

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
IDCA3703 - PROGRAMAÇÃO PARALELA

TARFA 3 - APROXIMAÇÃO MATEMÁTICA DE π
RELATÓRIO DE EXECUÇÃO

ERNANE FERREIRA ROCHA JUNIOR

NATAL/RN, 04 DE ABRIL DE 2025

SUMÁRIO

1. INTRODUÇÃO	3
2. METODOLOGIA	4
2.1. IMPLEMENTAÇÃO EM C	4
2.2. VISUALIZAÇÃO E ANÁLIZE COM PYTHON	4
3. RESULTADOS	6
Figure 1: Gráfico de Interações vs Aproximações de π	7
Figure 2: Gráfico de Interações vs Erro Absoluto	8
Figure 3: Interações vs Tempo de Execução	9
Figure 4: Gráfico de Interações vs Casas Decimais Corretas	10
4. CONCLUSÃO	11
5. ANEXOS	12

1. INTRODUÇÃO

A constante π (pi) é uma das mais importantes e conhecidas da matemática e aparece em diversas áreas do conhecimento, como geometria, física, estatística e engenharia. Sua definição clássica é a razão entre o comprimento de uma circunferência e seu diâmetro, mas seu valor exato não pode ser representado de forma finita, sendo um número irracional com infinitas casas decimais não periódicas. Devido a essa característica, é comum utilizar aproximações de π em aplicações computacionais que demandam alto desempenho e precisão.

Diversas séries matemáticas foram desenvolvidas ao longo da história para calcular aproximações cada vez mais precisas de π . Dentre essas, a série de Leibniz é uma das mais simples e didáticas, embora não seja a mais eficiente em termos de convergência. Ela permite ilustrar de maneira clara o conceito de aproximação por séries infinitas e o impacto do número de iterações sobre a precisão do resultado obtido.

Neste trabalho, foi implementado um programa em linguagem C para calcular aproximações de π utilizando a série de Leibniz, variando o número de iterações e medindo o tempo de execução de cada etapa. Os resultados obtidos foram analisados quanto à acurácia (número de casas decimais corretas) e ao tempo necessário para alcançá-la. A partir dessa análise, buscou-se refletir sobre a relação entre processamento computacional e precisão, tema recorrente em áreas como simulações físicas e inteligência artificial, onde a obtenção de resultados precisos é essencial para a confiabilidade dos sistemas.

2. METODOLOGIA

Para este experimento, foi utilizada a série de **Leibniz** para calcular aproximações sucessivas do valor de π . A série é representada pela fórmula:

$$\pi = 4 \cdot \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

Essa série alterna sinais positivos e negativos e converge lentamente para o valor de π , o que a torna adequada para demonstrar como o aumento no número de iterações influencia tanto na precisão quanto no tempo de execução de um algoritmo.

2.1. IMPLEMENTAÇÃO EM C

Foi desenvolvido um programa na linguagem C que:

1. Calcula o valor aproximado de π utilizando a série de Leibniz.
2. A cada nova casa decimal correta obtida (comparando com o valor real de π disponível via *M_PI* da biblioteca *math.h*), o programa:
 - 2.1. Registra o número de iterações necessárias.
 - 2.2. Calcula o erro absoluto em relação ao valor real de π .
 - 2.3. Mede o tempo de execução acumulado desde o início do cálculo.
3. Os dados obtidos são armazenados em um arquivo `.csv` com os seguintes campos:
 - 3.1. Número de iterações
 - 3.2. Aproximação de π
 - 3.3. Erro absoluto
 - 3.4. Tempo de execução (em segundos)
 - 3.5. Número de casas decimais corretas

A medição de tempo foi feita utilizando a biblioteca *sys/time.h*, permitindo maior precisão no acompanhamento do desempenho da execução.

2.2. VISUALIZAÇÃO E ANÁLISE COM PYTHON

Um script em Python foi criado para ler o arquivo *CSV* gerado e produzir gráficos que relacionam:

- Aproximação de π versus número de iterações
- Erro absoluto versus número de iterações
- Tempo de execução versus número de iterações
- Casas decimais corretas versus número de iterações

Os gráficos utilizam escalas logarítmicas para destacar a evolução exponencial do número de iterações necessárias conforme aumenta a precisão desejada.

Desse modo conseguimos não apenas visualizar a relação entre precisão e custo computacional, mas também comparar diretamente a eficiência da série de Leibniz com as exigências crescentes de precisão em aplicações reais.

3. RESULTADOS

A execução do programa desenvolvido em C resultou em uma série de aproximações sucessivas de π , utilizando a série de Leibniz. A cada incremento no número de casas decimais corretas em relação ao valor real de π , o algoritmo registrou a quantidade de iterações necessárias, o tempo acumulado de execução e o erro absoluto em relação a M_PI , constante da biblioteca matemática da linguagem C.

Os dados coletados revelam, de forma clara, o comportamento de convergência extremamente lenta da série. Por exemplo, para alcançar 5 casas decimais corretas foram necessárias cerca de 1.688 iterações, enquanto que, para atingir 10 casas, o número de iterações ultrapassou 155 milhões, com um tempo de execução superior a dois minutos. Essa discrepância crescente é um reflexo direto da natureza da série de Leibniz, que embora simples de implementar, não é eficiente para cálculos de alta precisão.

Iterations	Approximation of PI	Absolute Error	Execution Time (s)	Correct Decimal Places
1	4.000000000000000	0.8584073464	0.000022	0
3	3.466666666666667	0.3250740131	0.000079	1
19	3.194187909231943	0.0525952556	0.000117	2
119	3.149995866593470	0.0084032130	0.000267	3
1688	3.141000236580159	0.0005924170	0.002559	4
10794	3.141500009528466	0.0000926441	0.014958	5
136121	3.141599999994786	0.0000073464	0.134407	6
1530012	3.141592000000233	0.0000006536	1.401601	7
18660304	3.141592600000001	0.0000000536	17.149694	8
155973051	3.141592659999999	0.0000000064	143.128878	9
1362988896	3.141592653000000	0.0000000006	1255.245897	10
1584294516	3.141592653500000	0.0000000001	1459.173499	11
1608612706	3.141592653580000	0.0000000000	1481.370748	12
1611227808	3.141592653589000	0.0000000000	1483.765222	13
1611428360	3.141592653589700	0.0000000000	1483.949505	14

O gráfico apresentado abaixo mostra como a aproximação de π evolui em relação ao número de iterações. Nele, é possível observar que, mesmo com um grande número de iterações, a aproximação tende lentamente a estabilizar em torno de 3.14159. A escala logarítmica no eixo y ressalta a explosão no número de iterações à medida que se busca mais precisão.

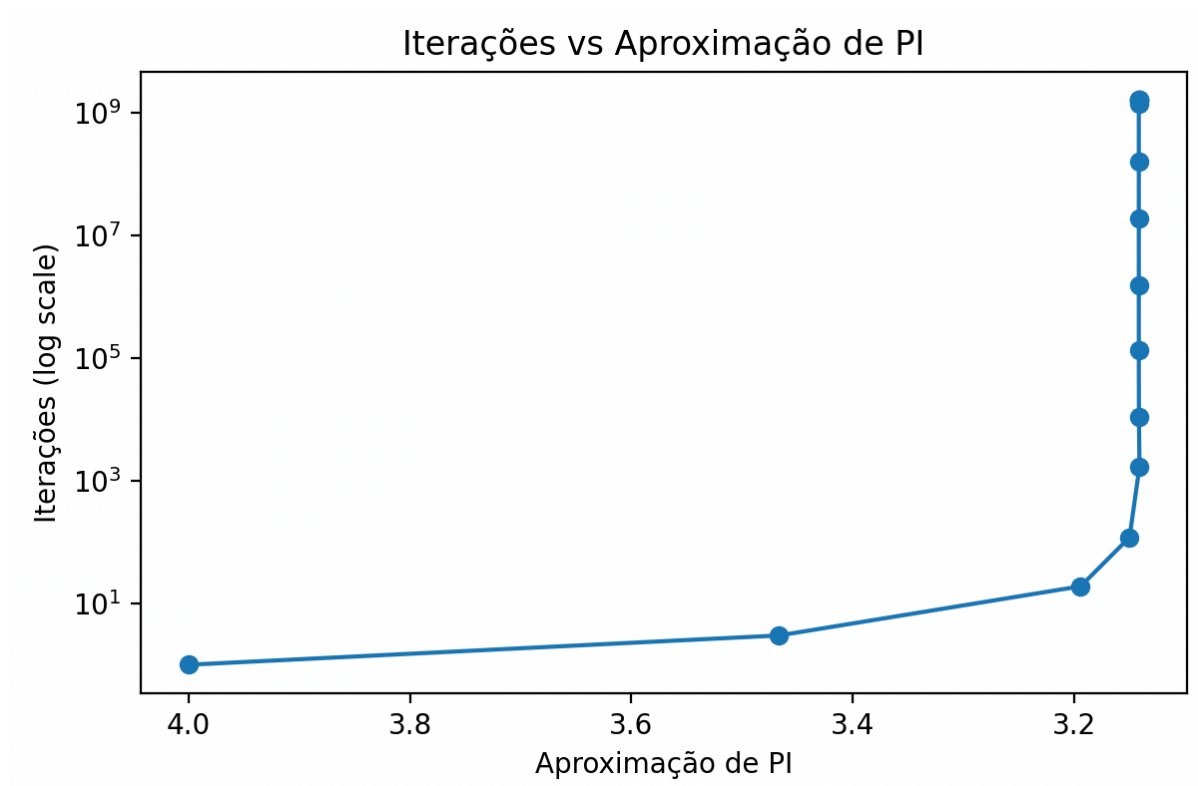


Figure 1: Gráfico de Interações vs Aproximações de π

A relação entre o número de iterações e o erro absoluto também foi analisada. Como ilustrado na imagem abaixo, há uma queda constante no erro à medida que as iterações aumentam. No entanto, essa queda é bastante suave, evidenciando novamente a baixa eficiência da série para grandes precisões. O uso de escalas logarítmicas em ambos os eixos facilita a visualização dessa tendência de queda lenta e contínua.

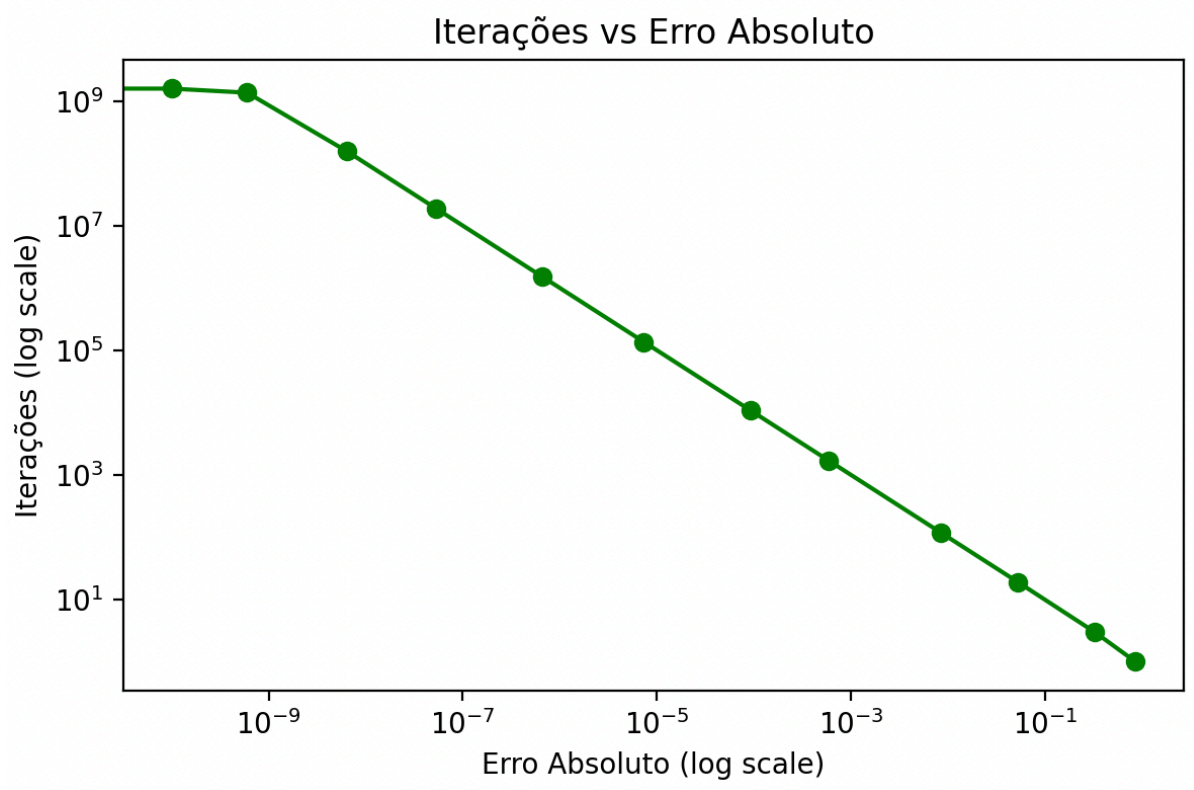


Figure 2: Gráfico de Iterações vs Erro Absoluto

Outro aspecto fundamental analisado foi o tempo de execução. A imagem abaixo apresenta a evolução do tempo em relação ao número de iterações, e, como esperado, o crescimento é praticamente linear. Essa característica reforça que, embora o algoritmo em si seja simples, o custo computacional aumenta significativamente à medida que se busca maior precisão.

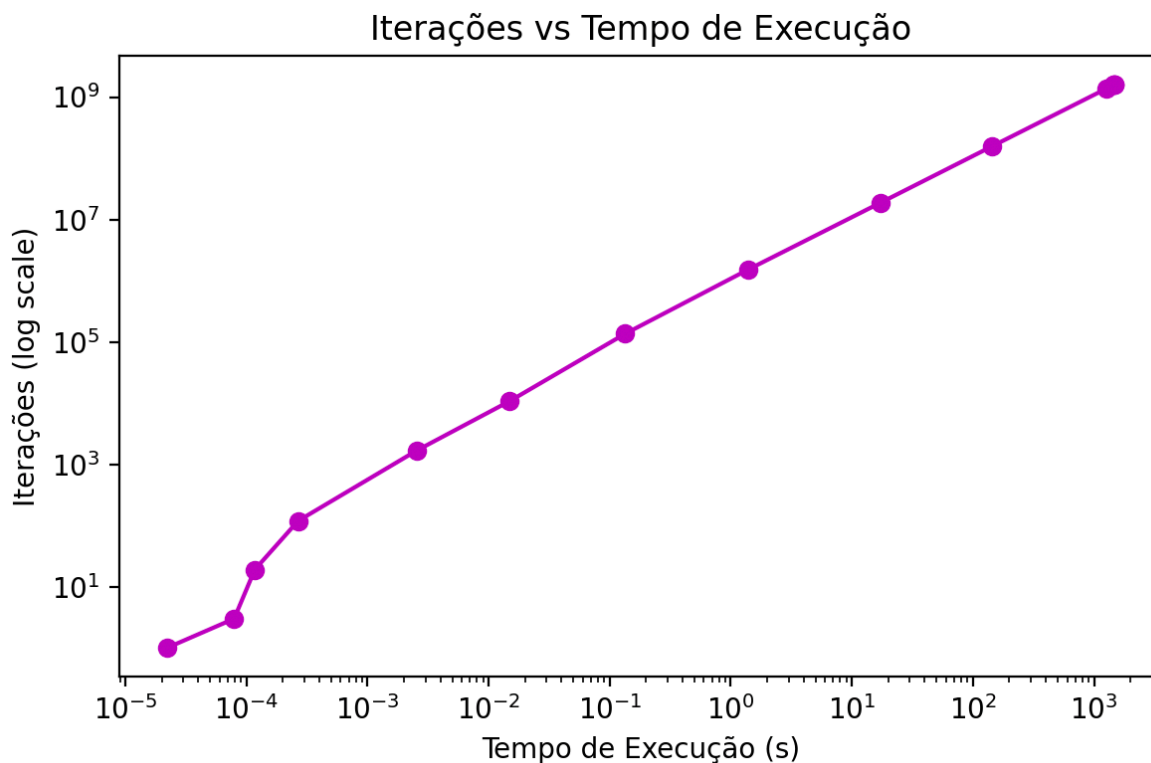


Figure 3: Iterações vs Tempo de Execução

Por fim, a imagem abaixo mostra a relação entre o número de casas decimais corretas e o número de iterações necessárias para alcançá-las. Esse gráfico evidencia de forma impactante o crescimento exponencial do esforço computacional exigido para obter cada nova casa decimal correta. A curva cresce abruptamente, deixando claro que métodos mais eficientes seriam necessários para aplicações que exigem precisão elevada.

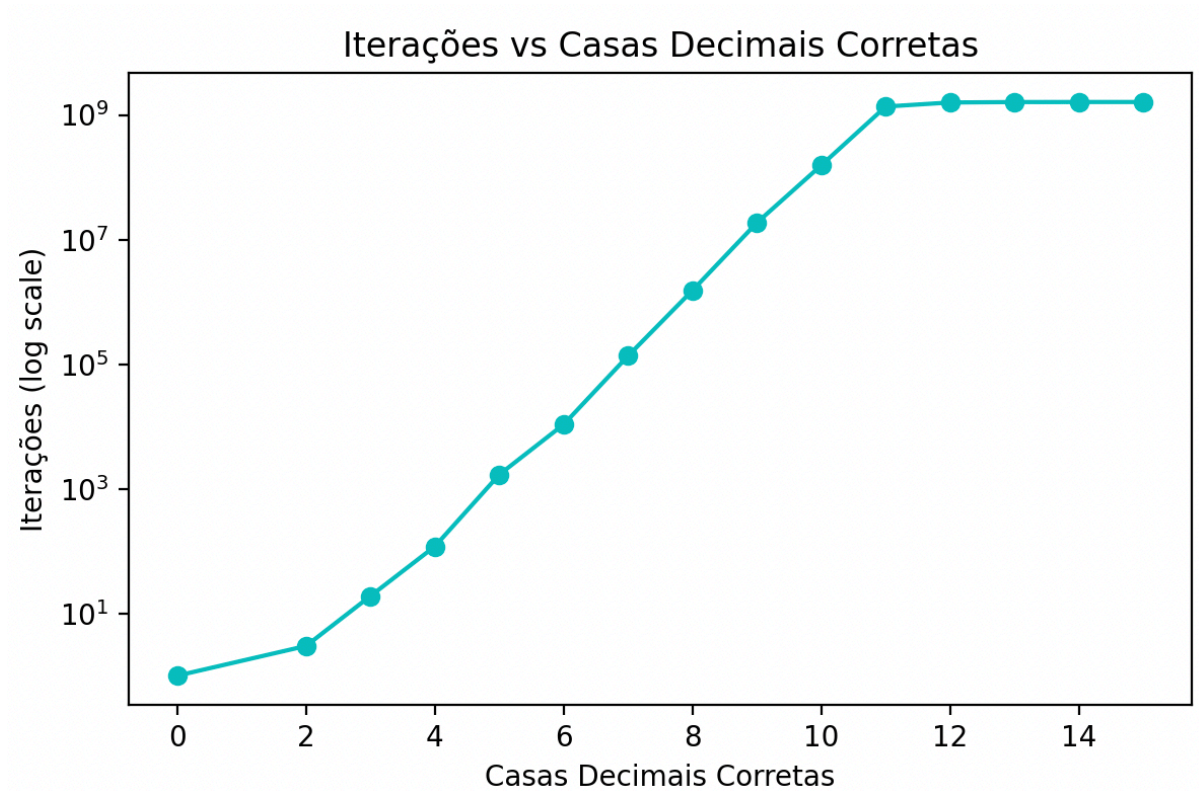


Figure 4: Gráfico de Iterações vs Casas Decimais Corretas

Esses resultados ilustram não apenas os limites da série de Leibniz, mas também a relação direta entre processamento computacional e precisão numérica. Em contextos práticos, como simulações físicas ou aplicações em inteligência artificial, esse tipo de análise é essencial para entender os trade-offs entre acurácia e desempenho, e para justificar a adoção de algoritmos mais sofisticados quando necessário.

4. CONCLUSÃO

A aproximação de π utilizando a série de Leibniz, embora didaticamente interessante por sua simplicidade, revelou-se extremamente ineficiente para obtenção de resultados com alta precisão. A análise demonstrou que, para cada nova casa decimal correta, o número de iterações e o tempo de execução crescem de forma exponencial, tornando o método inviável para aplicações que exigem precisão elevada em curto espaço de tempo.

Apesar dessas limitações, o experimento cumpriu seu objetivo de ilustrar a relação entre esforço computacional e acurácia numérica. Através da implementação em C e da análise gráfica em Python, foi possível visualizar claramente como a precisão cresce com o número de iterações e como isso impacta diretamente o tempo de processamento.

Essa observação tem implicações práticas importantes. Em áreas como simulações físicas, modelagem computacional e inteligência artificial, a precisão dos cálculos influencia diretamente a qualidade dos resultados. No entanto, há sempre um custo associado: quanto maior a precisão desejada, maior o consumo de recursos computacionais. Isso reforça a importância de escolher algoritmos mais eficientes e métodos de aproximação mais rápidos, principalmente quando se trabalha com grandes volumes de dados ou em tempo real.

Portanto, embora a série de Leibniz não seja indicada para aplicações de precisão, ela serve como excelente ferramenta pedagógica para compreender conceitos fundamentais de séries infinitas, erros numéricos e análise de desempenho, além de oferecer uma introdução prática ao estudo da eficiência computacional.

5. ANEXOS

- Repositório no Github com o programa desenvolvido: <https://github.com/ErnaneJ/parallel-programming-dca3703>

