1 Lista de Exercícios de MAC 315 - Segundo semestre de 2001

Leônidas de Oliveira Brandão

1.1 Exercícios "teóricos"

Exercício 1.1 *Sejam* $\mathbf{A}$ *e* $\mathbf{B}$ *matrizes quadradas de ordem* n. *Prove que:*
(a) *Se* $\mathbf{AB} = \mathbf{I}_n$, *então* $\mathbf{A}$ *e* $\mathbf{B}$ *são não singulares e* $\mathbf{A} = \mathbf{B}^{-1}$ *e* $\mathbf{B} = \mathbf{A}^{-1}$;
(b) $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$;
(c) $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$;
(d) $\mathbf{A}$ *é não singular se, e somente se pode ser transformada na matriz identidade por transformações elementares (vide definição no exercício 1.1).*

Exercício 1.2 *Sejam* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\boldsymbol{x}$ *e* $\boldsymbol{y}$, *respectivamente,* $\mathbb{R}^{m\times n}$, $\mathbb{R}^{m\times n}$, $\mathbb{R}^{n\times t}$, $\mathbb{R}^n$ *e* $\mathbb{R}^n$. *Prove que:*
(a) $(\mathbf{A}')' = \mathbf{A}$;
(b) $(\mathbf{A}+\mathbf{B})' = \mathbf{A}' + \mathbf{B}'$;
(c) $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$;
(d) $(\mathbf{A}\boldsymbol{x})'\boldsymbol{y} = \boldsymbol{x}'(\mathbf{A}'\boldsymbol{y})$ $\qquad (<\mathbf{A}\boldsymbol{x},\boldsymbol{y}>=<\boldsymbol{x},\mathbf{A}'\boldsymbol{y}>)$.

Exercício 1.3 *Prove a desigualdade de Cauchy-Schwartz:* $|\boldsymbol{a}'\boldsymbol{b}| \leqq \|\boldsymbol{a}\|\|\boldsymbol{b}\|$, *para todo* $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{R}^n \times \mathbb{R}^n$.
Dica: Note que, para $\lambda \in \mathbb{R}$, $\|\boldsymbol{a} + \lambda\boldsymbol{b}\|^2 \geqq 0$ e pode examinado como um polinômio em λ.

Exercício 1.4 *Sejam* $B := \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|^2 \leqq r^2\}$, $\mathbf{1} := (1,1,\dots,1)' = \sum_{i=1}^n \boldsymbol{e}_i$ *e* $H := \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{1}'\boldsymbol{x} \leqq r\sqrt{n}\}$. *Prove que* $B \subset H$ *(use desenho no* $\mathbb{R}^2$ *para ilustrar sua prova).*

Exercício 1.5 *Sejam* $V := \{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n\}$ *l.i.,* $\mathbf{A} := [\boldsymbol{a}_1|\boldsymbol{a}_2|\cdots|\boldsymbol{a}_n]$ *e* $\mathbf{B} := \mathbf{A}^{-1}$. *Para algum* $\boldsymbol{b} \in \mathbb{R}^n$, *se* $\{\boldsymbol{a}_j\}_{j=1}^n \bigcup \{\boldsymbol{b}\}$ *é l.i., mostre como obter* $\mathbf{C}$ *a inversa de* $\mathbf{D} := [\boldsymbol{a}_1|\boldsymbol{a}_2|\cdots|\boldsymbol{a}_{n-1}|\boldsymbol{b}]$ *em* $O(n^2)$ *operações elementares.*

Exercício 1.6 *Para* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *e* $\boldsymbol{b} \in \mathbb{R}^n$, *mostre que o sistema abaixo não pode ter solução* $(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^m$,

$$\begin{aligned} \mathbf{A}\boldsymbol{x} &\leqq \mathbf{0} && (1)\\ \mathbf{A}'\boldsymbol{y} &= \boldsymbol{b} && (2)\\ \boldsymbol{b}'\boldsymbol{x} &> 0 && (3)\\ \boldsymbol{y} &\geqq \mathbf{0} && (4) \end{aligned}$$

justificando cada passagem empregada (use os rótulos das equações para isso).

Exercício 1.7 *Dê uma interpretação geométrica para o resultado obtido no exercício acima (utilize um desenho em* $\mathbb{R}^2$ *para facilitar sua explicação).*

Exercício 1.8 *Prove ou ache um contra-exemplo para a seguinte implicação: para* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *e* $\boldsymbol{b} \in \mathbb{R}^m$,

$$\left.\begin{array}{l} \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{x} > \mathbf{0} \end{array}\right\rangle \implies \boldsymbol{b}'\mathbf{A} > \mathbf{0}.$$

Exercício 1.9 *Seja* $\mathcal{S}$ *o conjunto das matrizes quadradas tais que* $\mathbf{A} \in \mathcal{S} \iff \left\{\begin{array}{lll} a_{ij} > 0 & \Leftrightarrow & i = j, \quad e \\ i < j & \Rightarrow & a_{ij} = 0. \end{array}\right.$
Prove que $\mathbf{A} \in \mathcal{S}$ *implica que* $\mathbf{A}$ *é não singular e que* $|\mathbf{A}| > 0$.
Dica: Utilize indução na dimensão n das matrizes $\mathbf{A}$.

1.2 Exercícios no Scilab

Exercício 1.1 *Uma* TRANSFORMAÇÃO ELEMENTAR *sobre uma matriz* **A** *equivale a somar numa linha (coluna) de* **A** *o produto de um escalar por outra linha (coluna). Uma matriz* **B** *que realiza uma tranformação elementar (***BA***) é dita de* TRANSFORMAÇÃO ELEMENTAR. *Podemos dizer também que uma matriz que seja o produto de transformações elementares é de* TRANSFORMAÇÃO.

Utilizando o programa Scilab *no apêndice A.2 (estude-o para entender sua sintaxe) e a matriz "default" no mesmo, construa (deduza) as três matrizes de transformação* `T1`, `T2` *e* `T3` *que, respectivamente, "pivota" a coluna 1 (os elementos abaixo) por b_{11}, coluna 2 por b_{22} e coluna 3 por b_{33} (***B*** são as matrizes obtidas em cada iteração do algoritmo).*

Entregue um código Scilab*, definindo as matrizes* `T1`, `T2` *e* `T3` *e imprimindo* $|\mathbf{T1}|$, $|\mathbf{T2}|$, $|\mathbf{T3}|$, $|\mathbf{A}|$ *e* $|\mathbf{B}|$ *e também o produto* $\mathbf{T3} * \mathbf{T2} * \mathbf{T1} * \mathbf{A}$.

Exercício 1.2 *Construa um programa* Scilab *que realize a operação do exercício 1.5 (obtendo as correspondentes* **D** *e* **C***) e, ao final, imprima as matrizes* **C** *e* $\mathbf{C} * \mathbf{D}$. *Use o exemplo do apêndice A.2 como modelo, notando que uma operação efetuada numa linha de matriz (como* `B(i,:)=B(i,:)/B(i,i)`*) gasta da $O(n)$ operações.*

Exercício 1.3 *Teste o programa de seu exercício acima para as matrizes abaixo, aplicando seu algoritmo para obter* **C** *e, ao final, imprima as matrizes* **C** *e* $\mathbf{C} * \mathbf{D}$.

$$(\mathbf{A}, \boldsymbol{b}) := \left(\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 \\ 3 & 0 & 2 & 0 & 3 \\ 0 & 2 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right), \quad (\mathbf{A}, \boldsymbol{b}) := \left(\begin{bmatrix} 2 & -1 & 1 & 1 & 0 \\ -1 & 2 & 0 & 2 & 0 \\ 2 & 4 & -1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 3 \\ 0 & 3 & 0 & 1 & 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right),$$

$$(\mathbf{A}, \boldsymbol{b}) := \left(\begin{bmatrix} 1 & -1 & 1 & 1 & 0 \\ -1 & 2 & 0 & 2 & -1 \\ 1 & 1 & -1 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \right).$$

A Exemplos de programas no Scilab

A.1 Atribuições, recorrências e instâncias

```
function []=nada()
//
// MAC 315 - 2001 - SciLab - Prof. Leônidas
// Exemplo: funções contadoras, métodos recursivos e iterativos
//
// ----------------------------------------------------------------------
// Algumas dicas básicas sobre o Scilab (mais detalhes consulte o "Help")
//
// A primeira linha é necessária, pois uma função SciLab NÃO pode começar
// com comentários, precisa do 'function'
//
// Para definir variáveis 'a' e 'b' com os valores '1' e '2.5': a=1, b=2.5
// Para definir variáveis 'a' e 'b' sem "echo" na tela:        a=1; b=2.5;
// Para ver todas as variáveis e funções já definidas, digite:        who
// Para "limpar" o Scilab, removendo todas as variáveis e funções:   clear
// Para remover apenas a variável 'a':                             clear a
// Para "rodar" este exemplo, pegue-o na página de MAC315 e digite:
//      ;getf("/todo_o_caminho_no_Linux/ex-contadores.sci");
// Agora você dispõe das funções: contador_it, contador_rec e galho_rec.
// Para testar a função 'contador_it', somando de 2 até 10, digite
//      contador_it(2,10)
// se quiser o resultado na variável 's' digite
//      s=contador_it(2,10)
// ----------------------------------------------------------------------

function [cont]=contador_it(k,fim)
 cont=0;
 for j = k:fim, cont=cont+1; end;

// Para usar esta função, após o 'getf', digite por exemplo
//      soma=contador_rec(3,10), soma
//
function [c]=contador_rec(k,fim)
 if (k<fim)
    c=contador_rec(k+1,fim)+1; // Cuidado: retirando-se "c=...", a chamada
                               // externa ficará com o valor da primeira
    // printf("%f ",cont);     // chamada !!
 else c=1;
 end;

// Cuidado com a recorrência: a = a+EXP => cria nova variável a no nível
//                                         atual da recursão
// Para usar esta função, após o 'getf', digite por exemplo
//      cont=0, galho_rec(1,10)
//
function []=galho_rec(k,fim)
 if (k<fim)
    cont = cont+1;          // (*)
    printf("E: %f ",cont);
    galho_rec(k+1,fim);     // apesar de cont ter sido acresc. em recursão
                            // acima, aqui ela volta a ter o valor de (*)
    printf("S: %f ",cont);
 end;

end
```

A.2 Como "pivotar" uma matriz

```
function []=pivo()
//
// MAC 315 - 2001 - Prof. Leônidas O. Brandão
//
// Pivoteamento abaixo da diagonal principal
//
// Dicas;
// - acentos não são reconhecidos nos print's !!!
// - obrigatório a primeira linha começar comando e não com comentário.
// - operações em linhas (colunas)
//   B(i,:) = B(i,:) / B(i,i) dividiria toda a linha i de B por b_{ii}

A = [-1,1,0,0; 1,0,1,0; 3,0,0,3; -3,0,1,1];
dim=size(A); n=dim(1); m=dim(2);  // pega as dimensões de A
if (n<>m)
   printf("Erro: matriz A(%d,%d) deve ser quadrada",m,n)
   exit;
   end;

// Entrada de uma matriz via "diálogo gráfico"
row='linha ' ; labelv = row(ones(1,n))+string(1:n);
col='coluna '; labelh = col(ones(1,m))+string(1:m);
new=evstr(x_mdialog('<< Edite a matriz A >>', labelv, labelh, string(A)))

B = A // matrix("empty",n) definiria uma matriz quadrada vazia

for i=1:n
 if (B(i,i)==0)
    printf("Erro: B(%d,%d)=%f",i,i,B(i,i))
    return
    end
 for k=i+1:n // pivoteamento segundo B(i,i)
  B(k,:) =  B(k,:) - B(k,i)/B(i,i) * B(i,:)
  end
end

print(%io(2),B,det(B),det(A));

end
```