Relatório Lab 3 - ALN

Eduarda Mesquita

April 2024

1 Criando a função que implementa o método da potência:

```
function [lambda, x1, k, n_erro] = Metodo_potencia(A, x0, epsilon,
    M)
    // Inicializa  o de vari veis
    k = 0;
    n = length(x0);
    x0 = x0 / norm(x0, 'inf');
    x1 = A * x0;
    n_erro = epsilon + 1;

    // Loop principal
    while k <= M & n_erro >= epsilon
        // Calculando lambda usando o Quociente de Rayleigh
        lambda = x1' * x0 / (x0' * x0);
        // Corrigindo o sinal de x1, se necess rio
        if lambda < 0 then
            x1 = -x1;
        end
        // Normalizando x1
        x1 = x1 / norm(x1, 2);
        // Calculando o erro
        n_erro = norm(x1 - x0, 'inf');
        // Atualizando x0 para o pr ximo ciclo
        x0 = x1;
        // Atualizando x1
        x1 = A * x0;
        // Atualizando o contador de itera  es
        k = k + 1;
    end

    // Mensagem de retorno
    if n_erro < epsilon then
        disp('Converg ncia alcan ada.');
    else
        disp('N mero m ximo de itera  es atingido.');
    end

    // Retorno dos resultados
    lambda = x1' * x0 / (x0' * x0); // lambda final
    x1 = x1 / norm(x1, 2); // autovetor final normalizado
```

```
endfunction
```

O código apresenta uma implementação do Método da Potência, uma técnica fundamental na computação numérica para estimar o autovalor principal e o autovetor associado de uma matriz quadrada A. No início do algoritmo, as variáveis são inicializadas, incluindo o contador de iterações, o tamanho do vetor inicial e a normalização desse vetor. Em seguida, o método entra em um loop principal, que continua até que um critério de parada seja atendido, geralmente um número máximo de iterações ou uma tolerância para o erro.

Dentro do loop, o código calcula o autovalor estimado utilizando o Quociente de Rayleigh, ajusta o sinal do autovetor se necessário e o normaliza. O erro entre iterações sucessivas é avaliado para verificar a convergência. Ao final do processo iterativo, o algoritmo retorna o autovalor e o autovetor final, além de uma mensagem indicando se a convergência foi alcançada dentro da tolerância especificada. O Método da Potência é amplamente utilizado em problemas de análise numérica, física computacional e outras áreas, oferecendo uma abordagem eficaz para encontrar os maiores autovalores e seus correspondentes autovetores de matrizes simétricas e não simétricas.

2 Criando a função que implementa o método da potência deslocada com iteração inversa

```

function [lambda1, x1, k, n_erro] = Potencia_deslocada_inversa(A, x0, epsilon, alfa, M)
    // Inicializa  o de vari veis
    k = 0;
    n = length(x0);
    x0 = x0 / norm(x0, 2);
    n_erro = epsilon + 1;

    // Loop principal
    while k <= M & n_erro >= epsilon
        // Resolvendo o sistema (A - alfa*I) * x1 = x0
        x1 = linsolve(A - alfa*eye(n), x0);
        // Normalizando x1
        x1 = x1 / norm(x1, 2);
        // Calculando lambda usando o Quociente de Rayleigh
        lambda = x1' * A * x1;
        // Corrigindo o sinal de x1, se necess rio
        if x1' * x0 < 0 then
            x1 = -x1;
        end
        // Calculando o erro
        n_erro = norm(x1 - x0, 2);
        // Atualizando x0 para o pr ximo ciclo
        x0 = x1;
        // Atualizando o contador de itera   es
        k = k + 1;
    end

```

```
    // Mensagem de retorno
    if n_erro < epsilon then
        disp('Converg ncia alcan ada.');
    else
        disp('N mero m ximo de itera  es atingido.');
    end

    // Retorno dos resultados
    lambda1 = lambda; // autovalor mais pr ximo de alfa
endfunction
```

Este código implementa o Método da Potência Deslocada com Iteração Inversa, uma técnica eficiente para encontrar o autovalor mais próximo de um valor desejado α e seu respectivo autovetor correspondente de uma matriz A. Inicialmente, as variáveis são inicializadas, incluindo o contador de iterações, o tamanho do vetor inicial e a normalização desse vetor. Em seguida, o algoritmo entra em um loop principal, que continua até que um critério de parada seja alcançado, como o número máximo de iterações ou uma tolerância para o erro.

Dentro do loop, o código resolve o sistema de equações $(A - \alpha I) \cdot x1 = x0$, onde A é a matriz original, α é o valor desejado de autovalor e I é a matriz identidade. O autovetor $x1$ é normalizado e o autovalor é calculado usando o Quociente de Rayleigh. Se necessário, o sinal do autovetor é corrigido. O erro entre iterações sucessivas é avaliado para verificar a convergência. Ao final do processo iterativo, o algoritmo retorna o autovalor mais próximo de α. Uma mensagem é exibida indicando se a convergência foi alcançada dentro da tolerância especificada. Essa abordagem é útil em várias aplicações, incluindo análise numérica e física computacional, proporcionando uma maneira eficaz de encontrar autovalores específicos de uma matriz.

3 Testando as funções anteriores para várias matrizes

```
// Defini  o da matriz A de ordem 3
A3 = [4 1 2; 1 5 3; 2 3 6];

// Vetor inicial x0
x0 = [1; 1; 1];

// Par metros comuns para os testes
epsilon = 1e-6; // precis o
M = 1000; // n mero m ximo de itera  es

// Testando M todo da Pot ncia
tic();
[lambda_potencia, x1_potencia, k_potencia, n_erro_potencia] =
    Metodo_potencia(A3, x0, epsilon, M);
tempo_potencia = toc();

disp('Resultado do M todo da Pot ncia:');
disp(lambda_potencia);
```

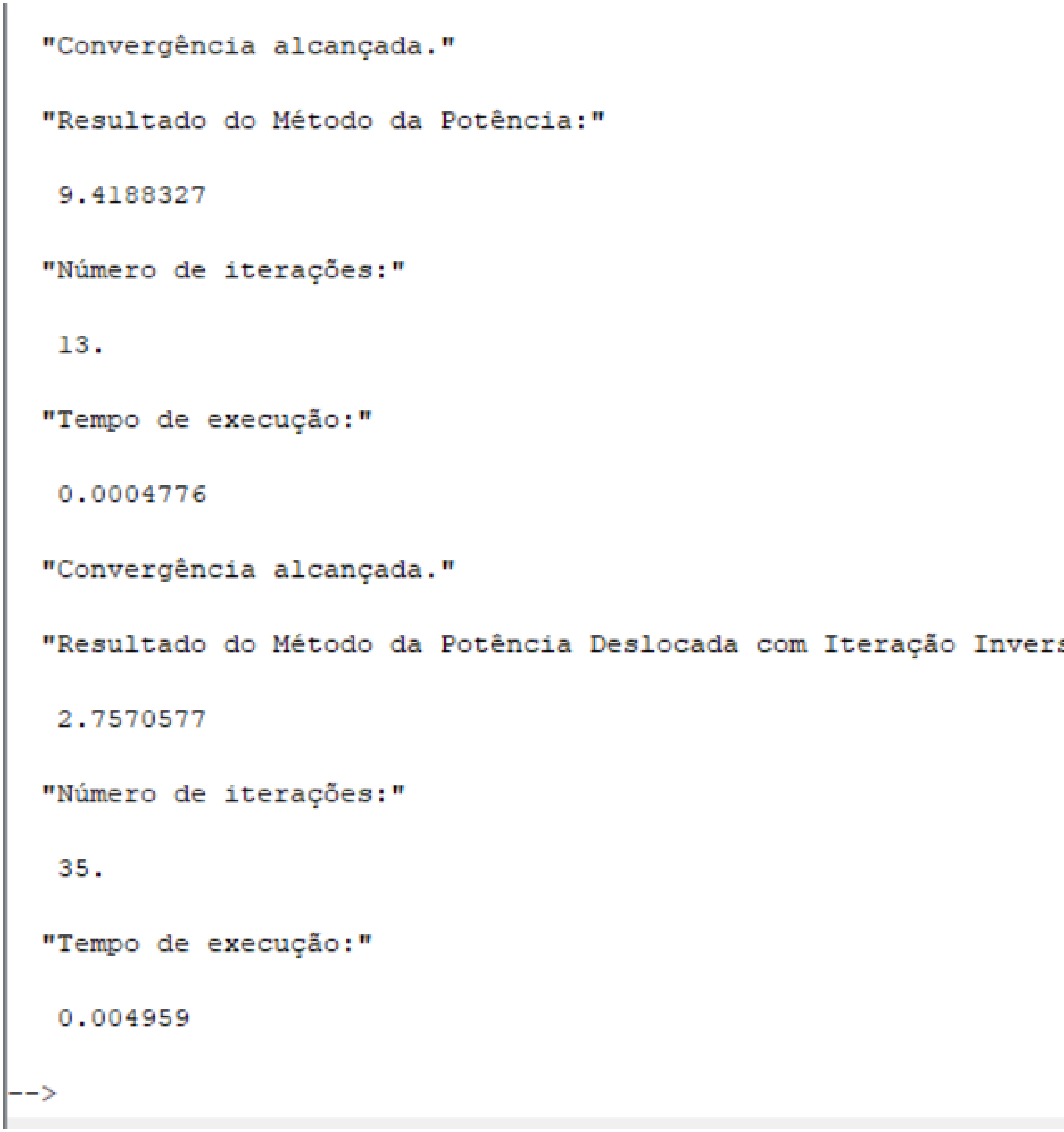
```
disp('N mero de itera  es:');
disp(k_potencia);
disp('Tempo de execu  o:');
disp(tempo_potencia);

// Testando M todo da Pot ncia Deslocada com Itera  o Inversa
alfa = 5; // valor desejado de autovalor
tic();
[lambda_deslocada, x1_deslocada, k_deslocada, n_erro_deslocada] = Potencia_deslocada_inversa(A3, x0, epsilon, alfa, M);
tempo_deslocada = toc();

disp('Resultado do M todo da Pot ncia Deslocada com Itera  o Inversa:');
disp(lambda_deslocada);
disp('N mero de itera  es:');
disp(k_deslocada);
disp('Tempo de execu  o:');
disp(tempo_deslocada);
```

O código realiza uma análise comparativa entre dois métodos numéricos para encontrar autovalores e autovetores de uma matriz A de ordem 3. Primeiramente, é definida a matriz A, seguida pelo vetor inicial x0 e pelos parâmetros comuns de precisão e número máximo de iterações.

Em seguida, o código testa o Método da Potência, um algoritmo iterativo que estima o maior autovalor e seu autovetor correspondente. O tempo de execução, o autovalor encontrado, o número de iterações e a precisão alcançada são exibidos. Posteriormente, o código testa o Método da Potência Deslocada com Iteração Inversa, onde um valor desejado de autovalor é especificado. Novamente, são exibidos os resultados semelhantes aos do Método da Potência. Essa abordagem permite comparar a eficiência e a precisão dos dois métodos na obtenção dos autovalores desejados para a matriz A.

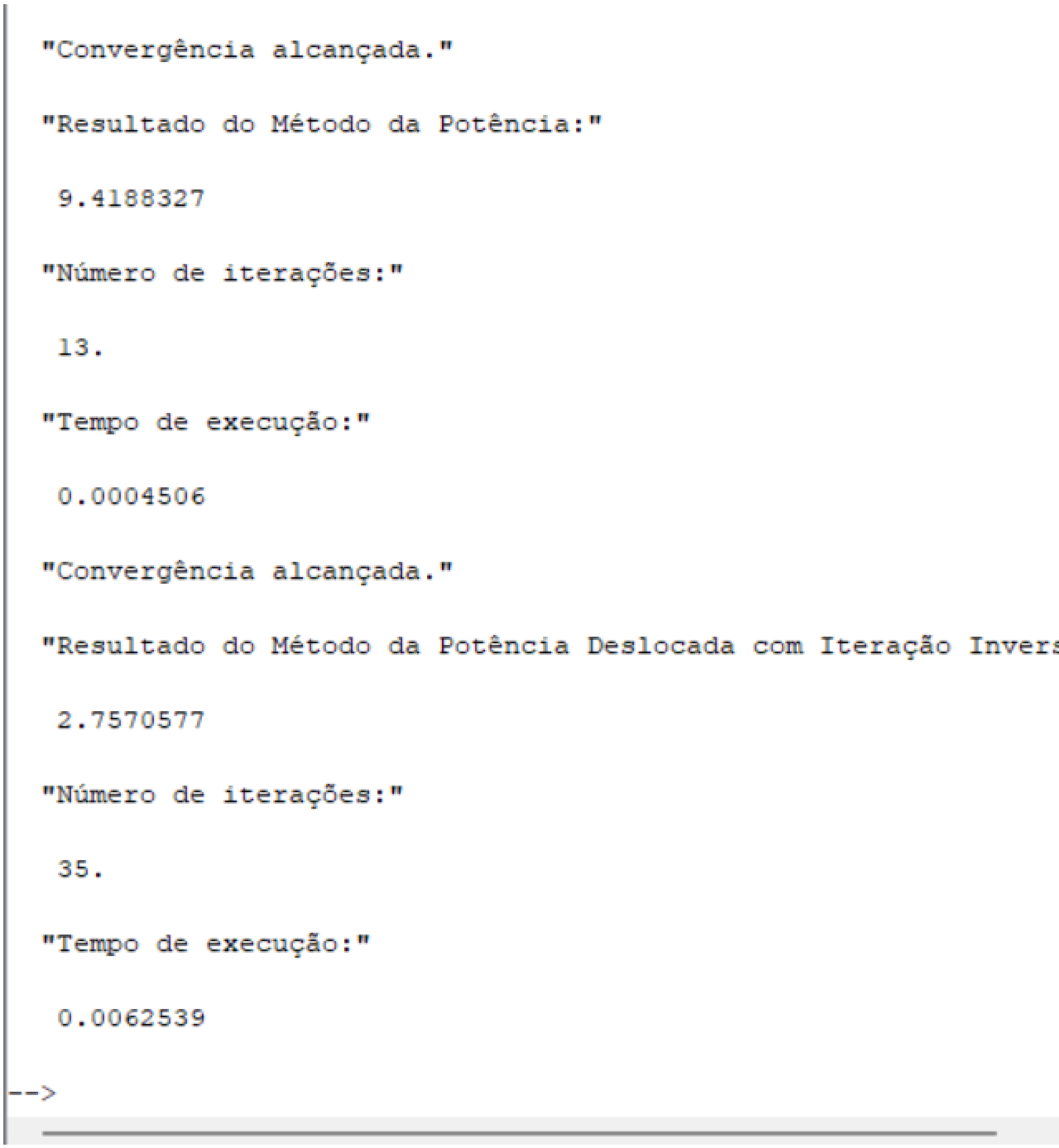
4 Construindo uma matriz simétrica e usando os Discos de Gerschgorin para estimar os autovalores

```
// Gerar uma matriz sim trica aleat ria
n = 5; // ordem da matriz
A = rand(n, n);
A = A + A'; // tornar a matriz sim trica

// Calcular os Discos de Gerschgorin para estimar os autovalores
r = sum(abs(A), 2) - abs(diag(A)); // raio de cada disc
centros = diag(A); // centro de cada disco
disp('Discos de Gerschgorin:');
for i = 1:n
    disp(sprintf('Autovalor estimado %d: Intervalo [%f, %f]', i, centros(i) - r(i), centros(i) + r(i)));
end

```



```
"Convergência alcançada."

"Resultado do Método da Potência:"

 9.4188327

"Número de iterações:"

 13.

"Tempo de execução:"

 0.0004776

"Convergência alcançada."

"Resultado do Método da Potência Deslocada com Iteração Invers

 2.7570577

"Número de iterações:"

 35.

"Tempo de execução:"

 0.004959

-->
```

Figure 1: Teste 1



```
"Convergência alcançada."

"Resultado do Método da Potência:"

 9.4188327

"Número de iterações:"

 13.

"Tempo de execução:"

 0.0004506

"Convergência alcançada."

"Resultado do Método da Potência Deslocada com Iteração Invers

 2.7570577

"Número de iterações:"

 35.

"Tempo de execução:"

 0.0062539

-->
```

Figure 2: Teste 2

```
"Convergência alcançada."

"Resultado do Método da Potência:"

 9.4188327

"Número de iterações:"

 13.

"Tempo de execução:"

 0.0004544

"Convergência alcançada."

"Resultado do Método da Potência Deslocada com Iteração Invers

 2.7570577

"Número de iterações:"

 35.

"Tempo de execução:"

 0.0057166

-->
```

Figure 3: Teste 3

```
// Par metros para o M todo da Pot ncia Deslocada com Itera  o Inversa
x0 = ones(n, 1); // vetor inicial
epsilon = 1e-6; // precis o
M = 1000; // n mero m ximo de itera  es

// Usar os autovalores estimados como valores iniciais para o M todo da Pot ncia Deslocada com Itera  o Inversa
for i = 1:n
    alfa = centros(i); // valor desejado de autovalor
    [lambda_deslocada, x1_deslocada, k_deslocada, n_erro_deslocada] = Potencia_deslocada_inversa(A, x0, epsilon, alfa, M);
    disp(sprintf('Autovalor %d calculado: %f', i, lambda_deslocada));
    disp(sprintf('N mero de itera  es para autovalor %d: %d', i, k_deslocada));
end
```

Este código realiza uma análise dos Discos de Gerschgorin para estimar os autovalores de uma matriz simétrica aleatória A. Primeiramente, a matriz simétrica é gerada aleatoriamente e os Discos de Gerschgorin são calculados a partir das somas das magnitudes dos elementos não diagonais em cada linha, que definem os raios dos discos, e os elementos diagonais, que são os centros dos discos. Em seguida, os autovalores estimados são apresentados, indicando os intervalos nos quais esses autovalores podem estar contidos.

Posteriormente, o código configura os parâmetros necessários para o Método da Potência Deslocada com Iteração Inversa, incluindo o vetor inicial, a precisão e o número máximo de iterações. Em um loop, cada autovalor estimado é utilizado como um valor deslocado para o Método da Potência Deslocada com Iteração Inversa. Os autovalores calculados, juntamente com o número de iterações necessárias para cada um, são exibidos. Essa abordagem permite uma análise mais precisa dos autovalores da matriz A e a eficácia do Método da Potência Deslocada com Iteração Inversa na sua obtenção.

Scilab 2024.0.0 Console

```
"Discos de Gerschgorin:"

"Autovalor estimado 1: Intervalo [-2.385778, 4.652662]"

"Autovalor estimado 2: Intervalo [-1.680010, 4.591699]"

"Autovalor estimado 3: Intervalo [-2.215211, 4.575440]"

"Autovalor estimado 4: Intervalo [-1.482275, 4.195858]"

"Autovalor estimado 5: Intervalo [-2.298837, 4.324610]"

"Convergência alcançada."

"Autovalor 1 calculado: -0.099911"

"Número de iterações para autovalor 1: 10"

"Convergência alcançada."

"Autovalor 2 calculado: -0.127893"

"Número de iterações para autovalor 2: 10"

"Convergência alcançada."

"Autovalor 3 calculado: -0.109977"

"Número de iterações para autovalor 3: 10"

"Convergência alcançada."

"Autovalor 4 calculado: -0.124770"

"Número de iterações para autovalor 4: 10"

"Convergência alcançada."
```

Scilab 2024.0.0 Console

```
"Autovalor estimado 2: Intervalo [-1.680010, 4.591699]"

"Autovalor estimado 3: Intervalo [-2.215211, 4.575440]"

"Autovalor estimado 4: Intervalo [-1.482275, 4.195858]"

"Autovalor estimado 5: Intervalo [-2.298837, 4.324610]"

"Convergência alcançada."

"Autovalor 1 calculado: -0.099911"

"Número de iterações para autovalor 1: 10"

"Convergência alcançada."

"Autovalor 2 calculado: -0.127893"

"Número de iterações para autovalor 2: 10"

"Convergência alcançada."

"Autovalor 3 calculado: -0.109977"

"Número de iterações para autovalor 3: 10"

"Convergência alcançada."

"Autovalor 4 calculado: -0.124770"

"Número de iterações para autovalor 4: 10"

"Convergência alcançada."

"Autovalor 5 calculado: 0.008082"

"Número de iterações para autovalor 5: 11"
```

5 Outros testes

```
// Fun  o para gerar uma matriz sim trica aleat ria com autovalores conhecidos
function A = gerar_matriz_simetrica_autovalores(autovalores)
    n = length(autovalores);
    A = zeros(n, n);
    for i = 1:n
        for j = i:n
            A(i, j) = rand() * 10 - 5; // N meros aleat rios entre -5 e 5
            if i == j then
                A(i, j) = autovalores(i); // Definir autovalores na diagonal principal
            else
                A(j, i) = A(i, j); // Preencher elementos sim tricos
            end
        end
    end
endfunction

// Teste dos algoritmos com diferentes matrizes e condi  es iniciais
n = 5; // Ordem da matriz
autovalores = [10, 8, 6, 4, 2]; // Autovalores conhecidos
A = gerar_matriz_simetrica_autovalores(autovalores); // Gerar matriz sim trica com autovalores conhecidos
x0 = rand(n, 1); // Vetor inicial aleat rio
epsilon = 1e-6; // Precis o
M = 100; // N mero m ximo de itera  es

// Testar M todo da Pot ncia
disp("Teste M todo da Pot ncia:");
[lambda1, x1, k1, n_erro1] = Metodo_potencia(A, x0, epsilon, M);
disp("Autovalor estimado:");
disp(lambda1);
disp("N mero de itera  es para converg ncia:");
disp(k1);

// Testar M todo da Pot ncia Deslocada com Itera  o Inversa
disp("Teste M todo da Pot ncia Deslocada com Itera  o Inversa:");
[lambda2, x2, k2, n_erro2] = Potencia_deslocada_inversa(A, x0, epsilon, 5, M);
disp("Autovalor estimado:");
disp(lambda2);
disp("N mero de itera  es para converg ncia:");
disp(k2);
```

```
"Teste Método da Potência:"

"Convergência alcançada."

"Autovalor estimado:"

 17.829617

"Número de iterações para convergência:"

 23.

"Teste Método da Potência Deslocada com Iteração Inversa:"

"Convergência alcançada."

"Autovalor estimado:"

 1.5532979

"Número de iterações para convergência:"

 13.

-->
```

Figure 6: Teste 5

Este código testa os algoritmos de Método da Potência e Método da Potência Deslocada com Iteração Inversa para calcular autovalores de matrizes simétricas com autovalores conhecidos. Primeiro, ele define uma função para gerar uma matriz simétrica aleatória com autovalores pré-definidos. Em seguida, ele testa esses algoritmos usando uma matriz gerada aleatoriamente e condições iniciais aleatórias. Para cada algoritmo, o código exibe o autovalor estimado e o número de iterações necessárias para convergência. Os autovalores conhecidos são inseridos manualmente na matriz simétrica, garantindo que o teste seja feito com uma matriz cujos autovalores são conhecidos a priori. Isso permite uma avaliação precisa da eficácia e precisão dos algoritmos de cálculo de autovalores em diferentes cenários.