

1 - Determinação da Raiz de $F(x) = \sin(x^2) + 1.1 - e^{-x}$

O objetivo deste projeto é determinar raízes da função

$$F(x) = \sin(x^2) + 1.1 - e^{-x}$$

usando os métodos numéricos iterativos **Método da Bisseção** e o **Método de Newton**, após identificar intervalos candidatos contendo a raiz.

O processo inclui duas etapas principais:

1. **Detecção do intervalo contendo a raiz**: feito pela função `Raiz`, que verifica subintervalos de tamanho ($k = 0.01$) onde a função muda de sinal.
 2. **Refinamento da raiz**: com métodos iterativos (Bisseção ou Newton), usando o intervalo detectado.
-

Implementação

A implementação segue a estrutura:

- `F(x)` – função alvo
- `F_derivada(x)` – derivada de `F(x)` para o método de Newton
- `Raiz(xmin, xmax)` – encontra um intervalo onde a função muda de sinal
- `Bissecao(a, b, eps)` – método da bisseção
- `Newton(a, b, eps, x0)` – método de Newton-Raphson

O código principal solicita ao usuário os limites do intervalo e depois aplica os métodos numéricos.

Parâmetros utilizados

- `xmin, xmax` – limites do intervalo fornecido pelo usuário
- `eps` – precisão desejada para o método iterativo
- `k = 0.01` – tamanho do subintervalo na detecção de raiz
- `x0` – valor inicial para o método de Newton (opcional; se não fornecido, é usado o centro do intervalo)

Resultados Obtidos

Executando o programa:

```
== Determinação automática do intervalo I ==
Raiz encontrada aproximadamente entre -0.11 e -0.10
Intervalo I definido automaticamente: [-0.1550, -0.0550] (amplitude =
0.10)
```

Método da Bisseção

$F(a)$ e $F(b)$ têm sinais opostos, e $F'(a)$, $F'(b)$ têm o mesmo sinal – condições verificadas para a bisseção.

Resultado Bisseção: $x = -0.105348853320$ | erro $\leq 7.45e-10$ | iterações = 26

Método de Newton

$F(a)$ e $F(b)$ têm sinais opostos, e $F'(a)$, $F'(b)$ têm o mesmo sinal – condições verificadas para Newton.

Resultado Newton: $x = -0.105348853348$ | erro $\leq 1.81e-15$ | iterações = 2

Análise dos Resultados

- O **método da Bisseção** encontrou a raiz em $x \approx -0.105348853320$ com 26 iterações e erro máximo menor que (7.45×10^{-10}).
 - O **método de Newton** convergiu muito mais rapidamente, obtendo $x \approx -0.105348853348$ em apenas 2 iterações com erro menor que (1.81×10^{-15}).
 - Ambos os métodos confirmam a consistência dos resultados e a eficácia da detecção automática do intervalo inicial.
 - O método de Newton se mostrou significativamente mais eficiente quando as condições de convergência são atendidas, graças à utilização da derivada da função.
-

Conclusão

- A combinação da **detecção automática do intervalo** com métodos iterativos garantiu a determinação precisa da raiz de $F(x)$.
- O método da Bisseção é robusto, porém mais lento, enquanto o método de Newton apresenta rápida convergência quando a derivada não se aproxima de zero.

- O projeto demonstra a importância de escolher o método numérico adequado dependendo da função e do intervalo inicial.
-

2 - Determinação da Raiz de $x^3 + 4x^2 - 10 = 0$

Método Iterativo Simples

O método iterativo simples tem como objetivo encontrar uma raiz de uma função $f(x)$ através da transformação:

$$x = g(x)$$

A partir de uma aproximação inicial x_0 , define-se a sequência:

$$x_{n+1} = g(x_n)$$

A sequência converge para a raiz se existir um valor x^* tal que $x^* = g(x^*)$ e se a condição de convergência for satisfeita:

$$|g'(x^*)| < 1$$

Caso contrário, o processo diverge.

Implementação

O programa em linguagem **C** aplica o método iterativo simples para cada uma das funções $g(x)$ e mostra o comportamento da sequência de aproximações.

Parâmetros utilizados

- Valor inicial: $x_0 = 1.5$
 - Precisão: $\epsilon = 10^{-12}$
 - Máximo de iterações: 1000
-

Resultados Obtidos

g₁(x)

```
x1 = -0.8750000000
x2 = 6.7324218750
x3 = -469.7200120017
x4 = 102754555.1873851120
x5 = -1084933870531746352594944.0000000000
x6 =
1277055591444378074254579861314550183250535909418315265493330570988486656.
000000000
x7 =
-2082712908581024997457179183627809033549016221738366759709954481305758017
67295649567106614767785401768557086338743510717499427593895666237434169140
7014825950224181484247938321688218219358704615541966960319844958863360.000
0000000
x8 = -nan
Divergiu
```

Conclusão: Divergiu rapidamente. Os valores explodem numericamente até se tornarem **NaN**.

Motivo: $|g'(x)| > 1$ próximo da raiz, tornando o método instável.

g₂(x)

```
x1 = 0.8164965809
x2 = 2.9969088058
x3 = -nan
Divergiu
```

Conclusão: Divergiu.

Motivo: o argumento de $\sqrt{10/x - 4x}$ torna-se negativo, saindo do domínio da função real.

g₃(x)

```
x1 = 1.2869537676
x2 = 1.4025408035
x3 = 1.3454583740
...
x40 = 1.3652300134
x41 = 1.3652300134
Convergiu em 41 iterações
```

Conclusão: Convergiu lentamente, mas de forma estável.

Motivo: $|g'(x)|$ está próximo de 1, reduzindo a velocidade de convergência.

g₄(x)

```
x1 = 1.3483997249  
x2 = 1.3673763720  
x3 = 1.3649570154  
...  
x12 = 1.3652300134  
x13 = 1.3652300134  
x14 = 1.3652300134  
Convergiu em 14 iterações
```

Conclusão: Convergência rápida e estável.

Motivo: $|g'(x)| < 1$ no intervalo de interesse, método eficiente.

g₅(x)

```
x1 = 1.3733333333  
x2 = 1.3652620149  
x3 = 1.3652300139  
x4 = 1.3652300134  
x5 = 1.3652300134  
Convergiu em 5 iterações
```

Conclusão: Convergência extremamente rápida e precisa.

Em apenas 5 iterações, a aproximação atingiu o erro solicitado (10^{-12}).

Resultados e Análise

1. **g₁(x)** diverge porque a derivada $g_1'(x)$ é grande em módulo próximo da raiz, violando $|g'(x)| < 1$.
 2. **g₂(x)** diverge devido a problema de domínio: a raiz quadrada recebe valor negativo.
 3. **g₃(x)** converge lentamente, pois o fator $1/2$ reduz oscilações, mas $|g'(x)|$ ainda está próximo de 1.
 4. **g₄(x)** apresenta convergência rápida e estável.
 5. **g₅(x)** converge quase imediatamente, demonstrando que a forma iterativa é ideal (semelhante a Newton-Raphson).
-

Conclusão

- As funções $g_1(x)$ e $g_2(x)$ **divergem**, não sendo adequadas para o método iterativo simples.
- As funções $g_3(x)$, $g_4(x)$ e $g_5(x)$ **convergem**, com diferentes velocidades.
- A **melhor forma iterativa** é $g_5(x)$, apresentando **convergência extremamente rápida** (5 iterações) e alta precisão.

A raiz real aproximada é:

$$x^* \approx 1.3652300134$$

Portanto, a forma iterativa mais eficiente para resolver a equação é:

$$g_5(x) = (2x^3 + 4x^2 + 10) / (3x^2 + 8x)$$