PQI-5884 - Programação Inteira Mista aplicada à Otimização de Processos
3º Período 2020

Data	Atividade	Conteúdo
17/09	Aula 1	Introdução, formulação, classes, representação
24/09	Aula 2	Condições de otimalidade
01/10	Aula 3	Condições KKT, multiplicadores
08/10	**Aula 4**	**Otimização irrestrita**
15/10	Aula 5	LP
22/10	Aula 6	NLP
29/10	Aula 7	MILP
05/10	Aula 8	MILP, problemas clássicos
12/11	Aula 9	MILP, problema de scheduling
19/11	Aula 10	MINLP, problema de síntese
26/11	Aula 11	Apresentações
03/12	-	-

OTIMIZAÇÃO SEM RESTRIÇÕES
pág. 32

I) Otimização unidimensional sem restrições (*line search*)

min $f(x)$
s.a. $x \in \Re^1$

NGL = 1

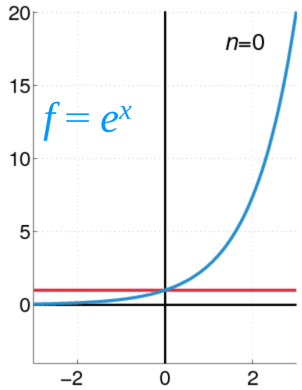
Métodos:
- analítico ($\partial f/\partial x = 0$ e $\partial^2 f/\partial x^2 > 0$)
- redução de intervalo
- aproximação polinomial
- baseados em derivadas (analíticas ou numéricas)

Método de Newton-Raphson

Série de Taylor para aproximação de $f(x)$ em um ponto a:

$f(x) = f(a + \Delta x) \cong$

$$f(a) + \frac{f'(a)}{1!}\underbrace{(x-a)}_{\Delta x} + \frac{f''(a)}{2!}\underbrace{(x-a)}_{\Delta x}^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots,$$



$n = 0$: $f(x) = f(a)$, constante

$n = 1$: aproximação linear em a

$n = 2$: aproximação quadrática em a

$n = 3$: aproximação cúbica em a

...

Método de Newton-Raphson

min $f(x)$
s.a. $x \in \Re^1$

Como x^* é ponto estacionário: $f'(x^*) = 0$.
A primeira derivada pode ser aproximada por uma série:
$f'(x) \approx f'(x_k) + f''(x_k).(x - x_k) + \ldots$

Para $f'(x^*) = 0 \rightarrow$ $x^* \approx x_k - \dfrac{f'(x_k)}{f''(x_k)}$

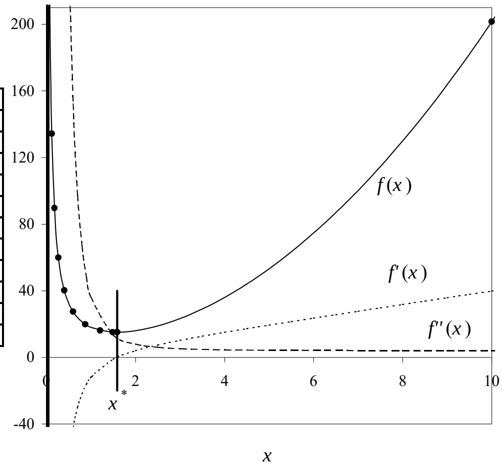
$$x_{k+1} \approx x_k - \frac{f'(x_k)}{f''(x_k)}$$

Desvantagem : as iterações podem divergir

Método da Newton-Raphson

EXEMPLO: obter o mínimo de $f(x) = 2.x^2 + 16/x$ partindo de $x_0 = 10$.
Derivadas analíticas: $f'(x) = 4.x - 16/x^2$
$f''(x) = 4 + 32/x^3$

iter.	x	f(x)	f'(x)	f"(x)
1	10.000	201.60	39.84	4.03
2	0.119	134.43	-1128.48	18970.53
3	0.179	89.68	-501.26	5627.27
4	0.268	59.93	-222.35	1673.71
5	0.400	40.28	-98.17	502.29
6	0.596	27.56	-42.67	155.23
7	0.871	19.89	-17.62	52.46
8	1.207	16.17	-6.16	22.22
9	1.484	15.19	-1.33	13.79
10	1.580	15.12	-0.08	12.11
11	1.587	15.12	0.00	12.00



33 cálculos de função

Método Quase-Newton

Aproximações numéricas das derivadas

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2.h}$$

$$f''(x) \approx \frac{f(x+h) - 2.f(x) + f(x-h)}{h^2}$$

Método Quasi-Newton

<u>EXEMPLO</u>: obter o mínimo de $f(x) = 2.x^2 + 16/x$ partindo de $x_0 = 10$ usando derivadas numéricas com $h = 1.10^{-5}$.

h	1E-5

iter.	x	f(x)	f(x+h)	f(x-h)	f'(x)	f''(x)
1	10.000	201.60	202.00	201.20	39.84	4.03
2	0.119	134.43	124.02	146.75	-1136.50	19105.25
3	0.179	89.68	84.94	94.99	-502.83	5644.92
4	0.268	59.93	57.79	62.24	-222.66	1676.02
5	0.400	40.27	39.32	41.28	-98.23	502.59
6	0.596	27.56	27.14	27.99	-42.69	155.26
7	0.871	19.89	19.72	20.07	-17.62	52.46
8	1.207	16.17	16.11	16.23	-6.16	22.22
9	1.484	15.19	15.17	15.20	-1.33	13.79
10	1.580	15.12	15.12	15.12	-0.08	12.11
11	1.587	15.12	15.12	15.12	0.00	12.00

33 cálculos de função

OTIMIZAÇÃO SEM RESTRIÇÕES
pág.42

II) Otimização multivariável sem restrições

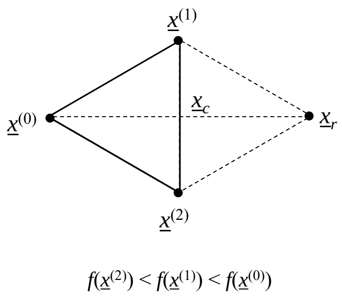
min $f(\underline{x})$
s.a. $\underline{x} \in \Re^n$

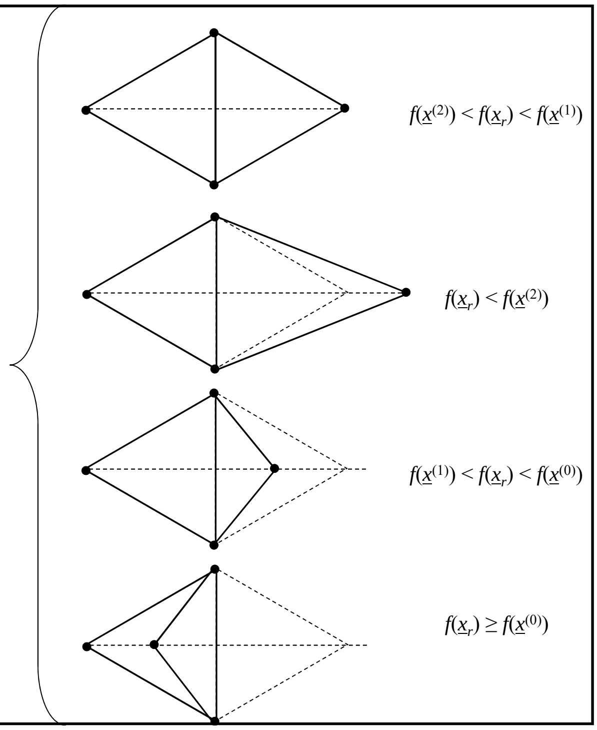
NGL = n

Métodos:

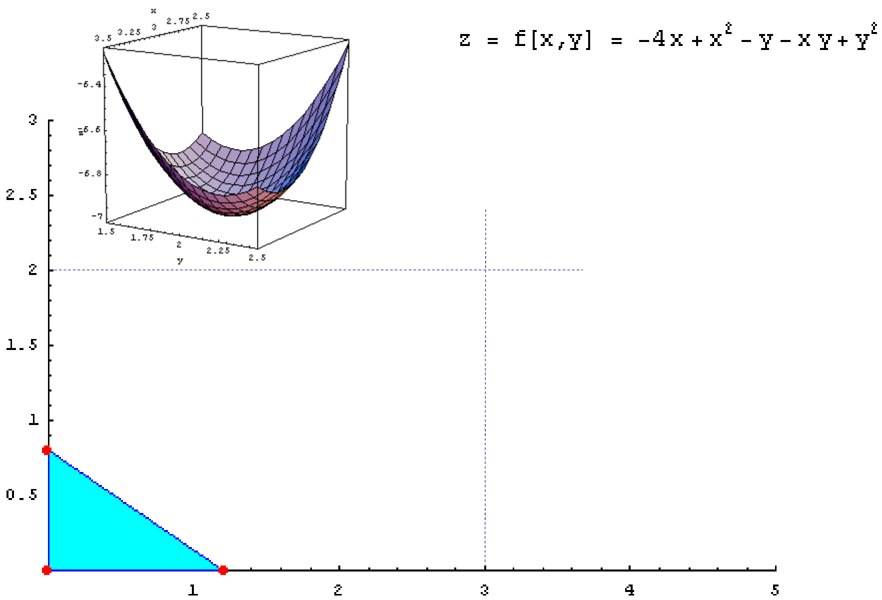
- analítico
- busca direta
- baseados em derivadas (analíticas ou numéricas)

Busca Direta Simplex (Nelder e Mead)

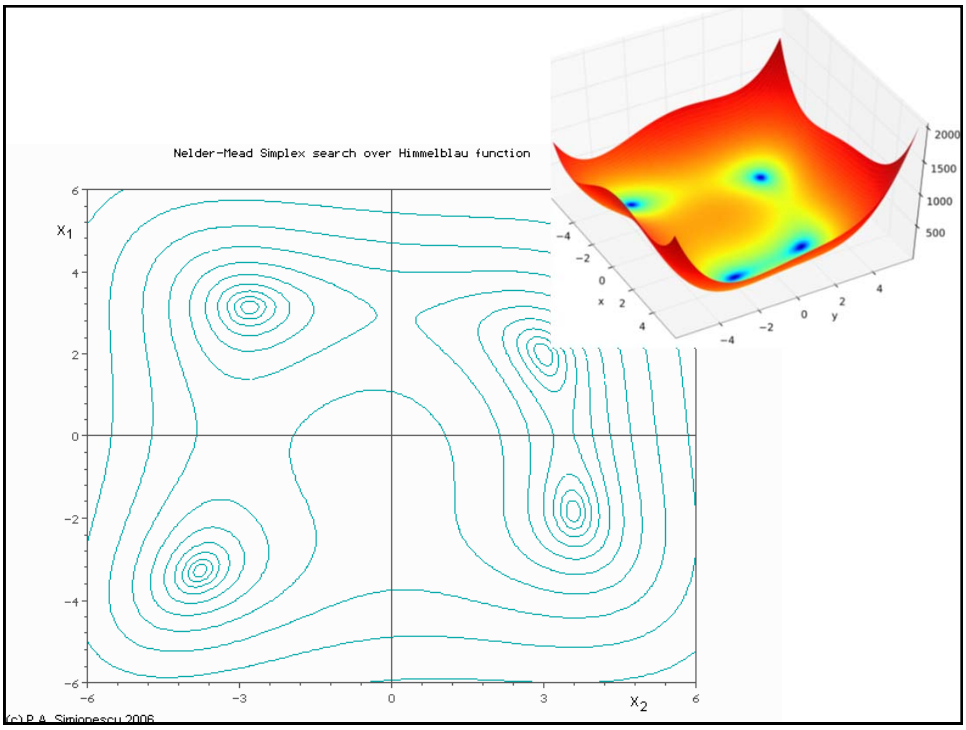




Método de Busca Direta Simplex (Nelder e Mead)



http://mathfaculty.fullerton.edu/mathews/n2003/neldermead/NelderMeadMod/Links/NelderMeadMod_lnk_5.html



MATLAB

GNU Octave

`fminsearch` Simplex modificado (Nelder-Mead)

<u>EXEMPLO</u>: Minimizar a função $f(x_1,x_2) = x_1^4 - 2.x_1^2.x_2 + x_2^2 + x_1^2 - 2.x_1 + 5$
tendo como ponto inicial $\underline{x} = [1 \;\; 2]^T$

```
x0 = [1 2]
[x,f] = fminsearch(@(x) x(1)^4 -2*x(1)^2*x(2) + x(2)^2 + x(1)^2 -2*x(1) +5, x0)
```

```
x =
     1.0000    1.0000
f =
     4.0000
```

exemplo_fun.m

```
function f = exemplo_fun(x)
f =  x(1)^4 -2*x(1)^2*x(2) + x(2)^2 + x(1)^2 -2*x(1) +5;
```

```
x0 = [1 2];
[x,f] = fminsearch(@(x)exemplo_fun(x),x0)
f = exemplo_fun(x)
```

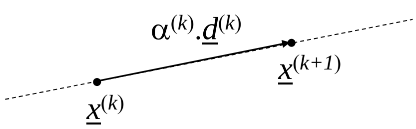
```
x =
      1.0000    1.0000

f =
      4.0000
```

Métodos baseados em gradientes (pág.40 e 47)

1) Determinar direção $\underline{d}$ que diminui $f(\underline{x})$
2) *Line search* para determinar o passo α nesta direção

   $$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha^{(k)}.\underline{d}^{(k)}$$

3) Verificar convergência



- Newton
- Steepest descent
- Marquardt
- Quasi-Newtonianos

Line Search (pág.40)

EXEMPLO: Minimizar a função $f(x_1,x_2) = x_1^4 - 2.x_1^2.x_2 + x_2^2 + x_1^2 - 2.x_1 + 5$
tendo como ponto inicial $\underline{x} = [1\ \ 2]^T$

Direção de busca: $\underline{d}^{(k)} = -\nabla f(\underline{x}^{(k)})$
(*steepest descent*)

$$\underline{d} = -\nabla f = -\begin{bmatrix} 4.x_1^3 - 4.x_1.x_2 + 2.x_1 - 2 \\ -2.x_1^2 + 2.x_2 \end{bmatrix}$$

Iteração 1:

$\underline{x}^{(0)} = [1\ \ 2]^T$ → $f(\underline{x}^{(0)}) = 5$
→ $\underline{d}^{(1)} = [4\ \ -2]^T$

$$\underline{x}^{(1)} = \underline{x}^{(0)} + \lambda.\underline{d}^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \lambda.\begin{bmatrix} 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 1+4.\lambda \\ 2-2.\lambda \end{bmatrix}$$

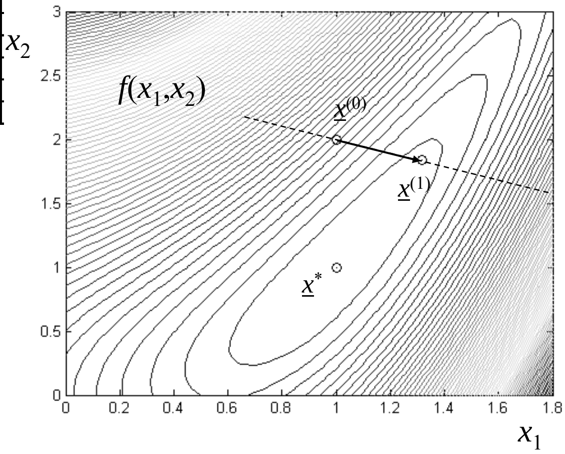
min $z(\lambda) = f(\underline{x}^{(1)}) = (1+4.\lambda)^4 - 2.(1+4.\lambda)^2.(2-2.\lambda) + (2-2.\lambda)^2 + (1+4.\lambda)^2 - 2.(1+4.\lambda) + 5$

$z(\lambda) = 256.\lambda^4 + 320.\lambda^3 + 84.\lambda^2 - 20.\lambda + 5$
$z'(\lambda) = 1024.\lambda^3 + 960.\lambda^2 + 168.\lambda - 20$
$z''(\lambda) = 3072.\lambda^2 + 1920.\lambda + 168$

Newton-Raphson partindo de $\lambda_0 = 0$

iter.	λ	f(l)
1	0,0000	5,00
2	0,1190	4,40
3	0,0842	4,12
4	0,0798	4,11
5	0,0797	4,11

$$\underline{x}^{(1)} = \begin{bmatrix} 1+4.\lambda \\ 2-2.\lambda \end{bmatrix} = \begin{bmatrix} 1{,}319 \\ 1{,}841 \end{bmatrix}$$



Algoritmos tipo-Newton (pág.48)

0. $k = 0$
1. Estimar $\underline{x}^{(k)}$, $f(\underline{x}^{(k)})$ e $\nabla f(\underline{x}^{(k)})$
2. Cálculo da direção de pesquisa $\underline{d}^{(k)}$ em $\underline{x}^{(k)}$

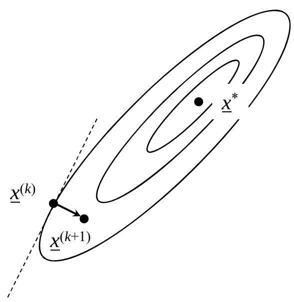
   Sistema linear de equações em $\underline{d}^{(k)}$: $\underline{\underline{B}}^{(k)}.\underline{d}^{(k)} = -\nabla f(\underline{x}^{(k)})$
3. Line Search

   Cálculo do tamanho do passo $\alpha^{(k)}$ que melhore $f(\underline{x})$ ao longo de $\underline{d}^{(k)}$
4. Obtenção no novo ponto: $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha^{(k)}.\underline{d}^{(k)}$
5. Se $| f(\underline{x}^{(k+1)}) - f(\underline{x}^{(k)}) | \leq \varepsilon_f$ e $\| \underline{x}^{(k+1)} - \underline{x}^{(k)} \| \leq \varepsilon_x$ então PARE

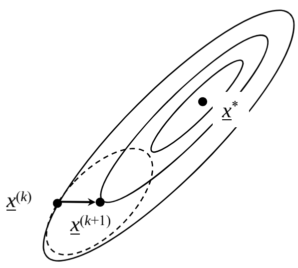
   Senão, $k = k + 1$ e vá para 1

A escolha de $\underline{\underline{B}}^{(k)}$:

$\underline{\underline{B}}^{(k)} = \underline{\underline{I}}$	Método *steepest descent*
$\underline{\underline{B}}^{(k)} = \underline{\underline{H}}(f(\underline{x}^{(k)}))$	Método de Newton
$\underline{\underline{B}}^{(k)} = \underline{\underline{H}}(f(\underline{x}^{(k)})) + \beta.\underline{\underline{I}}$	Métodos de Marquardt

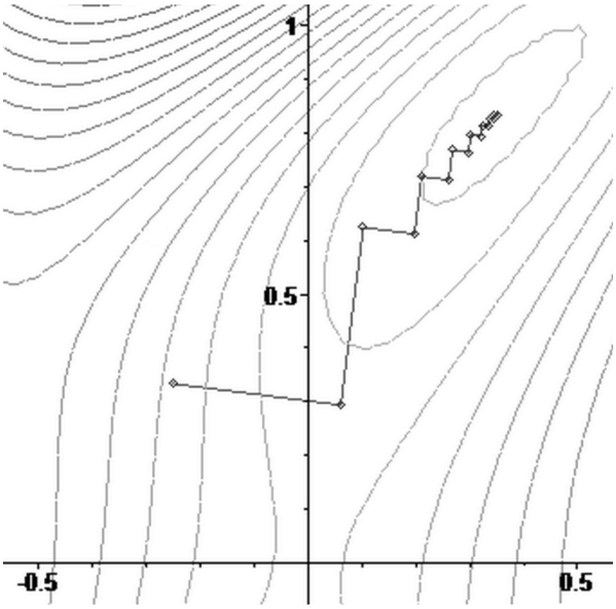


Steepest descent:
aproximação de 1ª ordem em $\underline{x}^{(k)}$



Newton:
aproximação de 2ª ordem em $\underline{x}^{(k)}$

Steepest descent



MÉTODOS QUASI-NEWTONIANOS

Fórmula DFP (Davidon-Fletcher-Powell, 1964)

$$\underline{\underline{B}}^{(k+1)} = \underline{\underline{B}}^{(k)} + \frac{\left(\underline{g} - \underline{\underline{B}}^{(k)}.\underline{d}\right)\underline{g}^T + \underline{g}.\left(\underline{g} - \underline{\underline{B}}^{(k)}.\underline{d}\right)^T}{\underline{g}^T.\underline{d}} - \frac{\left(\underline{g} - \underline{\underline{B}}^{(k)}.\underline{d}\right)^T.\underline{d}.\underline{g}.\underline{g}^T}{\left(\underline{g}^T.\underline{d}\right)\left(\underline{g}^T.\underline{d}\right)}$$

$$g = \nabla f\left(\underline{x}^{(k+1)}\right) - \nabla f\left(\underline{x}^{(k)}\right) \qquad d = \underline{x}^{(k+1)} - \underline{x}^{(k)}$$

$$\underline{\underline{H}}^{(k+1)} = \left[\underline{\underline{B}}^{(k+1)}\right]^{-1}$$

$$\underline{\underline{H}}^{(k+1)} = \underline{\underline{H}}^{(k)} + \frac{\underline{d}.\underline{d}^T}{\underline{d}^T.\underline{g}} - \frac{\underline{\underline{H}}^{(k)}.\underline{g}.\underline{g}^T.\underline{\underline{H}}^{(k)}}{\underline{g}^T.\underline{\underline{H}}^{(k)}.\underline{g}}$$

MÉTODOS QUASI-NEWTONIANOS

Fórmula BFGS (Broyden-Fletcher-Goldfarb-Shanno, 1970)

$$\underline{\underline{B}}^{(k+1)} = \underline{\underline{B}}^{(k)} + \frac{\underline{g}.\underline{g}^T}{\underline{g}^T.\underline{d}} - \frac{\underline{\underline{B}}^{(k)}.\underline{d}.\underline{d}^T.\underline{\underline{B}}^{(k)}}{\underline{d}^T.\underline{\underline{B}}^{(k)}.\underline{d}}$$

$$g = \nabla f\left(\underline{x}^{(k+1)}\right) - \nabla f\left(\underline{x}^{(k)}\right) \qquad d = \underline{x}^{(k+1)} - \underline{x}^{(k)}$$

$$\underline{\underline{H}}^{(k+1)} = \left[\underline{\underline{B}}^{(k+1)}\right]^{-1}$$

$$\underline{\underline{H}}^{(k+1)} = \underline{\underline{H}}^{(k)} + \frac{\left(\underline{d} - \underline{\underline{H}}^{(k)}.\underline{g}\right)\underline{d}^T + \underline{d}.\left(\underline{d} - \underline{\underline{H}}^{(k)}.\underline{g}\right)^T}{\underline{g}^T.\underline{d}} - \frac{\left(\underline{d} - \underline{\underline{H}}^{(k)}.\underline{g}\right)^T.\underline{g}.\underline{d}.\underline{d}^T}{\left(\underline{g}^T.\underline{d}\right)\left(\underline{g}^T.\underline{d}\right)}$$

MATLAB e OCTAVE – Otimização irrestrita

`fminsearch`	Simplex modificado (Nelder-Mead)
`fminbnd`	Line search unidimensional
`fminunc`	Quasi-Newtoniano

Matlab: Optimization Toolbox Examples

Minimization of the Banana Function (Rosenbrock function)

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2$$

função de difícil minimização

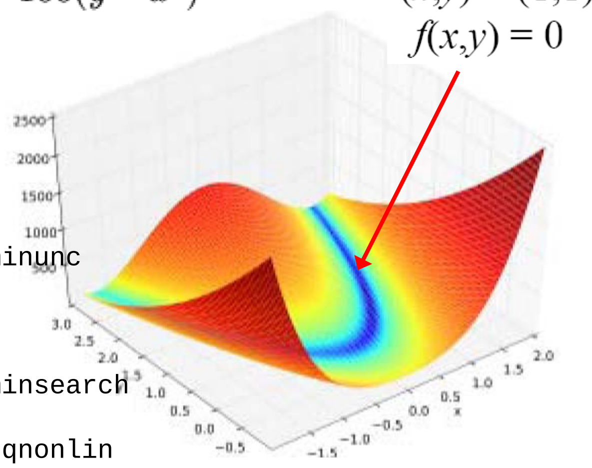
$(x,y) = (1,1)$

$f(x,y) = 0$

Quasi-Newton
Quasi-Newton
Steepest
} `fminunc`

Busca Simplex `fminsearch`

Marquardt `lsqnonlin`



Optimization Toolbox

\>> help optim → lista de funções

Help → Demos → Toolboxes → Optimization

http://www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf

```
>> help optim
 Optimization Toolbox   Version 4.2 (R2009a) 15-Jan-2009
```

Nonlinear minimization of functions.
- fminbnd - Scalar bounded nonlinear function minimization.
- fmincon - Multidimensional constrained nonlinear minimization.
- fminsearch - Multidimensional unconstrained nonlinear minimization, by Nelder-Mead.
- fminunc - Multidimensional unconstrained nonlinear minimization.
- fseminf - Multidimensional constrained minimization, semi-infinite constraints.

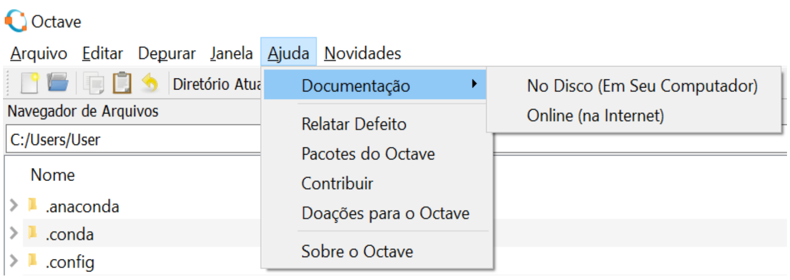
Nonlinear minimization of multi-objective functions.
- fgoalattain - Multidimensional goal attainment optimization
- fminimax - Multidimensional minimax optimization.

Linear least squares (of matrix problems).
- lsqlin - Linear least squares with linear constraints.
- lsqnonneg - Linear least squares with nonnegativity constraints.

Nonlinear least squares (of functions).
- lsqcurvefit - Nonlinear curvefitting via least squares (with bounds).
- lsqnonlin - Nonlinear least squares with upper and lower bounds.

Nonlinear zero finding (equation solving).
- fzero - Scalar nonlinear zero finding.
- fsolve - Nonlinear system of equations solve (function solve).

Minimization of matrix problems.
- bintprog - Binary integer (linear) programming.
- linprog - Linear programming.
- quadprog - Quadratic programming.



20.2 Minimizers

- fminbnd - Scalar bounded nonlinear function minimization.
- fminsearch - Multidimensional unconstrained nonlinear minimization, by Nelder-Mead.
- fminunc - Multidimensional unconstrained nonlinear minimization.

25 Optimization

- glpk - Linear Programming Kit.
- qp - Quadratic Programming.
- sqp - Multidimensional constrained nonlinear minimization, SQP.