**Fminunc**

Calcula o minimizante de uma função multidimensional onde $x$ é um vector e $f(x)$ uma função que retorna um escalar

$$min\ f(x)$$

# Fminunc

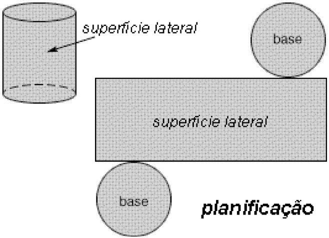
- `x = fminunc(fun,x0)`
- `x = fminunc(fun,x0,options)`
- `[x,fval]= fminunc(...)`
- `[x,fval,exitflag] = fminunc(...)`
- `[x,fval,exitflag,output] = fminunc(...)`
- `[x,fval,exitflag,output,grad] = fminunc(...)`
- `[x,fval,exitflag,output,grad,hessian] = fminunc(...)`

- x0 é a aproximação inicial
- **options** é uma estrutura específica que contém as opções (optimset para actualizar as opções)
- optimset('fminunc'): opções de fminunc
- Largescale on/off: algoritmo de Newton com regiões de confiança/ Quasi-Newton com procura unidimensional

## Exemplo 3

Um recipiente cilíndrico, aberto em cima, deve ter a capacidade de $375\,\pi cm^3$. O custo do material usado para a base é $0.15$ € por $cm^2$. O custo do material usado na parte lateral é $0.05$ € por $cm^2$. Calcule as dimensões que o recipiente deve ter para que o custo do material seja minimizado; calcule esse custo. Utilize para ponto inicial $x = 2$. Forneça as primeiras e segundas derivadas. Altere alguns parâmetros da estrutura options.

# MATLAB (fminunc)- Resolução




## Formulação

$x_1$ - raio do círculo; $x_2$ - altura do cilindro;
Área a minimizar: $\pi x_1^2 + 2\pi\, x_1\, x_2$
sujeito a $\pi\, x_1^2\, x_2 = 375\,\pi$
Eliminando uma variável fica:

$$\min \quad 0.15\,(\pi x_1^2) + 0.05\,(2\pi\,\frac{375}{x_1})$$

## M-file

```
function [f,g,h]=cilindro(x)
f=0.15*pi*x(1)^2+0.05*(2*pi*375/x(1));
if (nargout>1)
    g=0.3*pi*x(1)-0.1*pi*375/(x(1)^2);
    if (nargout>2)
        h=0.3*pi+0.2*pi*375/(x(1)^3);
    end
end
```

## Comandos

```
>> x0=[2];
>> options=optimset('Gradobj', 'on', 'Hessian', 'on');
>> [x,fval,exitflag,output]=fminunc('cilindro',x0, options)
```

```
x =
    5.0000
fval =
   35.3429
exitflag =
     1
output =
       iterations: 6
        funcCount: 7
     cgiterations: 6
    firstorderopt: 4.0681e-010
        algorithm: 'large-scale: trust-region Newton'
```

**Solução**

O cilindro deverá ter as seguintes dimensões: $x_1 = 5.0000$, $x_2 = 375/x_1^2 = 15.0000$. O custo será de $35.3429$ €

Considere a seguinte função:

$$f(x_1, x_2) = -sen(x_1)cos(x_2)$$

Calcule o seu máximo. Utilize como aproximação inicial o ponto $x_0 = (1, 1)$. Forneça as primeiras e segundas derivadas.

# Exemplo académico

## M-file

```
function [f,g,h]=teste(x)
f=sin(x(1))*cos(x(2))
if (nargout>1)
    g=[cos(x(1))*cos(x(2)), -sin(x(1))*sin(x(2))]
    if (nargout>2)
       h=[-sin(x(1))*cos(x(2)), -sin(x(2))*cos(x(1));
       -cos(x(1))*sin(x(2)), -cos(x(2))*sin(x(1))]
    end
end
```

## Comandos

```
>> x0=[1,1];
>> options=optimset('Gradobj', 'on', 'Hessian', 'on');
>> [x,fval,exitflag,output]=fminunc('teste',x0, options)
```

# Solução

```
x =
    7.8540    9.4248
fval =
    -1
exitflag =
     1
output =
       iterations: 5
        funcCount: 6
     cgiterations: 5
    firstorderopt: 1.2140e-011
        algorithm: 'large-scale: trust-region Newton'
```

# MATLAB - optimização com restrições

## Formulação geral do problema

| | | |
|---|---|---|
| $\min_{x \in R^n}$ | $f(x)$ | (função objectivo) |
| sujeito a | $c(x) \leq \ 0$ | (não lineares de desigualdade) |
| | $ceq(x) = 0$ | (não lineares de igualdade) |
| | $A\,x \leq b$ | (lineares de desigualdade) |
| | $Aeq\,x = beq$ | (lineares de igualdade) |
| | $lb \leq x \leq ub$ | (limites simples) |

Rotina usada - **fmincon** baseada em SQP.

Calcula o mínimo de uma função de várias variáveis sujeita a restrições.

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - inputs

- **x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon, OPTIONS)**
- fun e x0 - m-file com a função objectivo e aproximação inicial
- A - matriz dos coeficientes das restrições lineares de desigualdade
- b - vector independente das restrições lineares de desigualdade
- Aeq - matriz dos coeficientes das restrições lineares de igualdade
- beq - vector independente das restrições lineares de igualdade
- lb, ub - vectores dos limites inferior e superior de x
- Se alguma destas estruturas for vazia - colocar []
- nonlcon - m-file com a informação relativa às restrições não lineares (de igualdade e desigualdade)
- OPTIONS - estrutura de opções

# fmincon - outputs

- [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN]= fmincon(FUN,x0,...)
- x, FVAL, EXITFLAG, OUTPUT - explicadas na sessão anterior
- Retorna em LAMBDA os multiplicadores de Lagrange na solução x
- Retorna em GRAD e HESSIAN o valor do gradiente e Hessiana de FUN na solução x

- [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN]= fmincon(FUN,x0,...)
- x, FVAL, EXITFLAG, OUTPUT - explicadas na sessão anterior
- Retorna em LAMBDA os multiplicadores de Lagrange na solução x
- Retorna em GRAD e HESSIAN o valor do gradiente e Hessiana de FUN na solução x

# fmincon - outputs

- [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN]= fmincon(FUN,x0,...)
- x, FVAL, EXITFLAG, OUTPUT - explicadas na sessão anterior
- Retorna em LAMBDA os multiplicadores de Lagrange na solução x
- Retorna em GRAD e HESSIAN o valor do gradiente e Hessiana de FUN na solução x

# fmincon - outputs

- [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN]= fmincon(FUN,x0,...)
- x, FVAL, EXITFLAG, OUTPUT - explicadas na sessão anterior
- Retorna em LAMBDA os multiplicadores de Lagrange na solução x
- Retorna em GRAD e HESSIAN o valor do gradiente e Hessiana de FUN na solução x

# FMINCON

## Dimensões das estruturas

- x(n), f é escalar
- g(n), H(n,n)
- c(m), ceq(p), gc(n,m), gceq(n,p)

## Dimensões das estruturas

- x(n), f é escalar
- g(n), H(n,n)
- c(m), ceq(p), gc(n,m), gceq(n,p)

### Dimensões das estruturas

- x(n), f é escalar
- g(n), H(n,n)
- c(m), ceq(p), gc(n,m), gceq(n,p)

## Dimensões das estruturas

- x(n), f é escalar
- g(n), H(n,n)
- c(m), ceq(p), gc(n,m), gceq(n,p)

Função objectivo:

$$\min_{\underset{\sim}{x} \in \mathbb{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

Restrição

$$|x_1| + |x_2| \leq 1$$

Região admissível com fronteira não suave pode ser descrita pelas seguintes restrições suaves:

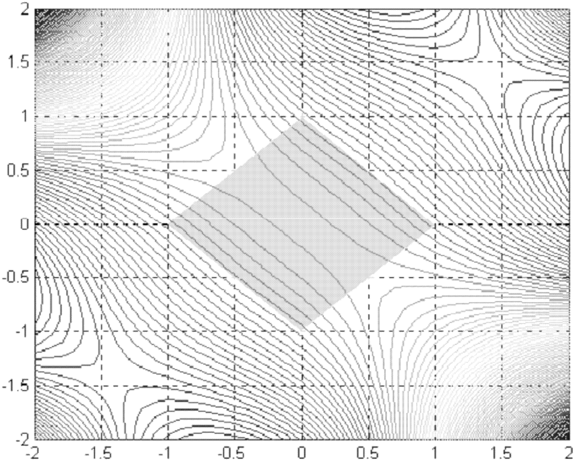
$$x_1 + x_2 \leq 1, \quad -x_1 - x_2 \leq 1, \quad -x_1 + x_2 \leq 1, \quad x_1 - x_2 \leq 1$$

# Continuação

$$\min_{\underset{\sim}{x} \in \mathbf{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

sujeito a

$$\begin{array}{rcl} x_1 + x_2 & \leq & 1 \\ -x_1 - x_2 & \leq & 1 \\ -x_1 + x_2 & \leq & 1 \\ x_1 - x_2 & \leq & 1. \end{array}$$


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

## Adicionando a restrição $(x_1 - 1)^2 + x_2^2 = 1$

$$\min_{\underset{\sim}{x} \in \mathbb{R}^2} f\left(\underset{\sim}{x}\right) = x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

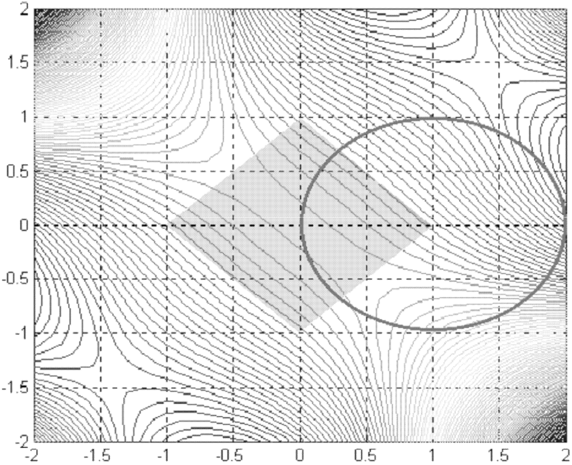
$$\begin{aligned} x_1 + x_2 &\leq 1 \\ -x_1 - x_2 &\leq 1 \\ -x_1 + x_2 &\leq 1 \\ x_1 - x_2 &\leq 1 \\ (x_1 - 1)^2 + x_2^2 &= 1 \end{aligned}$$

## Adicionando a restrição $(x_1 - 1)^2 + x_2^2 = 1$

$$\min_{\underset{\sim}{x} \in \mathbf{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

$$\begin{array}{rcl} x_1 + x_2 & \leq & 1 \\ -x_1 - x_2 & \leq & 1 \\ -x_1 + x_2 & \leq & 1 \\ x_1 - x_2 & \leq & 1 \\ (x_1 - 1)^2 + x_2^2 & = & 1 \end{array}$$


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

## Introduzindo a restrição $x_1^2 + x_2^2 \leq 0.5$

$$\min_{\underset{\sim}{x} \in \mathbb{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

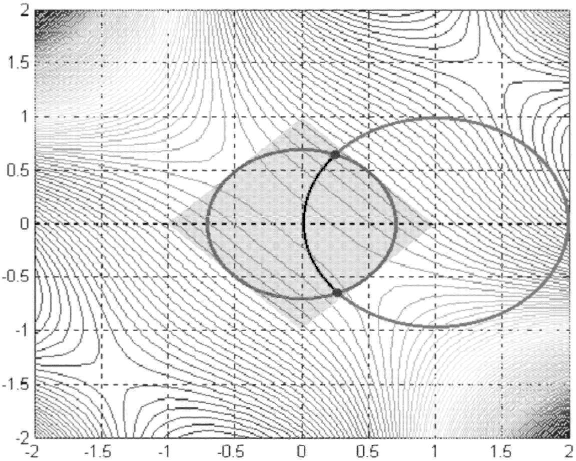
$$\begin{aligned} x_1 + x_2 &\leq 1 \\ -x_1 - x_2 &\leq 1 \\ -x_1 + x_2 &\leq 1 \\ x_1 - x_2 &\leq 1 \\ (x_1 - 1)^2 + x_2^2 &= 1 \\ x_1^2 + x_2^2 &\leq 0.5 \end{aligned}$$

## Introduzindo a restrição $x_1^2 + x_2^2 \leq 0.5$

$$\min_{\underset{\sim}{x} \in \mathbf{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

$$\begin{array}{rcl} x_1 + x_2 & \leq & 1 \\ -x_1 - x_2 & \leq & 1 \\ -x_1 + x_2 & \leq & 1 \\ x_1 - x_2 & \leq & 1 \\ (x_1 - 1)^2 + x_2^2 & = & 1 \\ x_1^2 + x_2^2 & \leq & 0.5 \end{array}$$


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

# Impondo um limite na variável $x_2 \leq -0.25$

$$\min_{\underset{\sim}{x} \in \mathbb{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

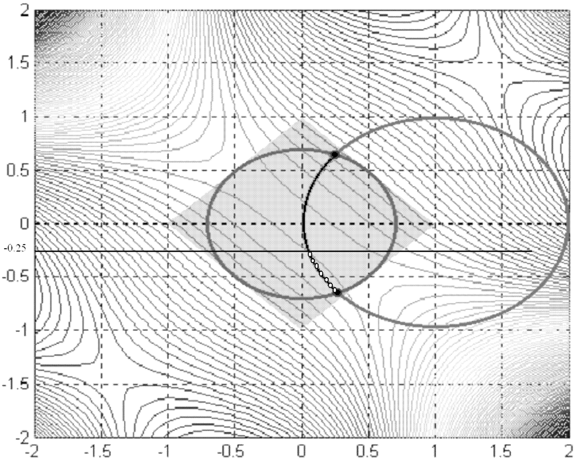
$$\begin{array}{rcl} x_1 + x_2 & \leq & 1 \\ -x_1 - x_2 & \leq & 1 \\ -x_1 + x_2 & \leq & 1 \\ x_1 - x_2 & \leq & 1 \\ (x_1 - 1)^2 + x_2^2 & = & 1 \\ x_1^2 + x_2^2 & \leq & 0.5 \\ x_2 \leq -0.25 & & \end{array}$$

## Impondo um limite na variável $x_2 \leq -0.25$

$$\min_{\underset{\sim}{x} \in \mathbf{R}^2} f\left(\underset{\sim}{x}\right) \equiv x_1^2 + x_2^2 - (1 - x_1 x_2)^2$$

s. a

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ -x_1 - x_2 &\leq 1 \\ -x_1 + x_2 &\leq 1 \\ x_1 - x_2 &\leq 1 \\ (x_1 - 1)^2 + x_2^2 &= 1 \\ x_1^2 + x_2^2 &\leq 0.5 \\ x_2 \leq -0.25 & \end{aligned}$$



2
1.5
1
0.5
0
-0.25
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

# M-files

## Função objectivo

```
function[f,g,H]=fobj(x)

f=x(1)^2+x(2)^2-(1-x(1)*x(2))^2;

if (nargout>1)

    g=[2*x(1)+2*x(2)*(1-x(1)*x(2));
    2*x(2)+2*x(1)*(1-x(1)*x(2))];

    if (nargout > 2)

        H=[2-2*x(2)^2 2-4*x(1)*x(2);
        2-4*x(2)*x(1) 2-2*x(1)^2];
    end
end
```

# M-files

## Restrições

```
function [c,ceq,gc,gceq]=frest(x)

c(1)=x(1)^2+x(2)^2-0.5;

ceq(1)=(x(1)-1)^2+x(2)^2-1;

if (nargout >2)

    gc=[2*x(1);2*x(2)];

    gceq=[2*(x(1)-1);2*x(2)];
end
```

## Janela de comandos

```
>> A=[1 1; -1 -1; -1 1; 1 -1];

>> b=[1;1;1;1];

>> lb=[-inf;-inf];

>> ub=[inf;-0.25];

>>options=optimset('TolCon',1.0e-5,'MaxIter',500,
'LargeScale','on','GradObj','on','Hessian','on',
'GradConstr','on');
```

# Solução

```
>> com x0=[0.1;0.1];

[x,fval,exitflag,output,lambda,grad,hessian]=
fmincon('fobj',x0,A,b,[],[],lb,ub,'frest',options)

x =
    0.0318
   -0.2500
fval =

   -0.9524
exitflag =
     1
output =
       iterations: 3
        funcCount: 7
         stepsize: 1
        algorithm: 'medium-scale: SQP, Quasi-Newton, lin
    firstorderopt: 5.5511e-017
```

# Soluções

```
com x0=[0.5;0.5];

[x,fval,exitflag,output,lambda,grad,hessian]=
fmincon('fobj',x0,A,b,[],[],lb,ub,'frest',options)
x =
    0.2500
   -0.6614
fval =
   -0.8581
exitflag =
     1
output =
       iterations: 5
        funcCount: 12
         stepsize: 1
        algorithm: 'medium-scale: SQP, Quasi-Newton, line-searc
    firstorderopt: 2.2204e-016
     cgiterations: []
          message: [1x143 char]
```