Todos os arquivos estão disponíveis no seguinte repositório do GitHub:
https://github.com/AnubisNL/numerico/tree/main/Lista%20Fevereiro

Exercício 1

1) $F(10,4,-9,9)$

- $10 = 1,0 \times 10^1$ é uma representação exata neste sistema
- $-101 = -1,01 \times 10^2$ é uma representação exata neste sistema
- $0,000125 = 1,25 \times 10^{-4}$ é uma representação exata neste sistema
- $1 = 1,0 \times 10^0$ é uma representação exata neste sistema
- $24,57400 = 2,4574 \times 10^1$ no sistema ficara $2,457 \times 10^1$ não sendo uma representação exata

o erro absoluto é $|EA_x| = |2,4574 \times 10^1 - 2,457 \times 10^1|$

$|EA_x| = 0,004$

o erro relativo é $|ER_x| = \dfrac{0,004}{2,457 \times 10^1} \approx 0,00016$

- $-2220,2 = -2,2202 \times 10^3$ no sistema ficara $-2,220 \times 10^3$ não sendo uma representação exata

o erro absoluto é $|EA_x| = -2,2202 \times 10^3 - (-2,220 \times 10^3)$

$EA_x = 0,2$

o erro relativo é $|ER_x| = \dfrac{0,2}{-2,220 \times 10^3} \approx 0,00009$

- $0,0013254 = 1,3254 \times 10^{-3}$ no sistema ficara $1,325 \times 10^{-3}$ não sendo uma representação exata

$|EA_x| = 1,3254 \times 10^{-3} - 1,325 \times 10^{-3} = 4 \times 10^{-7}$

$|ER_x| = \dfrac{4 \times 10^{-7}}{1,325 \times 10^{-3}} \approx 0,0003$

FORONI

② / /

* $-1,5 \times 10^{-11}$ não possui representação no sistema

* $1,5 \times 10^{10}$ não possui representação no sistema

* $45564,9 = 4,556 \times 10^{4}$ não possui representação exata no sistema

$|EA_x| = 4,55649 \times 10^{4} - 4,556 \times 10^{4} = 4,9$
$|ER_x| = \dfrac{|EA_x|}{4,556 \times 10^{4}} \approx 1,0755 \times 10^{-4}$

* $-0,000001 = -1,0 \times 10^{-6}$ é uma representação exata no sistema

* $-0,00000000011$ não possui representação no sistema

* $998999999 = 9,990 \times 10^{8}$ não possui representação exata no sistema

$|EA_x| = 998999999 - 9,990 \times 10^{8} = 1$
$|ER_x| = \dfrac{1}{9,990 \times 10^{8}} \approx 1,0 \times 10^{-9}$

* $99969999 = 9,997 \times 10^{7}$ não possui uma representação exata no sistema

* $\pi = 3,142 \times 10^{0}$ não possui uma representação exata no sistema
$|EA_x| \approx 4,07 \times 10^{-4}$ $|ER_x| \approx 1,29 \times 10^{-4}$

Exercício 2)

2) • $x = 6{,}127 \times 10^5$ • $y = 23{,}2$ • $z = -2{,}3456 \times 10^{-2}$

$\mu = x \times z - 2 \times (x + y)$

$\mu = 6{,}127 \times 10^5 \cdot -2{,}3456 \times 10^{-2} - 2 \cdot (6{,}127 \times 10^5 + 23{,}2)$

$\mu \simeq -1239817{,}8911$

$\bar{\mu} = 6{,}13 \times 10^5 \cdot -2{,}35 \times 10^{-2} - 2 \cdot (6{,}13 \times 10^5 + 2{,}32 \times 10^1)$

$\bar{\mu} = -1240451{,}9$

$|EA_\mu| = \mu - \bar{\mu} \simeq 634{,}0088$

$|ER_\mu| = \dfrac{634{,}0088}{-1240451{,}9} \approx 5{,}11 \times 10^{-4}$

Exercício 3)

A)

4 / /

3 *

a) $A * B - C$

para uma matriz A_{2x2} e B_{2x2} seu produto se da seguinte maneira

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$A * B = \begin{bmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} & a_{11} \cdot b_{12} + a_{12} \, b_{22} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} & a_{21} \cdot b_{12} + a_{22} \, b_{22} \end{bmatrix}$$

$$A * B - C = \begin{bmatrix} (a_{11} \cdot b_{11} + a_{12} \cdot b_{21}) - c_{11} & (a_{11} b_{12} + a_{12} b_{22}) - c_{12} \\ (a_{21} \cdot b_{11} + a_{22} \cdot b_{21}) - c_{21} & (a_{21} \cdot b_{12} + a_{22} b_{22}) - c_{22} \end{bmatrix}$$

FORONI

Executar Terminal Ajuda ex3.js - Lista

ex3.html | ex3.css | Extensão: IntelliCode | ex3.js

ex3 > ex3.js > ...

```javascript
// Definindo as matrizes A, B e C como arrays de arrays
let A = [[1, 2], [3, 4]];
let B = [[5, 6], [7, 8]];
let C = [[9, 10], [11, 12]];

// Função para multiplicar duas matrizes de ordem 2
function multiplicarMatrizes(matriz1, matriz2) {
  let resultado = [[0, 0], [0, 0]];
  for (let i = 0; i < 2; i++) {
    for (let j = 0; j < 2; j++) {
      for (let k = 0; k < 2; k++) {
        resultado[i][j] += matriz1[i][k] * matriz2[k][j];
      }
    }
  }
  return resultado;
}

// Calculando a matriz resultante da multiplicação A * B
let AB = multiplicarMatrizes(A, B);

// Subtraindo a matriz C da matriz resultante A * B
let resultado = [[0, 0], [0, 0]];
for (let i = 0; i < 2; i++) {
  for (let j = 0; j < 2; j++) {
    resultado[i][j] = AB[i][j] - C[i][j];
  }
}

// Imprimindo a matriz resultado
console.log(resultado);
```

PROBLEMAS 5 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro> cd ex3
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3> node ex3.js
[ [ 10, 12 ], [ 32, 38 ] ]
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3>
```

Exercício 3)

B)

Em resumo, o algoritmo pode não ser estável para todas as matrizes de entrada e pode gerar erros significativos devido à limitação de precisão e intervalo da máquina F(10, 3, -9, 9).

5

3) b

colocando as matrizes no ponto flutuante F(10, 3, -9, 9)

$$A = \begin{pmatrix} 1{,}00 \times 10^{0} & 0{,}0 \times 10^{0} \\ 1{,}99 \times 10^{0} & 4{,}0 \times 10^{0} \end{pmatrix}$$

$$B = \begin{pmatrix} 4{,}00 \times 10^{0} & 0 \\ 1{,}0 \times 10^{0} & -1{,}33 \times 10^{0} \end{pmatrix}$$

$$C = \begin{pmatrix} 4{,}0 \times 10^{0} & 2{,}0 \times 10^{0} \\ -8{,}0 \times 10^{0} & 1{,}0 \times 10^{-1} \end{pmatrix}$$

$$\text{resultado} = \begin{pmatrix} 0 & -2 \\ 15{,}96 & -5{,}42 \end{pmatrix}$$ O calculo foi feito pelo codigo

calculando erro temos

$$EA \approx \begin{pmatrix} 0{,}0024 & 0{,}0 \\ 0{,}041 & 0{,}013 \end{pmatrix}$$

Executar Terminal Ajuda ex3.js - Lista Fevereir

ex3.html | # ex3.css | JS ex3.js × | ~$sta fevereiro_natan.docx

ex3 > JS ex3.js > ...

```javascript
// Definindo as matrizes A, B e C como arrays de arrays
let A = [[1.0001, 0], [1.9995, 4]];
let B = [[4.002, 0], [0, -1.3333]];
let C = [[4, 2], [-8, 0.1]];

// Função para multiplicar duas matrizes de ordem 2
function multiplicarMatrizes(matriz1, matriz2) {
  let resultado = [[0, 0], [0, 0]];
  for (let i = 0; i < 2; i++) {
    for (let j = 0; j < 2; j++) {
      for (let k = 0; k < 2; k++) {
        resultado[i][j] += matriz1[i][k] * matriz2[k][j];
      }
    }
  }
  return resultado;
}

// Calculando a matriz resultante da multiplicação A * B
let AB = multiplicarMatrizes(A, B);

// Subtraindo a matriz C da matriz resultante A * B
let resultado = [[0, 0], [0, 0]];
for (let i = 0; i < 2; i++) {
  for (let j = 0; j < 2; j++) {
    resultado[i][j] = AB[i][j] - C[i][j];
  }
}

// Imprimindo a matriz resultado
console.log(resultado);
```

PROBLEMAS 5 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro> cd ex3
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3> node ex3.js
[ [ 10, 12 ], [ 32, 38 ] ]
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3> node ex3.js
[
  [ 0.0024001999999994084, -2 ],
  [ 16.001998999999998, -5.4331999999999999 ]
]
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3>
```

Executar Terminal Ajuda ex3b.js - ex3 - Visual Studio

ex3b.js U ×

ex3b.js > multiplicarMatrizes

```javascript
// Definindo as matrizes A, B e C como arrays de arrays
let A = [[1.00, 0], [1.99, 4]];
let B = [[4.00, 0], [0, -1.33]];
let C = [[4, 2], [-8, 0.1]];

// Função para multiplicar duas matrizes de ordem 2
function multiplicarMatrizes(matriz1, matriz2) {
  let resultado = [[0, 0], [0, 0]];
  for (let i = 0; i < 2; i++) {
    for (let j = 0; j < 2; j++) {
      for (let k = 0; k < 2; k++) {
        resultado[i][j] += matriz1[i][k] * matriz2[k][j];
      }
    }
  }
  return resultado;
}

// Calculando a matriz resultante da multiplicação A * B
let AB = multiplicarMatrizes(A, B);

// Subtraindo a matriz C da matriz resultante A * B
let resultado = [[0, 0], [0, 0]];
for (let i = 0; i < 2; i++) {
  for (let j = 0; j < 2; j++) {
    resultado[i][j] = AB[i][j] - C[i][j];
  }
}

// Imprimindo a matriz resultado
console.log(resultado);
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3> node ex3b.js
[ [ 0, -2 ], [ 15.96, -5.42 ] ]
PS C:\emergencia\estudos\2023\numérico\numerico\Lista Fevereiro\ex3>
```

Não podemos dizer que o algoritmo sempre será estável ao variar os coeficientes das matrizes. O problema é que alguns valores dos coeficientes podem levar a uma amplificação de erros de arredondamento que levarão a uma solução numérica imprecisa. Isso pode ocorrer, por exemplo, quando os coeficientes das matrizes são muito grandes ou muito pequenos em relação ao tamanho da matriz. Nesses casos, os erros de arredondamento que normalmente seriam insignificantes podem se tornar duradouros.

Um contra-exemplo numérico seria se tomarmos as matrizes A, B e C como:

A = [[1e15, 0], [0, 1]]

B = [[1, 0], [0, 1e-15]]

C = [[1, 0], [0, 1]]

Nesse caso, o algoritmo pode levar a um resultado impreciso devido à amplificação de erros de arredondamento. O produto A*B é igual a:

A*B = [[1e15, 0], [0, 1e-15]]

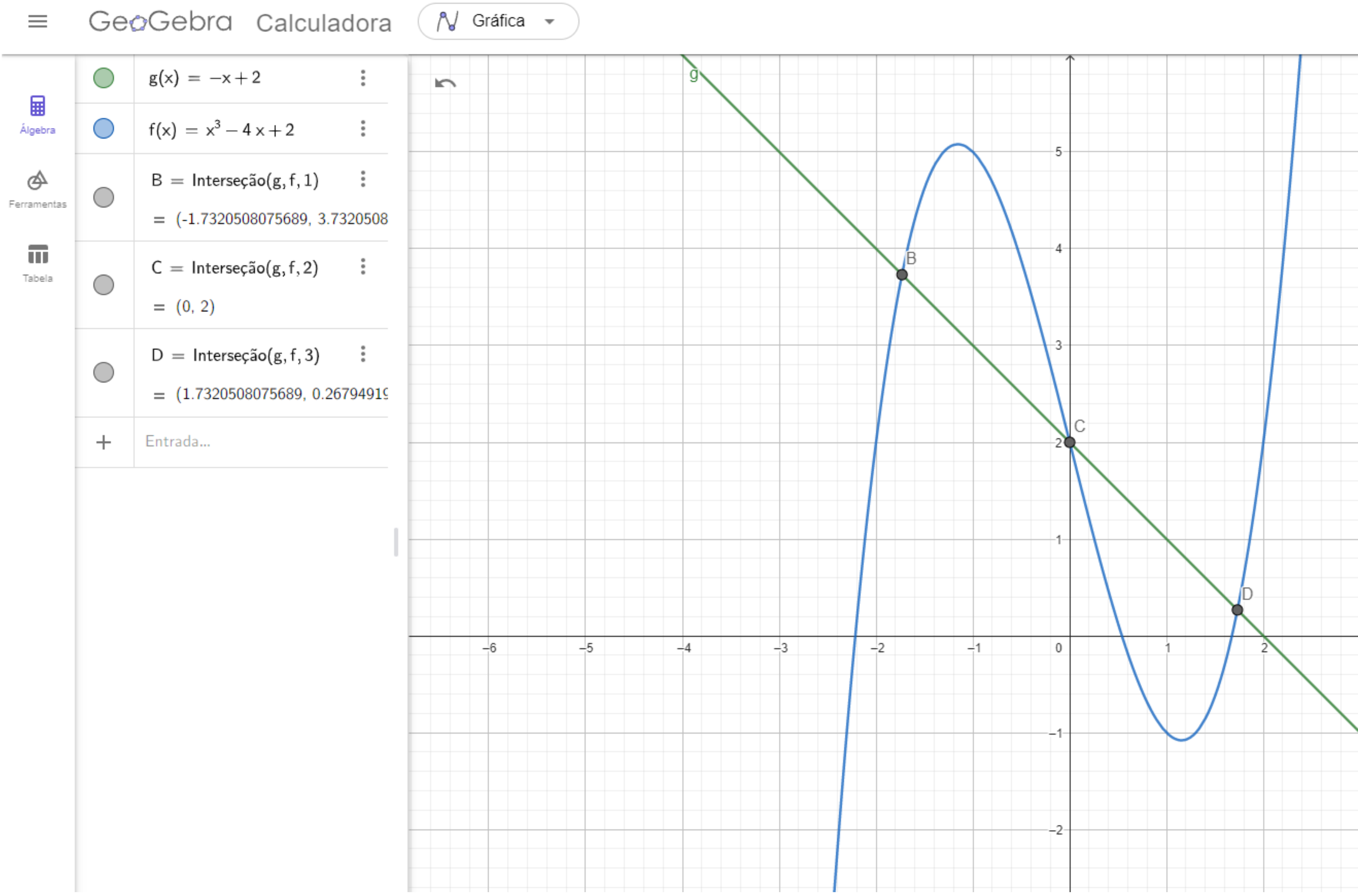
No entanto, a subtração A*B - C é igual a:

A*B - C = [[1e15 - 1, 0], [0, 1e-15 - 1]]

Observe que o elemento (1,1) da matriz resultante é muito menor do que o elemento correspondente na matriz A, mas ainda é grande em relação ao elemento correspondente na matriz C. Isso significa que o erro de arredondamento nesse elemento foi amplificado, gerado em uma solução imprecisa.

Exercício 4)

a)



b)

Algoritmo da bisseção:

1. Verifique se h(a) * h(b) < 0. Se não, informe que o intervalo não é adequado e pare o algoritmo.
2. Enquanto (b - a) / 2 > ϵ: a. Calcule o ponto médio c = (a + b) / 2. b. Se h(c) = 0 ou (b - a) / 2 < ϵ, pare o algoritmo. c. Se h(a) * h(c) < 0, atualize b = c. d. Senão, atualize a = c.
3. Retorne o valor de c como o zero da função h(x).

Código em Javascript:

```javascript
function bissecao(h, a, b, eps) {
    if (h(a) * h(b) >= 0) {
        console.log("Intervalo inadequado.");
        return null;
    }

    while ((b - a) / 2 > eps) {
        let c = (a + b) / 2;
        if (h(c) === 0 || (b - a) / 2 < eps) {
            return c;
        } else if (h(a) * h(c) < 0) {
            b = c;
        } else {
            a = c;
        }
    }

    return (a + b) / 2;
}

// Exemplo de uso
function h(x) {
    return x ** 3 - 3 * x;
}

console.log(bissecao(h, -1, 1, 0.0001));  // Saída esperada: 0.0
```

Algoritmo da falsa posição:

1. Verifique se h(a) * h(b) < 0. Se não, informe que o intervalo não é adequado e pare o algoritmo.
2. Enquanto |f(c)| > eps: a. Calcule o ponto c = (a * h(b) - b * h(a)) / (h(b) - h(a)). b. Se h(a) * h(c) < 0, atualize b = c. c. Senão, atualize a = c.
3. Retorne o valor de c como o zero da função h(x).

Código em Javascript:

```
function falsaPosicao(h, a, b, eps) {
    if (h(a) * h(b) >= 0) {
        console.log("Intervalo inadequado.");
        return null;
    }

    let c = a;
    while (Math.abs(h(c)) > eps) {
        c = (a * h(b) - b * h(a)) / (h(b) - h(a));
        if (h(a) * h(c) < 0) {
            b = c;
        } else {
            a = c;
        }
    }

    return c;
}

// Exemplo de uso
function h(x) {
    return x ** 3 - 3 * x;
}

console.log(falsaPosicao(h, -1, 1, 0.0001));  // Saída esperada: 0.0
```

c)

A fim de ler e executar o programa com as variáveis solidas pelo usuário, browser para rodar nosso script, com isso basta abrir o arquivo .HTML

- [−3,−1] -> o resultado usando o método de Bisseção é = -1.73199462890625

o resultado usando o método de Falsa Posição é = -1.7320365807780067

- [−1, 1] -> o resultado usando o método de Bisseção é = 0
  o resultado usando o método de Falsa Posição é = 0
- [1, 3] -> o resultado usando o método de Bisseção é = 1.73199462890625

o resultado usando o método de Falsa Posição é = 1.7320365807780067

- [− 1, 3]-> intervalo inadequado

d)

Não os algoritmos não dão os mesmos resultados.

Em geral, a escolha entre os dois métodos depende da natureza da função que está sendo avaliada. Se a função é simples e bem comportada, a Bisseção pode ser a melhor opção. Se a função é mais complexa ou não é monotônica, a Falsa Posição pode ser mais eficiente.

Exercício 5)

a)

Para verificar analiticamente que a função h(x) = sin(x) - x + 1 possui apenas um zero real, podemos analisar suas propriedades matemáticas.

Começando pela sua derivada, temos:

h'(x) = cos(x) - 1

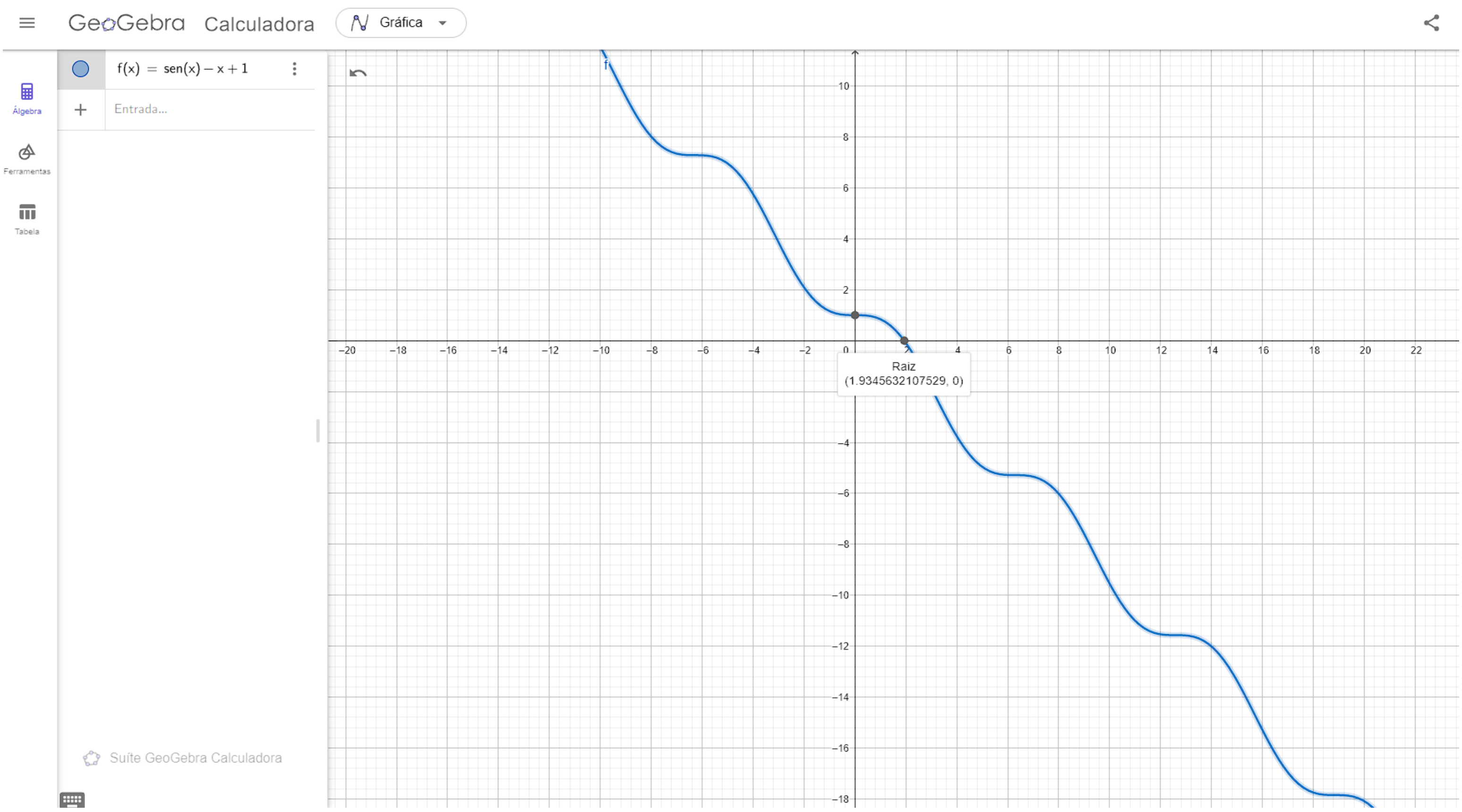
A derivada da função h(x) é h'(x) = cos(x) - 1. Notamos que esta derivada é sempre negativa para valores de x menores que π/2 e sempre positiva para valores maiores que π/2. Além disso, a derivada é zero apenas quando x = π/2.

Isso significa que a função h(x) é decrescente no intervalo (-∞,π/2] e crescente no intervalo [π/2,+∞). Como h(x) é uma função contínua, ela não pode mudar de sinal sem passar por zero, o que implica que há no máximo um zero em cada um desses intervalos.

Além disso, podemos observar que h(x) é estritamente crescente em [π/2,+∞), o que implica que há exatamente um zero em [π/2,+∞), já que h(π/2) > 0.

Já em (-∞,π/2], podemos notar que h(0) = 1 e h(π/2) > 0, o que implica que há pelo menos um zero neste intervalo. No entanto, como h(x) é uma função contínua, ela não pode ter mais do que um zero nesse intervalo, já que h(x) muda de sinal apenas uma vez.

Portanto, podemos concluir que a função h(x) = sin(x) - x + 1 possui apenas um zero real ξ, o que pode ser verificado graficamente observando que o gráfico da função intersecta o eixo x apenas uma vez.



b)

b) $h(x) = sen(x) - (x) + 1$

$\psi = sen(x) + 1$

Ponto	ψ	$x^k - x^{k-1}$
1,5	1,0261	1,0261
1,0261	1,0179	0,008
1,0179	1,0177	0,0002