Métodos Numéricos

Professor: Marcelo Zamith. Github: **https://github.com/marzam**

Leandro Bataglia. Github: https://github.com/eRRe-i

- Versão em notebook e PDF deste material: **github.com/eRRe-i/Metodos_Numericos_2021.1**
- Versão em HTML dos exercícios de Métodos Numéricos **aqui**

Lista de exercícios 1 - Métodos aproximativos para zero da função

Para executar esse script em seu notebook, assegure-se de estar com os pacotes **Plots**, **Calculus** e **PlutoUI**.

```
using PlutoUI
```

```
using Calculus
```

```
Plots.GRBackend()
```

```
begin
    using Plots
    gr()
end
```

Considere o $p(x) = x^5 - \frac{10}{9}x^3 + \frac{5}{21}x$, ache o zero da função conforme os casos abaixo, considerando $\epsilon = 10^{-5}$

Caso você esteja em um Pluto notebook, você pode usar as funcionalidades interativas do PlutoUI selecionando uma das opções abaixo. Caso deseje exportar para HTML, selecione a opção `GIF`. Caso queira exportar para PDF, selecione `PDF`.

atenção: caso esteja em HTML estático, não é possível interagir com o documento :(

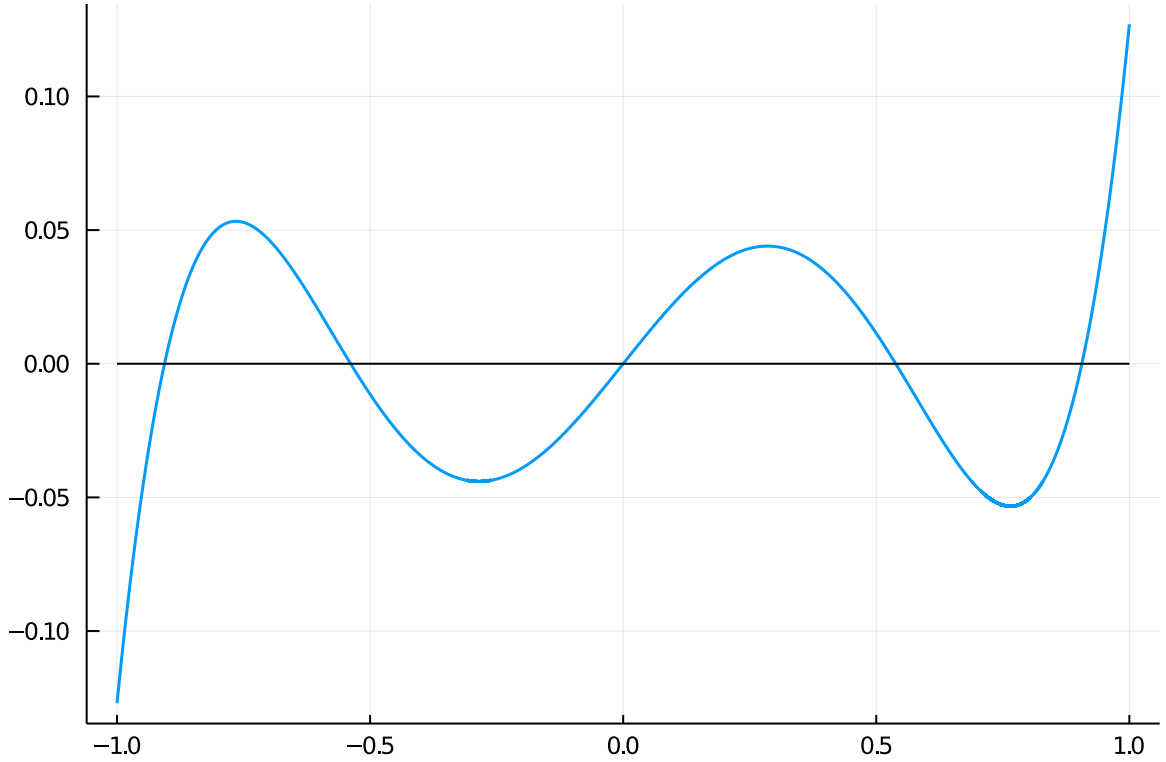
Selecione as opções abaixo:

☐ Slider | ☐ GIF | ☑ PDF

Passo 1: plotar o gráfico da função

```
f (generic function with 1 method)
```

```
f(x) = x^5-(10/9)*x^3 + (5/21)*x
```



Passo 2: fazer as aproximações de acordo com o método escolhido

1. Newton-Raphson [0.8]

```
newton_raphson (generic function with 1 method)
```

```
function newton_raphson(f::Function, x::Float64; ε=1e-5)
    x_ant = 0.0
    x_list = Vector{Float64}(undef, 0)
    push!(x_list, x)
    while(abs(x - x_ant) >= ε)
        x1 = x - f(x) / derivative(f)(x)
        push!(x_list, x1)
        x, x_ant = x1, x
    end
    x_list
end
```

```
Float64[0.8, 1.1321, 1.00839, 0.937329, 0.910227, 0.906261, 0.90618, 0.90618]
```

```
newton_raphson(f, 0.8)
```

```
line_equation (generic function with 1 method)
```

Slider Iterativo

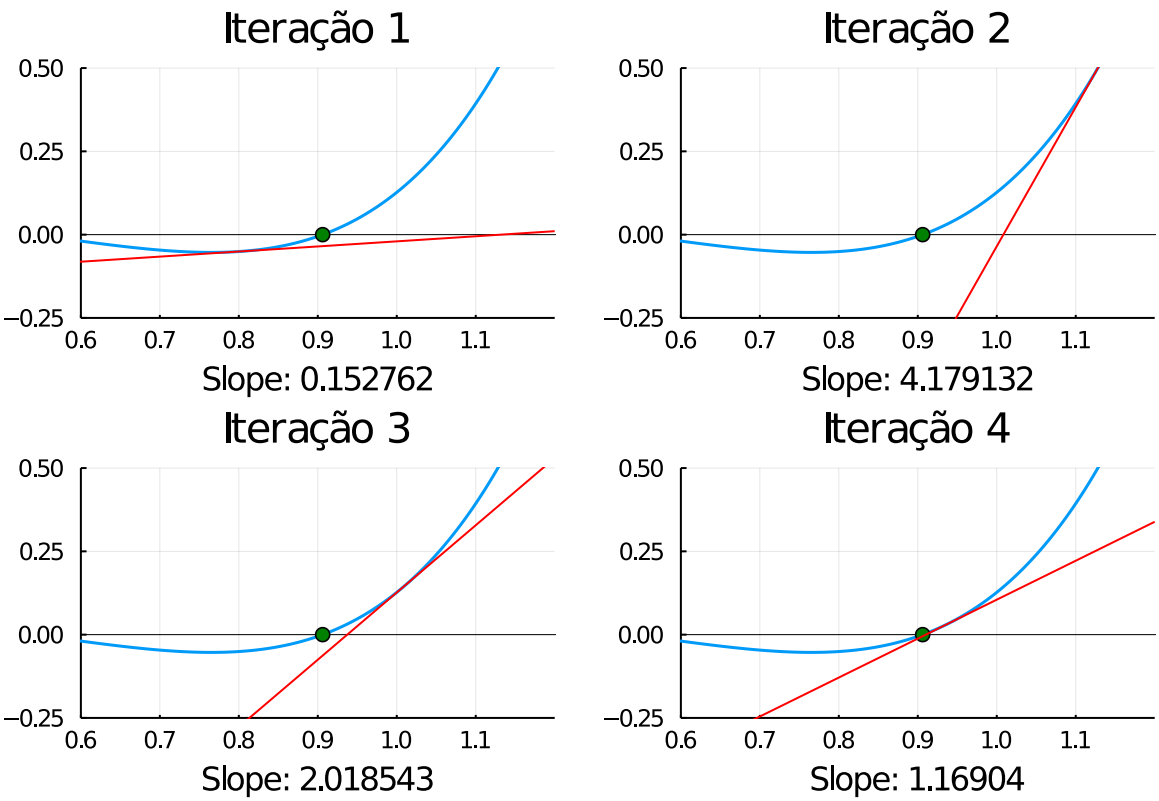
```
Essa célula ativará quando a checkbox estiver selecionada.
```

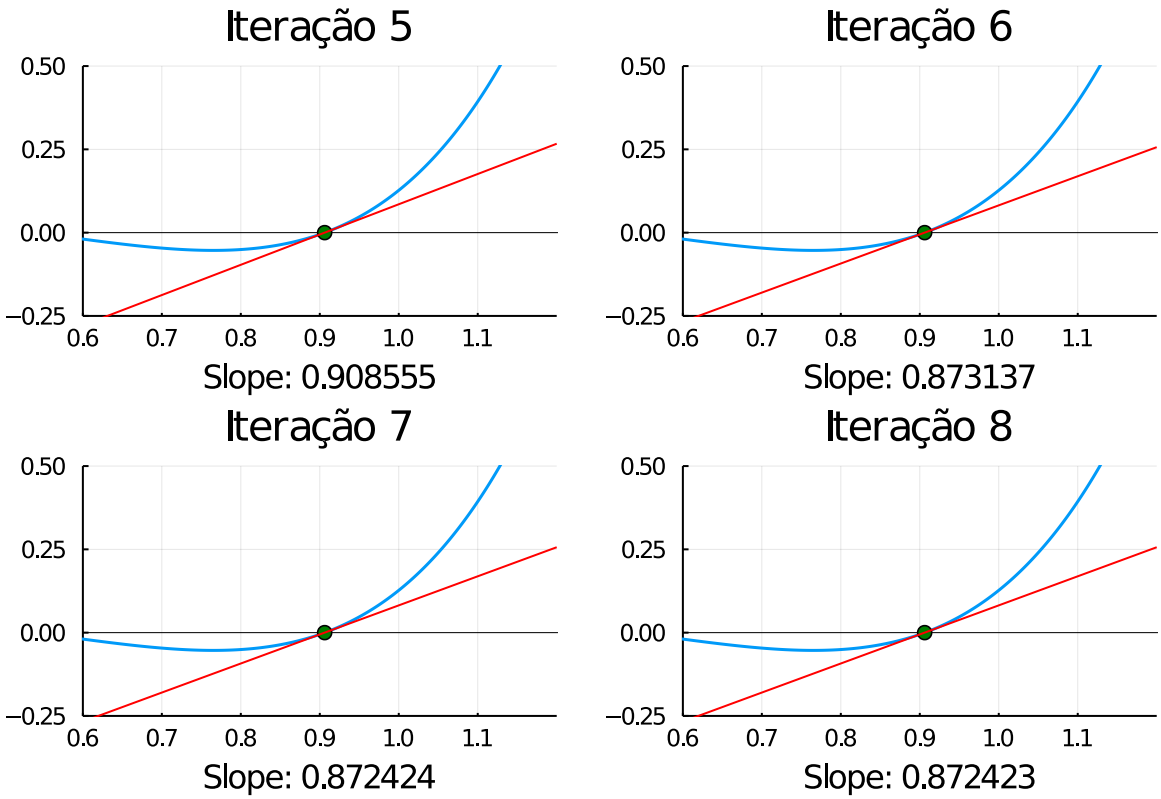
```
...
```

Animação em GIF

```
Essa célula ativará quando a checkbox estiver selecionada.
```

Imagens para PDF





2. Bicessão [-0.75, -0.25]

```
bisection (generic function with 2 methods)
```

```
function bisection(f::Function, a::Float64, b::Float64, ε=1e-5)
    range_list = Array{Array{Float64}}(undef,0)
    push!(range_list, [a, b])
    x = (a + b)/2
    while(f(x) != 0 && abs(a-b) > ε)
        if(f(x)*f(a) < 0)
            b = x
        else
            a = x
        end
        push!(range_list, [a, b])
        x = (a + b)/2
    end
    range_list
end
```

```
bisec_list =
  Array{Float64,N} where N[Float64[-0.75,  -0.25],  Float64[-0.75,  -0.5],  Float64[-0.625,
```

```
bisec_list = bisection(f, -0.75, -0.25)
```

Slider Iterativo

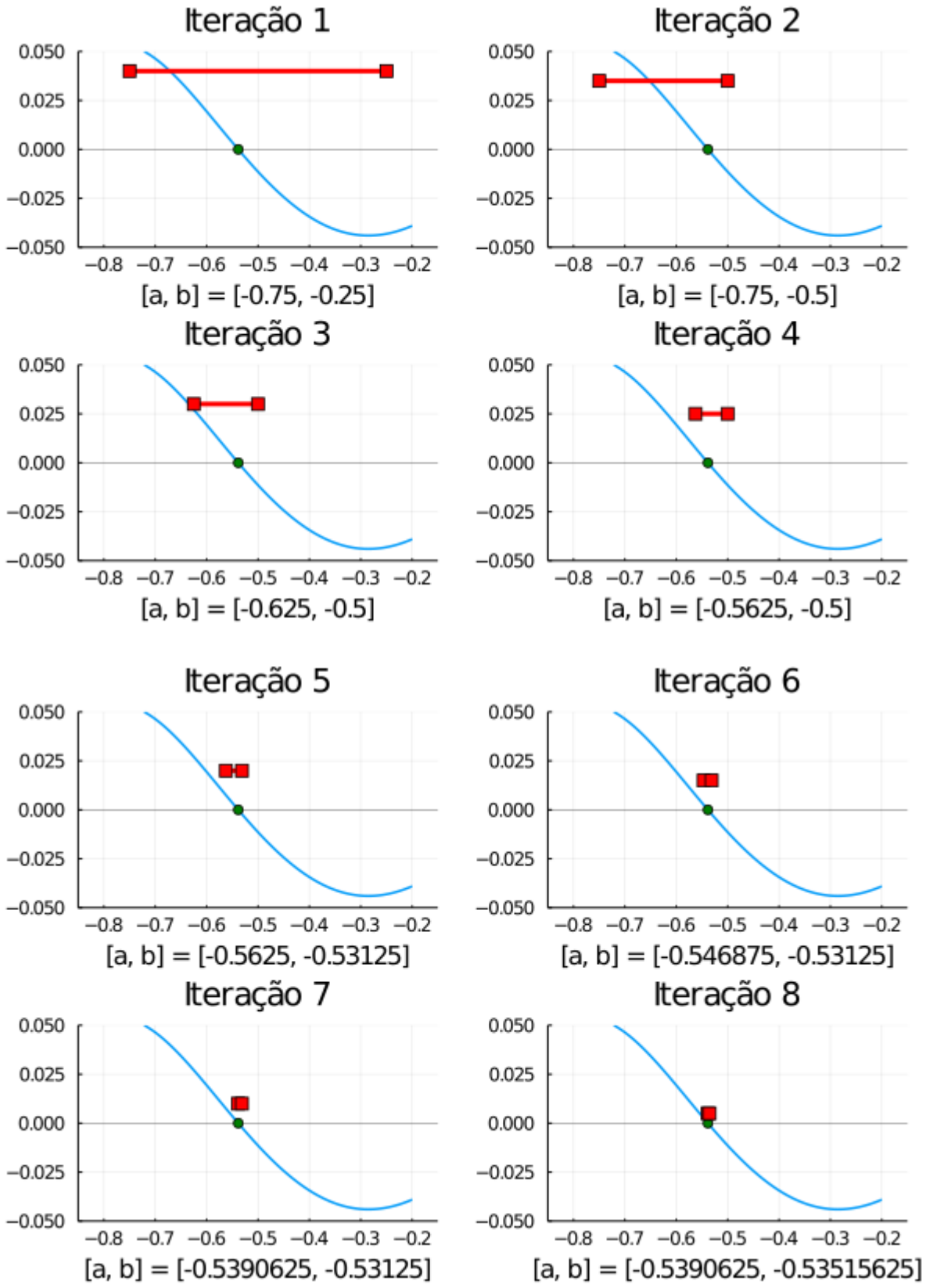
```
Essa célula ativará quando a checkbox estiver selecionada.
```

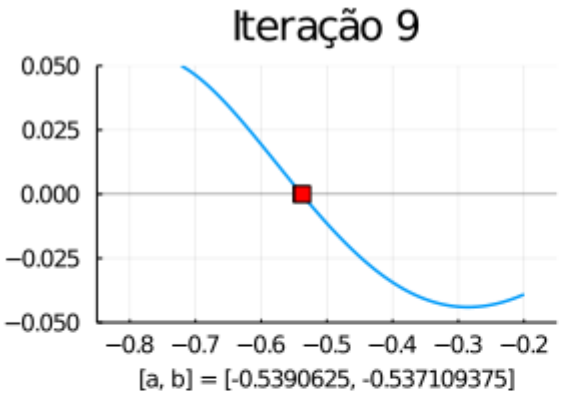
```
...
```

Animação em GIF

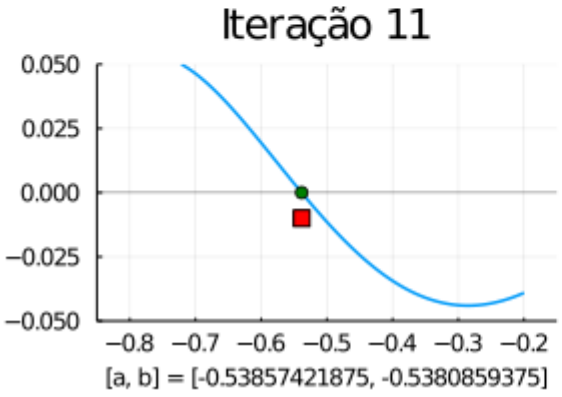
```
Essa célula ativará quando a checkbox estiver selecionada.
```

Imagens para PDF

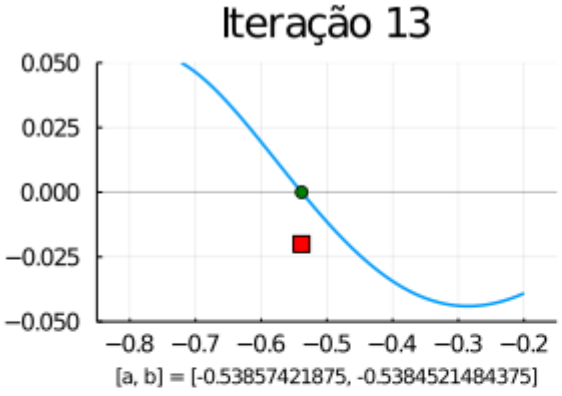



Iteração 9
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.5390625, -0.537109375]


Iteração 10
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.5390625, -0.5380859375]

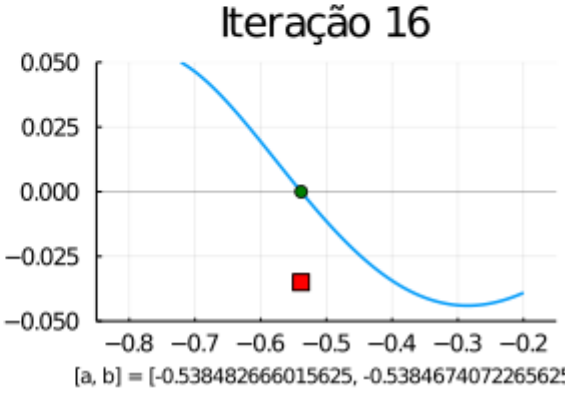

Iteração 11
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.53857421875, -0.5380859375]

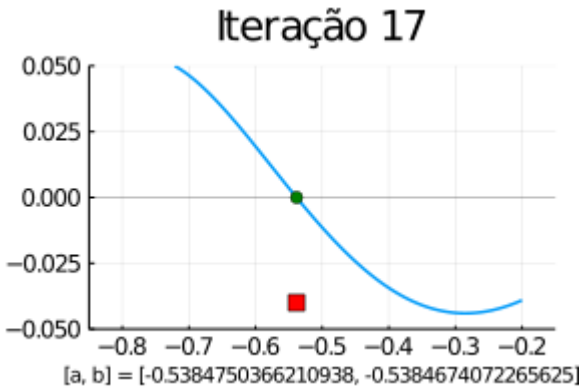

Iteração 12
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.53857421875, -0.538330078125]


Iteração 13
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.53857421875, -0.5384521484375]


Iteração 14
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.53851318359375, -0.5384521484375]


Iteração 15
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.538482666015625, -0.5384521484375]


Iteração 16
0.050
0.025
0.000
−0.025
−0.050
−0.8 −0.7 −0.6 −0.5 −0.4 −0.3 −0.2
[a, b] = [-0.538482666015625, -0.5384674072265625]



3. Falsa Posição [-0.25, 0.25]

```
false_pos (generic function with 2 methods)
```

```
function false_pos(f::Function, a::Float64, b::Float64, ε=1e-5)
    range_list = Array{Array{Float64}}(undef,0)
    push!(range_list, [a, b])
    x_ant=0.0
    x = (a * f(b) - b * f(a))/ (f(b) - f(a))
    while(f(x) != 0 &&  abs(x - x_ant)/abs(x) > ε)
        if(f(x) < 0)
            a = x
        else
            b = x
        end
        push!(range_list, [a, b])
        x, x_ant = a * f(b) - b * f(a)/ f(b) - f(a), x
    end
    range_list
end
```

```
false_pos_list =   Array{Float64,N} where N[Float64[-0.25,  0.25]]
```

```
false_pos_list = false_pos(f, -0.25, 0.25)
```

Slider Interativo

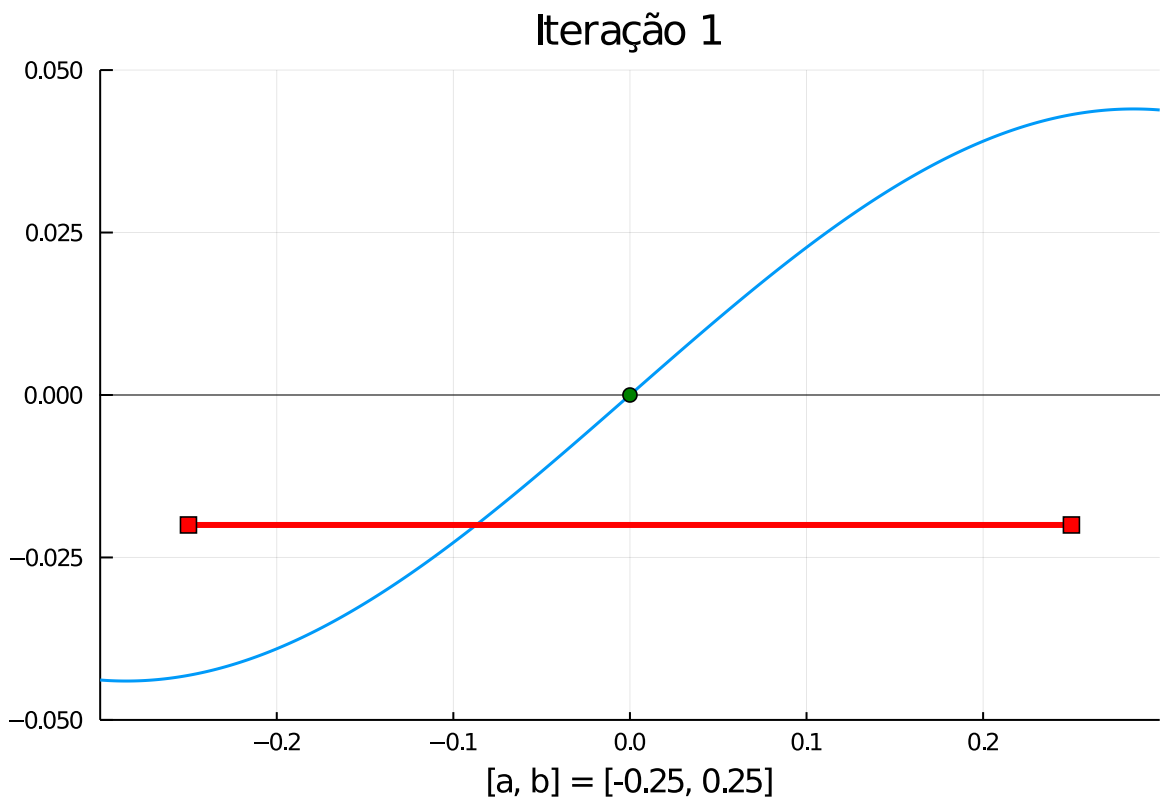
```
Essa célula ativará quando a checkbox estiver selecionada.
```

```
...
```

Animação em GIF

```
Essa célula ativará quando a checkbox estiver selecionada.
```

Imagens para PDF



4. Bicessão [0.2, 0.6]

```
bisec2 =
  Array{Float64,N} where N[Float64[0.2,  0.6],  Float64[0.4,  0.6],  Float64[0.5,  0.6],  Flo
```

Slider interativo

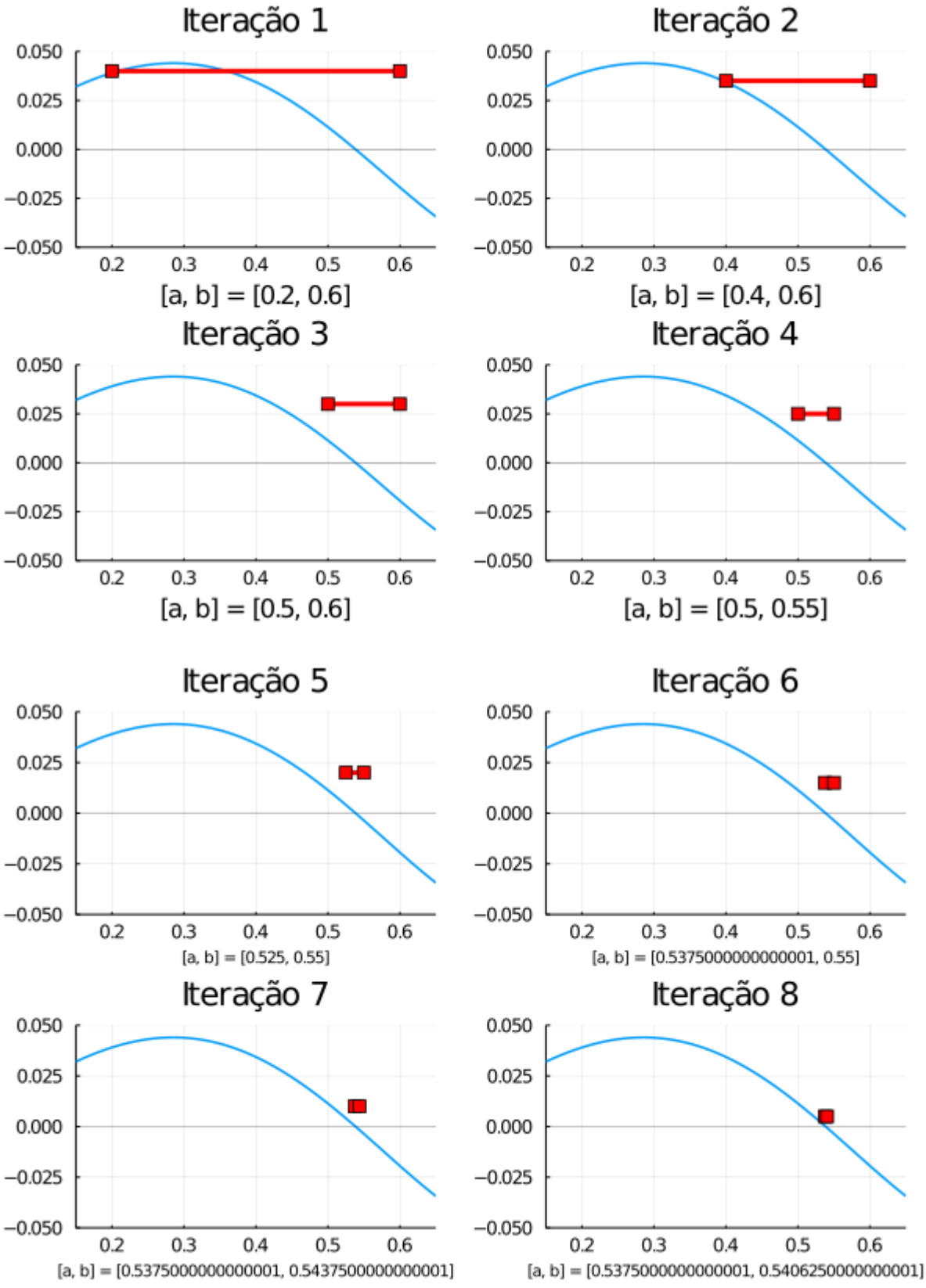
```
Essa célula ativará quando a checkbox estiver selecionada.
```

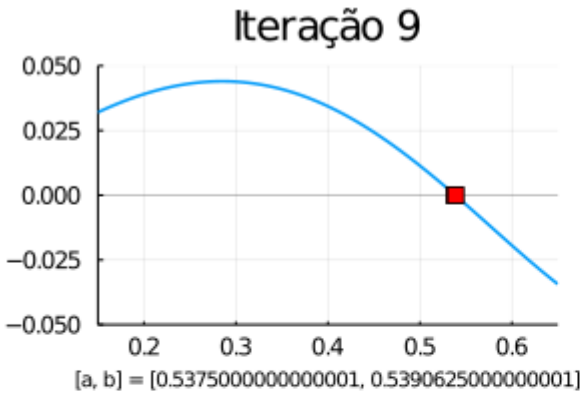
```
...
```

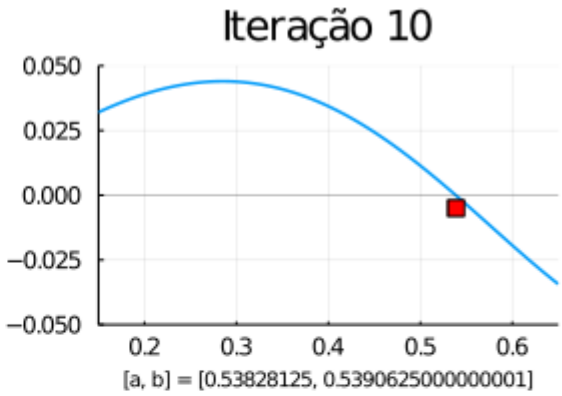
Animação em GIF

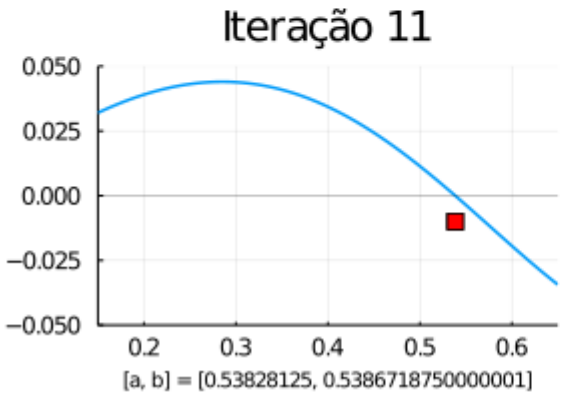
```
Essa célula ativará quando a checkbox estiver selecionada.
```

Imagens para PDF

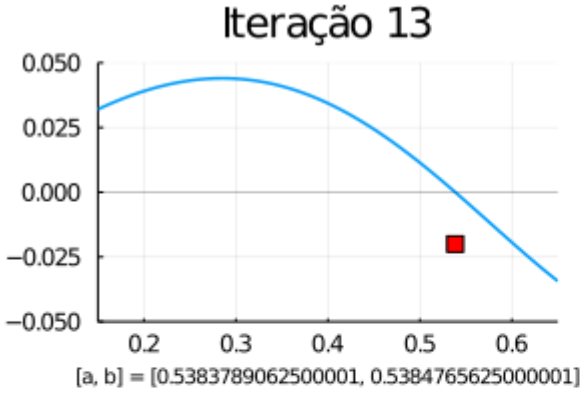

Iteração 1
[a, b] = [0.2, 0.6]
Iteração 2
[a, b] = [0.4, 0.6]
Iteração 3
[a, b] = [0.5, 0.6]
Iteração 4
[a, b] = [0.5, 0.55]
Iteração 5
[a, b] = [0.525, 0.55]
Iteração 6
[a, b] = [0.5375000000000001, 0.55]
Iteração 7
[a, b] = [0.5375000000000001, 0.5437500000000001]
Iteração 8
[a, b] = [0.5375000000000001, 0.5406250000000001]

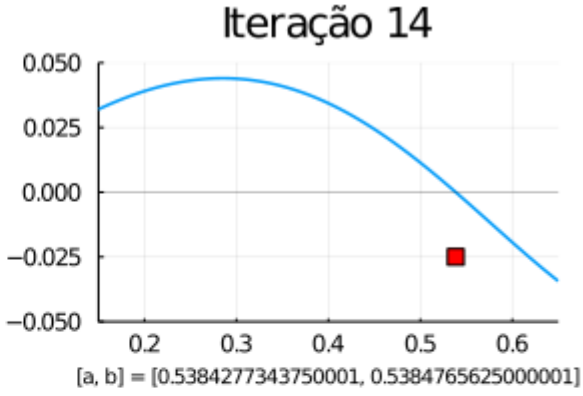

Iteração 9
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.5375000000000001, 0.5390625000000001]

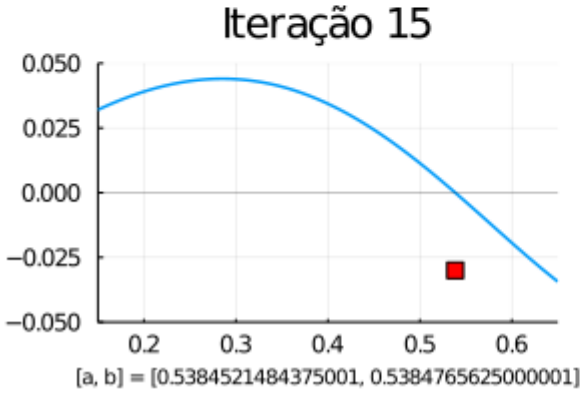

Iteração 10
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.53828125, 0.5390625000000001]


Iteração 11
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.53828125, 0.5386718750000001]


Iteração 12
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.53828125, 0.5384765625000001]


Iteração 13
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.5383789062500001, 0.5384765625000001]


Iteração 14
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.5384277343750001, 0.5384765625000001]


Iteração 15
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.5384521484375001, 0.5384765625000001]


Iteração 16
0.050
0.025
0.000
−0.025
−0.050
0.2
0.3
0.4
0.5
0.6
[a, b] = [0.53846435546875, 0.5384765625000001]

5. Secante [0.8, 1.0]

```
sec (generic function with 2 methods)
```

```
function sec(f::Function, a::Float64, b::Float64, ε=1e-5)
    x_list = Array{Array{Float64}}(undef,0)
    push!(x_list, [a , b])
    while(abs(f(a)) >= ε)
        x = b - (f(b) * (b - a))/(f(b) - f(a))
        a, b = b, x
        push!(x_list, [a, b])
    end
    x_list
end
```

```
line (generic function with 1 method)
```

```
sec_list =
  Array{Float64,N} where N[Float64[0.8,  1.0],  Float64[1.0,  0.857094],  Float64[0.857094,
```

```
sec_list = sec(f, 0.8, 1.0)
```

Slider interativo

```
Essa célula ativará quando a checkbox estiver selecionada.
```

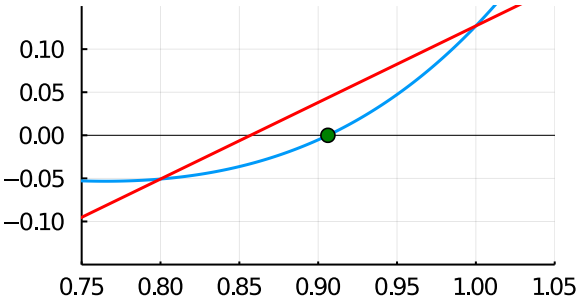
```
...
```

Animação em GIF

```
Essa célula ativará quando a checkbox estiver selecionada.
```

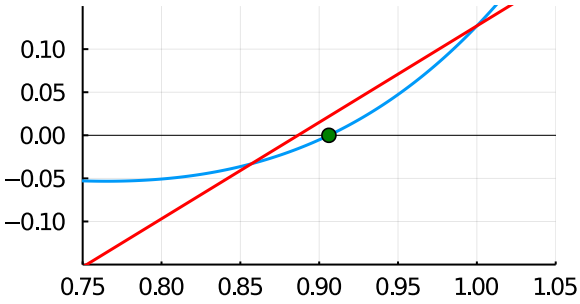
Imagens para PDF

Iteração 1



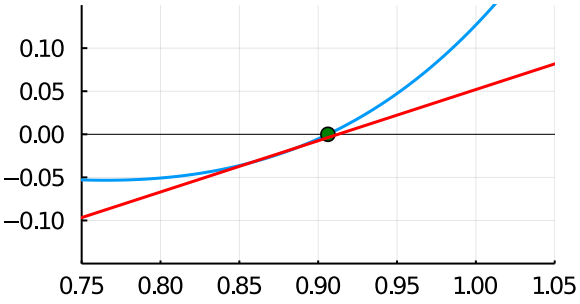
[a, b] = [0.8, 1.0]

Iteração 2



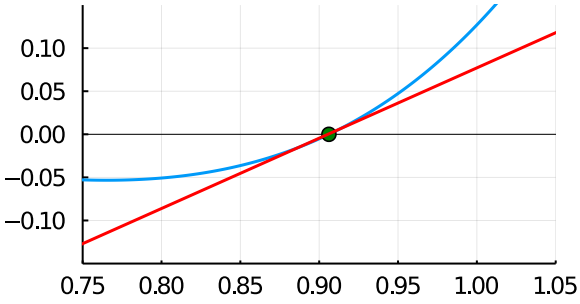
[a, b] = [1.0, 0.857093860752258]

Iteração 3



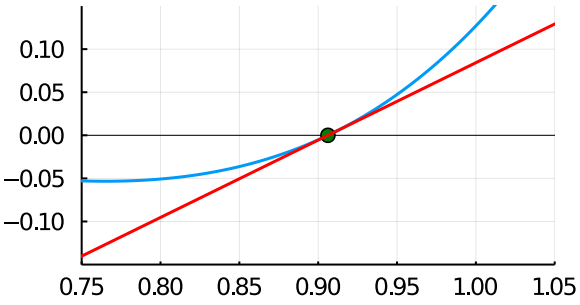
[a, b] = [0.857093860752258, 0.8865615030617445]

Iteração 4



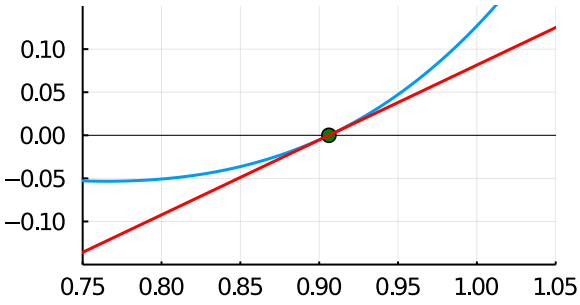
[a, b] = [0.8865615030617445, 0.9125772142039997]

Iteração 5



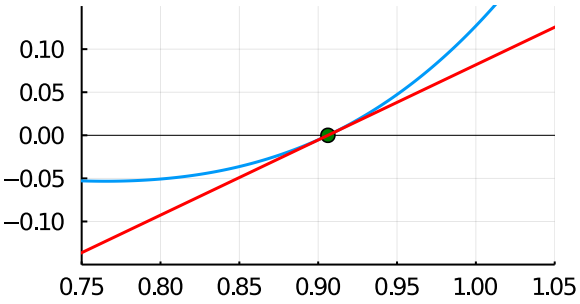
[a, b] = [0.9125772142039997, 0.9055142300797557]

Iteração 6



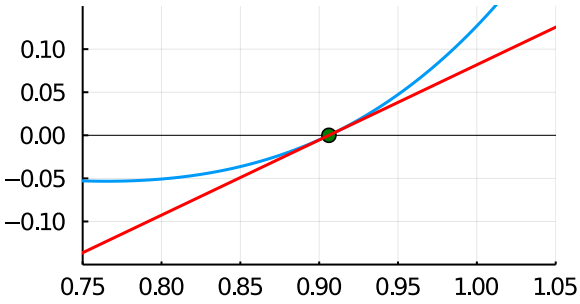
[a, b] = [0.9055142300797557, 0.9061586908755154]

Iteração 7



[a, b] = [0.9061586908755154, 0.9061799174583416]

Iteração 8



[a, b] = [0.9061799174583416, 0.906179845930997]