

1.10 Otimização

Prof. Dr. Sidney Bruce Shiki

e-mail: bruce@ufscar.br

Prof. Dr. Vitor Ramos Franco

e-mail: vrfranco@ufscar.br

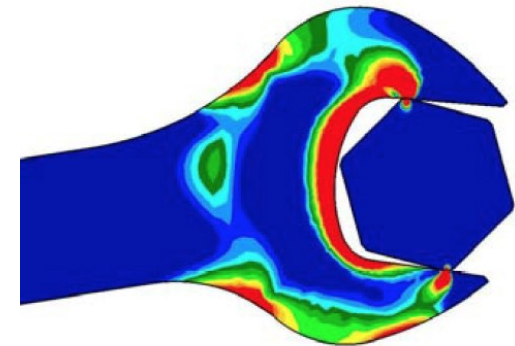
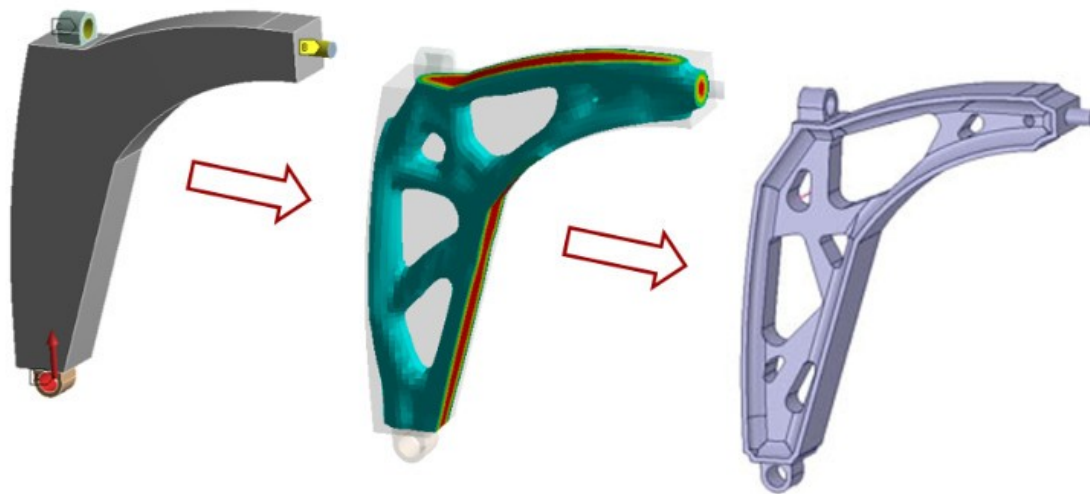
UFSCar – Universidade Federal de São Carlos

DEMec - Departamento de Engenharia Mecânica

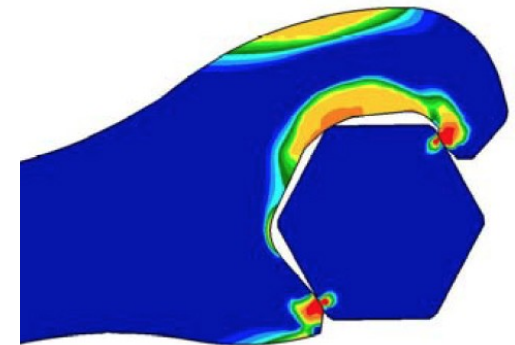
- Introdução
- Classificação
- Função objetivo/Função de custo
- Otimização linear
- Otimização não linear
- Ferramenta optimtool
- Noções de otimização global
- Exercícios

- Técnicas de otimização visam encontrar configurações ótimas para um determinado problema;
- Esses problemas devem ser formulados como problemas de:
 - Maximização: achar ponto de máximo;
 - Minimização: achar ponto de mínimo.
- O problema que será resolvido deve ser representado em uma função objetivo (ou função de custo).

- Aplicações:
 - Minimização de custos de um processo/maximização do lucro;
 - Maximização da capacidade de carga de uma estrutura mecânica;
 - Maximização da vida à fadiga;
 - Ajuste de modelos matemáticos à dados experimentais (e.g. ajuste de curvas).



Traditional Wrench



Optimized Wrench

- Será abordado nesse curso:
 - Formulação de problemas na forma de funções objetivo;
 - Otimização local com e sem restrições em MATLAB;
 - Noções de otimização global.

- Otimização:
 - Mono-objetivo: apenas uma função (e.g. minimizar custo de produção)
 - Multi-objetivo: múltiplas funções (e.g. maximizar resistência mecânica minimizando massa total)

- Otimização:

- Linear: função objetivo do problema é linear.

$$\min f(\mathbf{x}) = 20x_1 + 64x_2$$

$$25x_1 + 70x_2 \geq 2100$$

$$0 \leq x_1 \leq 70$$

$$0 \leq x_2 \leq 50$$

- Não linear: função objetivo é não linear.

$$\max f(\mathbf{x}) = 20x_1^2 + 64 \sin(x_2^3)$$

- Otimização:

Restrita: parâmetros tem restrição de valores

$$\min f(\mathbf{x}) = 20x_1^2 + 64x_2^3 \quad \left\{ \begin{array}{l} -70 \leq x_1 \leq 70 \\ -100 \leq x_2 \leq 50 \end{array} \right.$$

– Irrestrita: parâmetros não tem restrição de valores

$$\min f(\mathbf{x}) = 20x_1^2 + 64x_2^3$$

- Função objetivo ou função de custo são as funções matemáticas que descrevem o problema a ser otimizado em função de certos parâmetros;

$$\min f(\mathbf{x}) = \underbrace{20x_1 + 64x_2}_{\text{2 variáveis}}$$

$$25x_1 + 70x_2 \geq 2100$$

$$0 \leq x_1 \leq 70$$

$$0 \leq x_2 \leq 50$$

Restrições do problema

→ Problema de minimização

- Funções objetivos serão declaradas no MATLAB na forma de funções ou matrizes e vetores.

$$\min f(\mathbf{x}) = \underbrace{20x_1 + 64x_2}_{\text{2 variáveis}}$$

$$25x_1 + 70x_2 \geq 2100$$

$$0 \leq x_1 \leq 70$$

$$0 \leq x_2 \leq 50$$

Restrições do problema

→ Problema de minimização

- Exploraremos para otimização linear a função `linprog()`

```
X = linprog(f, A, b, Aeq, beq, LB, UB) ;
```

f – vetor descrevendo função objetivo

A e b – desigualdades tal que $A \cdot b \leq b$

A_{eq} e b_{eq} -igualdades tal que $A_{eq} \cdot b_{eq} = b_{eq}$

LB e UB – limite inferior (LB) e superior (UB) das variáveis

- Exemplo de criação – resolver o problema:

$$\min f(\mathbf{x}) = 20x_1 + 64x_2$$

$$25x_1 + 70x_2 \geq 2100$$

$$0 \leq x_1 \leq 70$$

$$0 \leq x_2 \leq 50$$

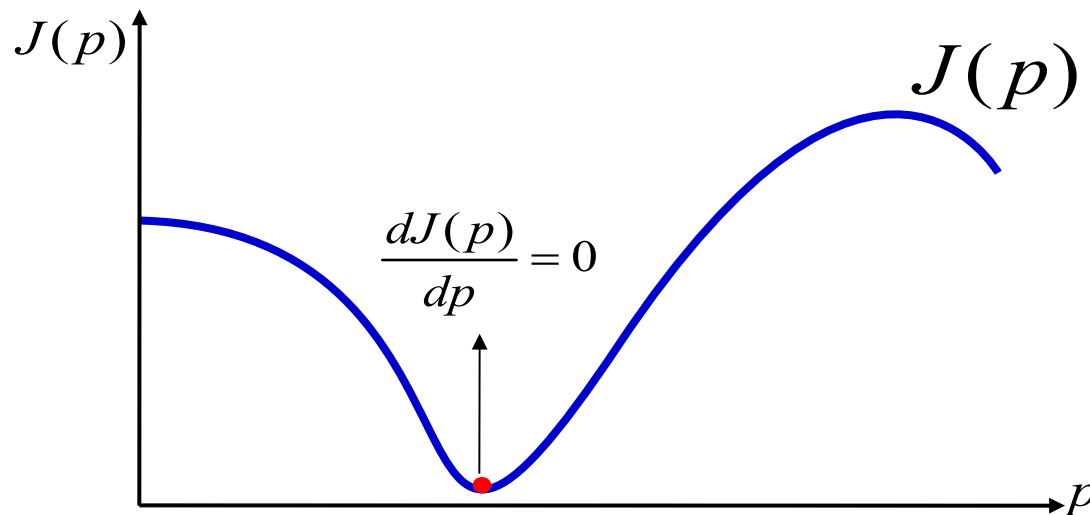
```
% Função objetivo
f = [20 64];
A = [-25 -70]; b = -2100;           % desigualdades
Aeq = []; beq = [];                % igualdades
LB = [0;0];                        % limite inferior
UB = [70;50];                      % limite superior
X = linprog(f,A,b,Aeq,beq,LB,UB);
% X contém as soluções x1 e x2
```

- Exercício 1 –Resolver o problema abaixo:

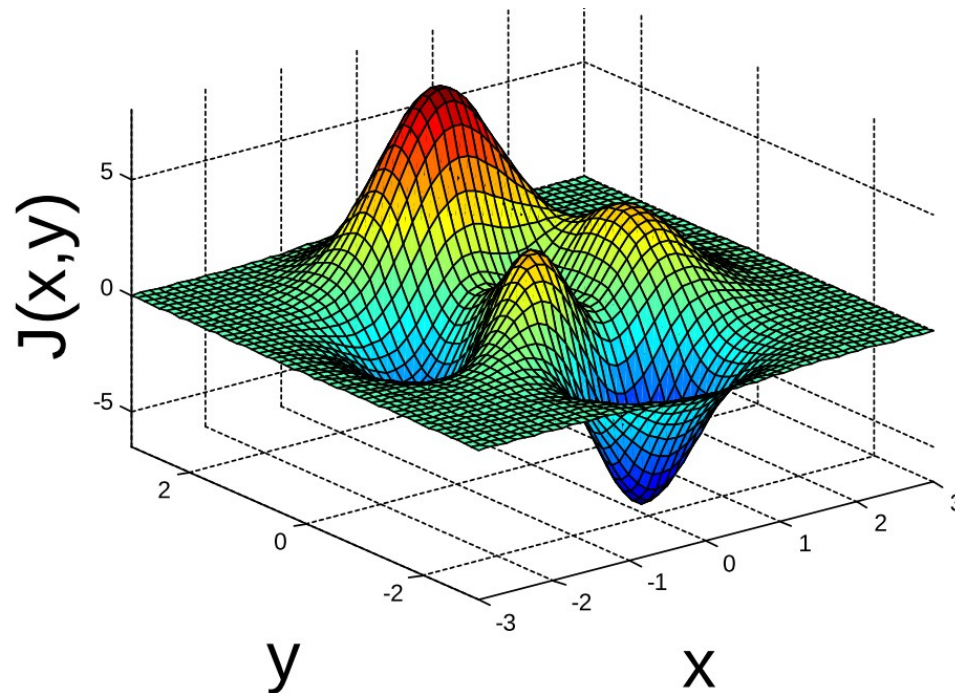
$$\min f(\mathbf{x}) = 0.5x_1 + 2x_2 + 2x_3 \quad \left\{ \begin{array}{l} 5x_1 + 10x_2 + 7x_3 \leq 230 \\ x_1 + x_2 + x_3 = 40 \\ -10 \leq x_1 \leq 90 \\ -10 \leq x_2 \leq 90 \\ -10 \leq x_3 \leq 90 \end{array} \right.$$

- Exercício 2 – Uma viga de concreto armado tem sua rigidez estipulada segundo a seguinte expressão: $f(\mathbf{v}) = 7v_1 + 8v_2$ v_1 – volume do componente 1 [L]
 v_2 – volume do componente 2 [L]
- O litro do componente 1 custa R\$ 10,00 reais e do componente 2 custa R\$ 80,00 reais, o custo total desses componentes deve ser no máximo de R\$ 5000,00 reais para seu projeto;
- O fornecedor dos componentes fornece esses componentes apenas de modo que a soma de seus volumes seja igual à 90 litros;
- O fornecedor pode fornecer no mínimo 5 litros, e no máximo 90 litros de cada componente;
- Calcular os volumes dos componentes 1 e 2 para maximizar a rigidez da viga de concreto dadas as restrições acima.

- Técnicas de otimização não linear são usadas para se minimizar/maximizar funções objetivo mais gerais;
- Boa parte desses métodos procura um ponto de gradiente (derivada) quase nulo na superfície da mesma (as vezes isso pode ser feito de forma analítica).



- Em funções simples é possível achar esses pontos facilmente, mas com múltiplas variáveis ou em funções mais elaboradas isso passa a ser mais complexo.





- Otimização irrestrita (`fminsearch`) – sintaxe:

```
X = fminsearch(fun,x0,options);
```

`fun` – função objetivo

`x0` – estimativa inicial

`options` – opções do algoritmo de otimização

- As opções podem ser configuradas com:

```
options = optimset();
```

- Exemplo de criação – resolver o problema:

$$\min f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

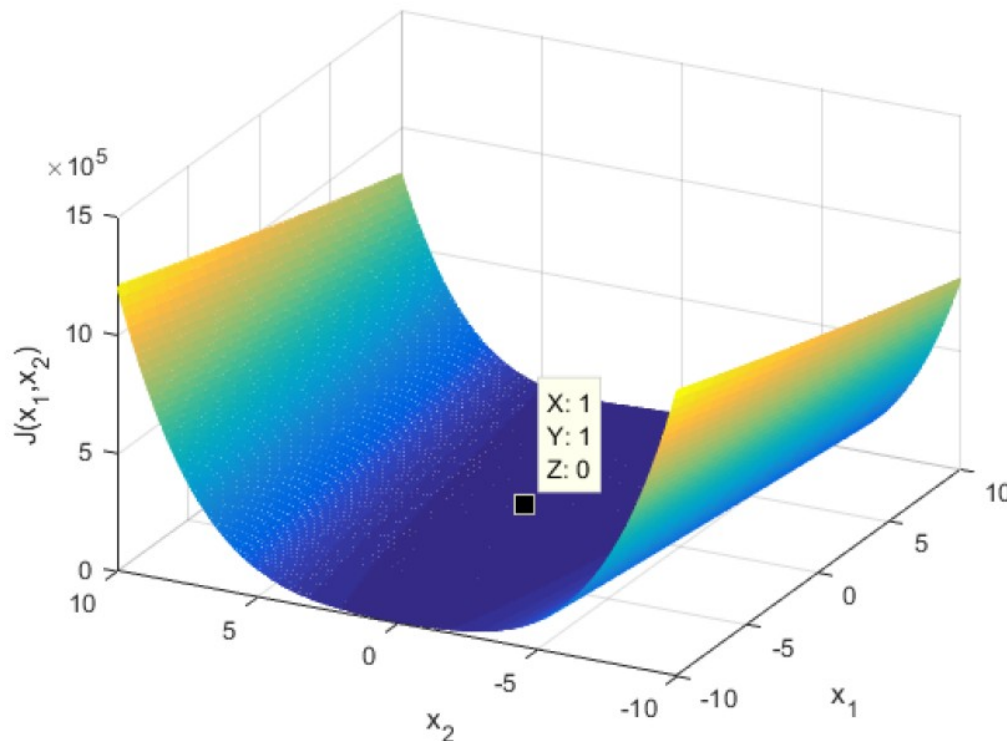
```
% Função objetivo
fun = @(x) 100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;
% Opções - display por iteração e plot da FObj
options =
optimset('Display','iter','PlotFcns',@optimplotfval);
x0 = [100,100];      % Chute inicial
x = fminsearch(fun,x0,options);
```

Como garantir que a solução achada é a ótima?

- Exemplo de criação – resolver o problema:

$$\min f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Como garantir que a solução achada é a ótima?



Mapeie as soluções possíveis!
(busca exaustiva)

- Otimização restrita (`fmincon`) – sintaxe:

```
X = fmincon(fun,x0,A,B,Aeq,Beq, LB,UB,NLCon,options);
```

`fun` – função objetivo

`x0` – estimativa inicial

`A` e `b` – desigualdades tal que $A \cdot b \leq b$

`Aeq` e `beq` -igualdades tal que $Aeq \cdot beq = beq$

`LB` e `UB` – limite inferior (LB) e superior (UB) das variáveis

`NLCon` – restrições não lineares

`options` – opções do algoritmo de otimização

- Exercício 3 – resolver o problema:

$$\min f(\mathbf{x}) = x_1^2 + 10x_2^2 - 3x_1x_2 \quad \left\{ \begin{array}{l} 2x_1 + x_2 \geq 2100 \\ x_1 + x_2 \geq -5 \\ -5 \leq x_1 \leq 5 \\ -5 \leq x_2 \leq 5 \end{array} \right.$$

```
X = fmincon(fun,x0,A,b,Aeq,beq, LB,UB,NLCon,options);
```

fun – função objetivo

x0 – estimativa inicial

A e b – desigualdades tal que $A \cdot b \leq b$

Aeq e beq -igualdades tal que $Aeq \cdot beq = beq$

LB e UB – limite inferior (LB) e superior (UB) das variáveis

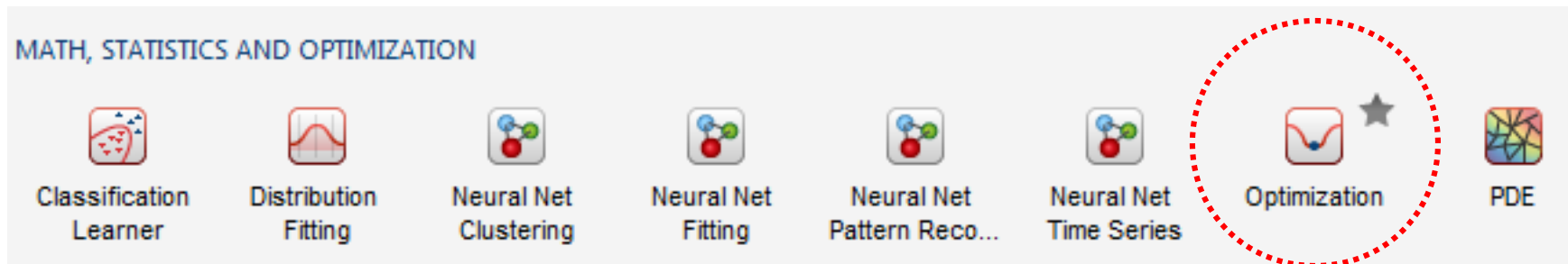
NLCon – restrições não lineares

options – opções do algoritmo de otimização

- Assim como para outras toolboxes do MATLAB, existe uma interface gráfica para lidar com problemas de otimização:
- A mesma pode ser chamada por:

```
optimtool;
```

- Ou ainda na aba APPS:



The screenshot displays the Optimization Tool interface, which is divided into three main sections: Problem Setup and Results, Options, and Quick Reference.

Problem Setup and Results:

- Solver:** fmincon - Constrained nonlinear minimization
- Algorithm:** Interior point
- Problem:**
 - Objective function: (empty field)
 - Derivatives: Approximated by solver
 - Start point: (empty field)
- Constraints:**
 - Linear inequalities: A: (empty field) b: (empty field)
 - Linear equalities: Aeq: (empty field) beq: (empty field)
 - Bounds: Lower: (empty field) Upper: (empty field)
 - Nonlinear constraint function: (empty field)
 - Derivatives: Approximated by solver
- Run solver and view results:**
 - Buttons: Start, Pause, Stop
 - Current iteration: (empty field) Clear Results
 - Final point: (empty field)

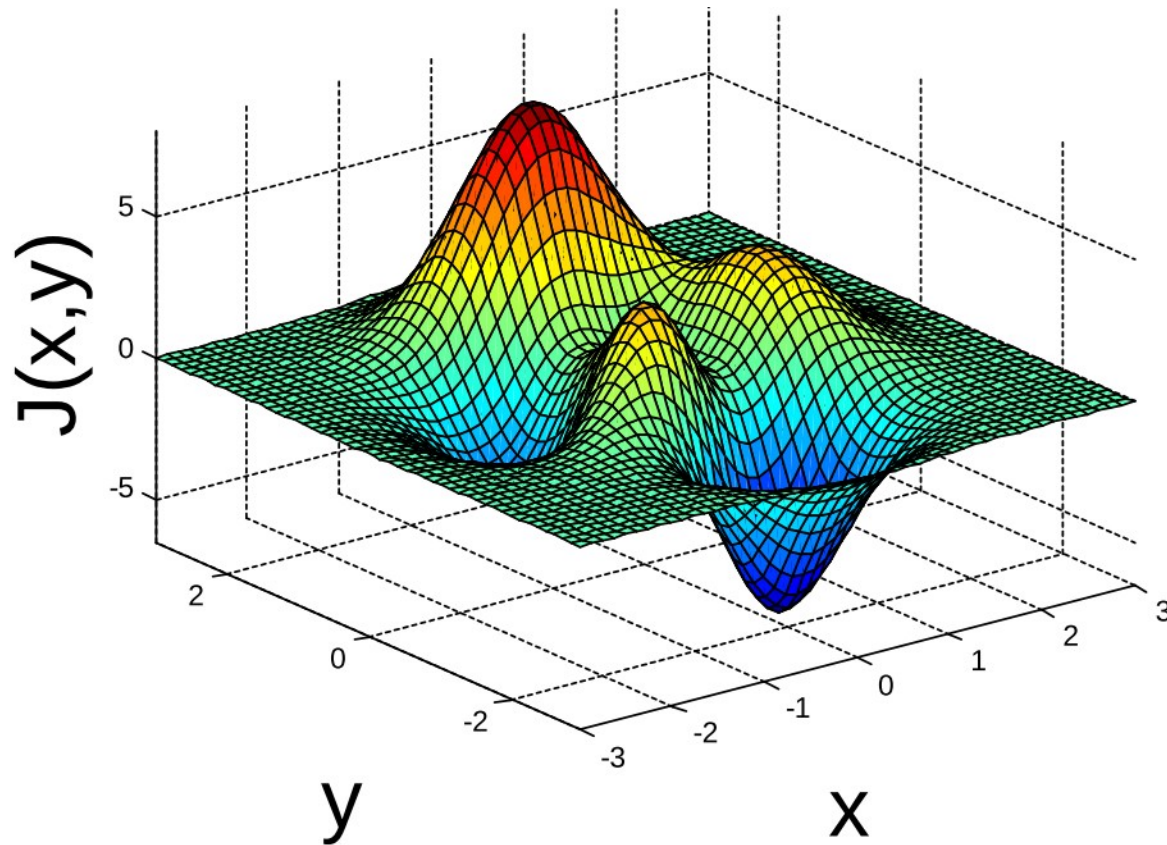
Options:

- Stopping criteria:**
 - Max iterations: ☒ Use default: 1000 ☐ Specify: (empty field)
 - Max function evaluations: ☒ Use default: 3000 ☐ Specify: (empty field)
 - X tolerance: ☒ Use default: 1e-10 ☐ Specify: (empty field)
 - Function tolerance: ☒ Use default: 1e-6 ☐ Specify: (empty field)
 - Constraint tolerance: ☒ Use default: 1e-6 ☐ Specify: (empty field)
 - SQP constraint tolerance: ☒ Use default: 1e-6 ☐ Specify: (empty field)
 - Unboundedness threshold: ☒ Use default: -1e20 ☐ Specify: (empty field)
- Function value check:**
 - ☐ Error if user-supplied function returns Inf, NaN or complex
- User-supplied derivatives:**
 - ☐ Validate user-supplied derivatives
- Hessian sparsity pattern: ☒ Use default: sparse(ones(numberOfVariables)) ☐ Specify: (empty field)
- Hessian multiply function: ☒ Use default: No multiply function ☐ Specify: (empty field)

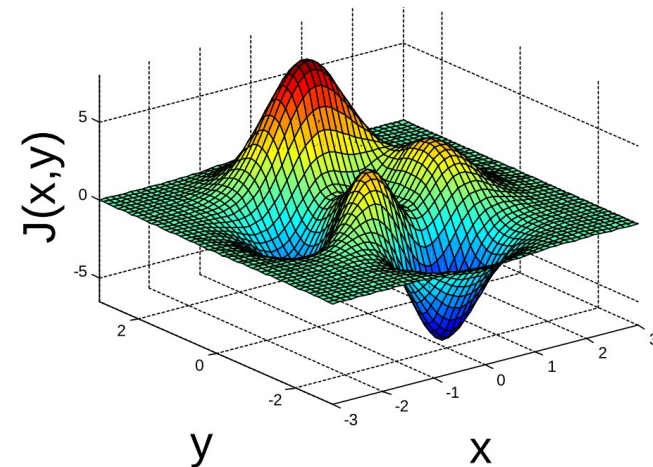
Quick Reference:

- fmincon Solver**
 - Find a minimum of a constrained nonlinear multivariable function using the interior-point algorithm.
 - Click to expand the section below corresponding to your task.
- Problem Setup and Results**
 - [Solver and Algorithm](#)
 - [Problem](#)
 - [Constraints](#)
 - [Run solver and view results](#)
- Options**
 - [Stopping criteria](#)
 - [Function value check](#)
 - [User-supplied derivatives](#)
 - [Approximated derivatives](#)
 - [Hessian](#)
 - [Algorithm settings](#)
 - [Inner iteration stopping criteria](#)
 - [Plot functions](#)
 - [Output function](#)
 - [Display to command window](#)
- Suggested Next Steps**
 - [Overview of Next Steps](#)
 - [When the Solver Fails](#)
 - [When the Solver Might Have Succeeded](#)

- O que fazer com um problema com vários mínimos locais?



- O que fazer com um problema com vários mínimos locais?
 - Fazer uma busca exaustiva (elevado custo computacional);
 - Usar vários chutes iniciais diferentes para tentar chegar na melhor resposta;
 - Utilizar algoritmos heurísticos (algoritmos genéticos, recozimento simulado, enxame de abelhas, colônia de formigas,...).



- Exercício 4 -

Perguntas ?