

---

# MATLAB - Comando fsolve

Ana Maria A. C. Rocha

Universidade do Minho, Escola de Engenharia, Departamento de Produção e Sistemas

---

O comando `fsolve` resolve um sistema de equações não lineares, na forma  $F(x) = 0$ , onde  $F$  é um vetor de funções e  $x$  é um vetor.

`[x,fval,exitflag,output] = fsolve(fun,x0,options)`

## Argumentos de entrada

- **fun** - é o sistema de funções cuja solução se pretende encontrar.

– Especificar **fun** como uma função:

```
[x,fval,exitflag,output] = fsolve(@fun,x0)
function F = fun(x)
    F(1) = ... ;
    F(2) = ... ;
end
```

– Especificar **fun** como um identificador de função anónima:

```
[x,fval,exitflag,output] = fsolve(@(x)[...;...],x0)
```

– Especificar a função **fun**:

```
fun = @(x) [...;...];
[x,fval,exitflag,output] = fsolve(fun,x0)
```

- **x0** - é uma aproximação inicial.
- **options** - (opcional) opções para controlo do processo de solução.

Options - definição de parâmetros	
Algorithm	Seleciona o algoritmo: 'trust-region-dogleg' (default), 'trust-region' e 'Levenberg-Marquardt'
Display	Nível de apresentação off - não apresenta nada iter - apresenta resultado em cada iteração iter-detailed - apresenta resultado em cada iteração e a mensagem de saída final - (default) apresenta resultado final e a mensagem de saída
FunValCheck	Verifica se os valores da função são válidos on - apresenta erro quando a função tem um valor complexo, Inf ou NaN off - (default) não apresenta erro
LargeScale	algoritmo de grande ou média escala (default = 'off')
MaxFunEvals	Número máximo de cálculos da função - (default = 100*numberofvariables)
MaxIter	Número máximo de iterações - (default = 400)
OutputFcn	Especifica uma ou mais funções que o processo de optimização pode invocar, em cada iteração.
PlotFcns	Representa graficamente o progresso do algoritmo @optimplotx - solução $x$ , ao longo das iterações @optimplotfunccount - número de cálculos da função, ao longo das iterações @optimplotfval - valor da função, ao longo das iterações @optimplotstepsize - comprimento do passo, ao longo das iterações @optimplotfirstorderopt - medida de optimalidade de 1ª ordem, ao longo das iterações
TolFun	Tolerância de paragem relativamente à função (default = 1e-6)
TolX	Tolerância de paragem relativamente a $x$ (default = 1e-6)

Para ver as opções disponíveis para o **fsolve** fazer:

```
optimset('fsolve')
```

```
optimoptions('fsolve')
```

Nestes casos, apresenta as opções disponíveis para o algoritmo, por defeito, 'trust-region-dogleg'.

### Argumentos de saída:

- **x** - é o vetor solução do problema.
- **fval** - é o valor da função na solução **x**).
- **exitflag**

exitflag - descreve valores de saída do <b>fsolve</b>	
1	sistema resolvido - a função convergiu para uma solução <i>x</i>
2	sistema resolvido - uma alteração em <i>x</i> é menor que a tolerância <b>TolX</b>
3	sistema resolvido - uma alteração no resíduo é menor que a tolerância especificada
4	sistema resolvido - a magnitude da direcção de procura é menor que a tolerância especificada
0	o número de iterações excedeu o <b>MaxIter</b> ou o <b>MaxFunEvals</b>
-1	o algoritmo terminou pela função de saída ou pela função de gráfico
-2	sistema não resolvido - ver a mensagem de saída
-3	sistema não resolvido - o Trust radius tornou-se muito pequeno

- **output**

output - estrutura que contém informação acerca do processo de procura da solução	
<b>iterations</b>	número de iterações realizadas
<b>funcCount</b>	número de cálculos da função
<b>algorithm</b>	algoritmo usado
<b>cgiterations</b>	número total de iterações PCG ('trust-region' algorithm only)
<b>stepsize</b>	comprimento do passo final (not in 'trust-region-dogleg')
<b>firstorderopt</b>	medida de optimalidade de 1ª ordem
<b>message</b>	mensagem de saída

### Algoritmos de otimização

#### • Medium-scale optimization

Por defeito ('LargeScale' a 'off'), o **fsolve** usa o algoritmo 'trust-region-dogleg' que é uma variante do método 'dogleg' de Powell.

Alternativamente, pode ser usado o método de 'Levenberg-Marquardt'.

#### • Large-scale optimization

Com o 'LargeScale' a 'on', o **fsolve** usa o algoritmo 'trust-region' que se baseia no método de Newton "interior-reflective" com a técnica de globalização baseada nas regiões de confiança. Cada iteração envolve a solução aproximada de um sistema linear de grandes dimensões utilizando o método método pre-condicionado dos gradientes conjugados (PCG).

- Os métodos 'Levenberg-Marquardt' e 'trust-region' são baseados nos algoritmos de mínimos quadrados não lineares. Usar um destes métodos se o sistema não tiver um zero.