Análise Numérica – Lista 2

*Patrick Duarte Pimenta*

*Questão 1.*

*// Nome: Patrick Duarte - 2023 (Ciência da computação)*

clc

*// Todo: parâmetros de entrada: n, A, b, Toler, IterMax (ordem, matriz, vetor independente, tolerância e número maximo de iterações)*

*// Todo: Parâmetros de saída: x, Iter, Erro (vetor solução, número de iterações e condição de erro)*

function [**x**, **Iter**, **Erro**]=gauss\_Seidel(**n**, **A**, **b**, **Toler**, **IterMax**)

*// Construção das matrizes para as iterações*

for i = 1 : **n**

r = 1 / **A**(i, i)

for j = 1 : **n**

if i ~= j then

**A**(i, j) = **A**(i, j) \* r

end

end

**b**(i) = **b**(i) \* r

**x**(i) = **b**(i)

end

*// Inicializando um vetor nulo*

**x** = zeros(length(**b**), 1);;

*// Interações de Gauss-Seidel*

for **Iter** = 1 : **IterMax**

for i = 1 : **n**

Soma = 0

for j = 1 : **n**

if i ~= j then

Soma = Soma + **A**(i, j) \* **x**(j)

end

end

v(i) = **x**(i)

**x**(i) = **b**(i) - Soma

end

Norma1 = 0

Norma2 = 0

for i = 1 : **n**

if abs(**x**(i) - v(i)) > Norma1 then

Norma1 = abs(**x**(i) - v(i))

end

if abs(**x**(i)) > Norma2 then

Norma2 = abs(**x**(i))

end

end

DifMax = Norma1 / Norma2

*//disp("Interação: " + string(Iter) + " - x: " + string(x) + " - DifMax = " + string(DifMax))*

*//disp(string(DifMax) + " < " + string(Toler));*

if DifMax < **Toler** || **Iter** >= **IterMax** then

*//disp("Convergência alcançada após " + string(Iter) + " iterações.")*

break

end

end

**Erro** = DifMax >= **Toler**

*// Variável lógica: Se verdadeiro há erro e se falso não há erro*

endfunction

*// Resolve por Gauss Ingenua.*

*// TODO: Retorna um vetor de solucoes. Recebe como parâmetro uma matriz A e seu vetorde termos independentes*

function [**x**]=gaussElimIngenua(**A**, **b**);

*// número de linha (e colunas) da matriz A*

n = size(**A**, 1);

*// etapa = quantidade de etapas (de 1 até (:) n - 1)*

for etapa = 1 : (n - 1)

*// i = linhas alteradas na etapa k de (k + 1 até n)*

for i = (etapa + 1) : n

*// Para cada linha deve ser calculado o multiplicador*

multiplicador = **A**(i, etapa) / **A**(etapa, etapa);

**A**(i, etapa) = 0

*// j = elementos que serão alterados na linha i na etapa k de k até n*

for j = (etapa + 1) : n

*// Atualizar o coeficiente a[i, j]*

**A**(i, j) = **A**(i, j) - multiplicador \* **A**(etapa, j);

end

*// Atualizar o coeficiente b[i];*

**b**(i) = **b**(i) - multiplicador \* **b**(etapa);

end

*//printf("\nEtapa: %d", etapa);*

*//disp([A, b]);*

end

*//disp("Matriz triangular superior: ")*

*//disp([A, b]);*

*// Substituição retroativa*

**x** = zeros(n, 1);

**x**(n) = **b**(n)/ **A**(n, n);

for i = (n - 1) : -1 : 1

soma = 0;

for j = (i + 1) : n

soma = soma + **A**(i, j) \* **x**(j);

end

**x**(i) = (**b**(i) - soma) / **A**(i, i);

end

endfunction

*// Número de linhas e colunas*

n = 10;

*// Matriz de 0 de ordem 10*

A = zeros(n - 1, n - 1);

*// Vetor de termos independentes por zeros*

b = zeros(n - 1, 1);

h= 1 / n;

*// Constrói o sistema de equações discretizadas*

for i=(1 : n - 1)

if (i == 1) then

A(i, 1) = ((h^2) - 2);

A(i, 2) =(1 + h);

b(i) = h^3;

elseif (i== (n - 1)) then

A(i, n - 2) = (1 - h);

A(i, n - 1) = ((h^2) - 2);

b(i) = (n - 1) \* h^3 + (h + 1);

else

A(i, i - 1) = (1 - h);

A(i, i) = ((h^2) - 2);

A(i, i + 1) = (1 + h);

b(i) = h^3;

end

end

disp(" ### Letra A ###");

disp("Matriz, aumentada: ");

disp([A, b]);

[x] = gaussElimIngenua(A, b);

disp("- ELIMINAÇÃO DE GAUSS (ingênua) -");

disp("x = ");

disp(x);

[x2, Iter, Erro] = gauss\_Seidel(n-1, A, b, 10e-3, 100)

disp("- GAUSS SEIDEL -");

disp("x2 = ");

disp(x2);

disp(" ### Letra B ###");

solReal = [

-0.2720;

-0.4911;

-0.6641;

-0.7969;

-0.8947;

-0.9620;

-1.0029;

-1.0208;

-1.0190

];

disp("- Comparativo entre x (gaussIngenua), x2 (gauss\_Seidel) e solReal (tabela)");

disp([x, x2, solReal]);

xData = linspace(1,9,9);

y1Data = x;

y2Data = x2;

y3Data = solReal;

xlabel("n");

ylabel("x(n)");

title("Comparativo entre Eliminação de Gauss Ingênua (azul), Gauss Seidel (vermelho) e solReal (verde)");

plot(xData, y1Data,'x-b');

plot(xData,y2Data,'x-r');

plot(xData,y3Data,'x-g');

a)

"Matriz, aumentada: "

-1.99 1.1 0. 0. 0. 0. 0. 0. 0. 0.001

0.9 -1.99 1.1 0. 0. 0. 0. 0. 0. 0.001

0. 0.9 -1.99 1.1 0. 0. 0. 0. 0. 0.001

0. 0. 0.9 -1.99 1.1 0. 0. 0. 0. 0.001

0. 0. 0. 0.9 -1.99 1.1 0. 0. 0. 0.001

0. 0. 0. 0. 0.9 -1.99 1.1 0. 0. 0.001

0. 0. 0. 0. 0. 0.9 -1.99 1.1 0. 0.001

0. 0. 0. 0. 0. 0. 0.9 -1.99 1.1 0.001

0. 0. 0. 0. 0. 0. 0. 0.9 -1.99 1.109

"- ELIMINAÇÃO DE GAUSS (ingênua) -"

"x = "

-0.254

-0.459

-0.622

-0.749

-0.844

-0.914

-0.962

-0.991

-1.006

"- GAUSS SEIDEL -"

"x2 = "

-0.209

-0.386

-0.535

-0.660

-0.764

-0.849

-0.914

-0.962

-0.992

b)

"- Comparativo entre x (gaussIngenua), x2 (gauss\_Seidel) e solReal (tabela)"

-0.254 -0.209 -0.272

-0.459 -0.386 -0.491

-0.622 -0.535 -0.664

-0.749 -0.66 -0.797

-0.844 -0.764 -0.895

-0.914 -0.849 -0.962

-0.962 -0.914 -1.003

-0.991 -0.962 -1.021

-1.006 -0.992 -1.019

Gráfico:

