UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

CIÊNCIA DA COMPUTAÇÃO

MATEMÁTICA COMPUTACIONAL (6900/1)
RELATÓRIO SOBRE MÉTODO PARA CÁLCULO DA RAIZ QUADRADA

Professor Dr. Airton Marco Polidório

Discentes:
Luca Mattosinho Teixeira RA 124316
Paula Fernandes Torres RA 123565

MARINGÁ
2024

1. INTRODUÇÃO

A busca por métodos mais eficientes na computação de funções matemáticas fundamentais é constante na área da Matemática Computacional, cujo principal objetivo é efetuar cálculos com precisão e menor custo de tempo possível. No entanto, alcançar esse objetivo muitas vezes é um desafio, já que as máquinas possuem limitações físicas e muitos problemas matemáticos, por natureza, exigem processamentos que extrapolam a capacidade de armazenar números muito grandes ou muito pequenos, os quais as máquinas enfrentam dificuldades para manipular e armazenar com precisão. Assim, os métodos existentes na Matemática Computacional servem para que seja possível contornar essas limitações. Logo, essa procura constante por soluções mais eficientes é fundamental não apenas para o avanço da Matemática Computacional, mas também para sua aplicação prática em diversos campos em que cálculos precisos e rápidos são essenciais.

2. OBJETIVO

No âmbito deste trabalho, iremos tratar sobre o problema da raiz quadrada: como calcular a raiz quadrada de um número de acordo com os objetivos da Matemática Computacional? Ou seja, como computar uma raiz quadrada de um número qualquer com precisão e menor custo de tempo? Para isto, seguiremos os princípios da Matemática Computacional:

- Redução do número de multiplicações
- Redução do número do valor do argumento.

Portanto, o objetivo deste trabalho é utilizar um método para calcular a raiz quadrada de um número N utilizando como suporte os conceitos da Matemática Computacional. Posteriormente os resultados obtidos pelo método serão comparados com aqueles obtidos com o uso da operação de cálculo da raiz quadrada da linguagem de programação utilizada, como a função sqrt.

3. MÉTODO PARA APROXIMAÇÃO DA RAIZ QUADRADA

Utilizaremos o padrão de ponto flutuante IEEE754 para nosso método de cálculo da raiz quadrada. Neste formato temos um número N com a seguinte estrutura:

$$n = 1^{s} \times (1 + f) \times 2^{e+B}$$

Em que:

- s é um bit de sinal (0 ou 1);
- f é a fração da mantissa (23 bits);
- e é o expoente (8 bits);
- B é o viés, sendo 127 para ponto flutuante de precisão simples e 1023 para ponto flutuante de precisão dupla.

Assim, podemos verificar que $\sqrt{n} = \sqrt{(1 + f) \times 2^{e}}$, em que n é nosso número decimal que queremos calcular a raiz. Portanto, temos que:

$$\sqrt{n} = \sqrt{(1 + f) \times 2^{e}}$$
$$\sqrt{n} = \sqrt{(1 + f)} \times \sqrt{2^{e}}$$

Para calcular $\sqrt{(1 + f)}$ podemos utilizar uma aproximação. Podemos considerar um chute inicial:

$$\sqrt{(1 + f)} \simeq 1 + \frac{f}{2}$$

Este chute pode ser conferido através de exemplos para verificar se a aproximação está adequada ou não:

f	$\sqrt{1 + f}$	$1 + \frac{f}{2}$
1	1,414213562	1.5
2	1,732050808	2

3	2	2.5

Através desse padrão, é notório que há uma diferença entre os valores aproximados com $1 + \frac{f}{2}$ e o resultado de $\sqrt{1+f}$. Iremos considerar $a = 1 + \frac{f}{2}$ e x essa diferença entre o valor aproximado e o valor real:

$$\sqrt{(1+f)} = a - x$$
$$(\sqrt{(1+f)})^2 = (a-x)^2$$
$$1 + f = a^2 - 2ax + x^2$$
$$1 + f = a^2 - 2ax + 0$$
$$1 + f = (1 + f/2)^2 - 2(1 + f/2)x + 0$$
$$1 + f = 1 + f + f^2/4 - 2x - fx$$
$$f^2/4 - 2x - fx = 0$$
$$f^2/4 - x(2 + f) = 0$$
$$f^2/4 = x(2 + f)$$
$$x = \frac{f^2/4}{(2+f)}$$
$$x = \frac{f^2}{(8+4f)}$$
$$\sqrt{(1+f)} = 1 + \frac{f}{2} - \frac{f^2}{(8+4f)}$$

Será utilizado o método multiplicativo de Horner para reduzir o número de multiplicações da expressão:

$$\sqrt{(1+f)} = 1 + \frac{f}{2} - \frac{f^2}{(8+4f)} = 1 + (\frac{f}{2} \times (1 - \frac{f}{4+2f}))$$

Portanto, com o resultado acima podemos calcular $\sqrt{(1+f)}$. Agora, o cálculo de 2^e será feito levando em consideração a paridade do expoente: Considerando um número x^k, temos que:

$$x^k = \left\{ (x^{\frac{k}{2}})^2 \text{ , se } k \text{ for par; } (x)(x^{\frac{k-1}{2}})^2 \text{ , se } k \text{ for ímpar} \right\}$$

Dito isso, caso k seja par, temos que $\sqrt{(2^{\frac{k}{2}})^2} = 2^{\frac{k}{2}}$. Assim, a aproximação da raiz quadrada será calculada conforme a seguinte expressão, em que k é o expoente $e - 127$:

$$2^{\frac{k}{2}} \times (1 + (\frac{f}{2} \times (1 - \frac{f}{4+2f})))$$

Caso contrário:

$$2^{\frac{k-1}{2}} \times \sqrt{2} \times (1 + (\frac{f}{2} \times (1 - \frac{f}{4+2f})))$$

Vale ressaltar que $\frac{k}{2}$ será calculado com a operação shift (k >> 1), para evitar a propagação de erro com a operação de divisão.

4. CÓDIGO FONTE

```python
from IEEE754 import Dec2IEEE
from math import sqrt
import matplotlib.pyplot as plt

'''
MATEMÁTICA COMPUTACIONAL (6900/1)
MÉTODO PARA CÁLCULO DA RAIZ QUADRADA

Professor Dr. Airton Marco Polidório

Discentes:
Luca Mattosinho Teixeira RA 124316
Paula Fernandes Torres RA 123565
'''

# Constantes para o intervalo dos valores de argumento
raiz_dois = 1.414213562
inicio = 0
fim = 1000
```

```
passo = 1

# Função para obter a fração da mantissa
def fracao_da_mantissa(mantissa_binaria):
  decimal = 0.0
  expoente = -1

  for bit in mantissa_binaria:
    if bit == '1':
      decimal += 2**expoente
    expoente -= 1

  return decimal

def decimal_para_binario(numero_decimal):
  if numero_decimal == 0:
    return '0'

  binario = ''
  while numero_decimal > 0:
    resto = numero_decimal % 2
    binario = str(resto) + binario
    numero_decimal //= 2

  return binario

def par(n):
  if n&1:
    return False
  return True

# Função para calcular a raiz quadrada com a aproximação
especificada
def raiz_aproximada(f):
  return ((1 - (f / (4 + 2 * f))) * (f/2)) + 1

# Função geral para calcular a raiz quadrada
def raiz(x):
```

```
  y = Dec2IEEE(x)
  expoente = y.Fbits.e - 127
  f = decimal_para_binario(y.Fbits.f)
  f = fracao_da_mantissa(f)

  if par(expoente):
    return (2**(expoente>>1)) * raiz_aproximada(f)
  else:
    return 2**((expoente-1)>>1) * raiz_dois * raiz_aproximada(f)

# Listas para armazenar os valores necessários
valores_aproximados = []
valores_exatos = []
erros = []
argumentos = []

# Função que calcula os resultados para um intervalo de valores
de argumento
def calculo_raizes():
  indice_lista = 0
  for i in range(inicio,fim,passo):
    valores_aproximados.append(raiz(i/100))
    valores_exatos.append(sqrt(i/100))
    erros.append(abs(valores_aproximados[indice_lista] -
valores_exatos[indice_lista]))
    argumentos.append(i/100)
    indice_lista += 1

def graficos_resultados():
  plt.plot(argumentos, valores_exatos, label='Corretos',
linestyle='-', color='red')
  plt.plot(argumentos, valores_aproximados, label='Aproximados',
linestyle='-', color='blue')
  plt.title('Valores obtidos do cálculo da raiz quadrada')
  plt.xlabel('Argumento')
  plt.ylabel('Valores')
  plt.legend()
  plt.grid(True)
```

```
  plt.show()

def grafico_erros():
  plt.plot(argumentos, erros)
  plt.title('Análise de erros da aproximação da raiz quadrada')
  plt.xlabel('Argumento')
  plt.ylabel('Erro')
  plt.show()

# Chamadas das funções para calcular os valores e gerar os
gráficos dos erros e resultados
if __name__ == "__main__":
  calculo_raizes()
  grafico_erros()
  graficos_resultados()
```

O código foi feito na linguagem Python contendo como única dependência a biblioteca *matplotlib* que pode ser obtida para Windows ou Linux através do comando no terminal:

```
Unset
pip install matplotlib
```

5. AVALIAÇÃO DOS RESULTADOS

Os resultados obtidos pela aproximação da raiz quadrada foram comparados com os resultados obtidos utilizando a função sqrt através dos gráficos abaixo:

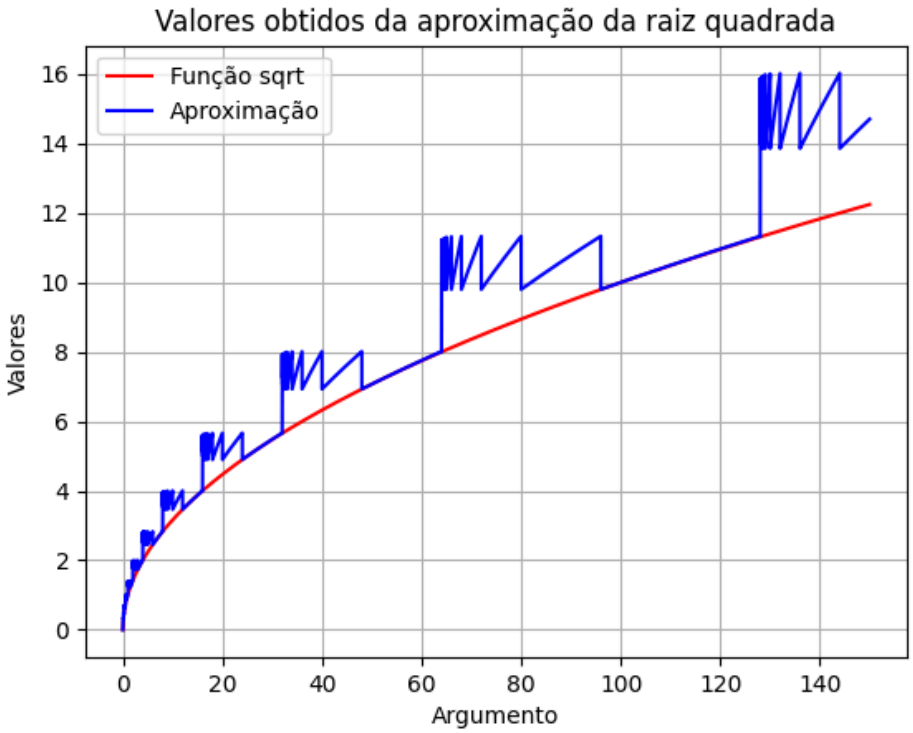

Figura 1: valores obtidos para valores de argumento de 0 a 150.

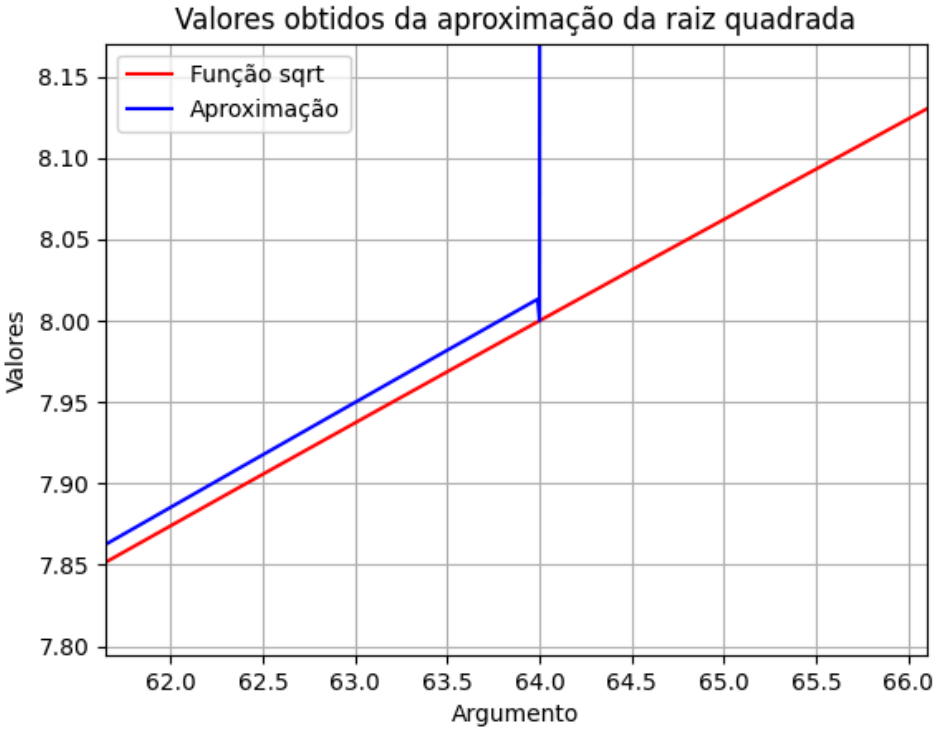

Figura 2: valores obtidos para para valores de argumento de 62 a 66.

Observa-se que no intervalo de 62 a 66 os valores obtidos pela aproximação são próximos aos valores de sqrt até o ponto 64, em que há uma raiz exata de valor 8. A função de aproximação consegue calcular esse valor exato com precisão, porém após esse ponto a função de aproximação distancia-se da função sqrt e o erro aumenta em sua magnitude.

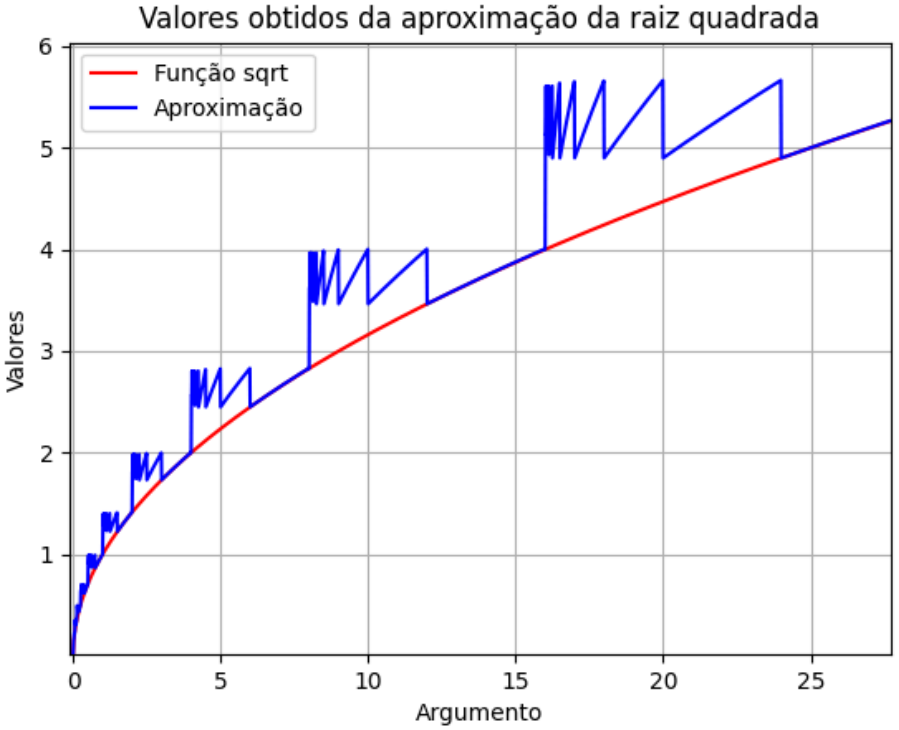

Figura 3: valores obtidos para valores de argumento de 0 a 25.

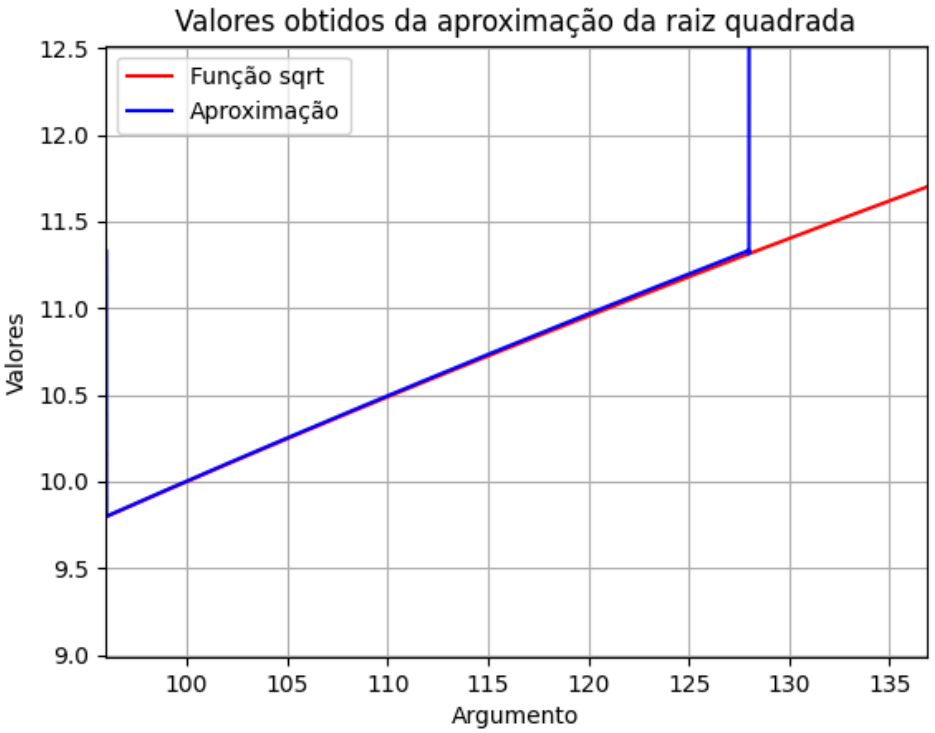

Figura 4: valores obtidos para valores de argumento de 100 a 135.

No intervalo de 100 a 135 é exibido como a função de aproximação calcula uma sequência de valores da raiz quadrada com precisão, incluindo as raízes exatas de 100 e 121. Novamente, a função de aproximação segue esse padrão de calcular alguns valores com precisão, porém após um determinado ponto a função de aproximação distancia-se da função sqrt e a magnitude do erro aumenta.

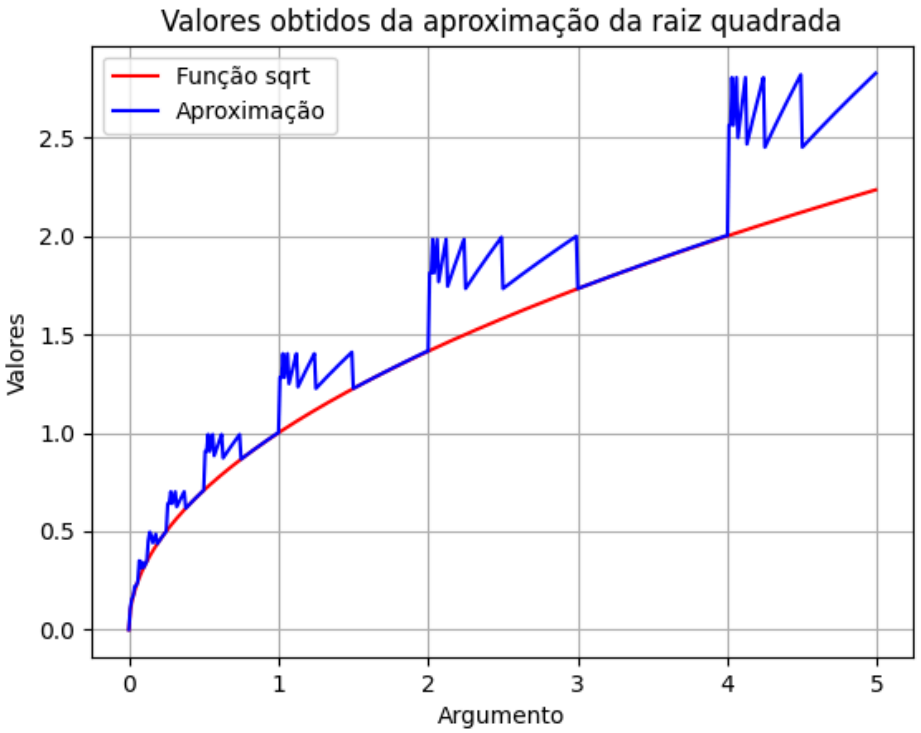

Figura 5: valores obtidos para valores de argumento de 0 a 5.

Através desses gráficos foi possível perceber que a função segue um padrão de comportamento, em que há uma sequência de cálculos com precisão para um determinado intervalo e em seguida uma sequência com maior propagação do erro para outro intervalo, e este comportamento se repete ao longo de todo o intervalo selecionado para os valores de argumento.

6. AVALIAÇÃO DO ERRO

Os resultados obtidos através da análise do erro calculado entre os valores aproximados da raiz quadrada e utilizando a função sqrt são descritos através dos gráficos abaixo:

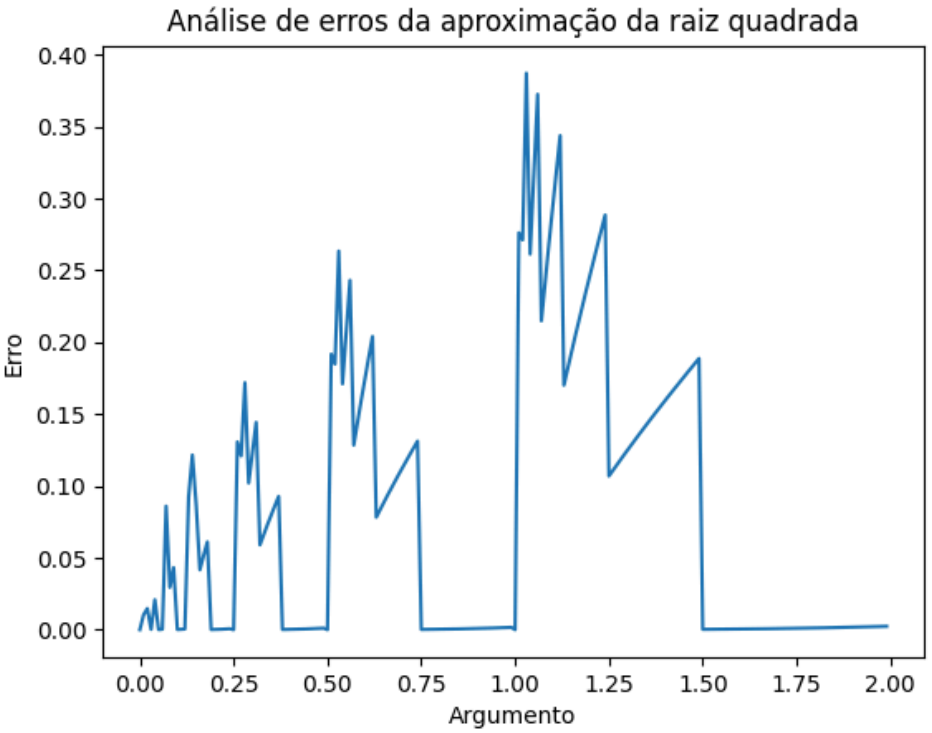

Figura 6: análise do erro para valores de argumento de 0 a 2.

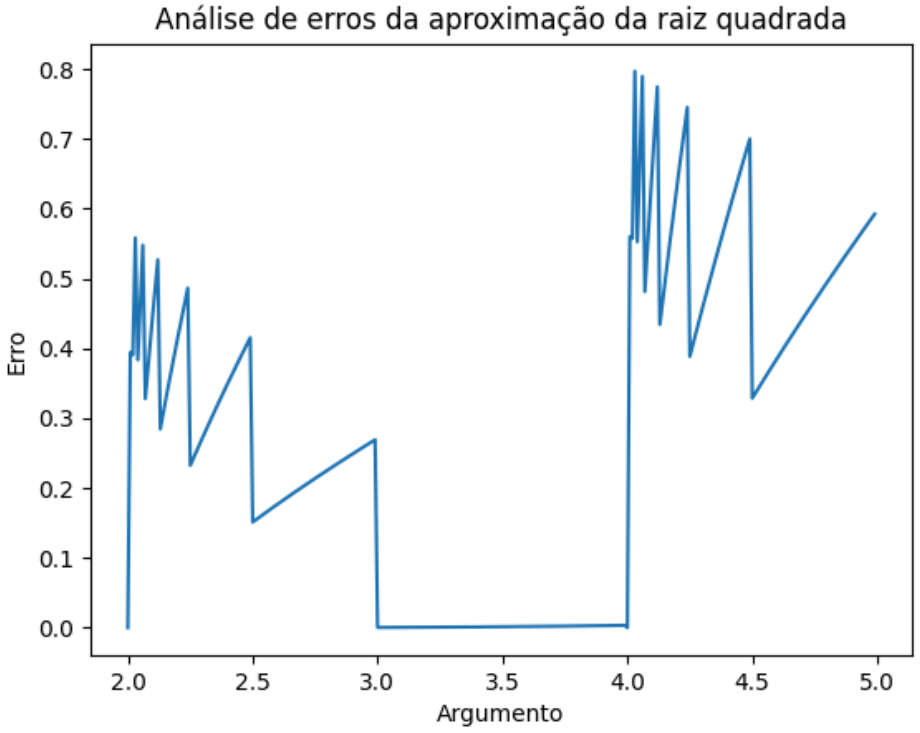

Figura 7: análise do erro para valores de argumento de 2 a 5.

Figura 8: análise do erro aproximando o gráfico para o argumento de valor 4.

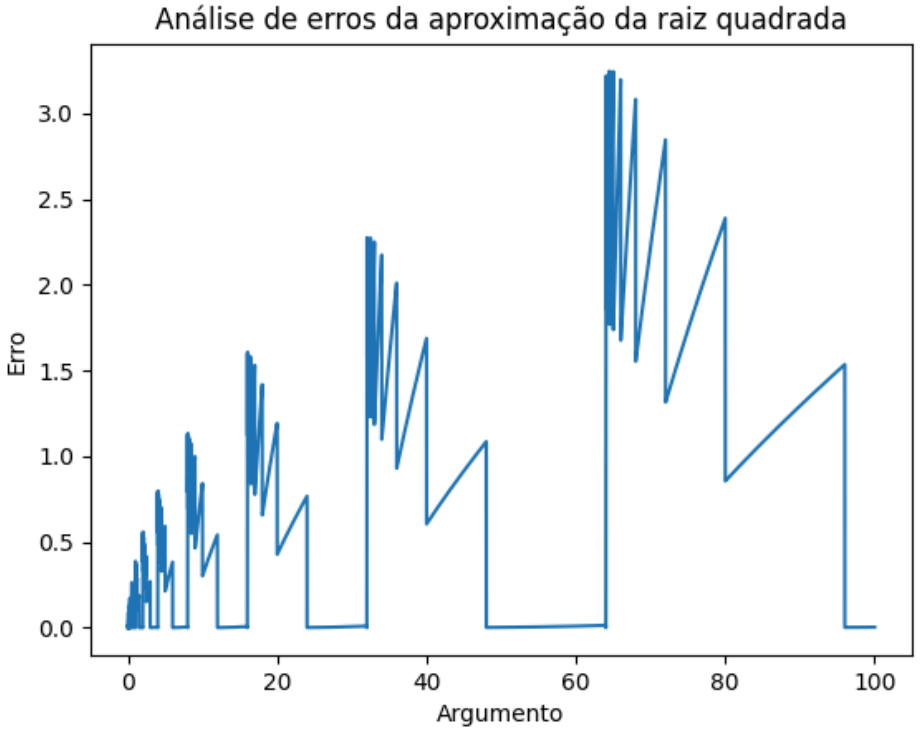

Figura 9: análise do erro para valores de argumento de 0 a 100.

Através dos gráficos nota-se novamente o mesmo padrão para o erro: durante certos intervalos o erro permanece em 0, indicando uma baixa magnitude do erro, e em seguida a magnitude do erro aumenta chegando na primeira casa decimal. O erro atinge a segunda casa decimal em valores próximos ao 0, e já atinge a primeira casa decimal em um ponto próximo ao valor de argumento 0.25. Ao aproximar o gráfico, é possível ver mais detalhadamente a magnitude do erro na Figura 8, que mostra como o erro se propaga até a primeira casa decimal para uma variação relativamente pequena de argumentos.

7. CONCLUSÃO

Em conclusão, reduzir o argumento para calcular a raiz quadrada e utilizar a aproximação especificada é uma estratégia eficaz para calcular a raiz quadrada de um número qualquer minimizando o número de multiplicações e os erros causados por limitações na representação de números de ponto flutuante.

Os resultados alcançados com a aproximação da raiz quadrada corresponderam a certos resultados obtidos pela função sqrt, havendo variações para determinados intervalos de valores de argumento e ocasionando maior magnitude do erro para esses intervalos.

Ao observar os resultados obtidos pelas simulações, nota-se um padrão em que a propagação do erro é melhor em alguns intervalos de valores de argumento, e posteriormente o erro é propagado rapidamente, chegando até a primeira casa decimal. Esse comportamento repete-se ao longo de todo o intervalo estabelecido para os valores de argumento. Percebe-se que, no geral, a precisão alcançada não é muito alta, considerando que para valores de magnitude baixa o erro já começa a ser visto na segunda casa decimal.

8. REFERÊNCIAS

[1] American National Standards Institute / Institute of Electrical and Electronics Engineers: IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985, New York, 1985.