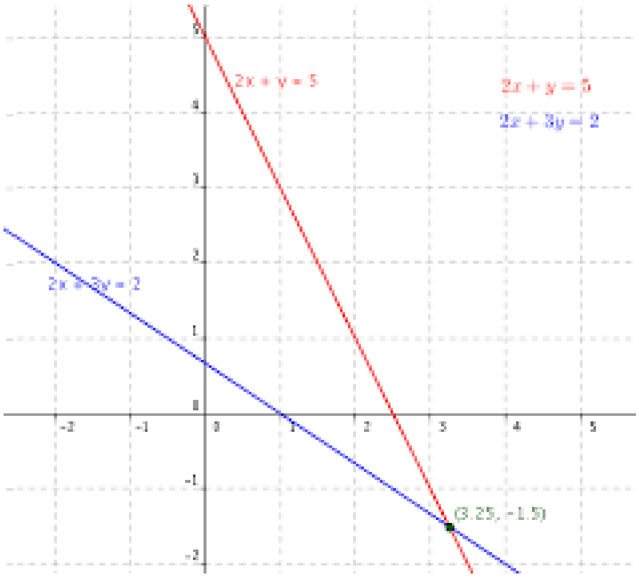
$$\left\{\begin{array}{ccccccccccc}
a_{1,1}x_1 & + & a_{1,2}x_2 & + & a_{1,3}x_3 & + & ... & + & a_{1,n}x_n & = & b_1 \\
a_{2,1}x_1 & + & a_{2,2}x_2 & + & a_{2,3}x_3 & + & ... & + & a_{2,n}x_n & = & b_2 \\
 & & & ... & ... & & \vdots & & ... & = & ... \\
a_{i,1}x_1 & + & a_{i,2}x_2 & + & a_{i,3}x_3 & + & ... & + & a_{i,n}x_n & = & b_i \\
 & & & ... & ... & & \vdots & & ... & = & ... \\
a_{n,1}x_1 & + & a_{n,2}x_2 & + & a_{n,3}x_3 & + & ... & + & a_{n,n}x_n & = & b_n
\end{array}\right.$$

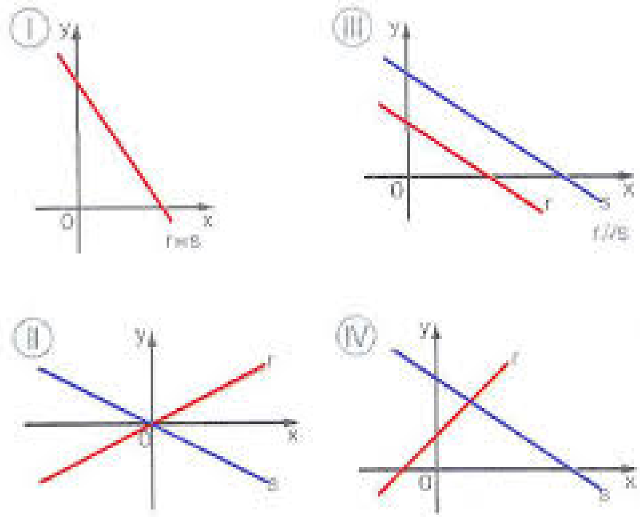
Exemplos:

Seja o sistema $2 \times 2$:

$$\begin{cases} 2x + y = 5.0 \\ 2x + 3y = 2.0 \end{cases}$$

Geometricamente, pode ser represntado por:



Sistema linear $n \times n$:

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad\qquad \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n & = & b_n \end{array}$$

Na forma matricial

$$Ax = b$$

$$Ax = b \Rightarrow \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

# 3 A substituição regressiva ou retrosubstituição

Se o sistema for particular do tipo, triangular superior $n \times n$:

$$\left\{\begin{array}{ccccccccccc}
a_{1,1}x_1 & + & a_{1,2}x_2 & + & a_{1,3}x_3 & + & & ... & + & a_{1,n}x_n & = & b_1 \\
 & + & a_{2,2}x_2 & + & a_{2,3}x_3 & + & & ... & + & a_{2,n}x_n & = & b_2 \\
 & & & ... & ... & & & \vdots & & ... & = & ... \\
 & & & & a_{i,i}x_i & + & a_{i,i+1}x_{i+1} & & & +a_{i,n}x_n & = & b_i \\
 & & & ... & ... & & & \vdots & & ... & = & ... \\
 & & & & & & a_{n-1,n-1}x_n & & + & a_{n-1,n}x_n & = & b_{n-1} \\
 & & & & & & & & & a_{n,n}x_n & = & b_n
\end{array}\right.$$

Então a solução pode ser obtida, falcilmemte, dado que na última equação há apenas uma incógnita, na penúltima há apenas duas e assim por diante.

Assim, o processo consiste em:

- obter $x_n$ da última equação ( da equacão $n$)

- obter $x_{n-1}$ da penúltima equação (pois já sabe $x_n$, então agora só há uma "nova" incógnita $x_{n-1}$

- obter $x_{n-2}$ da ante penúltima equação (pois já sabe $x_{n-1}$ e $x_n$, então agora só há uma "nova" incógnita $x_{n-2}$

- e assim por diante, até obter o $x_1$ pela primeira equação

A substituição regressiva é também chamada de substituição retroativa, ou ainda de retrosubstituição.

## 3.1 O algoritmo de substituição regressiva

Dada a matriz A na forma triangular superior e o vetor b, obtém a solução do sistema $Ax = b$.

```
Objetivo: Resolver sist. triangular superior
INICIO
 Ler(a,b,n)
 x=b(n)/a(n,n)
 Para  i = (n-1) até  1, passo(-1)
       s=0
       Para j= (i+1):n
                 s = s + a(i,j)*x(j)
       Fim {para j}
       x(i) = ( b(i)- s )/a(i,i)
 Fim {para i}
 Mostrar(x)
FIM
```

# 4 A triangularização

Dado uma matriz (quadrada) $A$, de ordem $n$, não singular, a triangularização consiste em transformar o sistema original $Ax = b$ em um o sistema equivalente que seja triangular superior.

O algoritmo abaixo (em pseudo-linguagem) refere-se à etapa de triangularização ingênua (também chamada de substituição progressiva ingênua).

```
INICIO
   Para k de 1 até (n-1)
      Para i de (k+1) até n
          m= A(i,k)/A(k,k)
          Para j de (k+1) até n
               A(i,j)= A(i,j)- m*A(k,j)
          Fim {para j}
     Fim {para i}
   Fim {para k}
FIM
```

# 5 Análise do erros de arrendondamento

Em uma etapa $k$, na linha $i$, a atualização de um elemento $a_{ij}$ é dada pela expressão (atualização exata):

$$a_{ij}^{novo} = a_{ij}^{ant} - ma_{ik}$$

Devido à limitação da mantissa, os valores $a_{ij}^{ant}$ e $a_{ik}$ que, de fato, ficam armanezados (que serão denominados por $(a_{ij}^{ant})_{Maq}$ e $(a_{ik})_{Maq}$ ) possuem erros em relação aos valores corretos (os exatos). Estes erros serão denotados de $\epsilon_{ij}^{ant}$ e $\epsilon_{ik}$, respectivamente para $a_{ij}^{ant}$ e $a_{ik}$. Assim pode-se escrever:

$$\left(a_{ij}^{ant}\right)_{Maq} = a_{ij}^{ant} + \epsilon_{ij}^{ant}$$

e

$$\left(a_{ik}\right)_{Maq} = a_{ik} + \epsilon_{ik}.$$

*Obs: estes erros podem eventualmente ser nulos (quando os valores $a_{ij}^{ant}$ e $a_{ik}$ tiverem sido representados de forma exata).*

O mesmo pode ser escrito para $a_{ij}^{novo}$ que, em relação ao valor exato, é dado por

$$\left(a_{ij}^{novo}\right)_{Maq} = a_{ij}^{novo} + \epsilon_{ij}^{novo}$$

**O erro contido em $a_{ij}^{novo}$**

Quer-se analisar como se comporta o erro $\epsilon_{ij}^{novo}$, ou seja, o erro contido em $a_{ij}^{novo}$.

Para isto é preciso ver que a atualização de $a_{ij}^{novo}$ efetivamente realizada pelo computador é diferente daquela teórica, já que os valores estão armazenados com erros. Portanto, na "Máquina", tem-se que

$$\left(a_{ij}^{novo}\right)_{Maq} = \left(a_{ij}^{ant}\right)_{Maq} - m(a_{ik})_{Maq}.$$

Substituindo os valores obtidos pela máquina pelos valores exatos e seus respectivos erros, tem se que:

$$a_{ij}^{novo} + \epsilon_{ij}^{novo} = a_{ij}^{ant} + \epsilon_{ij}^{ant} - m(a_{ik} + \epsilon_{ik}).$$

Assim, usando a expressão acima e lembrando que os valores exatos estão relacionados pela equação $a_{ij}^{novo} = a_{ij}^{ant} - ma_{ik}$ é possível escrever que:

$$\epsilon_{ij}^{novo} = \epsilon_{ij}^{ant} - m\epsilon_{ik}$$

O erro $|\epsilon_{ij}^{novo}|$ fica, portanto, limitado por

$$|\epsilon_{ij}^{novo}| = |\epsilon_{ij}^{ant} - m\epsilon_{ik}| \leq |\epsilon_{ij}^{ant}| + |m\epsilon_{ik}|.$$

Pode-se, então, perceber que o valor do **multiplicador** $m$ é de fundamental importância para a não ampliação dos erros de arredondamento.

Valores de $|m|$ elevados podem acarretar uma ampliação dos erros de arredondamento, no entanto, se o valor de $m$ for mantido entre $-1$ e $1$ (isto é $|m| \leq 1$) há um controle do erro de arredondamento.

Assim, a necessidade implementar a triangularização **com o pivoteamento**.