

# INTRODUÇÃO À PROGRAMAÇÃO

Aula 19

# Funções handle

- ” Um function handle é uma forma de referenciar uma dada função
- ” Para criar um handle para qualquer função é só utilizar o sinal @ antes do nome da função

$$y = x + 2e^{-x} - 3$$

```
function y = f1(x)
y = x + 2*exp(-x) - 3;
```

```
>> fh1 = @f1
```



Referenciar a função

```
fh1 =
```

```
function_handle with value:
```

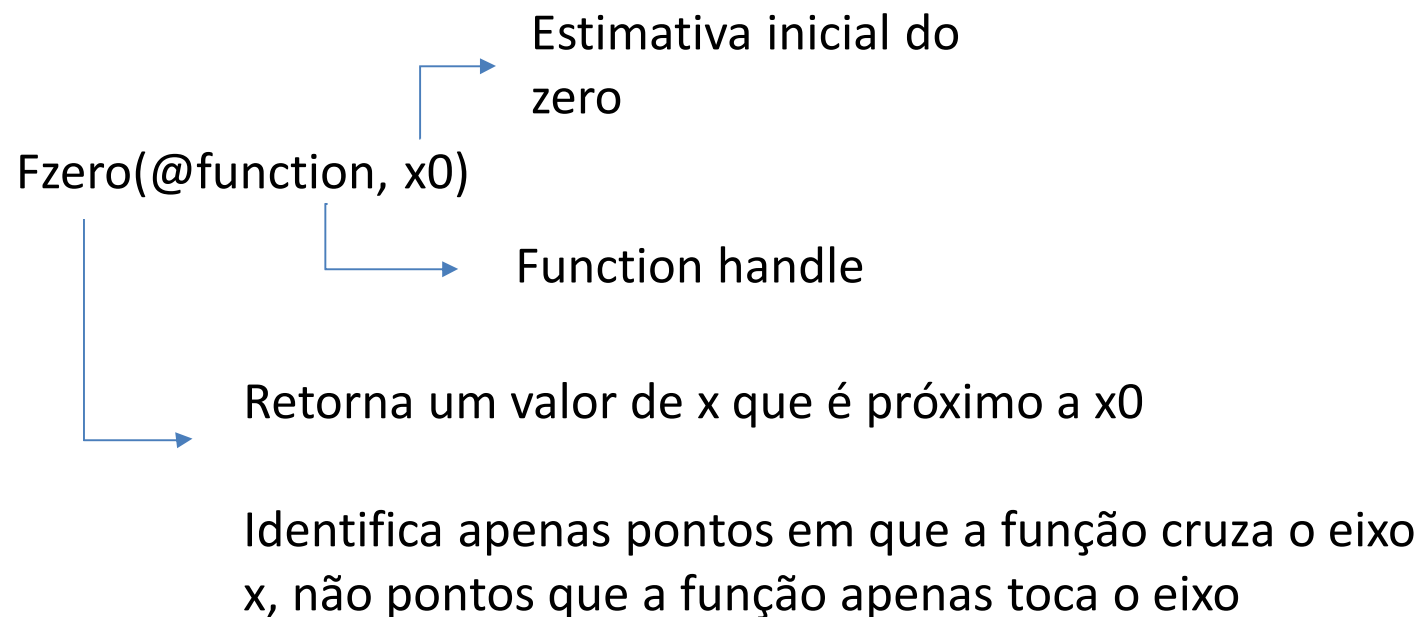
```
@f1
```

# Funções de funções

” É possível passar uma função para função principal utilizando um handle.

## Exemplo

fzero – Encontrar zeros de uma função de uma única variável



## Funções de funções

```
>> fh1 = @f1
```

```
fh1 =
```

```
function_handle with value:
```

```
@f1
```

```
>> fzero(fh1,-0.5)
```

```
ans =
```

```
-0.5831
```

```
>> clear
```

```
>> fzero(@f1,-0.5)
```

```
ans =
```

```
-0.5831
```

## Funções de funções

```
>> fh1 = @f1
```

```
fh1 =
```

```
function_handle with value:
```

```
@f1
```

```
>> fzero(fh1,-0.5)
```

```
ans =
```

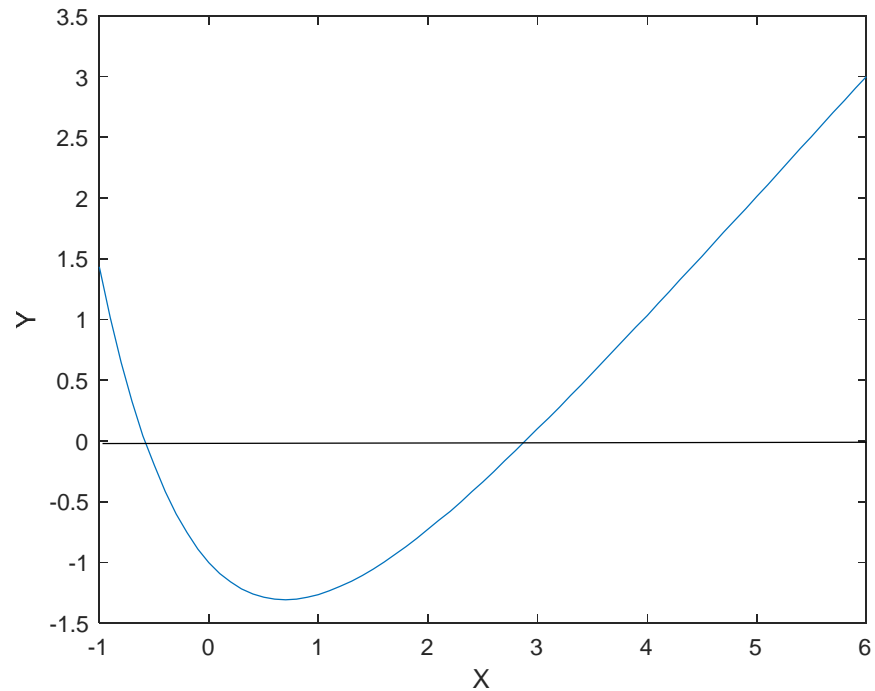
```
-0.5831
```

```
>> clear
```

```
>> fzero(@f1,-0.5)
```

```
ans =
```

```
-0.5831
```



```
>> x = -1:0.1:6;
```

```
>> y = x + 2*exp(-x) - 3;
```

```
>> plot(x,y)
```

**OBS: Plotar a função antes é uma boa maneira de se obter os valores para o vetor x0 !**

## Funções de funções

```
>> fzero(@f1, 3)
```

```
ans =
```

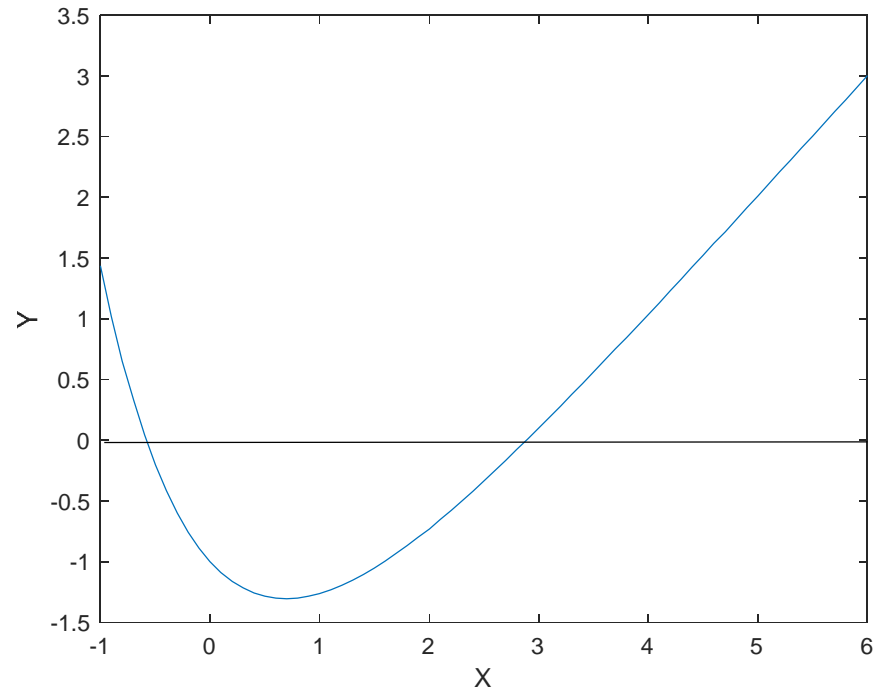
```
2.8887
```

```
>> fzero('f1',3)
```

```
ans =
```

```
2.8887
```

É recomendável o uso de 1  
opção, devido a melhor  
desempenho computacional !



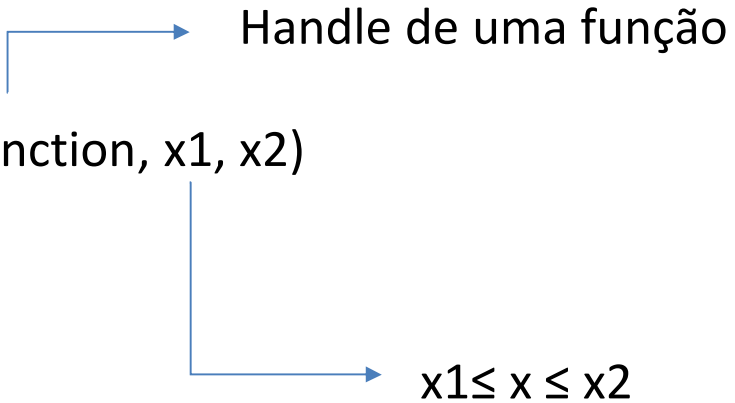
```
>> x = -1:0.1:6;
```

```
>> y = x + 2*exp(-x) - 3;
```

```
>> plot(x,y)
```

# Minimizando uma função de uma variável

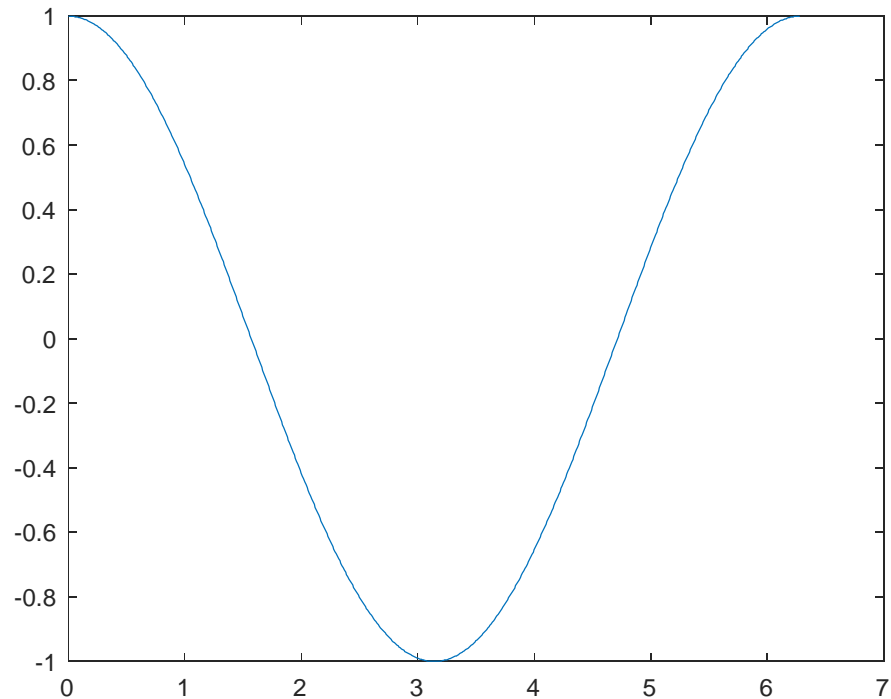
fminbnd – Encontra o mínimo de uma função de uma única variável

  
fminbnd(@function, x1, x2)  
Handle de uma função  
 $x1 \leq x \leq x2$

The diagram illustrates the arguments of the fminbnd function. A blue line originates from the '@function' argument and branches into two arrows: one pointing to the text 'Handle de uma função' and another pointing to the search interval  $x1 \leq x \leq x2$ .

## Minimizando uma função de uma variável

```
>> x = 0:0.01:2*pi;  
>> y = cos(x);  
>> plot(x,y)
```



```
>> fminbnd(@cos,0,4)
```

ans =

3.1416



## Minimizando uma função de uma variável

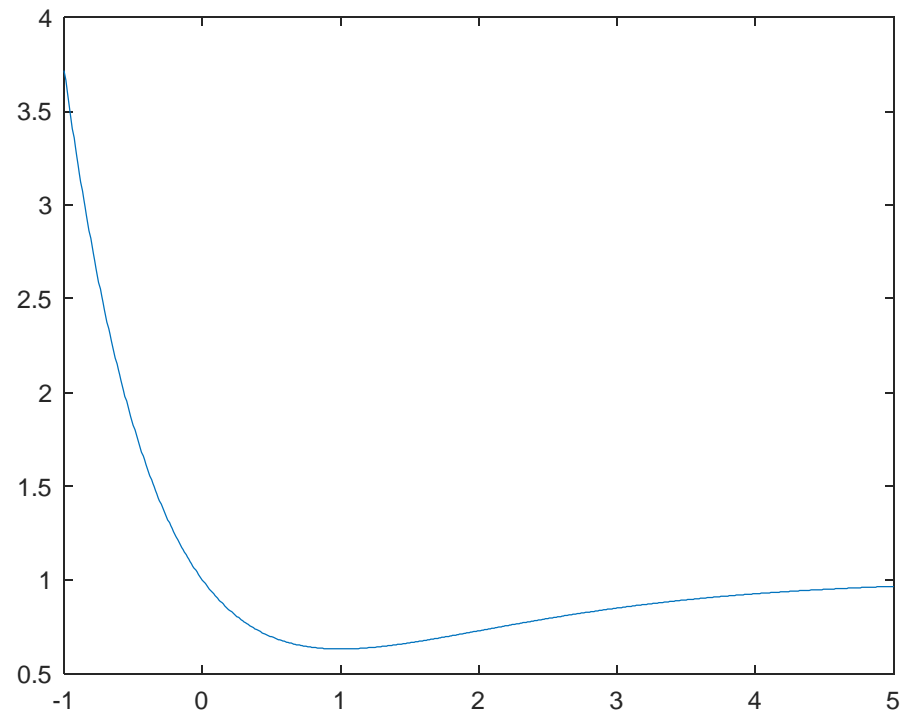
- ” Com funções mais complicadas, é recomendado criar um arquivo de função

$$y = 1 - xe^{-x}$$

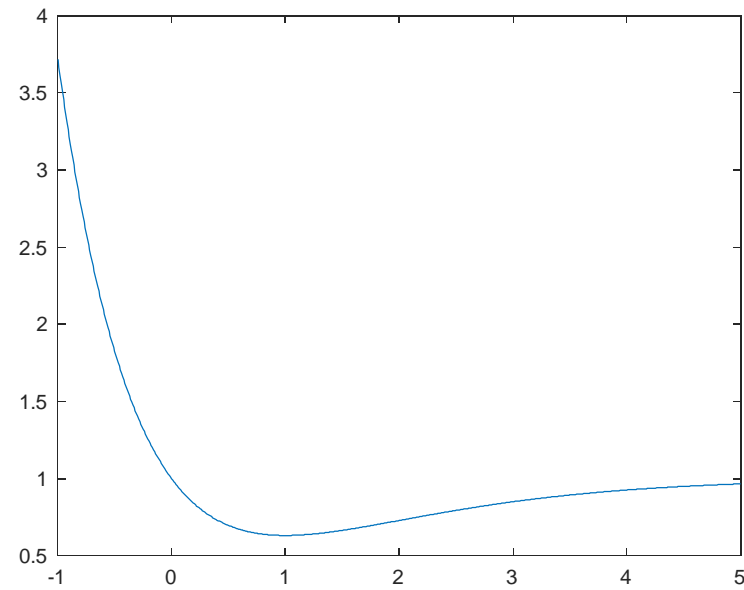
```
function y = f2(x)  
y = 1-x.*exp(-x);
```

# Minimizando uma função de uma variável

```
>> clear  
>> x = -1:0.01:5;  
>> y = 1-x.*exp(-x);  
>> plot(x,y)
```



# Minimizando uma função de uma variável



```
>> fminbnd(@f2,0,5)
```

```
ans =
```

```
1.0000
```

```
>> y = f2(1)
```

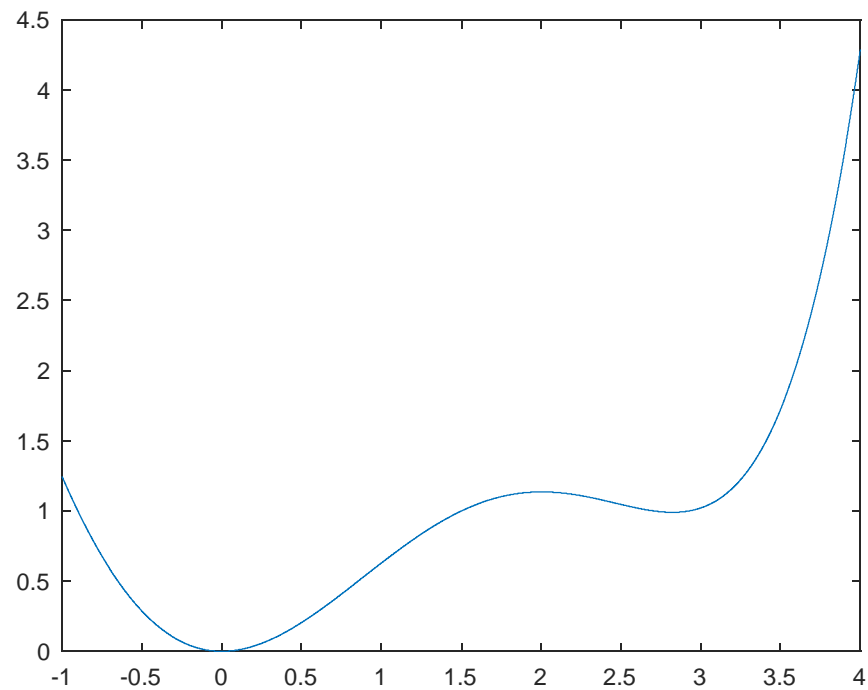
```
y =
```

```
0.6321
```

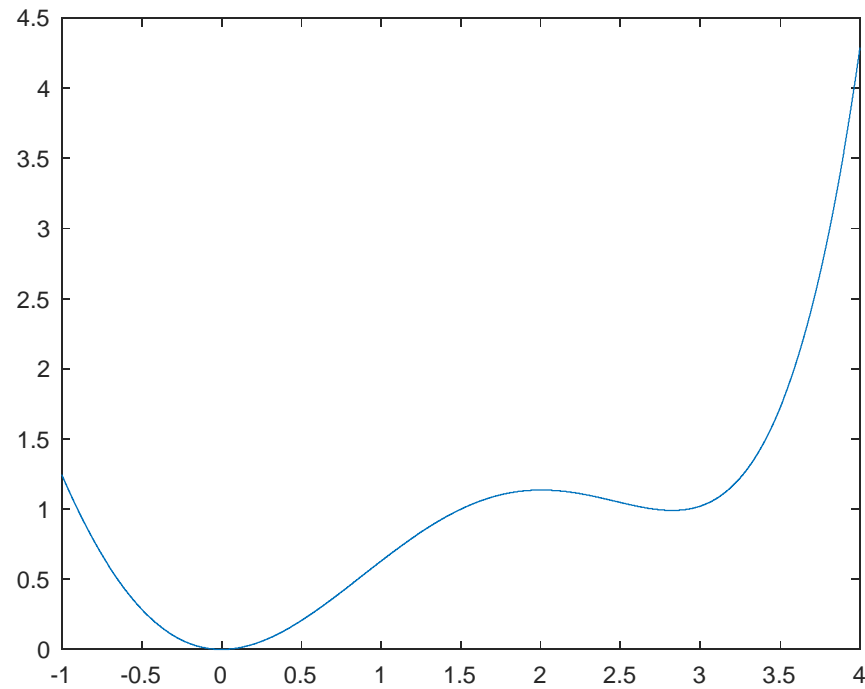
# Minimizando uma função de uma variável

$$y = 0.025x^5 - 0.0625x^4 - 0.333x^3 + x^2$$

```
>> x = -1:0.001:4;  
>> y = polyval([0.025, -0.0625, -0.333, 1, 0, 0], x);  
>> plot(x,y)
```



## Minimizando uma função de uma variável



```
>> fminbnd(@f3,2.5,4)
```

```
ans =
```

```
2.8236
```

```
>> fminbnd(@f3,-1,1)
```

```
ans =
```

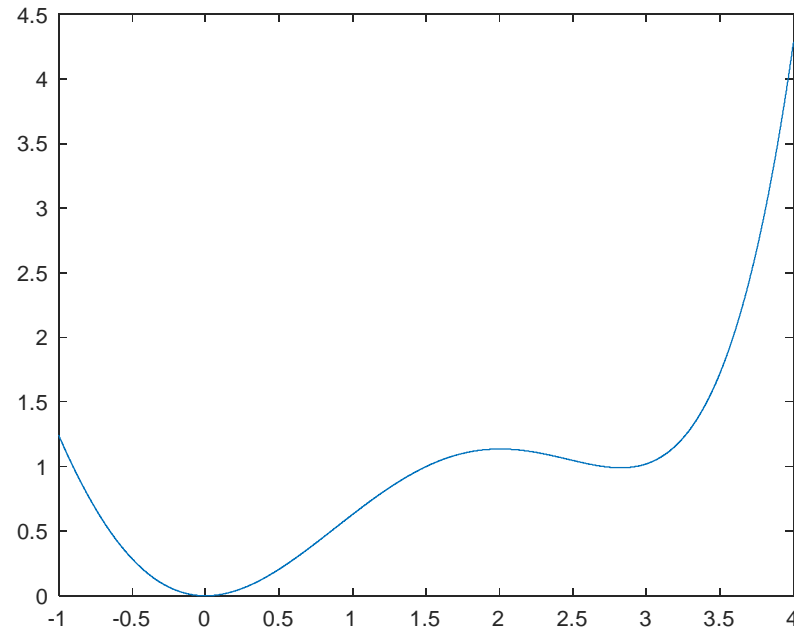
```
-1.4586e-05
```

```
>> fminbnd(@f3,-1,4)
```

```
ans =
```

```
2.0438e-06
```

# Minimizando uma função de uma variável



```
>> fminbnd(@f3,1,4)
```

```
ans =
```

```
2.8236
```

→ Neste intervalo o mínimo é igual a 1