EXERCÍCIO 01 (ZERO DE FUNÇÕES): Um servidor Cloud executa tarefas críticas e seu consumo de energia deve ser monitorado continuamente. A taxa de consumo energético, $E(u)$, depende da taxa de utilização da CPU, t, que varia de 0 (ocioso) a 1 (uso máximo). O responsável pela gestão do servidor - usando dados armazenados - modelou empiricamente por interpolação polinomial que a função de consumo é como:

$$E(t) = 162,5t^3 - 243,75t^2 + 130t + 16,25$$

O engenheiro de computação do setor identificou que, um consumo máximo de $60w$ equilibra a eficiência e a temperatura do servidor em níveis seguros durante o período de pico. Então determine em que taxa de utilização da CPU o servidor atinge exatamente tal consumo.

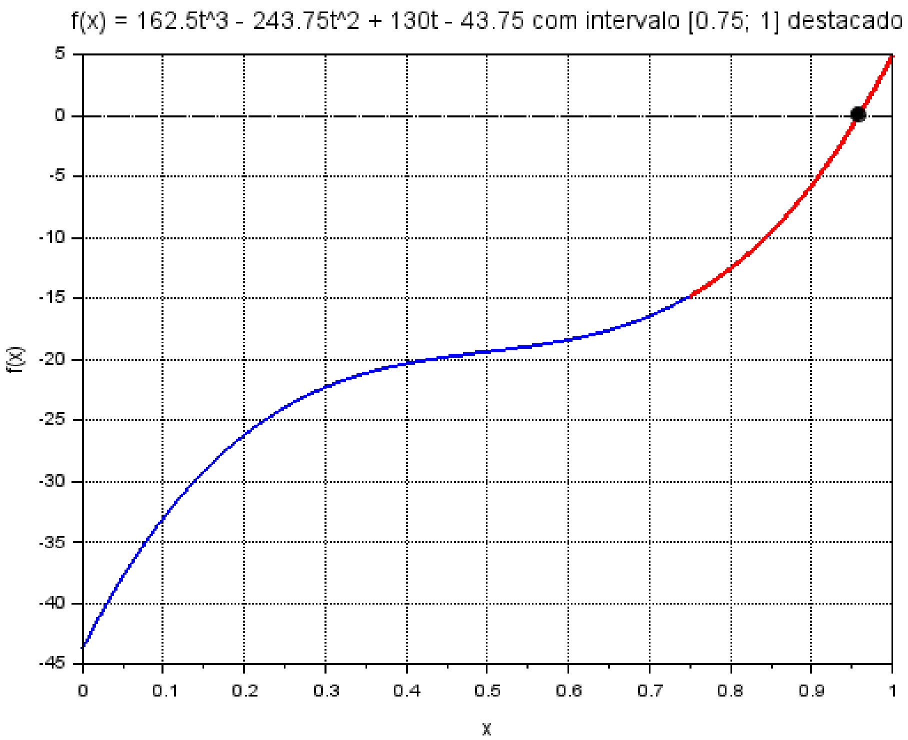
Solução:

1. **Análise por inspeção dos intervalos candidatos:**
   - **Resolvendo** $60 = 162,5t^3 - 243,75t^2 + 130t + 16,25$, **obtemos** $E(t) = 162,5t^3 - 243,75t^2 + 130t - 43,75$
   - **Fazendo um estudo de sinal no intervalo** $[0,\ 1] \subset \mathbb{R}$ **obtém-se:**

t	0	0,25	0,5	0,75	1
E(t)	-43,75	-23.945	-19.375	-14.805	5

   - **Como no intervalo** $[0,\ 1]$ **a função muda de sinal, ele é candidato a conter as raízes**

2. **Plotando o gráfico da função no intervalo** $[0,\ 1]$ **obtém-se:**



3. **Escolhendo o subintervalo $[0,75;\ 1]$ para o estudo numérico, para mostrar a existência da raiz nesse intervalo deve-se mostrar que $E(0{,}75) \cdot E(1) < \mathbf{0}$, e para mostrar que é única neste subintervalo deve- se mostrar que $E^1(0{,}75) \cdot E^1(1) > \mathbf{0}$. Como $E(0{,}75) = 162{,}5*0{,}75^3 - 243{,}75*0{,}75^2 + 130*0{,}75 - 43{,}75 = -14{,}805$ e $E(1) = 162{,}5*1^3 - 243{,}75*1^2 + 130*1 - 43{,}75 = 5$ de modo que $E(0{,}75) \cdot E(1) = -74{,}025 < \mathbf{0}$. Analogamente, $E^1(t) = 487{,}5t^2 - 487{,}5t + 130$, então $E^1(0{,}75) = 487{,}5*0{,}75^2 - 487{,}5*0{,}75 - 130 = 38.594$ e $E^1(1) = 487{,}5*1^2 - 487{,}5*1 - 130 = 130$, de modo que $E^1(0{,}75) \cdot E^1(1) = 5.017{,}22 > \mathbf{0}$. Assim, existe uma única raiz no subintervalo escolhido.**

4. **Para o subintervalo $[0,75;\ 1]$ obtém-se, via métodos de refinamento:**

*** APROXIMAÇÃO PELO MÉTODO DA BISSEÇÃO ***

EXERCÍCIO 01

k	xm	abs(bk-ak)	f(xm)
1	0.875000	0.250000	-7.758789
2	0.937500	0.125000	-2.212524
3	0.968750	0.062500	1.170578
4	0.953125	0.031250	-0.574903
5	0.960938	0.015625	0.284122
6	0.957031	0.007812	-0.148790
7	0.958984	0.003906	0.066812
8	0.958008	0.001953	-0.041202
9	0.958496	0.000977	0.012752
10	0.958252	0.000488	-0.014238
11	0.958374	0.000244	-0.000746
12	0.958435	0.000122	0.006002
13	0.958405	0.000061	0.002627
14	0.958389	0.000031	0.000940
15	0.958382	0.000015	0.000097
16	0.958378	0.000008	-0.000325
17	0.958380	0.000004	-0.000114
18	0.958381	0.000002	-0.000008

Aproximadamente: 0.958381 é a raiz, com 18 iterações

*** APROXIMAÇÃO PELO MÉTODO DA FALSA-POSIÇÃO ***

EXERCÍCIO 01

k	xk	abs(b-a)	f(xk)
01	0.936884	0.250000	-2.274965
02	0.956621	0.063116	-0.193878
03	0.958240	0.043379	-0.015547
04	0.958370	0.041760	-0.001241
05	0.958380	0.041630	-0.000099
06	0.958381	0.041620	-0.000008
07	0.958381	0.041619	-0.000001

Aproximação "0.958381" à raiz, com "07" iterações

*** APROXIMAÇÃO PELO MÉTODO DA SECANTE ***

EXERCÍCIO 01

k	xk	abs(xk-x1)	f(xk)
01	1.191848	0.320053	40.058936
02	0.925086	0.266761	-3.439143
03	0.946178	0.021091	-1.316127
04	0.959253	0.013075	0.096587
05	0.958359	0.000894	-0.002413
06	0.958381	0.000022	-0.000004

Aproximação "0.958381" à raiz, com "06" iterações

*** APROXIMAÇÃO PELO MÉTODO DE NEWTON-RAPHSON ***

EXERCÍCIO 01

k	xk	abs(xk-x0)	f(xk)	f1(xk)
01	0.976184	0.101184	2.040029	118.666402
02	0.958993	0.017191	0.067781	110.828907
03	0.958382	0.000612	0.000084	110.555394
04	0.958381	0.000001	0.000000	110.555056

Aproximação "0.958381" à raiz, com "04" iterações

5. **Como o valor médio das raízes é 0,958381, então $f(0{,}958381) = 162{,}5*0{,}958381^3 - 243{,}75*0{,}958381^2 + 130*0{,}958381 - 43{,}75 = 0.00002 \approx 0$, isso mostra que a solução aproximada é suficientemente boa para os mais diversos propósitos.**

O servidor atinge o consumo de 60w quando a taxa de utilização da CPU é de 0,958381

EXERCÍCIO 02 (ZERO DE FUNÇÕES): Durante testes de escalabilidade de uma API web, foi observado que a latência média de resposta $L(u)$ (em milissegundos, ms), depende da carga simultânea de usuários u (em centenas). Uma modelagem do comportamento empírico (que também foi obtida usando os dados e interpolação polinomial) gerou a função:

$$L(u) = 4c^3 - 30c^2 + 85c + 40$$

O time responsável pela rede estabeleceu que a latência aceitável não deve ultrapassar 200 ms para manter a qualidade do serviço. Então determine a carga de usuários simultâneos (em centenas) em que a latência atinge esse limiar.

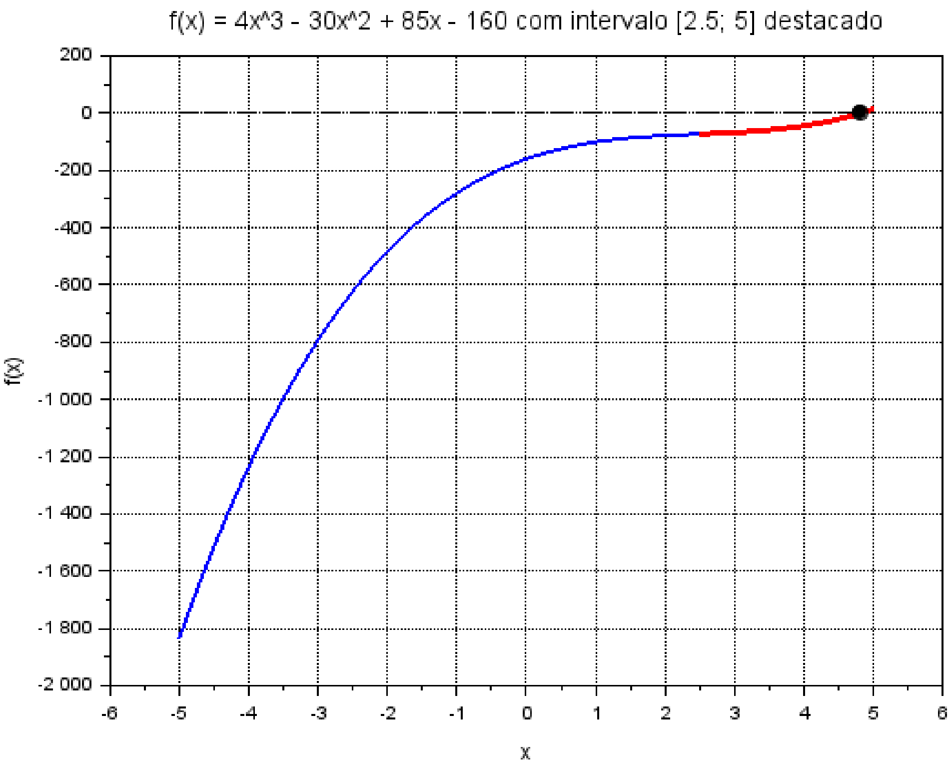
Solução:

1. **Análise por inspeção dos intervalos candidatos:**
   - Resolvendo $200 = 4u^3 - 30u^2 + 85u + 40$, obtemos $L(u) = 4u^3 - 30u^2 + 85u - 160$
   - Fazendo um estudo de sinal no intervalo $[-5; 5] \subset \mathbb{R}$ obtém-se:

u	-5	-2,5	0	2,5	5
L(u)	-1835	-622.5	-160	-72.5	15

   - Como no intervalo $[2,5; 5]$ a função muda de sinal, ele é candidato a conter as raízes

2. **Plotando o gráfico da função no intervalo $[-5; 5]$ obtém-se:**



3. **Escolhendo o subintervalo** $[2,5;\ 5]$ **para o estudo numérico, para mostrar a existência da raiz nesse intervalo deve-se mostrar que** $L(2,5) \cdot L(5) < \mathbf{0}$**, e para mostrar que é única neste subintervalo deve- se mostrar que** $E^1(0,75) \cdot E^1(1) > \mathbf{0}$**. Como** $L(2,5) = 4{*}2,5^3 - 30{*}2,5^2 + 85{*}2,5 - 160 = -72,5$ **e** $L(5)\ 4{*}5^3 - 30{*}5^2 + 85{*}5 - 160 = 15$ **de modo que** $L(2,5) \cdot L(5) = -1.087,5 < \mathbf{0}$**. Analogamente,** $L^1(t) = 12u^2 - 60u + 85$**, então** $L^1(2,5) = 12{*}2,5^2 - 60{*}2,5 + 85 = 10$ **e** $L^1(5) = 12u^2 - 60u + 85 = 85$**, de modo que** $L^1(2,5) \cdot L^1(5) = 850 > \mathbf{0}$**. Assim, existe uma única raiz no subintervalo escolhido.**

4. **Para o subintervalo** $[2,5;\ 5]$ **obtém-se, via métodos de refinamento:**

*** APROXIMAÇÃO PELO MÉTODO DA BISSEÇÃO ***

EXERCÍCIO 02

k	xm	abs(bk-ak)	f(xm)
1	3.750000	2.500000	-52.187500
2	4.375000	1.250000	-27.382812
3	4.687500	0.625000	-8.754883
4	4.843750	0.312500	2.435913
5	4.765625	0.156250	-3.325424
6	4.804688	0.078125	-0.486956
7	4.824219	0.039062	0.963839
8	4.814453	0.019531	0.235793
9	4.809570	0.009766	-0.126242
10	4.812012	0.004883	0.054610
11	4.810791	0.002441	-0.035857
12	4.811401	0.001221	0.009366
13	4.811096	0.000610	-0.013248
14	4.811249	0.000305	-0.001942
15	4.811325	0.000153	0.003712
16	4.811287	0.000076	0.000885
17	4.811268	0.000038	-0.000528
18	4.811277	0.000019	0.000178
19	4.811273	0.000010	-0.000175
20	4.811275	0.000005	0.000002
21	4.811274	0.000002	-0.000087
22	4.811274	0.000001	-0.000042

Aproximadamente: 4.811274 é a raiz, com 22 iterações

*** APROXIMAÇÃO PELO MÉTODO DA FALSA-POSIÇÃO ***

EXERCÍCIO 02

k	xk	abs(b-a)	f(xk)
01	4.571429	2.500000	-16.233236
02	4.794175	0.428571	-1.259063
03	4.810114	0.205825	-0.086005
04	4.811196	0.189886	-0.005822
05	4.811270	0.188804	-0.000394
06	4.811275	0.188730	-0.000027
07	4.811275	0.188725	-0.000002
08	4.811275	0.188725	-0.000000

Aproximação "4.811275" à raiz, com "08" iterações

*** APROXIMAÇÃO PELO MÉTODO DE NEWTON-RAPHSON ***

EXERCÍCIO 02

k	xk	abs(xk-x0)	f(xk)	f1(xk)
01	5.565217	1.815217	73.349881	122.746692
02	4.967646	0.597571	12.281199	83.071336
03	4.819807	0.147839	0.634282	74.578058
04	4.811302	0.008505	0.002011	74.105410
05	4.811275	0.000027	0.000000	74.103904

Aproximação "4.811275" à raiz, com "05" iterações

*** APROXIMAÇÃO PELO MÉTODO DA SECANTE ***

EXERCÍCIO 02

k	xk	abs(xk-x1)	f(xk)
01	4.794175	0.222747	-1.259063
02	4.812904	0.018729	0.120817
03	4.811265	0.001640	-0.000776
04	4.811275	0.000010	-0.000000

Aproximação "4.811275" à raiz, com "04" iterações

5. **Como o valor médio das raízes é 0,958381, então** $f(4,811275) = 4{*}4,811275^3 - 30{*}4,811275^2 + 85{*}4,811275 - 160 = 0.000001 \approx 0$**, isso mostra que a solução aproximada é suficientemente boa para os mais diversos propósitos.**

A latência da API web atinge $200\ \text{ms}$ **quando existem aproximadamente** 481 usuários **conectados.**

EXERCÍCIO 03 (ZERO DE FUNÇÕES): Durante o projeto de cabos metálicos de alta precisão, que são utilizados em equipamento específico para a transmissão de sinais de alta frequência, é essencial garantir que certos fios suportem adequadamente a pressão mecânica resultante de vibrações termomecânicas. Um engenheiro de computação, em parceria com a equipe de hardware, modelou que a pressão máxima suportada, $\mathbf{p(d)}$, (em $\mathbf{kg/mm^2}$) em função do diâmetro, $\mathbf{d}$, do condutor metálico (em milímetros, mm), é como na expressão empírica:

$$\mathbf{p(d) = 25d^2 + \ln(d)}$$

O diâmetro do cabo deve estar na faixa de valores $\mathbf{0,2 \leq d \leq 0,3}$ **mm** . Então determine uma aproximação para seu valor que garante suportar uma pressão de $\mathbf{p = 1,5\ kg/mm^2}$.

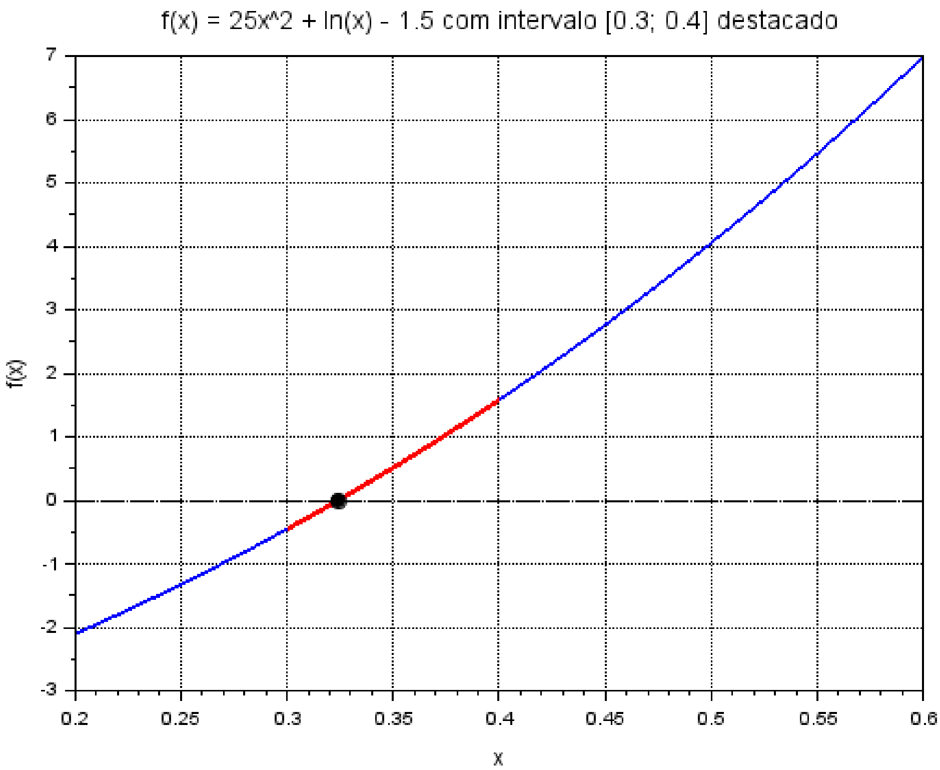
Solução:

1. **Análise por inspeção dos intervalos candidatos:**
   - **Resolvendo** $1,5 = 25d^2 + ln(d)$, **obtemos** $p(d) = 25d^2 + ln(d) - 1,5$
   - **Fazendo um estudo de sinal no intervalo** $[0,2;\ 0,6] \subset \mathbb{R}$ **obtém-se:**

d	0,2	0,3	0,4	0,5	0,6
p(d)	-2,109	-0,454	1,584	4,057	6,989

   - **No intervalo recomendado de** $[0,2;\ 0,3]$**, a função não muda de sinal, mas como no intervalo** $[0,3;\ 0,4]$ **ela muda, ele é candidato a conter as raízes.**

2. **Plotando o gráfico da função no intervalo** $[0,3;\ 0,4]$ **obtém-se:**



3. **Escolhendo o subintervalo $[0,3;\ 0,4]$ para o estudo numérico, para mostrar a existência da raiz nesse intervalo deve-se mostrar que $p(0,3) \cdot p(0,4) < \mathbf{0}$, e para mostrar que é única neste subintervalo deve- se mostrar que $p^1(0,3) \cdot p^1(0,4) > \mathbf{0}$. Como $p(0,3) = 25*0,3^2 + \ln(0,3) - 1,5 = -0,454$ e $p(0,4) = 25*0,4^2 + \ln(0,4) - 1,5 = 1,584$ de modo que $p(0,3) \cdot p(0,4) = -0,719 < \mathbf{0}$. Analogamente, $p^1(d) = 50d + 1/d$, então $p^1(0,3) = 50*0,3 + 1/0,3 = 18,333$ e $p^1(0,4) = 50*0,4 + 1/0,4 = 22,5$, de modo que $p^1(0,3) \cdot p^1(0,4) = 412,5 > \mathbf{0}$. Assim, existe uma única raiz no subintervalo escolhido.**

4. **Para o subintervalo $[0,3;\ 0,4]$ obtém-se, via métodos de refinamento:**

*** APROXIMAÇÃO PELO MÉTODO DA BISSEÇÃO ***

EXERCÍCIO 03

k	xm	abs(bk-ak)	f(xm)
1	0.350000	0.100000	0.512678
2	0.325000	0.050000	0.016695
3	0.312500	0.025000	-0.221745
4	0.318750	0.012500	-0.103309
5	0.321875	0.006250	-0.043504
6	0.323437	0.003125	-0.013454
7	0.324219	0.001563	0.001608
8	0.323828	0.000781	-0.005926
9	0.324023	0.000391	-0.002160
10	0.324121	0.000195	-0.000276
11	0.324170	0.000098	0.000666
12	0.324146	0.000049	0.000195
13	0.324133	0.000024	-0.000041
14	0.324139	0.000012	0.000077
15	0.324136	0.000006	0.000018
16	0.324135	0.000003	-0.000011
17	0.324136	0.000002	0.000004

Aproximadamente: 0.324136 é a raiz, com 17 iterações

*** APROXIMAÇÃO PELO MÉTODO DA FALSA-POSIÇÃO ***

EXERCÍCIO 03

k	xk	abs(b-a)	f(xk)
01	0.322279	0.100000	-0.035746
02	0.323994	0.077721	-0.002720
03	0.324125	0.076006	-0.000206
04	0.324135	0.075875	-0.000016
05	0.324135	0.075865	-0.000001
06	0.324135	0.075865	-0.000000

Aproximação "0.324135" à raiz, com "06" iterações

*** APROXIMAÇÃO PELO MÉTODO DE NEWTON-RAPHSON ***

EXERCÍCIO 03

k	xk	abs(xk-x0)	f(xk)	f1(xk)
01	0.324816	0.025184	0.013136	19.319459
02	0.324136	0.000680	0.000009	19.291920
03	0.324135	0.000000	0.000000	19.291900

Aproximação "0.324135" à raiz, com "03" iterações

*** APROXIMAÇÃO PELO MÉTODO DA SECANTE ***

EXERCÍCIO 03

k	xk	abs(xk-x1)	f(xk)
01	0.323994	0.001716	-0.002720
02	0.324136	0.000141	0.000005
03	0.324135	0.000000	-0.000000

Aproximação "0.324135" à raiz, com "03" iterações

5. **Como o valor médio das raízes é 0,324135, então $f(0,324135) = 25*0,324135^2 + \ln(0,324135) - 1,5 = -0.000008 \approx 0$, isso mostra que a solução aproximada é suficientemente boa para os mais diversos propósitos.**

Para suportar uma pressão de $1,5\ \mathrm{kg/mm^2}$, o cabo deve ter diâmetro de $0,324135\ \mathrm{mm}$

EXERCÍCIO 05 (RESOLUÇÃO DE SISTEMAS LINEARES): Você é responsável por abastecer uma loja de informática e precisa verificar o estoque dos quatro principais componentes vendidos: HDs (h), pendrives (p), SSDs (s) e memórias RAM (r). O almoxarife, que era aluno do Curso de Matemática, deixou indicações como deve ser abastecido o estoque. Ele indicou que:

- Três vezes o número de HDs, mais duas vezes o número de pendrives, menos o número de SSDs, mais o número de memórias RAM resulta em 9 componentes.
- Duas vezes o número de HDs, menos duas vezes o número de pendrives, mais quatro vezes o número de SSDs, menos três vezes o número de memórias RAM resulta em 11 componentes.
- O número de HDs, mais o número de pendrives, mais o número de SSDs, menos o número de memórias RAM totaliza 8 unidades.
- Duas vezes o número de HDs, mais três vezes o número de pendrives, mais o número de SSDs, mais quatro vezes o número de memórias RAM somam 21 componentes.

SOLUÇÃO:

....

Sistema obtido:

$$3h + 2p - s + r = 9$$

$$2h - 2p + 4s - 3r = 11$$

$$h + p + s - r = 8$$

$$2h + 3p + s + 4r = 21$$

Método de Gauss sem pivoteamento:

```
EXERCÍCIO 05
*** MÉTODO DIRETO: GAUSS (ELIMINAÇÃO GAUSSIANA) SEM PIVOTEAMENTO ***

Entrada - Matriz A (original):
   3.   2.  -1.   1.
   2.  -2.   4.  -3.
   1.   1.   1.  -1.
   2.   3.   1.   4.

Entrada - Vetor B (original):
   9.
   11.
   8.
   21.

Dimensão de n: 4 variáveis

****Matriz A triangularizada:****
   3.   2.          -1.          1.
   0.  -3.3333333   4.6666667  -3.6666667
   0.   0.          1.8        -1.7
   0.   0.          0.          5.2777778

****Vetor B escalonado:****
   9.
   5.
   5.5
   5.2777778

Solução X do Sistema:
 x(1) = 2.000000
 x(2) = 3.000000
 x(3) = 4.000000
 x(4) = 1.000000

Verificação dos resultados (AX = B):
 (3*2.000000) +  (2*3.000000) +  (-1*4.000000) +  (1*1.000000) =  9.000000
 (2*2.000000) +  (-2*3.000000) +  (4*4.000000) +  (-3*1.000000) =  11.000000
 (1*2.000000) +  (1*3.000000) +  (1*4.000000) +  (-1*1.000000) =  8.000000
 (2*2.000000) +  (3*3.000000) +  (1*4.000000) +  (4*1.000000) =  21.000000

Erro absoluto (AX - B):
   0.
   0.
   0.
   0.

 ********** ELIMINAÇÃO GAUSSIANA FINALIZADA **********
```

Método da Fatoração LU por Crout:

```
EXERCÍCIO 05
***** MÉTODO DIRETO: FATORAÇÃO LU por CROUT *****
Entrada - Matriz A (original):
   3.   2.  -1.   1.
   2.  -2.   4.  -3.
   1.   1.   1.  -1.
   2.   3.   1.   4.

Entrada - Vetor B (original):
   9.
   11.
   8.
   21.

Dimensão de n: 4 variáveis

*****FATOR L:*****
   3.   0.          0.    0.
   2.  -3.3333333   0.    0.
   1.   0.3333333   1.8   0.
   2.   1.6666667   4.    5.2777778

*****FATOR U:*****
   1.   0.6666667  -0.3333333   0.3333333
   0.   1.         -1.4         1.1
   0.   0.          1.         -0.9444444
   0.   0.          0.          1.

Solução Y de LY=B:
   3.
  -1.5000000
   3.0555556
   1.

Solução X (UX = Y):
 x(1) = 2.000000
 x(2) = 3.000000
 x(3) = 4.000000
 x(4) = 1.000000

Verificação dos resultados (A*X ≈ B):
(3*2.000000) + (2*3.000000) + (-1*4.000000) + (1*1.000000) = 9.000000
(2*2.000000) + (-2*3.000000) + (4*4.000000) + (-3*1.000000) = 11.000000
(1*2.000000) + (1*3.000000) + (1*4.000000) + (-1*1.000000) = 8.000000
(2*2.000000) + (3*3.000000) + (1*4.000000) + (4*1.000000) = 21.000000

Erro absoluto (AX - B):
   0.
   0.
   0.
   0.

 ********** FATORAÇÃO LU FINALIZADA **********
```

- **O método de TDMA não é aplicável pois a matriz A do sistema não é tridiagonal**
- **O método iterativo de Gauss-Jacobi diverge ao infinito, pois o sistema não possui diagonal dominante**

Método iterativo de Gauss-Seidel:

```
EXERCÍCIO 05
*** MÉTODO ITERATIVO: GAUSS-SEIDEL (REORDENAÇÃO GULOSA) ***

 Matriz A original:
   3.   2.  -1.   1.
   2.  -2.   4.  -3.
   1.   1.   1.  -1.
   2.   3.   1.   4.

 Vetor B original:
   9.
   11.
   8.
   21.

 Entrada - Dimensão n da matriz quadrada: 4
 Reordenação Gulosa aplicada com sucesso.

 Ordem das linhas escolhida:
   1.   4.   2.   3.

 Matriz A após reordenação:
   3.   2.  -1.   1.
   2.   3.   1.   4.
   2.  -2.   4.  -3.
   1.   1.   1.  -1.

 Vetor B após reordenação:
   9.
   21.
   11.
   8.

 Número de iterações: 538

 Vetor solução aproximada:
 x(1) = 2.000000
 x(2) = 2.999999
 x(3) = 4.000000
 x(4) = 0.999999

 Verificação dos resultados (A*X ≈ B):
 (3.0*2.000000) + (2.0*2.999999) + (-1.0*4.000000) + (1.0*0.999999) = 8.999998
 (2.0*2.000000) + (3.0*2.999999) + (1.0*4.000000) + (4.0*0.999999) = 20.999996
 (2.0*2.000000) + (-2.0*2.999999) + (4.0*4.000000) + (-3.0*0.999999) = 11.000002
 (1.0*2.000000) + (1.0*2.999999) + (1.0*4.000000) + (-1.0*0.999999) = 8.000000

***** ENCERRAMENTO DO GAUSS-SEIDEL COM MÉTODO GULOSO *****
```

EXERCÍCIO 06 (RESOLUÇÃO DE SISTEMAS LINEARES): Um problema de condução de calor (muito simplificado mas utilizado em alguns equipamentos de informática) é o definido em uma barra unidimensional (1D) isolada termicamente nas laterais, mas não nas extremidades. A equação diferencial para condução de calor 1D (temperatura $\mathbf{T}$ $^{\mathbf{o}}\mathbf{C}$) em estado estacionário com sua respectiva discretização em diferenças finitas são, respectivamente, como:

$$\frac{d^2T}{dx^2} = 0 \Rightarrow T_{i-1} - 2\,T_i + T_{i+1} = 0$$

Considere que as temperatura nas extremidades da barra são conhecidas, sendo $b_0 = 100$ e $b_6 = 200$. As temperaturas em todos os elementos discretizado com $n = 5$ pontos são obtidas com a solução de um sistema tridiagonal. Resolva o sistema para mostrar as temperaturas em cada ponto. Inicie a solução com $i = 1$ e considere que $T_0 = 0 = T_6$.

Solução:

Para i = 1:

$b_0 - 2b_1 + b_2 = 0$

$100 - 2b_1 + b_2 = 0$

$-2b_1 + b_2 = -100$

Para i = 2:

$b_1 - 2b_2 + b_3 = 0$

Para i = 3:

$b_2 - 2b_3 + b_4 = 0$

Para i = 4:

$b_3 - 2b_4 + b_5 = 0$

Para i = 5:

$b_4 - 2b_5 + b_6 = 0$

$b_4 - 2b_5 + 200 = 0$

$b_4 - 2b_5 = -200$

Sistema obtido:

$$
\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -100 \\ 0 \\ 0 \\ 0 \\ -200 \end{bmatrix}
$$

Método de Gauss sem pivoteamento:

```
EXERCÍCIO 06
*** MÉTODO DIRETO: GAUSS (ELIMINAÇÃO GAUSSIANA) SEM PIVOTEAMENTO ***

Entrada - Matriz A (original):
  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

Entrada - Vetor B (original):
  -100.
   0.
   0.
   0.
  -200.

Dimensão de n: 5 variáveis

****Matriz A triangularizada:****
  -2.   1.    0.          0.     0.
   0.  -1.5   1.          0.     0.
   0.   0.   -1.3333333   1.     0.
   0.   0.    0.         -1.25   1.
   0.   0.    0.          0.    -1.2

****Vetor B escalonado:****
  -100.
  -50.
  -33.333333
  -25.000000
  -220.

Solução X do Sistema:
 x(1) = 116.666667
 x(2) = 133.333333
 x(3) = 150.000000
 x(4) = 166.666667
 x(5) = 183.333333

Verificação dos resultados (AX = B):
 (-2*116.666667) +  (1*133.333333) +  (0*150.000000) +  (0*166.666667) +  (0*183.333333) =  -100.000000
 (1*116.666667) +  (-2*133.333333) +  (1*150.000000) +  (0*166.666667) +  (0*183.333333) =  0.000000
 (0*116.666667) +  (1*133.333333) +  (-2*150.000000) +  (1*166.666667) +  (0*183.333333) =  0.000000
 (0*116.666667) +  (0*133.333333) +  (1*150.000000) +  (-2*166.666667) +  (1*183.333333) =  -0.000000
 (0*116.666667) +  (0*133.333333) +  (0*150.000000) +  (1*166.666667) +  (-2*183.333333) =  -200.000000

Erro absoluto (AX - B):
   0.
   0.
   5.684D-14
  -5.684D-14
   0.

 ********** ELIMINACÃO GAUSSIANA FINALIZADA **********
```

Método da Fatoração LU por Crout:

```
EXERCÍCIO 06
***** MÉTODO DIRETO: FATORAÇÃO LU por CROUT *****
Entrada - Matriz A (original):
  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

Entrada - Vetor B (original):
  -100.
   0.
   0.
   0.
  -200.

Dimensão de n: 5 variáveis

*****FATOR L:*****
  -2.   0.    0.          0.     0.
   1.  -1.5   0.          0.     0.
   0.   1.   -1.3333333   0.     0.
   0.   0.    1.         -1.25   0.
   0.   0.    0.          1.    -1.2

*****FATOR U:*****
   1.  -0.5   0.          0.     0.
   0.   1.   -0.6666667   0.     0.
   0.   0.    1.         -0.75   0.
   0.   0.    0.          1.    -0.8
   0.   0.    0.          0.     1.

Solução Y de LY=B:
   50.
   33.333333
   25.
   20.
   183.33333

Solução X (UX = Y):
 x(1) = 116.666667
 x(2) = 133.333333
 x(3) = 150.000000
 x(4) = 166.666667
 x(5) = 183.333333

Verificação dos resultados (A*X ≈ B):
(-2*116.666667) + (1*133.333333) + (0*150.000000) + (0*166.666667) + (0*183.333333) = -100.000000
(1*116.666667) + (-2*133.333333) + (1*150.000000) + (0*166.666667) + (0*183.333333) = 0.000000
(0*116.666667) + (1*133.333333) + (-2*150.000000) + (1*166.666667) + (0*183.333333) = 0.000000
(0*116.666667) + (0*133.333333) + (1*150.000000) + (-2*166.666667) + (1*183.333333) = -0.000000
(0*116.666667) + (0*133.333333) + (0*150.000000) + (1*166.666667) + (-2*183.333333) = -200.000000

Erro absoluto (AX - B):
   0.
   0.
   2.842D-14
  -2.842D-14
   0.

 ********** FATORACÃO LU FINALIZADA **********
```

Método de Thomas (TDMA):

```
EXERCÍCIO 06
*** MÉTODO DIRETO: THOMAS (TDMA) - SISTEMAS TRIDIAGONAIS ***

 Vetor a^T:
   1.
   1.
   1.
   1.
   1.

 Vetor b^T:
  -2.
  -2.
  -2.
  -2.
  -2.

 Vetor c^T:
   1.
   1.
   1.
   1.
   1.

 Vetor d^T:
  -100.   0.   0.   0.  -200.

O sistema possui 5 variáveis (dimensão da raiz x).

 Solução X do sistema:
  x(1) = 116.666667
  x(2) = 133.333333
  x(3) = 150.000000
  x(4) = 166.666667
  x(5) = 183.333333

Verificação dos resultados (A*X ≈ d):
 (-2.0*116.666667) + (1.0*133.333333) + (0.0*150.000000) + (0.0*166.666667) + (0.0*183.333333) = -100.000000
 (1.0*116.666667) + (-2.0*133.333333) + (1.0*150.000000) + (0.0*166.666667) + (0.0*183.333333) = 0.000000
 (0.0*116.666667) + (1.0*133.333333) + (-2.0*150.000000) + (1.0*166.666667) + (0.0*183.333333) = 0.000000
 (0.0*116.666667) + (0.0*133.333333) + (1.0*150.000000) + (-2.0*166.666667) + (1.0*183.333333) = -0.000000
 (0.0*116.666667) + (0.0*133.333333) + (0.0*150.000000) + (1.0*166.666667) + (-2.0*183.333333) = -200.000000

 Erro absoluto (A*X - d):

   0.
   0.
   2.842D-14
  -2.842D-14
   0.

 ********** TDMA FINALIZADO **********
```

Método iterativo de Gauss-Jacobi:

```
EXERCÍCIO 06
*** MÉTODO ITERATIVO: GAUSS-JACOBI (REORDENAÇÃO GULOSA) ***

 Matriz A original:

  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

 Vetor B original:

  -100.
   0.
   0.
   0.
  -200.

 Entrada - Dimensão n da matriz quadrada: 5
 Reordenação Gulosa aplicada com sucesso.

 Ordem das linhas escolhida:
   1.   2.   3.   4.   5.

 Matriz A após reordenação:
  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

 Vetor B após reordenação:
  -100.
   0.
   0.
   0.
  -200.

 Saída número de iterações: 124

 Saída - Vetor Solução aproximada:
 x(1) = 116.666665
 x(2) = 133.333331
 x(3) = 149.999996
 x(4) = 166.666664
 x(5) = 183.333332

 Verificação dos resultados (A*X ≈ B):
 (-2.0*116.666665) + (1.0*133.333331) + (0.0*149.999996) + (0.0*166.666664) + (0.0*183.333332) = -99.999999
 (1.0*116.666665) + (-2.0*133.333331) + (1.0*149.999996) + (0.0*166.666664) + (0.0*183.333332) = -0.000000
 (0.0*116.666665) + (1.0*133.333331) + (-2.0*149.999996) + (1.0*166.666664) + (0.0*183.333332) = 0.000002
 (0.0*116.666665) + (0.0*133.333331) + (1.0*149.999996) + (-2.0*166.666664) + (1.0*183.333332) = 0.000000
 (0.0*116.666665) + (0.0*133.333331) + (0.0*149.999996) + (1.0*166.666664) + (-2.0*183.333332) = -199.999999

***** ENCERRAMENTO DO GAUSS-JACOBI COM MÉTODO GULOSO *****
```

Método iterativo de Gauss-Seidel:

```
EXERCÍCIO 06
*** MÉTODO ITERATIVO: GAUSS-SEIDEL (REORDENAÇÃO GULOSA) ***

 Matriz A original:
  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

 Vetor B original:
  -100.
   0.
   0.
   0.
  -200.

 Entrada - Dimensão n da matriz quadrada: 5
 Reordenação Gulosa aplicada com sucesso.

 Ordem das linhas escolhida:
   1.   2.   3.   4.   5.

 Matriz A após reordenação:
  -2.   1.   0.   0.   0.
   1.  -2.   1.   0.   0.
   0.   1.  -2.   1.   0.
   0.   0.   1.  -2.   1.
   0.   0.   0.   1.  -2.

 Vetor B após reordenação:
  -100.
   0.
   0.
   0.
  -200.

 Número de iterações: 64

 Vetor solução aproximada:
 x(1) = 116.666665
 x(2) = 133.333331
 x(3) = 149.999998
 x(4) = 166.666665
 x(5) = 183.333332

 Verificação dos resultados (A*X ≈ B):
 (-2.0*116.666665) + (1.0*133.333331) + (0.0*149.999998) + (0.0*166.666665) + (0.0*183.333332) = -99.999999
 (1.0*116.666665) + (-2.0*133.333331) + (1.0*149.999998) + (0.0*166.666665) + (0.0*183.333332) = 0.000001
 (0.0*116.666665) + (1.0*133.333331) + (-2.0*149.999998) + (1.0*166.666665) + (0.0*183.333332) = 0.000001
 (0.0*116.666665) + (0.0*133.333331) + (1.0*149.999998) + (-2.0*166.666665) + (1.0*183.333332) = 0.000000
 (0.0*116.666665) + (0.0*133.333331) + (0.0*149.999998) + (1.0*166.666665) + (-2.0*183.333332) = -200.000000

***** ENCERRAMENTO DO GAUSS-SEIDEL COM MÉTODO GULOSO *****
```