

MEC 2403 - Otimização e Algoritmos para Engenharia Mecânica

Aluno: Felipe da Costa Pereira (Mestrado)

Lista 1: Exercício Matlab

1 – Seja a função $f(x_1, x_2)$ dada por:

$$f(x_1, x_2) = x_1^3 + 2x_1x_2^2 - x_2^3 - 20x_1$$

Desenhar o gráfico da curva contida em um plano Cartesiano, cujo eixo das abscissas corresponde à reta que passa pelos pontos $P_1 = (-0.7, 1.6)$ e $P_2 = (3.7, -0.4)$ e o eixo das ordenadas corresponde aos valores da função f . Utilizar o MATLAB e considerar apenas o trecho entre os pontos P_1 e P_2 .

Solução:

Função que avalia f no ponto x_0

In []:

```
function f_value = f_func(x0)
    f_value = x0(1)^3 + 2*x0(1).*x0(2)^2 - x0(2)^3 - 20*x0(1);
end
```

Dados do problema

In []:

```
x1 = linspace(-2,4,50);
x2 = linspace(-2,2,50);
[x1,x2] = meshgrid(x1,x2);

f = x1.^3 + 2.*x1.*x2.^2 - x2.^3 - 20.*x1;

P1 = [-0.7 1.6];
P2 = [3.7 -0.4];
```

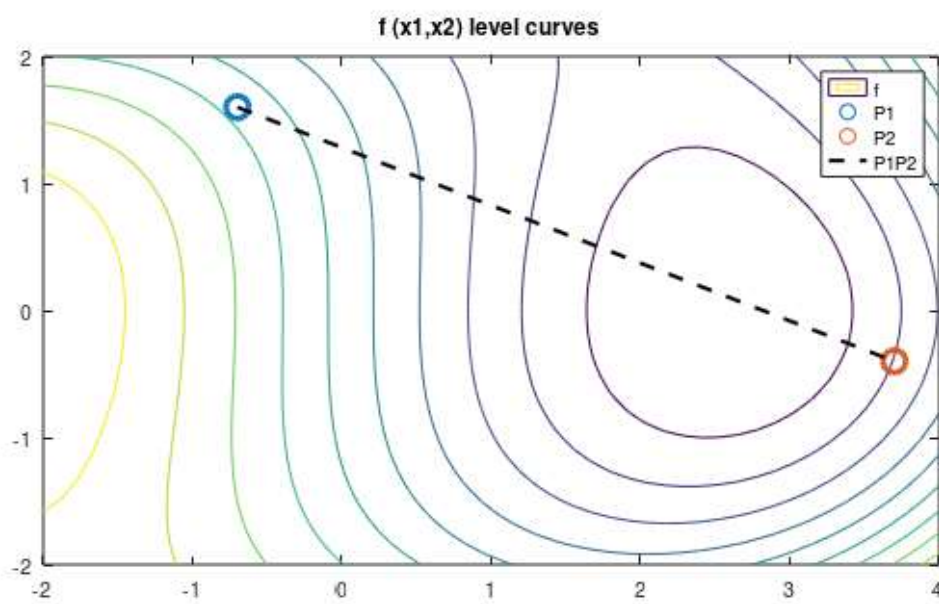
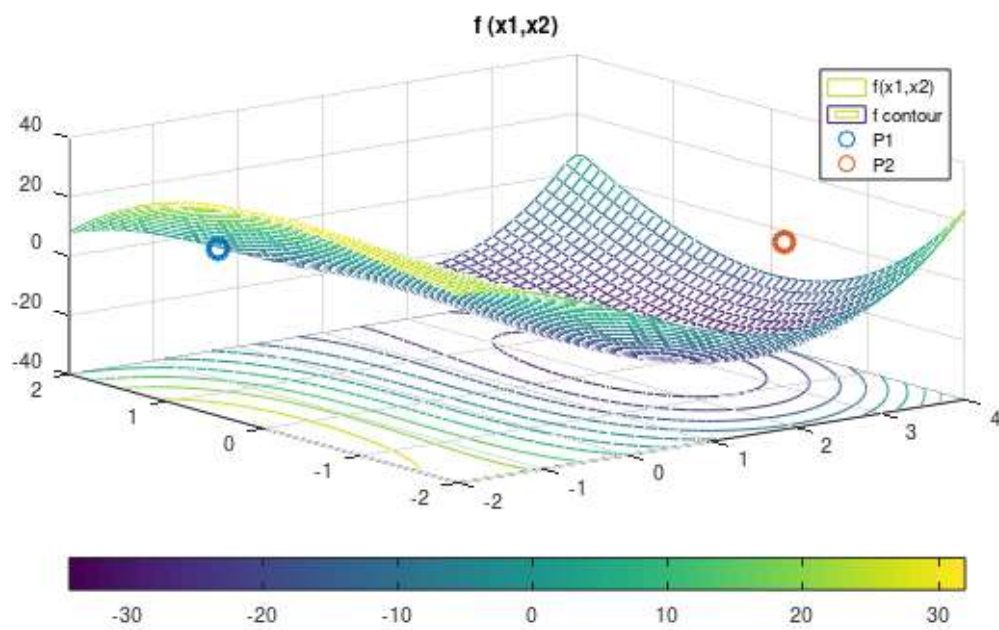
Plotando f , P_1 , P_2 e o segmento P_1P_2

In []:

```
set(gcf,'Position',[0 0 600 800])

% mesh with contours
subplot(2,1,1), meshc(x1,x2,f), title('f (x1,x2)');
hold on
plot3(P1(1), P1(2), 'o', 'LineWidth', 3, 'MarkerSize', 10);
plot3(P2(1), P2(2), 'o', 'LineWidth', 3, 'MarkerSize', 10);
legend('f(x1,x2)', 'f contour', 'P1', 'P2');
colorbar('southoutside')

subplot(2,1,2);
contour(x1,x2,f);
title('f (x1,x2) level curves');
hold on
plot(P1(1),P1(2),'o', 'LineWidth', 3, 'MarkerSize', 12)
plot(P2(1),P2(2),'o', 'LineWidth', 3, 'MarkerSize', 12)
plot ([P1(1);P2(1)], [P1(2);P2(2)], 'k--', 'LineWidth', 2)
legend('f', 'P1', 'P2', 'P1P2');
```



Número de divisões do segmento P_1P_2

In []:

n = 10;

Vetor \vec{d}

In []:

```
d = P2-P1
```

d =

```
4.4000 -2.0000
```

Cálculo de α e $f(\alpha)$

In []:

```
% inicialização
alpha = 0;
alpha_max = norm(d);
alphas = [];
fs = [];

while alpha <= alpha_max    % critério de parada: chegou em P2?
    % calcula x
    x = P1 + alpha * d/alpha_max;

    % calcula f(x), utilizando a funcao f definida anteriormente
    fs(end+1) = f_func(x);

    % salva alpha
    alphas (end+1) = alpha;

    % atualiza alpha
    alpha += alpha_max/n;
end
```

Valores de f nos pontos:

- $\alpha = 0$ (Ponto P_1)
- $\alpha = \alpha_{max}$ (Ponto P_2)

... e valor mínimo de f

In []:

```
fprintf('f(alpha=%i) = %i (Ponto P1) \n', alphas(1), fs(1))
fprintf('f(alpha=%i) = %i (Ponto P2) \n', alphas(end), fs(end))
fprintf('fmin = %i \n', min(fs))
```

```
f(alpha=0) = 5.977 (Ponto P1)
f(alpha=4.83322) = -22.099 (Ponto P2)
fmin = -33.9742
```

Plotando $f(\alpha)$

In []:

```
set(gcf,'Position',[0 0 1000 400])

subplot(1,2,1), contour(x1,x2,f), title('f (x1,x2)'), xlabel ('x1'), ylabel ('x2')
hold on
plot(P1(1),P1(2),'o', 'LineWidth', 3, 'MarkerSize', 12)
plot(P2(1),P2(2),'o', 'LineWidth', 3, 'MarkerSize', 12)
plot ([P1(1);P2(1)], [P1(2);P2(2)], 'k-', 'LineWidth', 2)
legend('f', 'P1', 'P2', 'P1P2');

subplot(1,2,2), plot(alphas, fs, 'k-o'), xlabel ('alpha'), ylabel ('f(alpha)'), title(
'f (alpha)')
```

