UFRJ

Universidade Federal do Rio de Janeiro

EQE-703: Métodos Matemáticos

Trabalho 1

Professor:
José Luiz de Medeiros

Grupo:
Gustavo L. R. Caldas
Oscar Chamberlain

21 de junho de 2021

Conteúdo

Item (i)

A partir da definição de forma quadrática:

$$\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i A_{ij} x_j \tag{1}$$

Aplicando a derivada de um índice k:

$$\frac{\partial \left(\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}\right)}{\partial x_k} = \frac{\partial}{\partial x_k} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} x_i A_{ij} x_j \right) \tag{2}$$

Desdobrando $\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}$ para facilitar:

$$\frac{\partial \left(\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}\right)}{\partial x_k} = \frac{\partial}{\partial x_k} \left(\sum_{i \neq k} \sum_{j \neq k} x_i A_{ij} x_j + \sum_{i \neq k} x_i A_{ik} x_k + \sum_{j \neq k} x_k A_{kj} x_j + A_{kk} x_k^2 \right) \tag{3}$$

Ao aplicar-se as derivadas, o primeiro termo é nulo e nos demais, torna-se:

$$\frac{\partial \left(\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}\right)}{\partial x_k} = \sum_{i \neq k} x_i A_{ik} + \sum_{j \neq k} A_{kj} x_j + 2 A_{kk} x_k \tag{4}$$

O último termo pode ser reincorporado ao primeiro, fazendo:

$$\frac{\partial \left(\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}\right)}{\partial x_k} = \sum_{i=1}^{n} x_i A_{ik} + \sum_{j=1}^{n} A_{kj} x_j = (\underline{\underline{A}} + \underline{\underline{A}}^T)\underline{\mathrm{X}} \tag{5}$$

Para a hessiana, fazemos:

$$\begin{aligned} \frac{\partial^2 \left(\underline{\mathrm{X}}^T \underline{\underline{A}}\ \underline{\mathrm{X}}\right)}{\partial x'_k \partial x_k} &= \frac{\partial}{\partial x_k} \left(\sum_{i=1}^{n} x_i A_{ik} + \sum_{j=1}^{n} A_{kj} x_j \right) = \\ = \frac{\partial}{\partial x'_k} \left(\sum_{i \neq k'}^{n} x_i A_{ik} + \sum_{j \neq k'}^{n} A_{kj} x_j + x_{k'} A_{k'k} + A_{kk'} x_{k'} \right) &= (\underline{\underline{A}} + \underline{\underline{A}}^T) \end{aligned} \tag{6}$$

Sendo A simétrica, logo:

$$\nabla_X(\underline{X}^T \underline{\underline{A}}\ \underline{X}) = (\underline{\underline{A}} + \underline{\underline{A}}^T)\underline{X} = 2\underline{\underline{A}}\ \underline{X} \tag{7a}$$

$$\nabla_x \nabla_x^T(\underline{X}^T \underline{\underline{A}}\ \underline{X}) = (\underline{\underline{A}} + \underline{\underline{A}}^T) = 2\underline{\underline{A}} \tag{7b}$$

Então aplicando os resultados acima para a função $F(\underline{X})$ e usando a regra da cadeia:

$$F(\underline{X}) = \frac{1}{4}(\underline{X}^T \underline{\underline{A}}\ \underline{X})^2 + \frac{1}{2}(\underline{X}^T \underline{\underline{A}}\ \underline{X}) + \underline{B}^T\underline{X} + C \tag{8a}$$

$$\nabla_X F(\underline{X}) = \frac{1}{2}[(\underline{X}^T \underline{\underline{A}}\ \underline{X})(2\underline{\underline{A}}\ \underline{X})] + \frac{1}{2}(2\underline{\underline{A}}\ \underline{X}) + \underline{B} = (\underline{X}^T \underline{\underline{A}}\ \underline{X})(\underline{\underline{A}}\underline{X}) + \underline{\underline{A}}\ \underline{X} + \underline{B} \tag{8b}$$

Note que o gradiente da função F(X) deve ser um vetor coluna, assim como é na eq. (7a), logo devemos ter B e não B^T na eq. (8b). Aplicando ∇_x mais uma vez:

$$\nabla_x \nabla_x^T F(\underline{X}) = 2(\underline{\underline{A}}\ \underline{X})(\underline{\underline{A}}\ \underline{X})^T + (\underline{X}^T \underline{\underline{A}}\ \underline{X})\underline{\underline{A}} + \underline{\underline{A}} \tag{8c}$$

Item (ii)

O código para aplicar o aplicar o Processo Schmidt à uma matriz $\underline{\underline{M}}$ está abaixo:

```
% Script file: conjugado_schmidt.m
% Objetivo: Este programa executa a Construção de Base Conjugada
%P1 , P2 , ..., Pn por Matriz Simétrica e Definida pelo
%método de Schmidt.
% Referência: S. Chapman, "Programação para Engenheiros"

% Definição do tamanho da matriz
function P = conjugado_schmidt(M)

n = length(M); %dimensão de M

P = zeros(n); % declaração da dimensão de P

W = M;
```

```
P(:,1) = W(:,1); %Fazer P1.

alfa = zeros(n);

    for k=2:n % ref. p.141
        for i=1:(k-1)
            alfa(k,i) =   - ( (P(:,i)).'*M*W(:,k) ) /
            ( (P(:,i)).'*M*P(:,i)  ); % ref. p. 46
        end
        soma = (P(:,:))*(alfa(k,:).');
        P(:,k) = W(:,k)+soma;
    end

    for k=1:n
        P(:, k) = (P(:,k))/(norm(P(:,k)));
    end
end
```

Logo, a matriz $\underline{\underline{P}}$ é:

$$\underline{\underline{P}} = \begin{pmatrix} 0.3536 & -0.6219 & 0.8276 & -0.9189 & 0.7009 & -0.3506 & 0.1266 & 0.1259 \\ 0.3536 & -0.5202 & 0.4894 & -0.1247 & -0.4833 & 0.6948 & -0.4641 & -0.4628 \\ 0.3536 & -0.4184 & 0.2273 & 0.2133 & -0.3566 & -0.2049 & 0.6139 & 0.6131 \\ 0.3536 & -0.3167 & 0.0413 & 0.2381 & 0.0856 & -0.3967 & -0.1920 & -0.1917 \\ 0.3536 & -0.2149 & -0.0687 & 0.0929 & 0.2758 & 0.1249 & -0.3560 & -0.3572 \\ 0.3536 & -0.1131 & -0.1026 & -0.0793 & 0.0740 & 0.3251 & 0.4347 & 0.4363 \\ 0.3536 & -0.0114 & -0.0604 & -0.1354 & -0.2318 & -0.2647 & -0.1950 & -0.1959 \\ 0.3536 & 0.0904 & 0.0579 & 0.0676 & 0.0740 & 0.0598 & 0.0327 & 0.0329 \end{pmatrix} \tag{9}$$

Item (iii)

O caráter diagonal da matriz $\underline{\underline{D}}$ que satisfaz a equação $\underline{\underline{P}}^T \underline{\underline{AP}}$ está dado pela eq. (10) abaixo.

$$\underline{\underline{D}} = 1.0e{+}03 * \begin{pmatrix} 1.6086 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0274 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.0000 & 0.0039 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0015 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0005 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0001 & -0.0000 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \end{pmatrix} \tag{10}$$

Item (iv)

O programa abaixo executa a diagonalização de forma quadrática com transformação de congruência:

```
%Congruência de uma matriz simétrica pelo método de Schmidt.
% Referência: S. Chapman, "Programação para Engenheiros"

function U = diagonalizacao_forma_quadratica(M)
n = length(M);
I = eye(n); % ref. p. 29
U = I;
alfa = zeros(n); %matriz do alfa é de zeros
soma = 0;
    for k=2:n % ref. p.141
        for i=1:(k-1)
            %Calculando os alfas
            alfa(k,i) = -( (U(:,i)).'*M*I(:,k) )/((U(:,i)).'*M*U(:,i));
        end
    %Calculando a soma
    soma = U(:,:)*alfa(k,:).';
    U(:,k) = U(:,k)+soma;
    end

end
```

Logo, o caráter da forma quadrática é:

$$\underline{\underline{Q}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

Como todos os valores da sua diagonal são positivos, ela é positiva definida.

Da mesma forma, caráter de $\underline{\underline{A}}$ é mostrado numa matriz cuja diagonal são seus autovalores:

$$carater(A) = \begin{pmatrix} 2.2009 \cdot 10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0067 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0837 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5119 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.9535 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11.9431 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 148.8048 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.5437 \cdot 10^{3} \end{pmatrix} \tag{12}$$

A matriz $\underline{\underline{A}}$ é, portanto, positiva definida, pois seus autovalores são maiores que zero.

Item (v)

O caráter da hessiana pode ser definida através da equação eq. (13a). Nesse caso, λ são os autovalores de $\underline{\underline{H}}$ e $\underline{vec}$ os autovetores de $\underline{\underline{H}}$

$$\underline{\underline{H}}\ \underline{vec} = \lambda\ \underline{vec} \tag{13a}$$

Para deduzirmos o caráter de $\underline{\underline{H}}$, fazemos:

$$\underline{\underline{H}}\ \underline{vec} = [2(\underline{\underline{A}}\ \underline{\text{X}})(\underline{\underline{A}}\ \underline{\text{X}})^T + (\underline{\text{X}}^T \underline{\underline{A}}\ \underline{\text{X}})\underline{\underline{A}} + \underline{\underline{A}}]\ \underline{vec} = \lambda\ \underline{vec} \tag{13b}$$

Considerando que $\underline{\text{X}}^T \underline{\underline{A}}\underline{\text{X}}$ é uma constante C e que:

$$(\underline{\underline{A}}\ \underline{X})^T = \underline{X}^T \underline{\underline{A}}^T = \underline{X}^T \underline{\underline{A}} \tag{13c}$$

Logo:

$$\underline{\underline{H}}\ \underline{vec} = [2(\underline{\underline{A}}\ \underline{X})\ \underline{X}^T + C + I]\ \underline{\underline{A}}\ \underline{vec} = \lambda\ \underline{vec} \tag{13d}$$

Portanto, o caráter de $\underline{\underline{A}}$ definirá o caráter da hessiana. Logo com os autovalores de $\underline{\underline{A}}$, tem-se o caráter da hessiana. Com $\underline{\underline{A}}$ é positiva definida, portanto a hessiana também é positiva definida.

Item (vi)

O código abaixo minimiza $F(\underline{X})$ pelo método de Newton-Raphson sem restrições dados $\underline{X}_0$, $\underline{\underline{A}}$ e $\underline{\underline{B}}$.

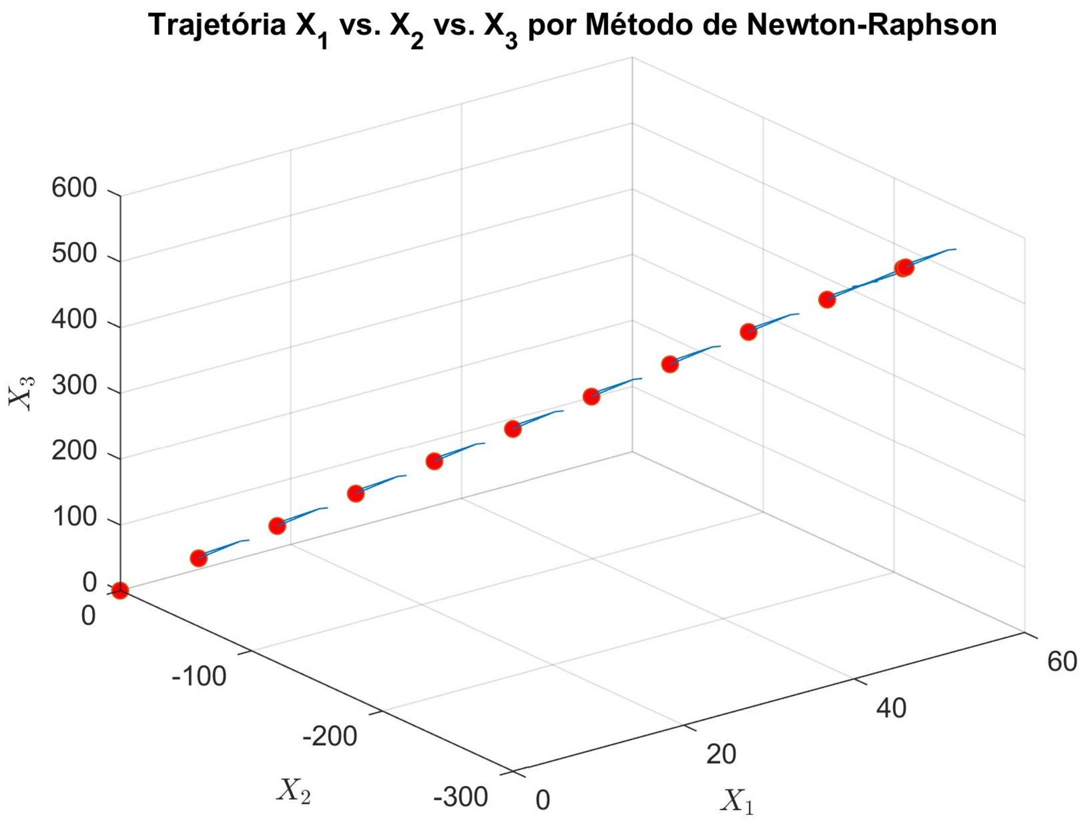
```
function [X,path,cont,Spath] = newton_raphson(X0,A,B)
%Sendo X vetor coluna
    X = X0; %chute inicial
    path = X0; %histórico
    epsilon = 100; %declarando a tolerância de forma a rodar o script
    Spath = zeros(size(X)); %o histórico do vetor busca
    cont= 0; % contador de iterações
    %condição de tolerância e condição de n máximo de iterações
    while (epsilon >= 0.01 && cont<10000)
        Xant = X; % X^(k) = X^(k-1)
        % Cálculo do gradiente
        G = (X.'*A*X)*A*X + A*X+B;
        % Cálculo da Hessiana
        H = 2*(A*X)*((A*X).')+(X.'*A*X)*A + A;
        % Cálculo do vetor de busca
        S = -H\G; % É mais rápido que inv(H)*G
        %Vetor S unitário para traçar graficamente e calcular o freio t
        Snorm = S/norm(S);
        %Limitando o passo t
        t = 1;
        while(norm(S)*t>100)
          t = 100/(norm(S)+0.1);
        end
        %Calculando o novo X
```

```
        X = X + t*S;
        %Tolerância
        epsilon = norm(X-Xant);
        %Tensor do histórico de X
        path = cat(2,X,path);
        %Histórico de S
        Spath = cat(2,Snorm,Spath);
        %Contador
        cont=cont+1;
    end
end
```

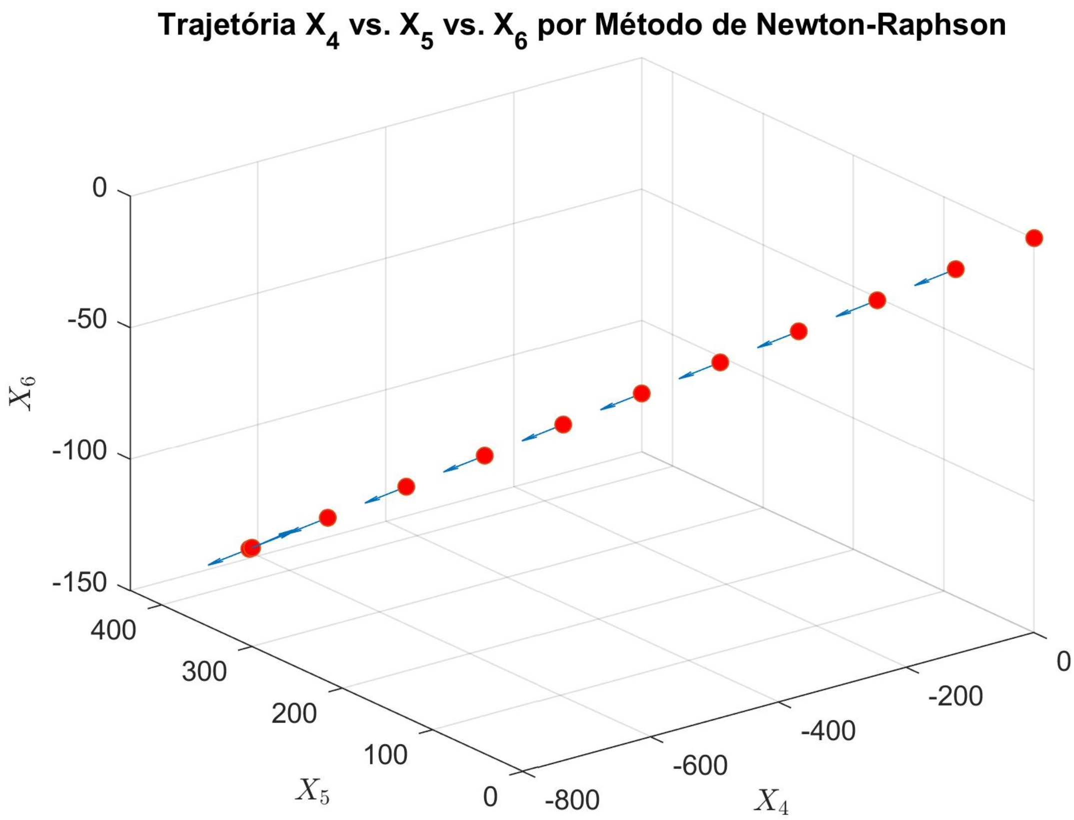
Partindo de $\underline{X}_0$ nulo, após treze iterações, o valor mínimo para $F(\underline{X})$ encontrado por esse método foi de $-8.1917 \cdot 10^5$, tal que:

$$\underline{X} = \begin{pmatrix} 50.1525 \\ -270.8236 \\ 561.7082 \\ -641.9522 \\ 411.2507 \\ -140.4271 \\ 20.0610 \\ 0.0000 \end{pmatrix} \tag{14}$$

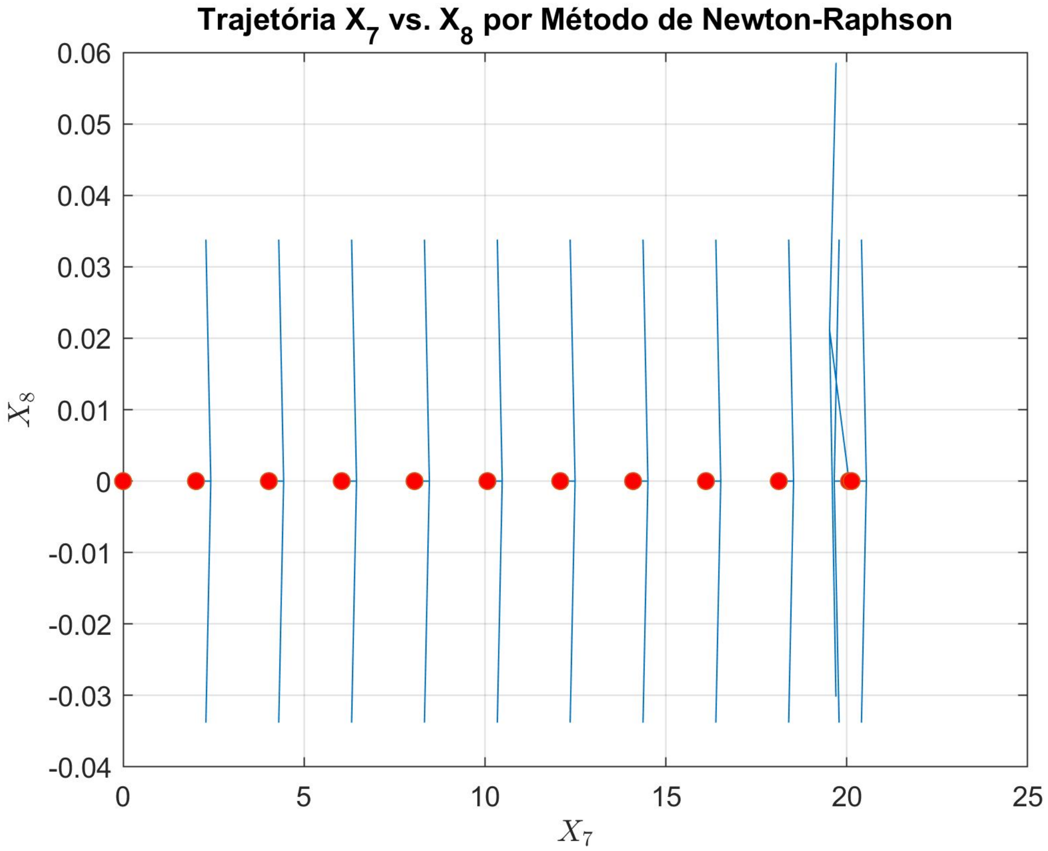
Considerando que F duas vezes é diferenciável, sendo a matriz hessiana positiva definida e sabendo que $G(\underline{X})$ é zero, tem-se uma condição necessária e suficiente (condição de segunda ordem) para afirmar que o valor de $\underline{X}$ é ponto de mínimo. A Figura 1 mostra as trajetórias de iterações por Newton-Raphson.



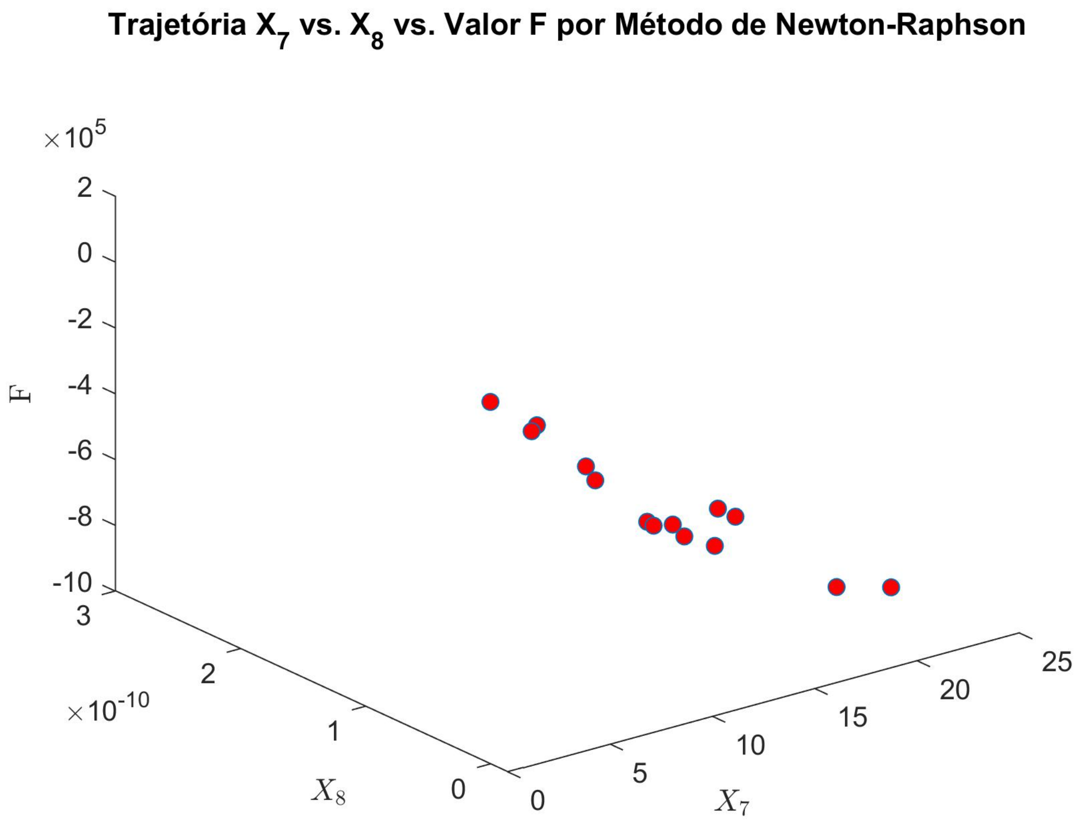
(a) Trajetória de X_1 vs. X_2 vs. X_3 por Método de Newton-Raphson.



(b) Trajetória de X_4 vs.X_5 vs. X_6 por Método de Newton-Raphson.



(c) Trajetória de X_7 vs. $X8$ por Método de Newton-Raphson.



(d) Trajetória de X_7 vs. $X8$ vs. valores de F por Método de Newton-Raphson.

Figura 1: Trajetórias por Método de Newton-Raphson.

Item (vii)

De modo semelhante, o código abaixo minimiza $F(\underline{X})$ pelo método de Powell sem restrições dados $\underline{X}_0$, $\underline{\underline{A}}$ e $\underline{\underline{B}}$.

```
% Script file: powell.m
function [X,path,cont,Spath] = powell(X0,A,B)
%Sendo X vetor coluna
    X = X0; %chute inicial
    epsilon = 100;  %declarando a tolerância de forma o rodar o script
    Spath = zeros(size(X)); %o histórico do vetor busca
    cont= 0; % contador de iterações
    path = X0; %histórico
```

```
        while(epsilon >= 0.001)
          Xant = X;
          %Cálculo da Hessiana em k
          H = 2*(A*X)*((A*X).')+(X.'*A*X)*A + A;
          % Cálculo da Base conjugada da Hessiana
          P = conjugado_schmidt(H);
          %Estimativa inicial de X(teta):
          Xteta = zeros(length(P)); % todos Xn pontos auxiliares
          Xteta(:,1) = X;
          for m=2:length(P)
              %Condição de pto estacionário
              pe =  @(teta) ponto_estacionario(P(:,m),Xteta(:,m-1),A,B,teta);
              teta = fzero(pe,0);
              %teta
              Xteta(:,m) = Xteta(:,m-1) + P(:,m)*teta;
          end
          X = Xteta(:,length(P));
        % Cálculo do vetor de busca
        S = X-Xant; % É mais rápido que inv(H)*G
        %Vetor S unitário para traçar graficamente e calcular o freio t
        Snorm = S/norm(S);
        %Tolerância
        epsilon = norm(X-Xant);
        %Tensor do histórico de X
        path = cat(2,X,path);
        %Histórico de S
        Spath = cat(2,Snorm,Spath);
        %Contador
        cont=cont+1;
        end
end
```

Repare que o código "conjugado_schmidt"descrito no Item (ii) também é chamado. Além disso, o código auxiliar "ponto_estacionario"abaixo é utlizado de forma a assegurar a condição de ponto estacionário unidimensional sobre a reta de busca. No código "ponto_estacionario", calcula-se o gradiente também. A função fzero encontra o teta que zera a equação.

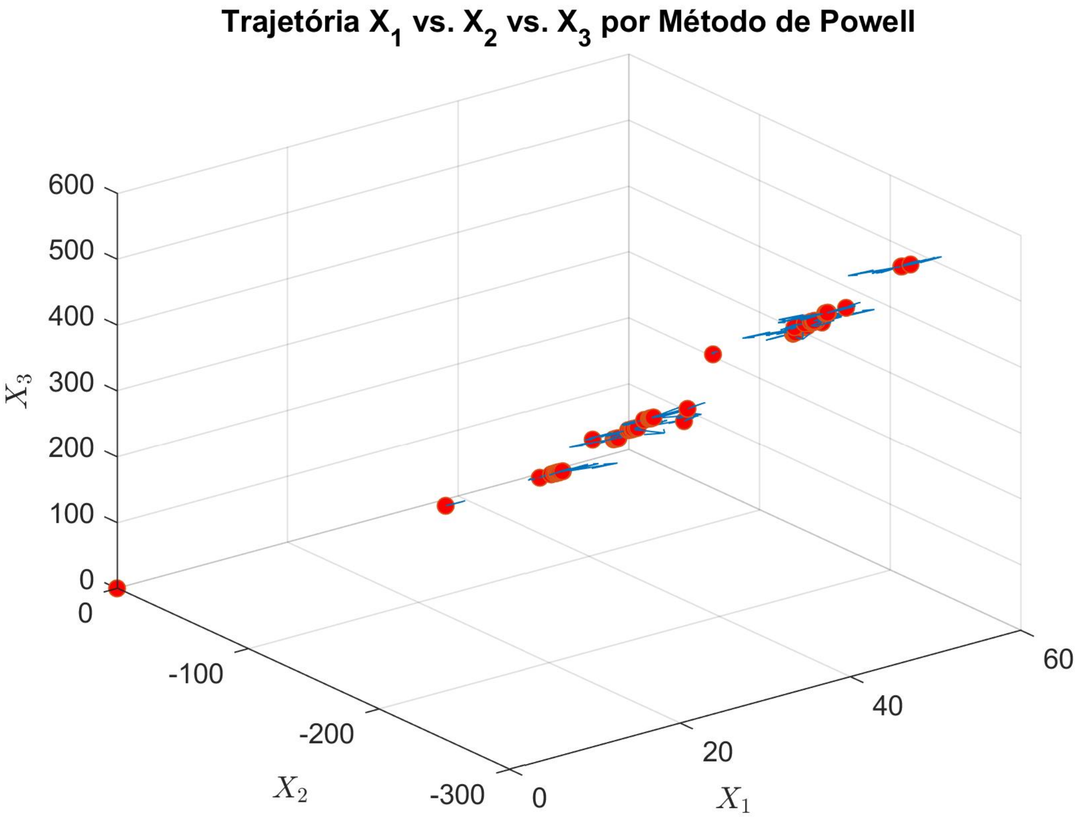
```
function f = ponto_estacionario(P,Xteta_ant,A,B,teta)
     Xteta = Xteta_ant + P*teta;
     %Cálculo do gradiente
```

```
    G = (Xteta.'*A*Xteta)*A*Xteta + A*Xteta+B;
    f = P.'*G;
end
```

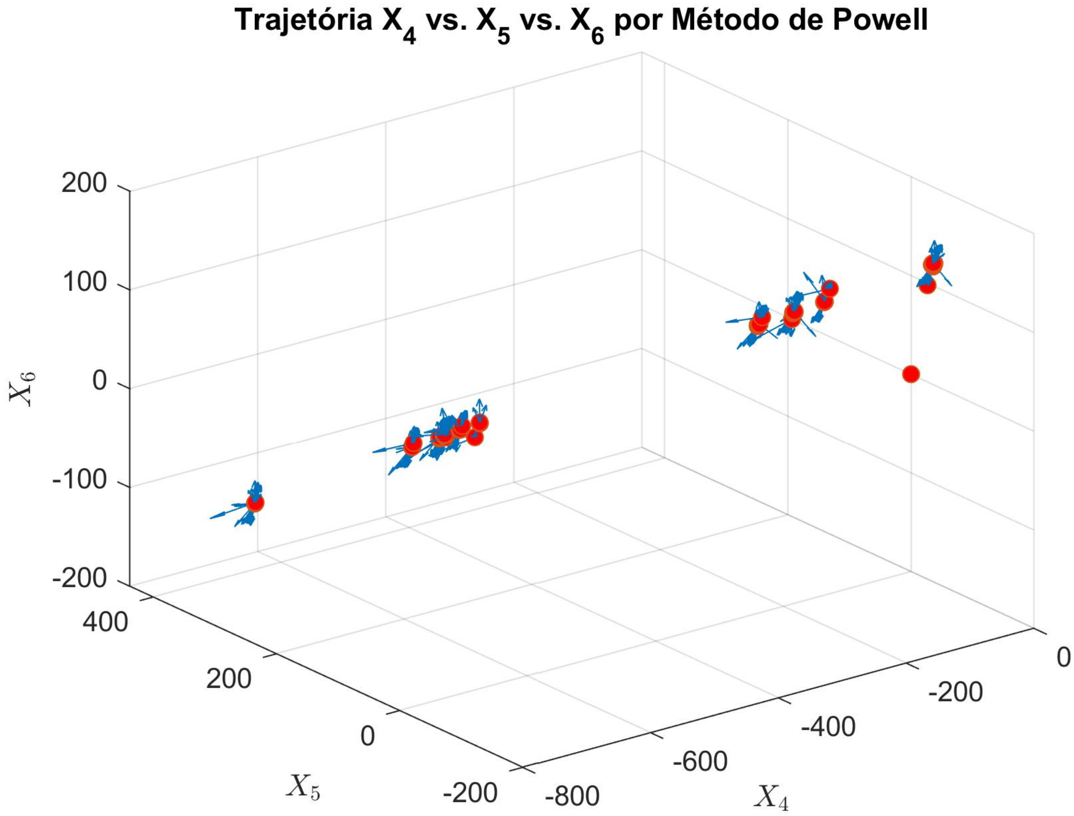
Partindo de $\underline{\mathrm{X}}_0$ nulo,após 483 iterações, o valor mínimo para $F(\underline{\mathrm{X}})$ encontrado por Powell foi de $-7.8928 \cdot 10^5$, tal que:

$$\underline{\mathrm{X}} = \begin{pmatrix} 50.3334 \\ -271.4481 \\ 560.6316 \\ -637.0854 \\ 403.9533 \\ -135.0151 \\ 18.0567 \\ 0.4191 \end{pmatrix} \tag{15}$$

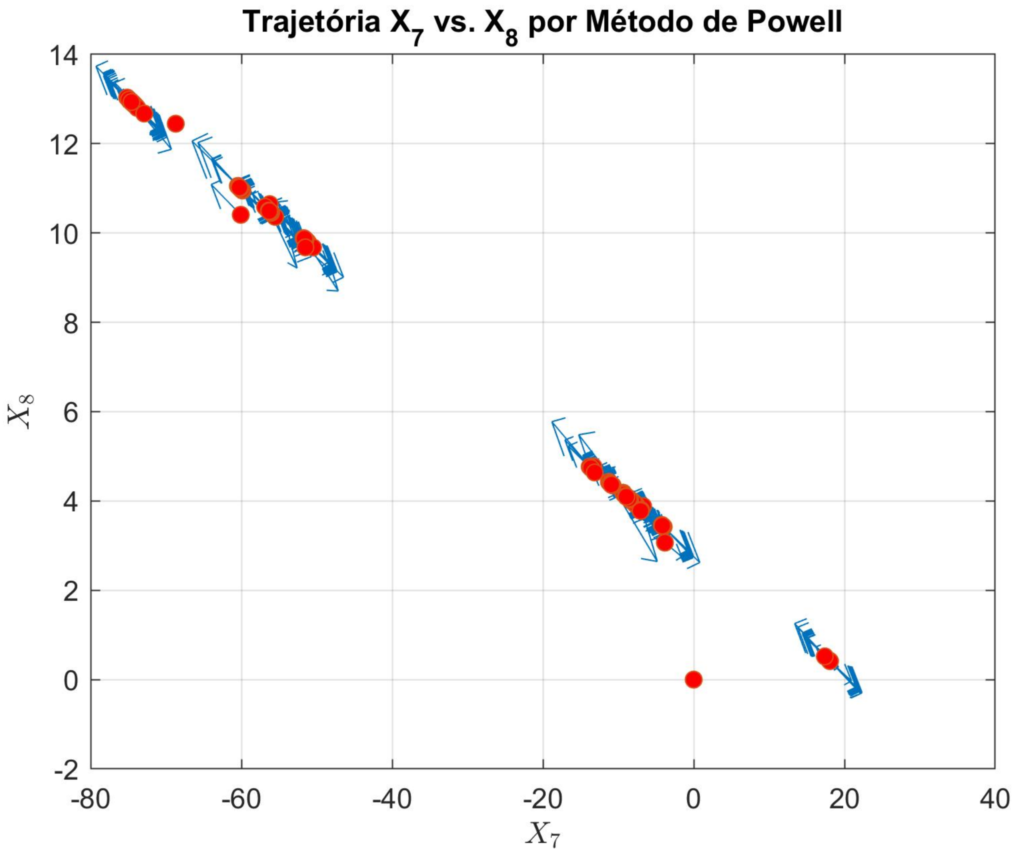
A Figura 2 mostra as trajetórias de iterações por Método de Powell.



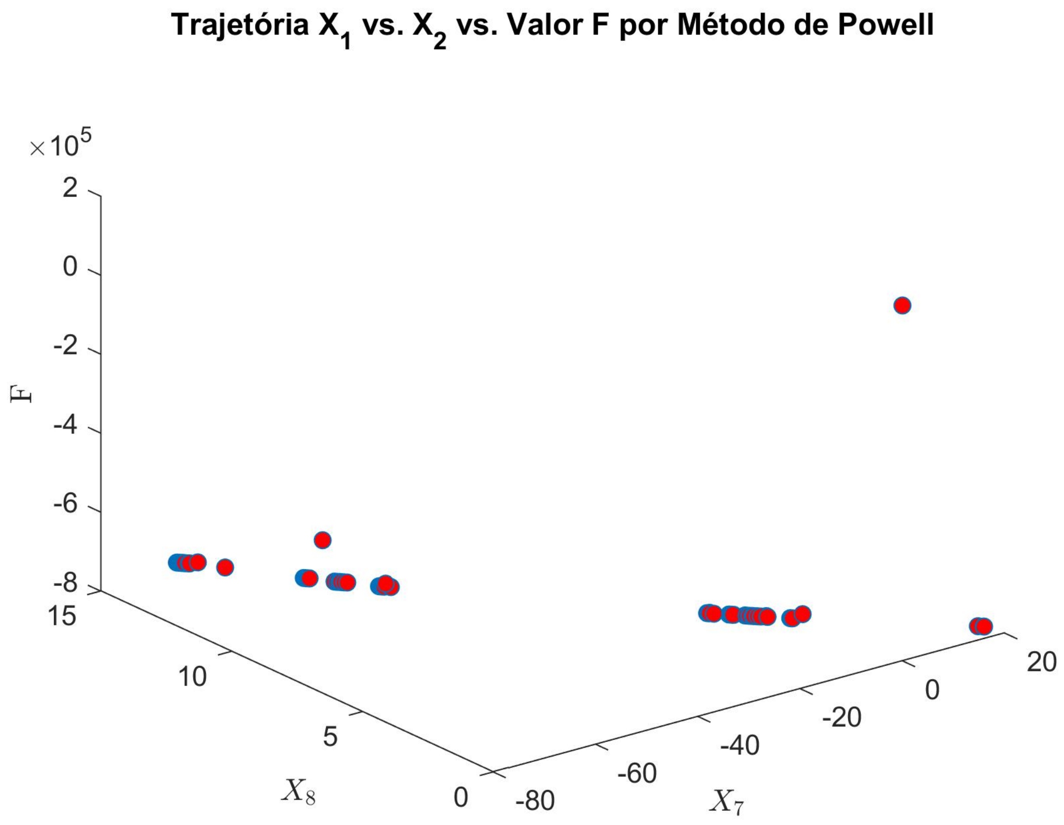
(a) Trajetória de X_1 vs. X_2 vs. X_3 por Método de Powell.



(b) Trajetória de X_4 vs.X_5 vs. X_6 por Método de Powell.



(c) Trajetória de X_7 vs. $X8$ por Método de Powell.



(d) Trajetória de X_7 vs. $X8$ vs. valores de F por Método de Powell.

Figura 2: Trajetórias por Método de Powell.

Item (viii)

No espaço reduzido $\underline{\underline{E}}\underline{X} = 0$, temos apenas dois graus de liberdade, diferentemente do caso anterior, que poderíamos resolver X_1 a X_8. Nesse caso, somente X_7 e X_8 podem ter qualquer valor. Os demais (X_1 e X_6) podem ser obtidos a partir da sua combinação linear, obedencendo a equação $\underline{X} = \underline{\underline{K}}\ \underline{R}$.

Resolvendo pelo script abaixo que aplica a pivotagem total:

```
% Metodologia para transformação da matriz
% Considerando a Matriz M (nxm)
function [Mt] = Trator(M)
Mt = M;
```

```
%Tamanho da matriz
t = size(M);
L=t(1);
C=t(2);

%Transformação das linhas
for i=1:L
    Mt(i,:) = Mt(i,:)/Mt(i,i);
    for k=1:L
        if k~=i
            Mt(k,:)= Mt(k,:)-Mt(i,:)*Mt(k,i);
        end
    end
end
end
```

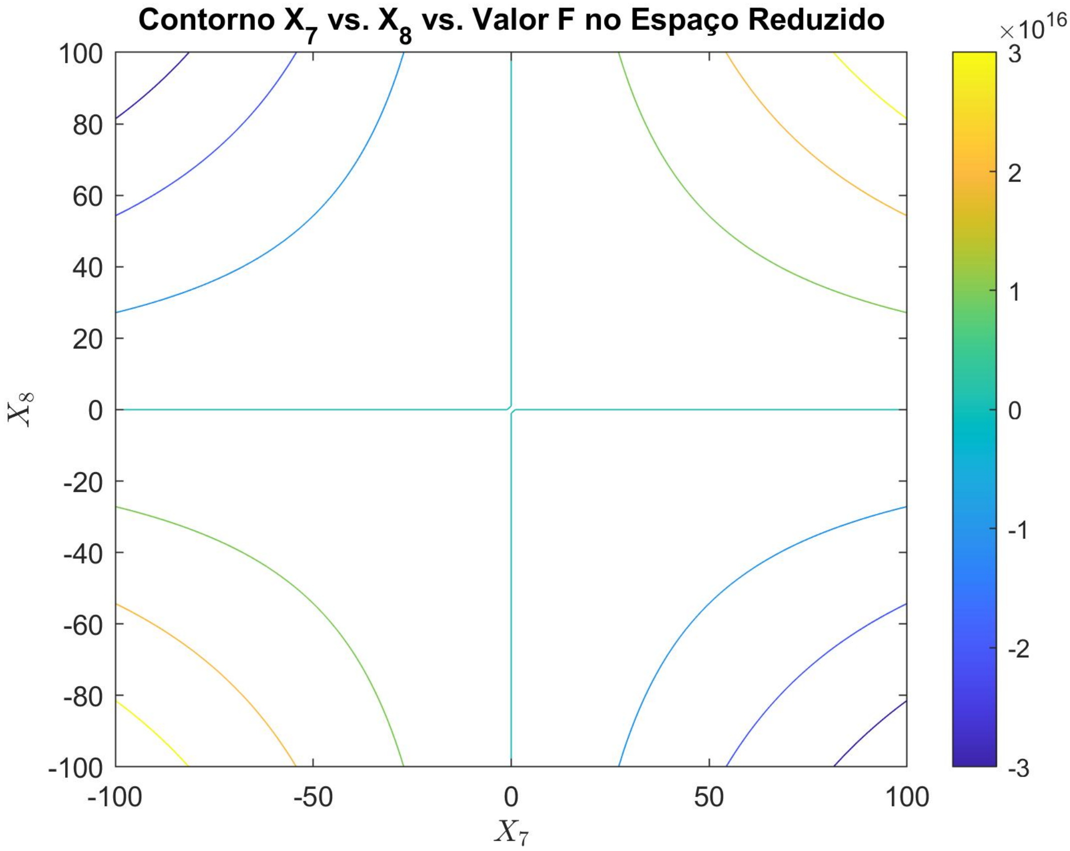
A resolução é

$$\underline{\underline{K}} = \begin{pmatrix} -2.0283 & -0.5621 \\ 1.0239 & -0.3022 \\ 0.7217 & 0.4657 \\ -0.5130 & -0.3150 \\ -0.3565 & 0.4647 \\ -0.2891 & -0.1949 \\ 1.0000 & 0 \\ 0 & 1.0000 \end{pmatrix} \tag{16}$$

E:

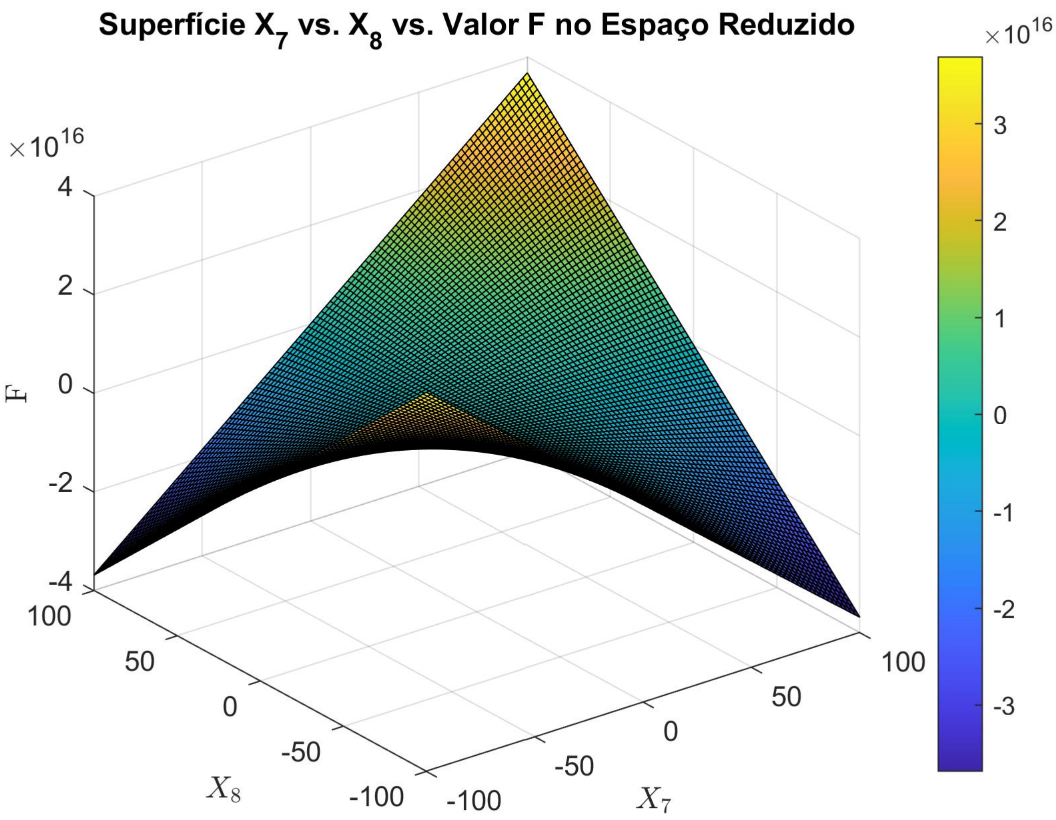
$$\underline{R} = \begin{pmatrix} X_7 \\ X_8 \end{pmatrix} \tag{17}$$

Conforme observado nas figuras 3a e 3b, a função F pode apresentar um ou mais mínimos parciais com duas extremidades sem limite superior e duas sem limite inferior.



Contorno X7 vs. X8 vs. Valor F no Espaço Reduzido
X_8
X_7
100
80
60
40
20
0
-20
-40
-60
-80
-100
-100
-50
0
50
100
×10^16
3
2
1
0
-1
-2
-3

(a)



(b)

Figura 3: Projeção de $F(\underline{X})$ sobre o Espaço Reduzido: (a) contorno, (b) superfície.

Item (ix)

Para o cálculo do gradiente no espaço reduzido:

$$\underline{\underline{G_R}} = \nabla_R(\underline{X}^T)\underline{G}(\underline{X}(\underline{R})) \tag{18}$$

Veja que:

$$\nabla_R(\underline{X}^T) = \nabla_R[(\underline{\underline{K}}\underline{R})^T] = \nabla_R(\underline{R}^T\underline{\underline{K}}^T) = \underline{\underline{K}}^T \tag{19}$$

Logo, a hessiana no espaço reduzido é dada por:

$$\underline{\underline{H_R}} = [\nabla_R(\underline{X}^T)][\nabla_x\nabla_x^T F][\nabla_R(\underline{X}^T)]^T = \underline{\underline{K}}^T\underline{\underline{H}}(\underline{X}(\underline{R}))\underline{\underline{K}} \tag{20}$$

Dessa forma, o script abaixo calcula a hessiana e o gradiente no espaço reduzido. A partir desse cálculo, temos o vetor de busca S. Com o vetor de busca S, pode-se, como no Item (vi), encontrar os novos valores de R.

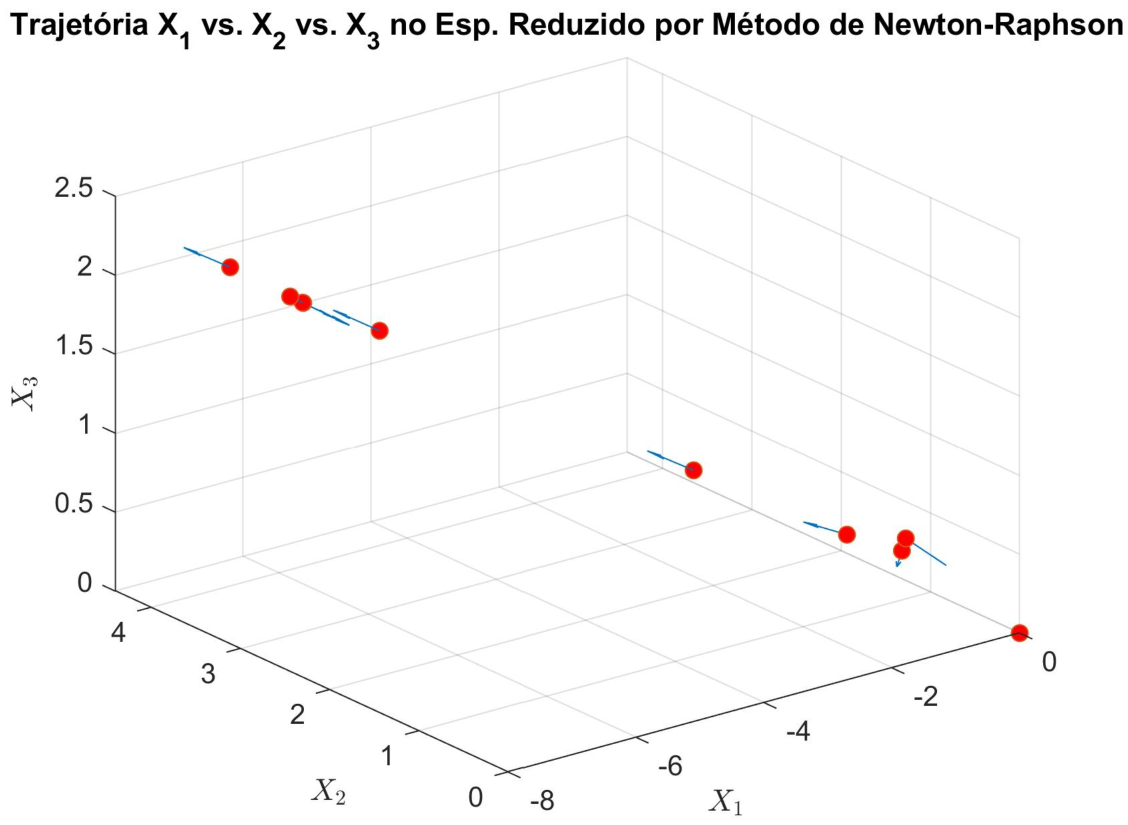
```
% Script file: newton_raphson_9.m
function [X,path,cont,Spath] = newton_raphson_9(X0,A,B,K)
%Sendo X vetor coluna
    X = X0; %chute inicial
    path = X0; %histórico
    epsilon = 100; %declarando a tolerância de forma o rodar o script
    Spath = zeros(size(X)); %o histórico do vetor busca
    cont= 0; % contador de iterações
    while (epsilon >= 0.001 && cont<10000)
    %condição de tolerância e condição de n máximo de iterações
        Xant = X; % X^(k) = X^(k-1)
        % Cálculo do gradiente
        G = (X.'*A*X)*A*X + A*X+B;
        % Gradiente reduzido
        Gr = K.'*G;
        % Cálculo da Hessiana
        H = 2*(A*X)*((A*X).')+(X.'*A*X)*A + A;
        Hr = K.'*H*K;
        % Cálculo do vetor de busca
        S = -Hr\Gr; % É mais rápido que inv(H)*G
        % Cálculo do vetor de busca S para todo X
        Stotal = K*S;
        Snorm = Stotal/norm(Stotal);
        %Limitando o passo t
        t = 1;
        while(norm(S)*t>100)
          t = 100/(norm(S)+0.1);
        end
        %Calculando o novo X
        R = X(7:8,1) + t*S(1:2,1); % S só tem duas posições
        X = K*R;
        %Tolerância
        epsilon = norm(X-Xant);
        %Tensor do histórico de X
        path = cat(2,X,path); %path não somente para R, mas para todo X
        %Histórico de S
        Spath = cat(2,Snorm,Spath);
```

```
        %Contador
        cont=cont+1;
    end
end
```

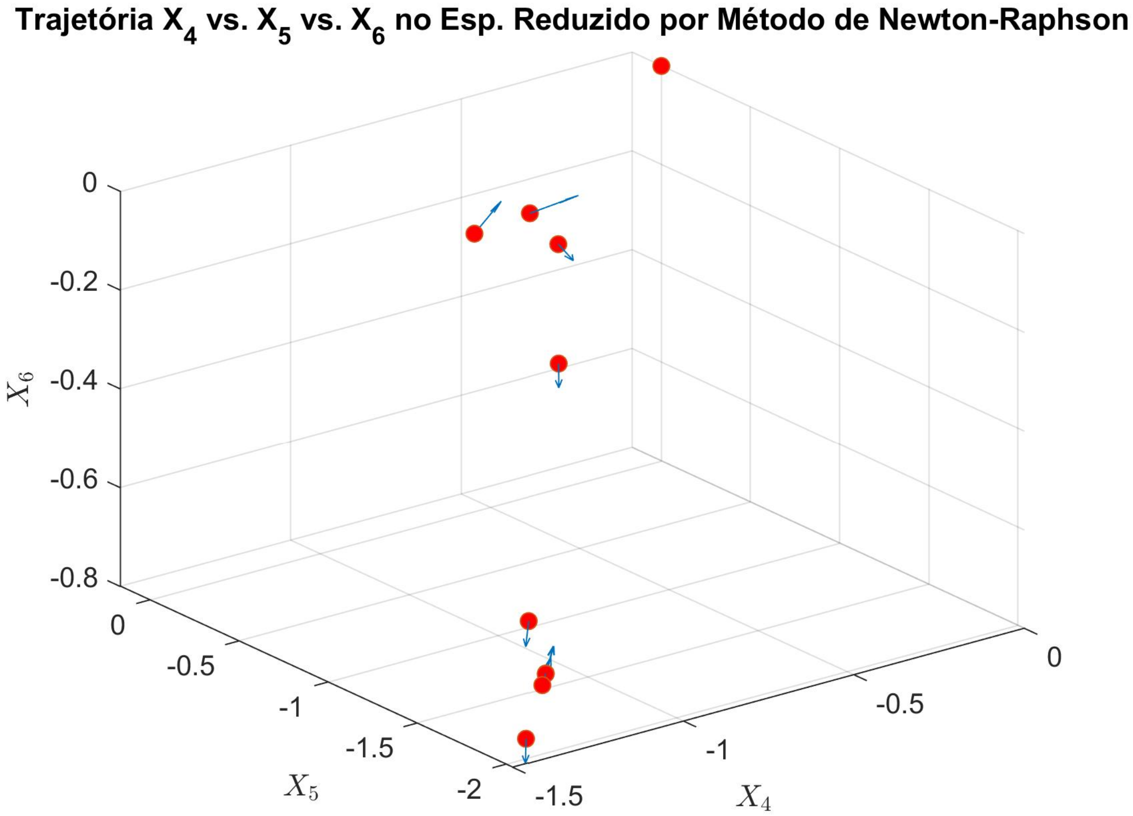
O valor de F minimizado no espaço reduzido por método de Newton-Raphson foi $8.1907 \cdot 10^4$. Foram necessárias dez iterações para a convergência. Os valores de $\underline{X}$ correspondentes foram:

$$\underline{X} = \begin{pmatrix} -5.9348 \\ 3.7708 \\ 1.7605 \\ -1.2727 \\ -1.7884 \\ -0.6942 \\ 3.2925 \\ -1.3223 \end{pmatrix} \tag{21}$$

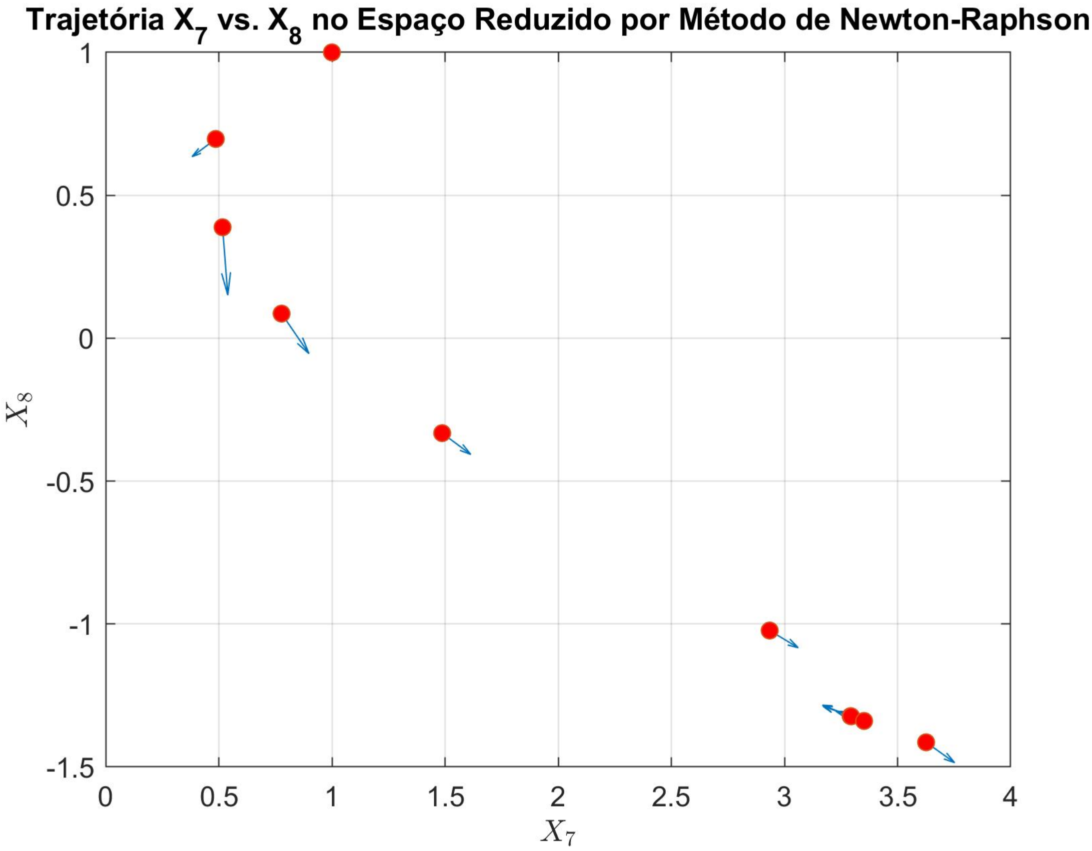
A figura 4 mostra as trajetórias de iterações por Newton-Raphson no Espaço Reduzido.



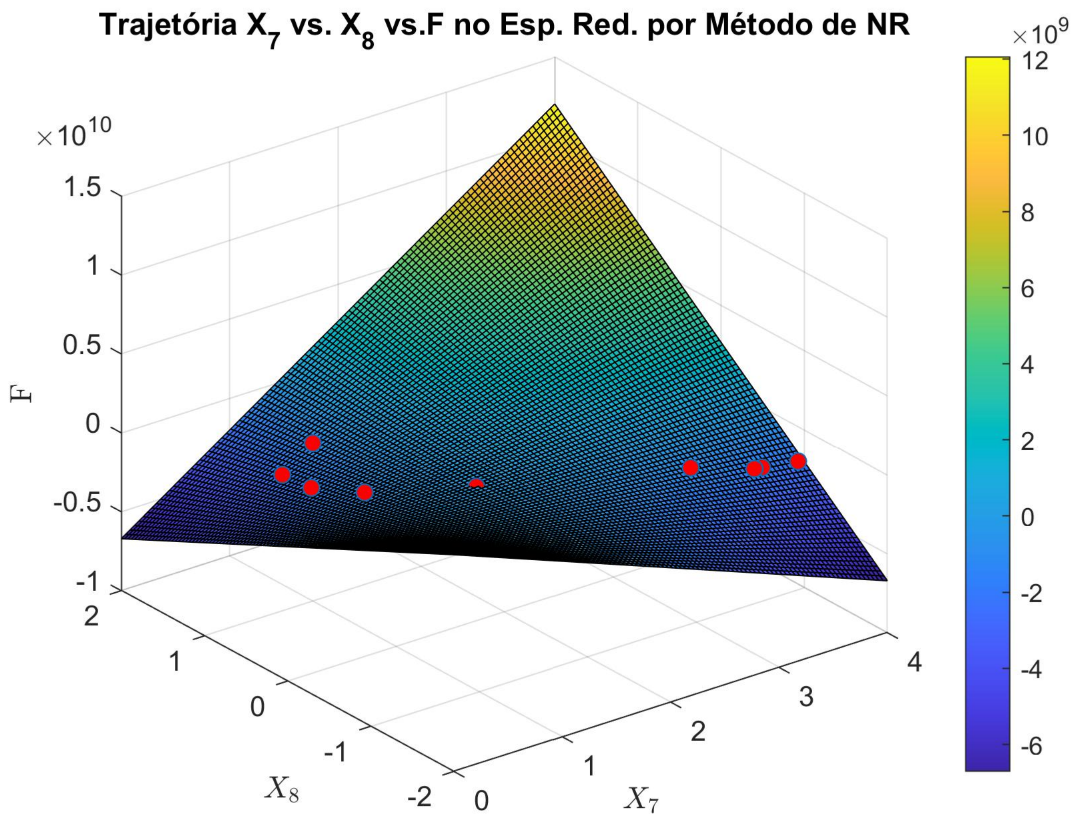
(a) Trajetória de X_1 vs. X_2 vs. X_3 por Método de Newton-Raphson no Espaço Reduzido.



(b) Trajetória de X_4 vs.X_5 vs. X_6 por Método de Newton-Raphson no Espaço Reduzido.



(c) Trajetória de X_7 vs. $X8$ por Método de Newton-Raphson no Espaço Reduzido.



(d) Trajetória de X_7 vs. $X8$ vs. valores de F por Método de Newton-Raphson no Espaço Reduzido.

Figura 4: Trajetórias por Método de Newton-Raphson no Espaço Reduzido.

Item (x)

Utilizando a hessiana no espaço reduzido conforme a eq. (20), procede-se de forma semelhante ao Item (vii), fazendo-se buscas unidimensionais nos vetores conjugados da hessiana.

```
function [X,path,cont,Spath] = powell_10a(X0,A,B,K)
%Sendo X vetor coluna
    X = X0; %chute inicial
    epsilon = 100;  %declarando a tolerância de forma o rodar o script
    cont= 0; % contador de iterações
    R = X(7:8);
```

```
    path = X0; %histórico
    Spath = zeros(size(X)); %o histórico do vetor busca
        while(epsilon >= 0.00001 && cont<10000)
          %Calculando o novo X
          Rant = R;
          X = K*R;
          %Cálculo da Hessiana em k
          Hr = K.'*(2*(A*X)*((A*X).')+(X.'*A*X)*A + A)*K;
          % Cálculo da Base conjugada da Hessiana
          P = conjugado_schmidt(Hr);
          %Estimativa inicial de X(teta):
          Rteta(:,1) = R(:,1);
          %Condição de pto estacionário
          pe =  @(teta) ponto_estacionario_10(P(:,2),Rteta(:,1),A,B,teta,K,X);
          teta = fzero(pe,0);
          %teta
          Rteta(:,2) = Rant(:,1) + P(:,2)*teta;
          R = Rteta(:,2);
        % Cálculo do vetor de busca
            S = R-Rant;
        % Cálculo do vetor de busca S para todo X
        Stotal = K*S;
        Snorm = Stotal/norm(Stotal);
        %Tolerância
            epsilon = norm(R-Rant);
        %Tensor do histórico de R
            path = cat(2,X,path); %path não somente para R, mas para todo X
        %Histórico de S
            Spath = cat(2,Snorm,Spath);
        %Contador
            cont=cont+1;
        end
end
```

De forma análoga ao Item (vii), o código auxiliar "ponto_estacionario_10" calcula o valor do gradiente no espaço reduzido, os valores na reta de busca reduzida de forma a encontrar o valor de θ que zera a equação pela função fzero.

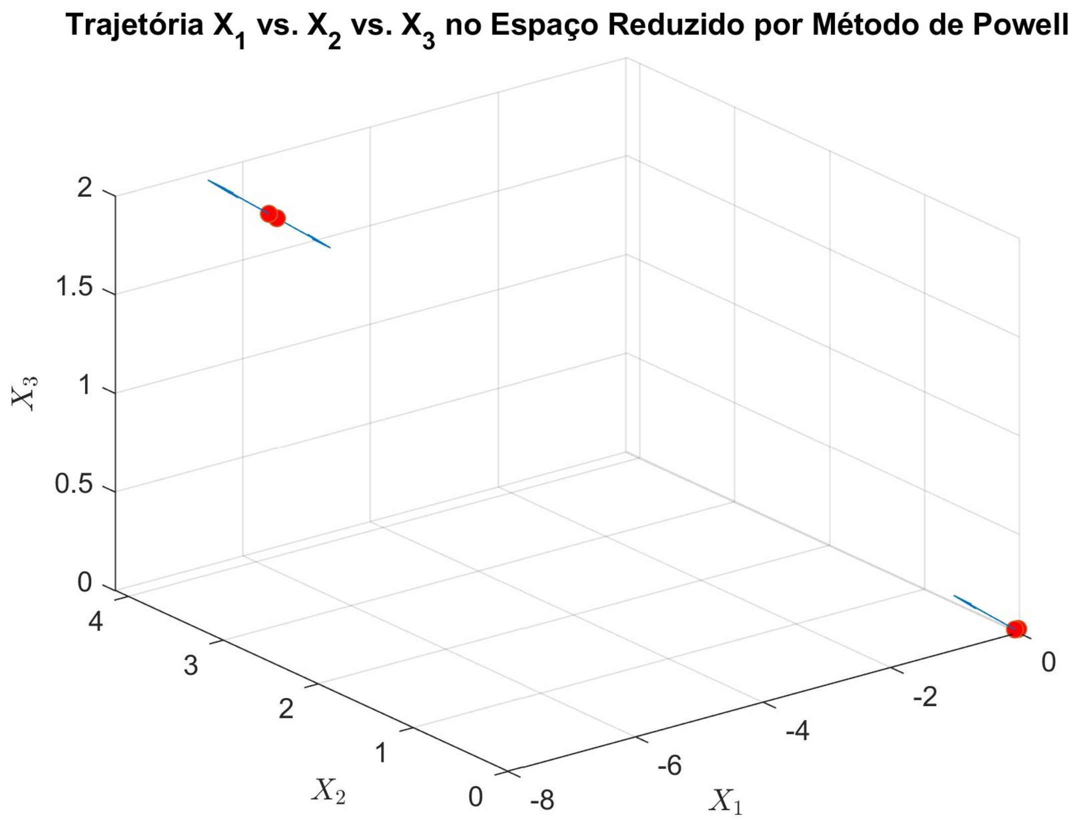
```
function f = ponto_estacionario_10(P,Rteta_ant,A,B,teta,K,X)
     Rteta = Rteta_ant + P*teta;
     X = K*Rteta;
```

```
    %Cálculo do gradiente
    G = (X.'*A*X)*A*X + A*X+B;
    Gr = K.'*G;
    f = P.'*Gr;
end
```

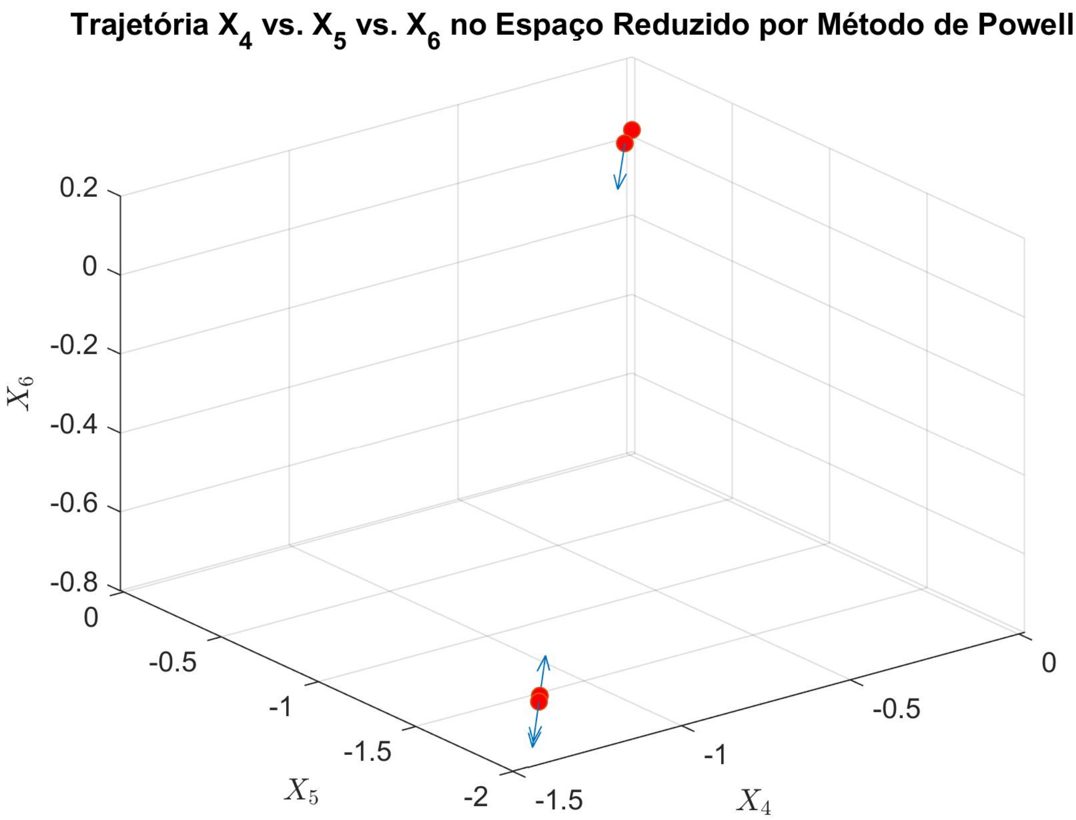
Após cinco iterações, o valor encontrado pelo Método Powell de Ponto Interior no Espaço Reduzido foi de $8.2345 \cdot 10^4$. O $\underline{\mathrm{X}}$ na região viável que satisfaz a $\underline{\underline{E}}\underline{\mathrm{X}} = 0$ é:

$$\underline{\mathrm{X}} = \begin{pmatrix} -6.0544 \\ 3.8198 \\ 1.8082 \\ -1.3063 \\ -1.7985 \\ -0.7135 \\ 3.3461 \\ -1.3029 \end{pmatrix} \tag{22}$$

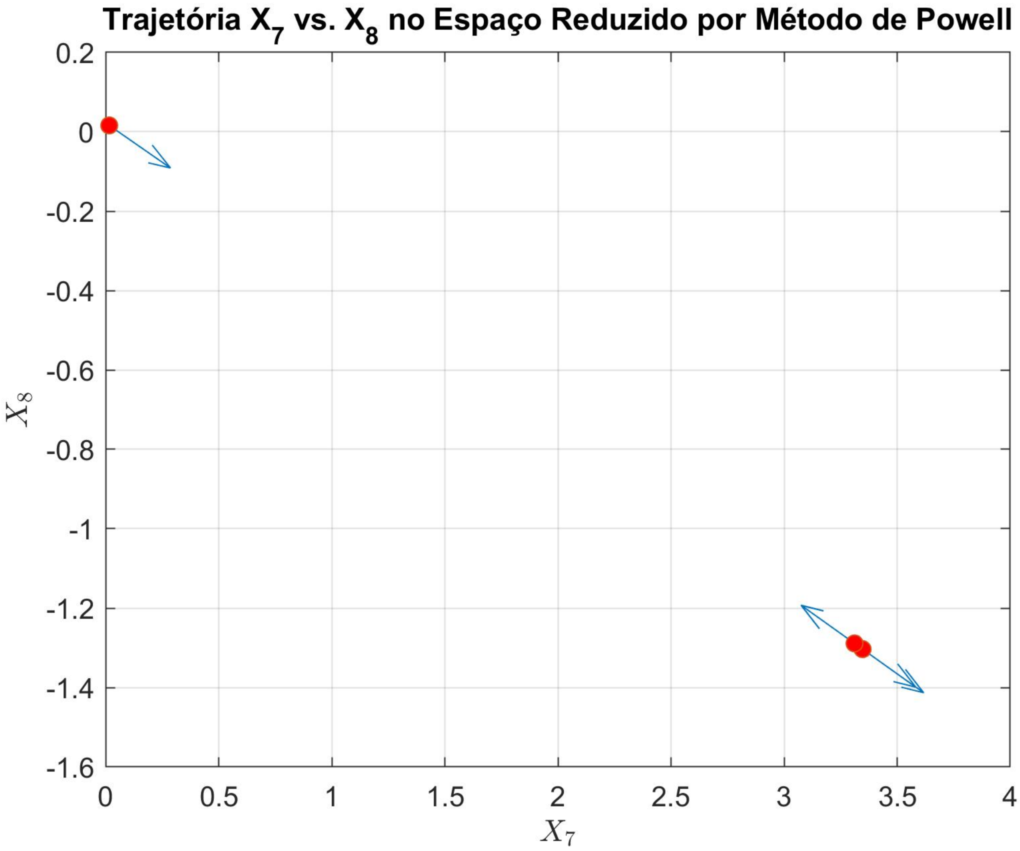
A figura 5 mostra as trajetórias de iterações por Método de Powell no espaço reduzido.



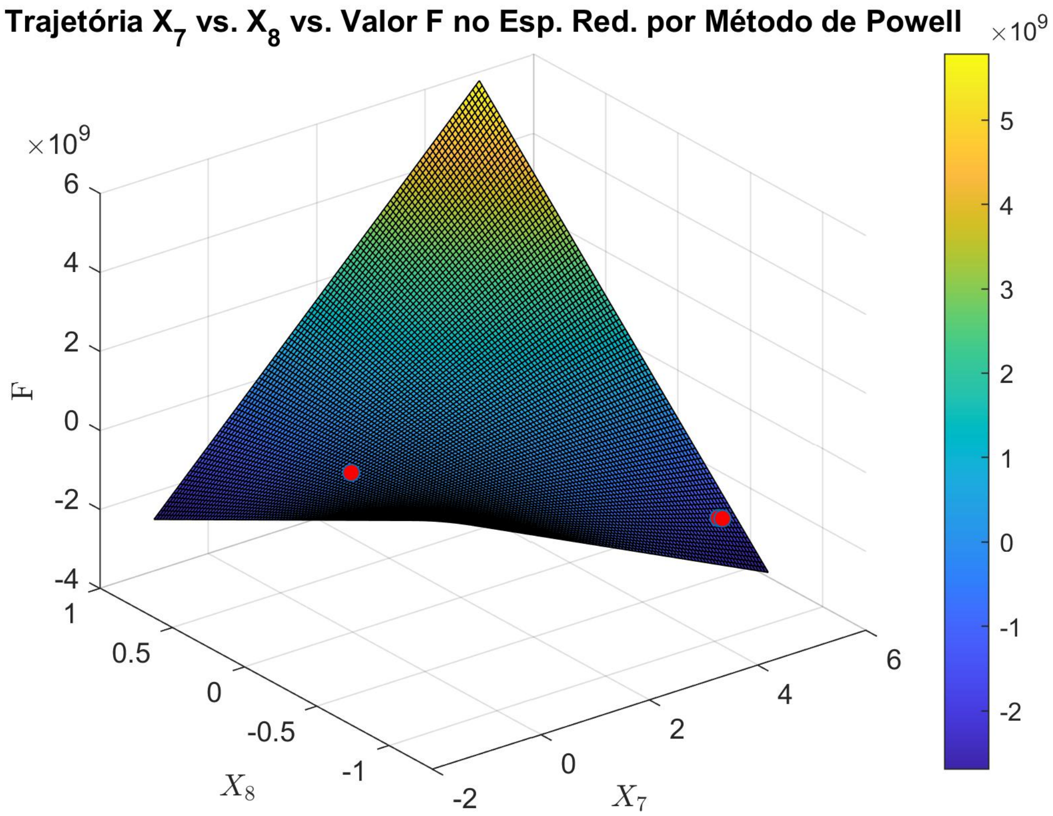
(a) Trajetória de X_1 vs. X_2 vs. X_3 por Método de Powell no Espaço Reduzido.



(b) Trajetória de X_4 vs.X_5 vs. X_6 por Método de Powell no Espaço Reduzido.



(c) Trajetória de X_7 vs. $X8$ por Método de Powell no Espaço Reduzido.



(d) Trajetória de X_7 vs. $X8$ vs. valores de F por Método de Powell.

Figura 5: Trajetórias por Método de Powell no Espaço Reduzido.

Apêndice

Código para executar o trabalho todo em MatLab.

Trabalho 1 - 2021/1 EQE 703 - Métodos Matemáticos Aplicados

Por Gustavo Caldas (gustavo.caldas@eq.ufrj.br) e Oscar Chamberlain (ochamberlain@eq.ufrj.br)

```
A = [ 1 1 1 1 1 1 1 1;
    1 2 3 4 5 6 7 8;
    1 3 6 10 15 21 28 36;
    1 4 10 20 35 56 84 120;
    1 5 15 35 70 126 210 330;
    1 6 21 56 126 252 462 792;
    1 7 28 84 210 462 924 1716;
    1 8 36 120 330 792 1716 3432 ];
```

Item ii

```
P = conjugado_schmidt(A);
```

Item iii

```
D = (P.'*A)*P;
```

Item iv

```
U = diagonalizacao_forma_quadratica(A);
Q = U.'*A*U;
autovalores = eig(A);
```

Item v - caráter da Matriz Hessiana

```
%H = 2*(A*X)*((A*X).')+(X.'*A*X)*A + A;

%H*P = P*lambda

% H*P = [2*(A*X)*((A*X)')+(X'*A*X)*A + A]*P = lambda*P
```

```
% Considerando que X'AX é um constante K e que (AX)' = X'A'=X'A
% H*P = [2*(A*X)X'+ K + I]AP = lambda*P
% O caracter de A definirá o caracter da hessiana.

% Para conhecer o caráter da matriz Hessiana, precisamos
obter o autovalor.
```

Item vi - Newton-Raphson

```
X0  = [0;0;0;0;0;0;0;0];
B=[11111;
22222;
33333;
44444;
44444;
33333;
22222;
11111];


% X é valor X^(n), onde n é a última iteração
% Iterations é o número de iterações
% path é o tensor histórico com os valores de X^(k): 8 x Iterations
% Spath é o tensor histórico de S^k: 8 x Iterations
[X,path,iterations,Spath] = newton_raphson(X0,A,B);

%Calculando os valores de F
C = 1e5;
F = (1/4)*(X.'*A*X)^2 +(0.5)*(X.'*A*X)+B.'*X + C; % último valor

% Cálculo do valor histórico de f
x = path;
f = zeros(size(x,2),1);
for i=1:size(x,2)
    f(i) = (1/4)*(x(:,i).'*A*x(:,i))^2
    +(0.5)*(x(:,i).'*A*x(:,i))+B.'*x(:,i) + C;
end

% Plotando!
figure(1);
```

```
quiver3(x(1,:),x(2,:),x(3,:),Spath(1,:),Spath(2,:),Spath(3,:),0.2);
grid on
hold on
plot3(x(1,:),x(2,:),x(3,:),'o','MarkerFaceColor','r');
xlabel({'$X_1$'},'Interpreter','latex');
ylabel({'$X_2$'},'Interpreter','latex');
zlabel({'$X_3$'},'Interpreter','latex');
title({'Trajetória X_1 vs. X_2 vs. X_3 por Método de
Newton-Raphson'},'Interpreter','tex');
print('x1x2x3-NR.jpg','-djpeg','-r300')

figure(2);
quiver3(x(4,:),x(5,:),x(6,:),Spath(4,:),Spath(5,:),Spath(6,:),0.2);
grid on
hold on
plot3(x(4,:),x(5,:),x(6,:),'o','MarkerFaceColor','r')
xlabel({'$X_4$'},'Interpreter','latex');
ylabel({'$X_5$'},'Interpreter','latex');
zlabel({'$X_6$'},'Interpreter','latex');
title({'Trajetória X_4 vs. X_5 vs. X_6 por Método de
Newton-Raphson'},'Interpreter','tex');
print('x4x5x6-NR.jpg','-djpeg','-r300')

figure(3);
quiver(x(7,:),x(8,:),Spath(7,:),Spath(8,:),0.1)
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
grid on
hold on
plot(x(7,:),x(8,:),'o','MarkerFaceColor','r')
title({'Trajetória X_7 vs. X_8 por Método de
Newton-Raphson'},'Interpreter','tex');
print('x7vsx8-NR.jpg','-djpeg','-r300')

figure(4);

plot3(x(7,:),x(8,:),f,'o','MarkerFaceColor','r')
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
zlabel({'F'},'Interpreter','latex');
```

```
title({'Trajetória X_7 vs. X_8 vs. Valor F por Método de
Newton-Raphson'},'Interpreter','tex');
print('x7vsx8vsF-NR.jpg','-djpeg','-r300')

%Calculando o traço de H;
H = 2*(A*X)*((A*X).')+(X.'*A*X)*A + A;
traco = trace(H);
```

Item vii - Método de Powell

```
%X0  = [100;10;101;10;10;10;200;5000];
[X_powell,path_powell,iterations_powell,Spath_powell] = powell(X0,A,B);

%Calculando os valores de F
C = 1e5;
F_powell = (1/4)*(X_powell.'*A*X_powell)^2
+(0.5)*(X_powell.'*A*X_powell)+B.'*X_powell + C; % último valor

% Cálculo do valor histórico de f
x_powell = path_powell;
f_powell = zeros(size(x_powell,2),1);
for i=1:size(x_powell,2)
    f_powell(i) = (1/4)*(x_powell(:,i).'*A*x_powell(:,i))^2
    +(0.5)*(x_powell(:,i).'*A*x_powell(:,i))+B.'*x_powell(:,i) + C;
end

% Plotando!
figure(5);
quiver3(x_powell(1,:),x_powell(2,:),x_powell(3,:),
Spath_powell(1,:),Spath_powell(2,:),Spath_powell(3,:),0.5);
grid on
hold on
plot3(x_powell(1,:),x_powell(2,:),x_powell(3,:),'o','MarkerFaceColor','r');
%contour(x(1,:),x(2,:),f);
%quiver(Spath(1,:),Spath(2,:),Spath(3,:))
xlabel({'$X_1$'},'Interpreter','latex');
ylabel({'$X_2$'},'Interpreter','latex');
zlabel({'$X_3$'},'Interpreter','latex');
title({'Trajetória X_1 vs. X_2 vs. X_3 por Método de
Powell'},'Interpreter','tex');
```

```
print('x1x2x3-powell.jpg','-djpeg','-r300')

figure(6);
quiver3(x_powell(4,:),x_powell(5,:),x_powell(6,:),
Spath_powell(4,:),Spath_powell(5,:),Spath_powell(6,:));
grid on
hold on
plot3(x_powell(4,:),x_powell(5,:),x_powell(6,:),'o','MarkerFaceColor','r')
xlabel({'$X_4$'},'Interpreter','latex');
ylabel({'$X_5$'},'Interpreter','latex');
zlabel({'$X_6$'},'Interpreter','latex');
title({'Trajetória X_4 vs. X_5 vs. X_6 por Método de
Powell'},'Interpreter','tex');
print('x4x5x6-powell.jpg','-djpeg','-r300')

figure(7);

quiver(x_powell(7,:),x_powell(8,:),Spath_powell(7,:),Spath_powell(8,:))
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
grid on
hold on
plot(x_powell(7,:),x_powell(8,:),'o','MarkerFaceColor','r')
title({'Trajetória X_7 vs. X_8 por Método de
Powell'},'Interpreter','tex');
print('x7vsx8-powell.jpg','-djpeg','-r300')

figure(8);

plot3(x_powell(7,:),x_powell(8,:),f_powell,'o','MarkerFaceColor','r')
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
zlabel({'F'},'Interpreter','latex');
title({'Trajetória X_1 vs. X_2 vs. Valor F por Método de
Powell'},'Interpreter','tex');
print('x7vsx8vsF-powell.jpg','-djpeg','-r300')
```

Item viii - Espaço Reduzido

```
E = [4 0 4 0 -3 -4 3 1
```

```
-2 -4 2 3 -2 2 0 -1
1 3 -3 -4 -3 0 -2 3
0 0 0 -1 1 4 1 0
3 1 4 3 -2 0 3 2
2 2 4 -4 -3 0 -4 0 ];

% Cálculo dos vetores com grau de liberdade livre
Mt = Trator(E);

% Cálculo da matriz constante K
K = -Mt(:,7:8);
K(7,1)=1;
K(8,2)=1;

% Plotando a projeção de F em função de F(X(R))
R(:,1) = linspace(-100,100);
R(:,2) = linspace(-100,100);

%Cálculo de X no Espaço Reduzido
X_reduz = K*R.';

% F no espaço reduzido
F_reduz = (1/4)*(X_reduz.'*A*X_reduz)^2
+(0.5)*(X_reduz.'*A*X_reduz)+B.'*X_reduz + C;

%Projeção no espaço reduzido

%Superfície
figure(9);

surf(R(:,1),R(:,2),F_reduz);
colorbar
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
zlabel({'F'},'Interpreter','latex');
title({'Superfície X_7 vs. X_8 vs. Valor F no Espaço
Reduzido'},'Interpreter','tex');
print('x7vsx8-surface_reduced.jpg','-djpeg','-r300')

% Contorno
```

```
figure(10);

contour(R(:,1),R(:,2),F_reduz);
colorbar
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
%zlabel({'F'},'Interpreter','latex');
title({'Contorno X_7 vs. X_8 vs. Valor F no Espaço
Reduzido'},'Interpreter','tex');
print('x7vsx8vsF-contour_reduced.jpg','-djpeg','-r300')
```

Item ix - Espaço Reduzido com Newton-Rapson

```
X0  = [0;0;0;0;0;0;1;1];
% X é valor X^(n), onde n é a última iteração
% Iterations é o número de iterações
% path é o tensor histórico com os valores de X^(k): 8 x Iterations
% Spath é o tensor histórico de S^k: 8 x Iterations
[X_nr_er,path_nr_er,iterations_nr_er,Spath_nr_er] =
newton_raphson_9(X0,A,B,K);

%Calculando os valores de F
C = 1e5;
F_nr_er = (1/4)*(X_nr_er.'*A*X_nr_er)^2
+(0.5)*(X_nr_er.'*A*X_nr_er)+B.'*X_nr_er + C; % último valor

% Cálculo do valor histórico de f
x_nr_er = path_nr_er;
f_nr_er  = zeros(size(x_nr_er,2),1);
for i=1:size(x_nr_er,2)
    f_nr_er(i) = (1/4)*(x_nr_er(:,i).'*A*x_nr_er(:,i))^2
    +(0.5)*(x_nr_er(:,i).'*A*x_nr_er(:,i))+B.'*x_nr_er(:,i) + C;
end

i_nr_er=size(x_nr_er,2);

% Plotando!
figure(11);
quiver3(x_nr_er(1,:),x_nr_er(2,:),x_nr_er(3,:),
Spath_nr_er(1,:),Spath_nr_er(2,:),Spath_nr_er(3,:),0.2);
```

```
grid on
hold on
plot3(x_nr_er(1,:),x_nr_er(2,:),x_nr_er(3,:),
'o','MarkerFaceColor','r');
xlabel({'$X_1$'},'Interpreter','latex');
ylabel({'$X_2$'},'Interpreter','latex');
zlabel({'$X_3$'},'Interpreter','latex');
title({'Trajetória X_1 vs. X_2 vs. X_3 no Espaço Reduzido
por Método de Newton-Raphson'},
'Interpreter','tex');
print('x1x2x3-NR_er.jpg','-djpeg','-r300')

figure(12);

quiver3(x_nr_er(4,:),x_nr_er(5,:),x_nr_er(6,:),
Spath_nr_er(4,:),Spath_nr_er(5,:),Spath_nr_er(6,:),0.2);
grid on
hold on
plot3(x_nr_er(4,:),x_nr_er(5,:),x_nr_er(6,:),
'o','MarkerFaceColor','r')
xlabel({'$X_4$'},'Interpreter','latex');
ylabel({'$X_5$'},'Interpreter','latex');
zlabel({'$X_6$'},'Interpreter','latex');
title({'Trajetória X_4 vs. X_5 vs. X_6 no Espaço Reduzido
por Método de Newton-Raphson'},
'Interpreter','tex');
print('x4x5x6-NR_er.jpg','-djpeg','-r300')

figure(13);

quiver(x_nr_er(7,:),x_nr_er(8,:),Spath_nr_er(7,:),Spath_nr_er(8,:),0.2)
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
grid on
hold on
plot(x_nr_er(7,:),x_nr_er(8,:),'o','MarkerFaceColor','r')
title({'Trajetória X_7 vs. X_8 no Espaço Reduzido
por Método de Newton-Raphson'},
'Interpreter','tex');
print('x7vsx8-NR_er.jpg','-djpeg','-r300')
```

```
% Plotando a projeção de F em função de F(X(R))
R_nr(:,1) = linspace(0,4);
R_nr(:,2) = linspace(-2,2);
%Cálculo de X no Espaço Reduzido
X_reduz_nr = K*R_nr.'; %valores de X para projeção
% F no espaço reduzido
F_reduz_nr = (1/4)*(X_reduz_nr.'*A*X_reduz_nr)^2
+(0.5)*(X_reduz_nr.'*A*X_reduz_nr)+B.'*X_reduz_nr + C;

%Projeção no espaço reduzido

figure(14);
surf(R_nr(:,1),R_nr(:,2),F_reduz_nr);
colorbar
grid on
hold on
plot3(x_nr_er(7,:),x_nr_er(8,:),f_nr_er,
'o','MarkerFaceColor','r')
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
zlabel({'F'},'Interpreter','latex');
title({'Trajetória X_7 vs. X_8 vs.F no Esp. Red. por Método de NR'},
'Interpreter','tex');
print('x7vsx8vsF-NR_er.jpg','-djpeg','-r300')
```

Item x - Espaço Reduzido com Método de Powell

```
[breaklines=true, breakanywhere=true]

E = [ 4 0 4 0 -3 -4 3 1
-2 -4 2 3 -2 2 0 -1
1 3 -3 -4 -3 0 -2 3
0 0 0 -1 1 4 1 0
3 1 4 3 -2 0 3 2
2 2 4 -4 -3 0 -4 0 ];

% Cálculo dos vetores com grau de liberdade livre
Mt = Trator(E);
```

```
% Cálculo da matriz constante K
K = -Mt(:,7:8);
K(7,1)=1;
K(8,2)=1;

%X0  = -[100;10;101;10;10;10;200;5000];
X0 = 0.016* [1 1 1 1 1 1 1 1]';% 0.3536
[X_powell_er,path_powell_er,iterations_powell_er,
Spath_powell_er] = powell_10a(X0,A,B,K);

%Calculando os valores de F
C = 1e5;
F_powell_er = (1/4)*(X_powell_er.'*A*X_powell_er)^2
+(0.5)*(X_powell_er.'*A*X_powell_er)+B.'*X_powell_er + C; % último valor

% Cálculo do valor histórico de f
x_powell_er = path_powell_er;
f_powell_er = zeros(size(x_powell_er,2),1);
for i=1:size(x_powell_er,2)
    f_powell_er(i) = (1/4)*(x_powell_er(:,i).'*A*x_powell_er(:,i))^2
    +(0.5)*(x_powell_er(:,i).'*A*x_powell_er(:,i))+B.'*
    x_powell_er(:,i) + C;
end

i_powell_er=size(x_powell_er,2);

% Plotando!
figure(15);
quiver3(x_powell_er(1,:),x_powell_er(2,:),x_powell_er(3,:),
Spath_powell_er(1,:),Spath_powell_er(2,:),
Spath_powell_er(3,:),0.2);
grid on
hold on
plot3(x_powell_er(1,:),x_powell_er(2,:),x_powell_er(3,:),
'o','MarkerFaceColor','r');
xlabel({'$X_1$'},'Interpreter','latex');
ylabel({'$X_2$'},'Interpreter','latex');
zlabel({'$X_3$'},'Interpreter','latex');
title({'Trajetória X_1 vs. X_2 vs. X_3 no Espaço Reduzido
por Método de Powell'},
```

```
'Interpreter','tex');
print('x1x2x3-powell_er.jpg','-djpeg','-r300')

figure(16);
quiver3(x_powell_er(4,:),x_powell_er(5,:),x_powell_er(6,:),
Spath_powell_er(4,:),
Spath_powell_er(5,:),Spath_powell_er(6,:),0.2);
grid on
hold on
plot3(x_powell_er(4,:),x_powell_er(5,:),x_powell_er(6,:),
'o','MarkerFaceColor','r')
xlabel({'$X_4$'},'Interpreter','latex');
ylabel({'$X_5$'},'Interpreter','latex');
zlabel({'$X_6$'},'Interpreter','latex');
title({'Trajetória X_4 vs. X_5 vs. X_6 no Espaço Reduzido
por Método de Powell'},
'Interpreter','tex');
print('x4x5x6-powell_er.jpg','-djpeg','-r300')

figure(17);

quiver(x_powell_er(7,:),x_powell_er(8,:),
Spath_powell_er(7,:),Spath_powell_er(8,:),0.2)
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
grid on
hold on
plot(x_powell_er(7,:),x_powell_er(8,:),'o','MarkerFaceColor','r')
title({'Trajetória X_7 vs. X_8 no Espaço Reduzido por
Método de Powell'},
'Interpreter','tex');
print('x7vsx8-powell_er.jpg','-djpeg','-r300')

% Plotando a projeção de F em função de F(X(R))
R_powell(:,1) = linspace(-1,5);
R_powell(:,2) = linspace(-1,1);
%Cálculo de X no Espaço Reduzido
X_reduz_powell = K*R_powell.'; %valores de X para projeção
% F no espaço reduzido
F_reduz_powell = (1/4)*(X_reduz_powell.'*A*X_reduz_powell)^2
```

```
+(0.5)*(X_reduz_powell.'*A*X_reduz_powell)+B.'*X_reduz_powell + C;

figure(18);
surf(R_powell(:,1),R_powell(:,2),F_reduz_powell);
colorbar
grid on
hold on
plot3(x_powell_er(7,:),x_powell_er(8,:),
f_powell_er,'o','MarkerFaceColor','r')
xlabel({'$X_7$'},'Interpreter','latex');
ylabel({'$X_8$'},'Interpreter','latex');
zlabel({'F'},'Interpreter','latex');
title({'Trajetória X_7 vs. X_8 vs. Valor F no Esp. Red.
por Método de Powell'},'Interpreter','tex');
print('x7vsx8vsF-powell_er.jpg','-djpeg','-r300')
```