# Cálculo Numérico - Exercício II

João Victor Colombari Carlet 5274502

12/09/2022

## 1 Introdução

Este é o segundo exercício proposto para SME0300 - Cálculo Numérico. Os exercícios propostos são semanais e cobrem o conteúdo exposto em sala. A orientação é a disponibilização de um documento em formato *pdf* na plataforma da disciplina até a data limite.

## 2 Desenvolvimento

### 2.1 Questão 01

**Seja o sistema de ponto flutuante F(2,3,6,7) utilizando as normas IEEE 754 (ou seja, o maior e menor expoente na base do sistema são separados para casos especiais, há um viés no expoente e há 1 dígito implícito na precisão):**

- **a) quantos bits são necessários para representá-lo?**
- **b) qual o maior número real representado?**
- **c) qual o menor número real maior que zero representado?**

O sistema $F(b, T, m, M)$ é organizado da seguinte maneira [1]:

$$X = (-1)^s.b^e.(0.d_1.d_2....dn)$$

Em que:

- **b** 2 é a base de representação;
- **T** é a precisão;
- **m** e **M** são respectivamente o menor e maior;
- **s** é sinal ( 0 se positivo e 1 se negativo);
- **e** é o expoente entre **m** e **M**.

Portanto, são necessários:

$$(s = 1) + (e = 4) + (f = 3) \quad bits = 8 \quad bits$$

O maior real é:

$$0 \quad 0111 \quad 111$$

$$(-1)^0.2^7.(0.111) = (1110000.00)_b = (112)_{10}$$

E o menor real maior que zero, pela mesma lógica:

$$(-1)^0.2^{-6}.(0.001) = (0.001953125)_{10}$$

### 2.2 Questão 02

**Escreva um código em Matlab/Octave que:**

- **crie um matriz triangular inferior unitária de tamanho 10 $\times$ 10 (i.e., lii = 1 para todo i, e lij = 0 para todo i < j), cujos elementos lij tal que j < i são números aleatórios de distribuição uniforme entre zero e 1; e**
- **calcule sua inversa utilizando a função inv(L) e exiba o resultado na tela.**

**Após executar o código algumas vezes, gerando diferentes números aleatórios, o que podemos afirmar a respeito da forma da matriz L1?**

O código é facilmente implementado por meio da criação de um a matriz vazia $A$ recursões seguindo o descrito no enunciado.

```
clc
clear all
close all

A=[];

for i=1:1:10
    for j=1:1:10
        if(i==j)
            A(i,j)=1;
        elseif(i<j)
            A(i,j)=0;
        else
            A(i,j)=rand;
        end
    end
end
```

Rodando o código acima uma vez se tem a seguinte matriz presente na Figura 1.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1.0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.9797 | 1.0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.4389 | 0.1111 | 1.0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.2581 | 0.4087 | 0.5949 | 1.0000 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.2622 | 0.6028 | 0.7112 | 0.2217 | 1.0000 | 0 | 0 | 0 | 0 | 0 |
| 0.1174 | 0.2967 | 0.3188 | 0.4242 | 0.5079 | 1.0000 | 0 | 0 | 0 | 0 |
| 0.0855 | 0.2625 | 0.8010 | 0.0292 | 0.9289 | 0.7303 | 1.0000 | 0 | 0 | 0 |
| 0.4886 | 0.5785 | 0.2373 | 0.4588 | 0.9631 | 0.5468 | 0.5211 | 1.0000 | 0 | 0 |
| 0.2316 | 0.4889 | 0.6241 | 0.6791 | 0.3955 | 0.3674 | 0.9880 | 0.0377 | 1.0000 | 0 |
| 0.8852 | 0.9133 | 0.7962 | 0.0987 | 0.2619 | 0.3354 | 0.6797 | 0.1366 | 0.7212 | 1.0000 |

Figura 1: Matriz A obtida com o código da Questão 02

Fonte: Disponibilizado pelo Autor

Utilizando o comando *inv()* se tem o que segue:

```
 1.0000         0         0         0         0         0         0         0         0         0
-0.9797    1.0000         0         0         0         0         0         0         0         0
-0.3300   -0.1111    1.0000         0         0         0         0         0         0         0
 0.3387   -0.3426   -0.5949    1.0000         0         0         0         0         0         0
 0.4880   -0.4478   -0.5793   -0.2217    1.0000         0         0         0         0         0
-0.1131    0.1115    0.2278   -0.3116   -0.5079    1.0000         0         0         0         0
 0.0554    0.1711   -0.4119    0.4043   -0.5579   -0.7303    1.0000         0         0         0
-0.4359   -0.1138    0.6837   -0.2856   -0.3946   -0.1662   -0.5211    1.0000         0         0
 0.0336   -0.2154    0.3065   -0.8656    0.3572    0.3604   -0.9683   -0.0377    1.0000         0
 0.1467   -0.6565   -0.6966    0.4523    0.0839   -0.0762    0.0898   -0.1093   -0.7212    1.0000
```

Figura 2: Matriz inversa de A obtida com o código da Questão 02 e o comando *inv()*

Fonte: Disponibilizado pelo Autor

Executando o código algumas vezes se pode dizer que a matriz invertida é também uma triangular inferior cuja primeira diagonal abaixo da diagonal principal é igual em módulo à respectiva diagonal na matriz original, sendo os elementos desta os mesmo daquela apenas com o sinal trocado.

### 2.3 Questão 03

**Escreva um código em Matlab/Octave que crie uma matriz 100 $\times$ 100 com todos os elementos aleatórios uniformes entre zero e um, e calcule:**

**a decomposição LU da matriz sem utilizar nenhuma função já programada, apenas a matriz dada e loops;**

**o determinante da matriz A utilizando o resultado da decomposição LU, sem utilizar nenhuma função já programada.**

A decomposição LU pôde ser implementada com o seguinte código:

```
function [L,U,P] = LU_ex3(A)

n=size(A,1);
m=zeros(n);
L=zeros(n);
P=eye(n);

for r=1:n-1
    [V,k]=max(abs(A(r:n,r)));
    k=k+r-1;
    if(k~=r)
        temp=A(k,:);
        A(k,:)=A(r,:);
        A(r,:)=temp;

        temp=P(k,:);
        P(k,:)=P(r,:);
        P(r,:)=temp;

        temp=L(k,:);
        L(k,:)=L(r,:);
        L(r,:)=temp;
    end
    L(r,r)=1;
    for i=r+1:n
        m(i,r)=-A(i,r)/A(r,r);
        L(i,r)=-m(i,r);
        for j=r:n
            A(i,j)=A(i,j)+m(i,r)*A(r,j);
        end
    end
end

L(n,n)=1;
U=A;
end
```

A matriz 100x100 com todos os elementos aleatórios uniformes entre zero e um, bem como a aplicação da função que calcula a decomposição LU é feita com o seguinte código:

```
clc
clear all
close all

A=[];

for i=1:1:4
    for j=1:1:4
        A(i,j)=rand;
    end
end

[L,U,P]=LU_ex3(A)
```

O leitor poderá conferir o código ao passo que o mesmo será enviado juntamente a este arquivo *pdf*, no entanto, um teste para conferir o bom funcionamento da função é a multiplicação das matrizes P com A e das matrizes L com U, que devem resultar na mesma matriz, como segue na Figura 3.

```
>> P*A

ans =

    0.8981    0.6218    0.4146    0.6476
    0.6684    0.1566    0.7743    0.2131
    0.4764    0.6028    0.5915    0.2253
    0.4893    0.0938    0.6373    0.9503

>> L*U

ans =

    0.8981    0.6218    0.4146    0.6476
    0.6684    0.1566    0.7743    0.2131
    0.4764    0.6028    0.5915    0.2253
    0.4893    0.0938    0.6373    0.9503
```

Figura 3: Teste de funcionamento da função que calcula a decomposição LU

Fonte: Disponibilizado pelo Autor

O determinante de A pode ser calculado de maneira simples utilizando a seguinte relação provada em aula:

$$det(A) = det(L\mho) = det(L) * det(\mho)$$

Uma vez que $L$ e $\mho$ são matrizes triangulares, seus respectivos determinantes são iguais ao produto dos elementos de suas diagonais principais. Em sendo

L uma triangular inferior, $det(L) = 1$, portanto $det(A) = det(\mathfrak{U})$.

O procedimento é executado com o seguinte código:

```
deter=1;
for i=1:1:100
    for j=1:1:100
        if(i==j)
            deter=deter*U(i,j);
        end
    end
end
```

Que, para a matriz A 100x100 retorna os seguintes valores:

```
>> det(A)

ans =

   8.6433e+25

>> deter

deter =

   8.6433e+25
```

Figura 4: Teste de funcionamento da função que calcula o determinante

Fonte: Disponibilizado pelo Autor

Foram então criadas duas versões do script - uma utilizando a função *lu()* e uma utilizando a função escrita e descrita aqui. Ambas foram rodadas com o comando *Run and Time*. Os resultados estão presentes na Figura 5. A função *lu()* roda mais rápido pois o Matlab não otimiza a execução da função com a escolha do melhor método numérico para a matriz em questão com base em suas proporcionalidades.

| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| ex3 | 1 | 0.055 s | 0.023 s | |

| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| ex3 | 1 | 0.042 s | 0.033 s | |

Figura 5: Comparação tempo de execução com função *lu()* (nativa) e função escrita, respectivamente.

Fonte: Disponibilizado pelo Autor

# Referências


[1] G. V. R. Viana, "Padrao ieee 754 para aritmética binária de ponto flutuante," *Revista CT*, vol. 1, no. 1, pp. 29–33, 1999.