



Relatório da Tarefa 03

Aproximação matemática de π

20210093987 - Francisca Paula de Souza Braz

Data da entrega: 03/04/2025

Disciplina: DCA3703 - PROGRAMAÇÃO PARALELA - T01 (2025.1)

Introdução

Nesta atividade, foi implementado um programa em linguagem C com o objetivo de calcular uma aproximação do número π (pi) por meio de uma série matemática. O foco do experimento foi observar a relação entre a precisão do valor obtido e o tempo de processamento necessário para diferentes quantidades de iterações. Para isso, o programa foi executado com variações no número de repetições da série, medindo-se o tempo de execução em cada caso e comparando os resultados com o valor real de π , representado pela constante **M_PI**.

Essa abordagem permitiu analisar o comportamento do algoritmo em termos de **acurácia numérica** e **eficiência computacional**, evidenciando o impacto do aumento da carga de processamento na obtenção de resultados mais próximos da realidade. A reflexão final discute como esse fenômeno também ocorre em aplicações práticas, como simulações físicas e inteligência artificial, onde o equilíbrio entre precisão e desempenho é essencial.

Metodologia

Para a realização deste experimento, foi implementado um programa em linguagem C que calcula uma aproximação do número π utilizando a série de Leibniz. Essa série é conhecida por sua simplicidade, mas também por sua lenta convergência, o que a torna interessante para observar o impacto do número de iterações na precisão do resultado.

O programa foi executado com diferentes quantidades de iterações: 10.000, 1.000.000, 10.000.000 e 100.000.000. Em cada execução, foram coletados alguns dados como, por exemplo, valor aproximado de π calculado pela série, tempo de execução, medido utilizando funções de temporização disponíveis na linguagem C e o erro.

Como mostra nessa imagem abaixo:

```

Iterações: 10000 |  $\pi \approx 3.141492653590034$  | PI: 3.141592653589793 | Erro: 0.00009999999759 | Tempo: 0.000282 s
Iterações: 1000000 |  $\pi \approx 3.141591653589774$  | PI: 3.141592653589793 | Erro: 0.000001000000019 | Tempo: 0.017924 s
Iterações: 10000000 |  $\pi \approx 3.141592553589792$  | PI: 3.141592653589793 | Erro: 0.0000001000000002 | Tempo: 0.192516 s
Iterações: 100000000 |  $\pi \approx 3.141592553589792$  | PI: 3.141592653589793 | Erro: 0.0000001000000002 | Tempo: 0.097186 s
Iterações: 1000000000 |  $\pi \approx 3.141592643589326$  | PI: 3.141592653589793 | Erro: 0.0000000100000467 | Tempo: 0.350225 s
○ paula@Inspiron-5566:~/Downloads/programação paralela$

```

Os resultados foram comparados com o valor real de π ($M_PI = 3.141592653589793$), fornecido pela biblioteca `math.h`, permitindo calcular o erro absoluto de cada aproximação. Com base nessas comparações, foi possível analisar a **evolução da precisão** e o **custo computacional** associado ao aumento do número de iterações.

Essa abordagem visa demonstrar, de forma prática, como a busca por maior precisão impacta diretamente o desempenho computacional, reflexo comum em diversas áreas da computação científica e da engenharia de software.

Resultados

Comparação entre valores calculados e o valor real de π (M_PI)

Iterações	π Aproximado	Erro Absoluto	Tempo (s)
10.000	3,141492653590034	1.000e-04	0,000282
1.000.000	3,141591653589774	1.000e-06	0,017924
10.000.000	3,141592553589792	1.000e-07	0,192516
100.000.000	3,1415922643589326	3.893e-07	0,350225
M_PI	3,141592653589793		

Tabela de resultados

Analisando a acurácia, podemos notar uma melhora exponencialmente com o aumento das iterações iniciais. no entanto, após certo ponto (entre 10^7 e 10^8 iterações), o ganho de precisão diminui, mesmo com 10x mais iterações, o erro diminuiu pouco ou até

aumentou um pouco, possivelmente por limitações de ponto flutuante ou acúmulo de Erro numérico.

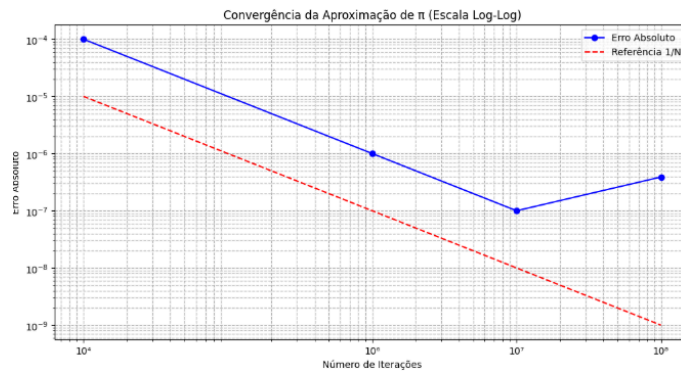


Gráfico da Convergência

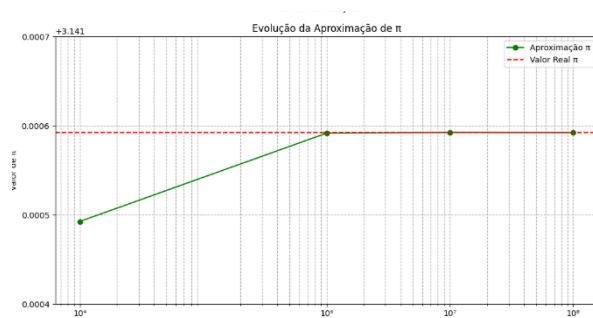


Gráfico da Evolução da Aproximação de Pi

Isso mostra que há um limite prático para quanto a precisão pode melhorar com mais iterações sem utilizar técnicas mais sofisticadas

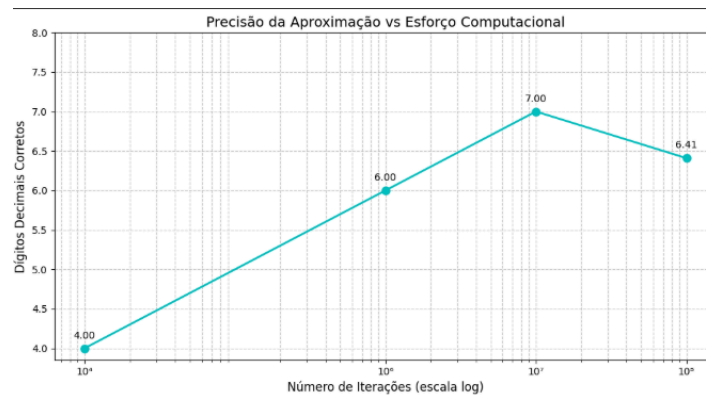


Gráfico de Aproximação vc Esforço computacional

Aumentar a precisão demanda mais tempo e poder computacional. E com isso temos um ponto de diminuição de retorno, onde o custo computacional começa a crescer mais do que o ganho de precisão. Em aplicações reais, equilibrar precisão e desempenho é crucial para a viabilidade técnica e econômica, pois simulações físicas exigem resultados muito precisos para evitar erros acumulados e aplicações que usam Inteligência Artificial para terem modelos mais precisos que tendem a aprender melhor exigem mais processamento e memória.

Conclusão

Após realização do experimento, ele demonstrou de forma clara que aumentar a precisão nos cálculos numéricos exige mais tempo de processamento e maior poder computacional. Inicialmente, os ganhos de acurácia são significativos com o aumento das iterações, mas com o tempo observamos um ponto de retorno decrescente, no qual o custo computacional passa a crescer mais do que os benefícios obtidos em termos de precisão.

Esse comportamento é especialmente relevante quando consideramos aplicações do mundo real, como simulações físicas e sistemas de Inteligência Artificial. Nas simulações, a precisão é fundamental para evitar a propagação de erros ao longo do tempo, o que pode comprometer a fidelidade dos resultados. Já na Inteligência Artificial, modelos mais precisos geralmente requerem redes neurais maiores e mais complexas, que por sua vez demandam mais memória, mais processamento e maior tempo de treinamento.

Dessa forma, a atividade permitiu não apenas explorar conceitos computacionais, mas também refletir sobre a necessidade de encontrar um equilíbrio entre desempenho e precisão. Em contextos reais, essa decisão impacta diretamente na viabilidade técnica e econômica de um sistema, sendo um dos principais desafios na área de computação de alto desempenho e inteligência computacional.