Introdução ao Cálculo Numérico

Eliminação gaussiana e sistemas triangulares

Instituto de MATEMÁTICA E ESTATÍSTICA
UFRGS

Prof. Rudnei Dias da Cunha

- Método de eliminação de Gauss
  - Objetivo: resolver o sistema não-singular de equações lineares $Ax = b$
  - O método consiste da aplicação de operações elementares às linhas de A e elementos de b correspondentes, de forma a transformar A numa matriz triangular superior

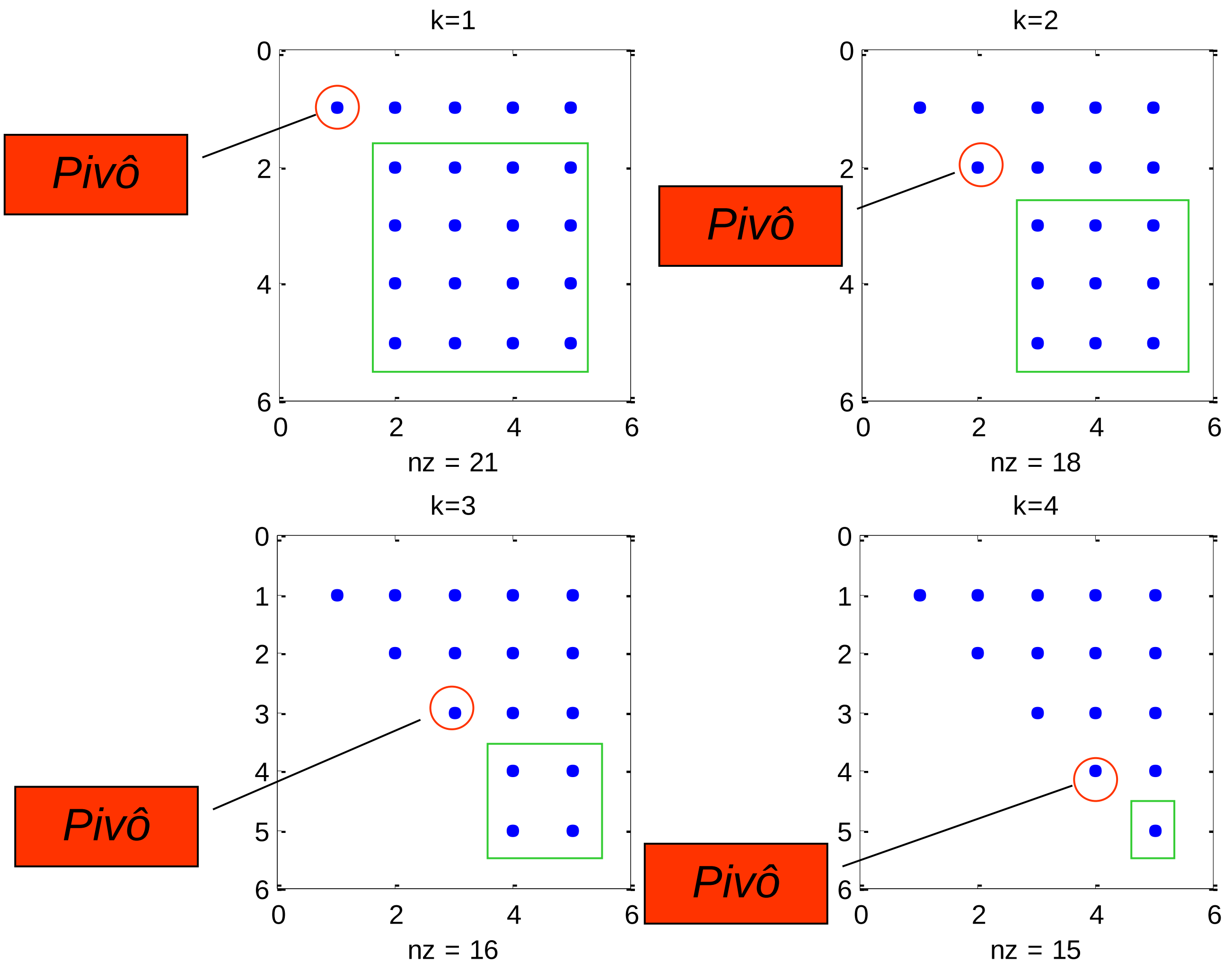
- Método de eliminação de Gauss
  - Para se eliminar os elementos abaixo da diagonal k (i.e. os elementos das linhas $k+1, k+2, \ldots, n$ na coluna k), usa-se o elemento $A_{k,k}$ como *pivô* e calcula-se o multiplicador $z = A_{i,k}/A_{k,k}$ para se multiplicar a linha k de A e subtraí-la da linha i de A (o mesmo é feito sobre os elementos b_k e b_i, resp.)

- Método de eliminação de Gauss
  - Ao final do método, obtém-se o sistema triangular superior $\tilde{A}x=y$
  - A solução do sistema, x, é então obtida através do processo de retro-substituição
  - O método pode ser escrito em forma algorítmica como segue:

```
proc eliminacao_gaussiana(entrada: n,A,b; saida: x,resultado)
  resultado:= 0
  para k:=1 até n-1 faça
    se A(k,k)==0 então
      resultado:= -k
      retorna
    fim se
    para i:=k+1 até n faça
      z:= A(i,k)/A(k,k)
      para j:=k+1 até n faça
        A(i,j):= A(i,j)-z*A(k,j)
      fim para
      b(i) = b(i)-z*b(k)
    fim para
  fim para
  se A(n,n)==0 então
    resultado:= -n
    retorna
  fim se
  ! Resolve sistema triangular inferior
  retro_substituicao(n,A,b,x)
end proc
```

- Método de eliminação de Gauss
  - Se nós efetuarmos a eliminação gaussiana com o algoritmo acima, e exibirmos o conteúdo da matriz A a cada iteração k (i.e., imediatamente após o término do laço), obteremos o seguinte gráfico:

* Método de eliminação de Gauss



- A retro-substituição
  - O processo de retro-substituição

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} A_{i,j} x_j}{A_{i,i}}$$

para $i = n, n-1, \ldots, 1$, desde que $A_{i,i} \neq 0\ \forall i$ e considerando que $\sum_{j=i+1}^{n} A_{i,j} x_j = 0$ se $i = n$, pode ser implementado como descrito a seguir:

•A retro-substituição

```
proc retro_substituicao(entrada: n,A,b; saida: x,resultado)
  para i:=n até 1 por -1 faça
    s:= 0.0
    para j:=i+1 até n faça
      s:= s+A(i,j)*x(j)
    fim para
    se A(i,i)/=0.0 então
      x(i):= (b(i)-s)/A(i,i)
    senão
      resultado:= -i
      retorna
    fim se
  fim para
fim proc
```

- Método de eliminação de Gauss
  - Retomando o método de eliminação de Gauss, o problema com o algoritmo visto anteriormente é que ele não leva em consideração os efeitos numéricos nefastos causados pela divisão por $A(k,k)$ quando $|A(k,k)| \cong 0$
  - Evidentemente que a troca de linhas também deve ser feita, para evitar o uso de pivôs nulos

- Método de eliminação de Gauss
  - Por exemplo, considere

$$\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, 0 < \epsilon \ll 1$$

cuja solução é exata é

$$x_1 = \frac{1}{1-\epsilon} \cong 1, x_2 = \frac{1-2\epsilon}{1-\epsilon} \cong 1$$

- Método de eliminação de Gauss
  - Aplicando o algoritmo acima, para $\epsilon \ll 1$, obteremos:

$$x_2 = \frac{2-\epsilon^{-1}}{1-\epsilon^{-1}} \cong 1, x_1 = (1-x_2)\epsilon^{-1} \cong 0$$

o que obviamente é errado!

- Método de eliminação de Gauss
    - Se implementarmos o algoritmo acima em Fortran 90, em precisão simples – REAL(KIND=4) – obteremos os seguintes valores de x:

$\boldsymbol{\epsilon}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$
10^{-4}	$1{,}000166$	$0{,}999000$
10^{-6}	$1{,}013279$	$0{,}999990$
10^{-8}	$0{,}000000$	$1{,}000000$

- Método de eliminação de Gauss
  - Por essa razão, o método de eliminação de Gauss deve considerar não só a troca de linhas mas, também, o escalonamento das mesmas, de forma a evitar problemas como o visto anteriormente
  - Evidentemente que a troca de linhas também deve ser feita, para evitar o uso de pivôs nulos:
    - A troca de linhas é feita de forma ***implícita***, utilizando um arranjo de índices para fazer o acesso ***indireto*** das linhas da matriz A e do vetor b

- Método de eliminação de Gauss
  - O algoritmo para o método de eliminação de Gauss, incorporando pivotamento parcial e escalonamento, pode ser descrito como segue:

```
proc eliminacao_gaussiana_pivo(entrada: n,A,b; saida:
   A,b,x,resultado)
  resultado:= 0
  ! Inicializa o arranjo de permutação e os valores de escalonamento
  para i:=1 até n faça
    p(i):= i
    s(i):= -MAXR
    para j:=1 até n faça
      s(i):= max(s(i),abs(A(i,j)))
    fim para
  fim para
  para k:=1 até n-1 faça
    ! Localiza o pivô, indicado pelo valor de j
    j:= k
    para i:=k+1 até n faça
      se abs(A(p(i),k))/s(p(i))>abs(A(p(j),k))/s(p(j)) então
        j:= i
        para
      fim se
    fim para
    ! Indica a troca entre as linhas k e j
    q:= p(k) ; p(k):= p(j) ; p(j) = q
```

```
    se A(p(k),k))/=0.0 então
      para i:=k+1 até n faça
        z:= A(p(i),k)/A(p(k),k)
        para j:=k+1 até n faça
          A(p(i),j):= A(p(i),j)-z*A(p(k),j)
        fim para
        b(p(i)):= b(p(i))-z*b(p(k))
      fim para
    senão
      resultado:= -p(k)
      retorna
    fim se
  fim para
  se A(p(n),n)==0.0 então
    resultado:= -p(n)
    retorna
  fim se
  ! Resolve sistema triangular inferior
  retro_substituicao_pivo(n,A,b,p,x)
end proc
```

- A retro-substituição com pivotamento

```
proc retro_substituicao_pivo(entrada: n,A,b,p; saida: x)
  para i:=n até 1 por -1 faça
    s:= 0.0
    para j:=n até i+1 por -1 faça
      s:= s+A(p(i),j)*x(j)
    fim para
    x(i):= (b(p(i))-s)/A(p(i),i)
  fim para
fim proc
```

- Método de eliminação de Gauss
  - Se implementarmos os algoritmos com pivotamento em Fortran 90, em precisão simples – REAL(KIND=4) – obteremos os seguintes valores de x, os quais são aceitáveis:

ϵ	x_1	x_2
10^{-4}	1,000100	0,999000
10^{-6}	1,000001	0,999990
10^{-8}	1,000000	1,000000

- Análise da quantidade de multiplicações
  - Considerando apenas as multiplicações (que são mais demoradas do que somas e subtrações), o algoritmo da eliminação gaussiana pode ser sumarizado como segue:

```
para k:=1 até n-1
    ...
    para i:=k+1 até n
        para j:=k+1 até n
            1 multiplicação
        fim para
        1 multiplicação
    fim para
fim para
```


Instituto de MATEMÁTICA E ESTATÍSTICA UFRGS

- Análise da quantidade de multiplicações
  - Escrevendo na notação de somatórios e expandindo os termos, vem:

$$\sum_{k=1}^{n-1}\sum_{i=k+1}^{n}\left(1+\sum_{j=k+1}^{n}1\right)=$$

$$\sum_{k=1}^{n-1}\sum_{i=k+1}^{n}(n-k+1)=$$

$$\frac{1}{3}(n^3-n)$$

- Análise da quantidade de multiplicações
  - A quantidade de multiplicações efetuadas na retrosubstituição é
$$\sum_{i=1}^{n}\sum_{j=1}^{i-1} 1 = \frac{1}{2}(n^2 - n)$$
  - Somando os resultados de ambas as expressões, a quantidade de multiplicações, $c_\times$, é dada por
$$c_\times = \frac{1}{3}(n^3 - n) + \frac{1}{2}(n^2 - n)$$

- Análise da quantidade de multiplicações
  - Para n grande, podemos escrever

$$c_{\times} \cong \frac{n^3}{3}$$

  - Logo, dependendo do valor de n, a solução do sistema utilizando eliminação gaussiana pode se tornar inviável, em termos do tempo computacional necessário para obter a solução.