Lista 2 - Decomposição QR

1. Métodos de decomposição QR

Considere a matriz

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 4 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

Utilize cada um dos métodos a seguir para calcular a decomposição QR de A:

a) Gram-Schmidt modificado.

b) Reflexões de Householder.

c) Rotações de Givens.

2. Complexidade computacional dos métodos de decomposição QR

Seja uma matriz $A_{n\times m}$, calcule a complexidade computacional de cada um dos algoritmos de decomposição QR a seguir.

a) Método utilizando o processo de Gram-Schmidt modificado:

```
function [Q,R] = mgs(A)
  [m,n] = size(A);
  V = A;
  Q = zeros(m,n);
  R = zeros(n,n);
  for i=1:n
    R(i,i) = norm(V(:,i));
    Q(:,i) = V(:,i)/R(i,i);
    for j=i+1:n
      R(i,j) = Q(:,i)'*V(:,j);
      V(:,j) = V(:,j)- R(i,j)*Q(:,i);
    end
  end
end
```

b) Método utilizando reflexões de Householder:

```
function [Q,R] = hqr(A)
  [m,n] = size(A);
  Q = eye(m);
  R = A;
  for k = 1:min(m-1,n)
    v = R(k:end,k);
    s = sign(v(1));
    if s == 0
      s = 1;
    end
    v(1) = v(1) - (-s * norm(v));
    v = v / norm(v);
    R(k:end,k:end) = R(k:end,k:end) - 2*v*(v'*R(k:end,k:end));
    Q(:,k:end) = Q(:,k:end) - 2*(Q(:,k:end)*v)*v';
  end
end
```

c) Método utilizando rotações de Givens:

```matlab
function [Q,R] = simplegivens(A)
  [m,n] = size(A);
  Q = eye(m);
  R = A;
  for j = 1:n
    for i = j+1:m
      if R(i,j) ~= 0
        % Calcula os parametros da rotacao de Givens.
        r = sqrt(R(j,j)^2 + R(i,j)^2);
        c = R(j,j)/r;
        s = -R(i,j)/r;
        % Aplica a rotacao em R (apenas linhas j e i).
        r1 = c*R(j,:) - s*R(i,:);
        r2 = s*R(j,:) + c*R(i,:);
        R(j,:) = r1;
        R(i,:) = r2;
        % Aplica a rotacao em Q (apenas colunas j e i).
        q1 = c*Q(:,j) - s*Q(:,i);
        q2 = s*Q(:,j) + c*Q(:,i);
        Q(:,j) = q1;
        Q(:,i) = q2;
      end
    end
  end
end
```

3. Comparação do erro dos métodos de decomposição QR [Matlab]

Para cada matriz abaixo, aplique os 3 métodos de decomposição QR da Questão 1, calcule o erro da decomposição e o erro de ortogonalidade de cada método, e faça uma comparação entre os valores obtidos.

a) Matriz densa aleatória (bem-condicionada):

$$A = \begin{bmatrix} 0.56 & -0.71 & 1.03 & 0.44 & 0.88 \\ 1.12 & 0.34 & -0.92 & 0.56 & 1.23 \\ -0.45 & 1.05 & 0.78 & -1.23 & 0.12 \\ 0.67 & -1.01 & 0.12 & 0.98 & -0.76 \\ 1.34 & 0.56 & -0.67 & 0.23 & 1.45 \end{bmatrix}$$

b) Matriz com colunas quase dependentes:

$$A = \begin{bmatrix} 1 & 1+10^{-8} & 2 & 3 & 4 \\ 2 & 2+10^{-8} & 3 & 5 & 7 \\ 3 & 3+10^{-8} & 4 & 7 & 10 \\ 4 & 4+10^{-8} & 5 & 9 & 13 \\ 5 & 5+10^{-8} & 6 & 11 & 16 \end{bmatrix}$$

c) Matriz mal escalonada:

$$A = \begin{bmatrix} 1 & 1 \cdot 10^{-8} & 1 \cdot 10^{8} & 1 \cdot 10^{-4} & 1 \cdot 10^{4} \\ 2 & 2 \cdot 10^{-8} & 2 \cdot 10^{8} & 2 \cdot 10^{-4} & 2 \cdot 10^{4} \\ 3 & 3 \cdot 10^{-8} & 3 \cdot 10^{8} & 3 \cdot 10^{-4} & 3 \cdot 10^{4} \\ 4 & 4 \cdot 10^{-8} & 4 \cdot 10^{8} & 4 \cdot 10^{-4} & 4 \cdot 10^{4} \\ 5 & 5 \cdot 10^{-8} & 5 \cdot 10^{8} & 5 \cdot 10^{-4} & 5 \cdot 10^{4} \end{bmatrix}$$

4. Solução de sistema linear

Resolva o sistema linear abaixo utilizando decomposição QR.

$$\begin{cases} x + 2y - z = 1, \\ 3x - y + 2z = 4, \\ 2x + y + z = 3. \end{cases}$$

5. Reflexão de Householder

Seja x um vetor $x = [x_1, \ldots, x_n]^T$. No algoritmo de decomposição QR utilizando reflexões de Householder, a reflexão de x é definida como:

$$\overline{x} = \begin{cases} \|x\|_2 \, e_1, & \text{se } x_1 < 0 \\ -\|x\|_2 \, e_1, & \text{caso contrário} \end{cases}$$

Qual a justificativa da escolha desta definição específica para $\overline{x}$?

6. Passos do método de Francis

Para o método de Francis, mostre que se uma matriz quadrada A é simétrica, então todas as matrizes A_k também serão simétricas. Em outras palavras, mostre que se A_{k-1} é simétrica, então A_k também será simétrica.

7. Cálculo de autovalores [Matlab]

Considere a matriz

$$A = \begin{bmatrix} 67 & -46 & -23 & -29 & 4 \\ -46 & 91 & -19 & 71 & 13 \\ -23 & -19 & 100 & 0 & -34 \\ -29 & 71 & 0 & 63 & 7 \\ 4 & 13 & -34 & 7 & 46 \end{bmatrix}.$$

a) Utilize o método de Francis abaixo para calcular todos os autovalores e autovetores de A. Registre o tempo de execução para tolerância de 10^{-8}.

```
function [V,D] = francis(A,tol)
  n = size(A,1);
  V = eye(n);
  erro = inf;
  while erro>tol
    [Q,R] = hqr(A);
    A = R*Q;
    V = V*Q;
    erro = max(max(abs(tril(A,-1))));
  end
  D = diag(A);
end
```

b) Utilize o método das potências inversas abaixo para calcular apenas os 2 menores autovalores (em módulo) de A, e os respectivos autovetores. Registre o tempo de execução para tolerância de 10^{-14}, e compare os resultados com o item anterior.

```
function [lambda,y] = potencia_inv(A,tol,y0)
  erro = inf;
  n = size(A,1);
  if(nargin==2)
    y0 = zeros(n,1);
    y0(1) = 1;
  end
  [L,U] = lu(A);
  while (erro>tol)
    x = U\(L\y0);
    y = x/norm(x);
    erro = abs(abs(y0'*y)-1);
    y0 = y;
  end
  lambda = y'*A*y;
end
```

OBS.: Para reduzir a volatilidade na medição, meça o tempo total de 100 execuções para cada caso e utilize o tempo médio obtido.