Avaliação de desempenho da equação do calor (Python vs Julia)

Enzo Lisbôa Peixoto
Nathan Mattes
Pedro Scholz Soares
Lucas Mello Schnorr (orientador)

Resumo

Este trabalho tem como objetivo comparar o desempenho das linguagens de programação Python e Julia na resolução de equações do calor em 1, 2 e 3 dimensões. Se atentando ao tempo percorrido na execução dos programas, assim como a memória utilizada pelos mesmos, a fim de determinar qual das duas é a mais eficiente para esta aplicação específica. Este trabalho é feito com o intuito de ser um estudo em cálculo computacional e análise de desempenho.

1. DESCRIÇÃO

O objeto de estudo deste trabalho é a solução numérica da equação do calor em uma, duas e três dimensões. Essa equação descreve a forma como o calor se espalha por um objeto. Serão implementados algoritmos em duas linguagens de programação distintas, julia e python. A justificativa vem do fato de python ser consistente para uso científico devido às suas bibliotecas e ser amplamente popular. Por outro lado, julia apresenta ótimos resultados em cálculo numérico.
Assim, o objetivo é fazer a análise desses algoritmos e, com isso, quantificar e compreender os fatores que influenciam no desempenho em cada linguagem. Para isso, métricas como tempo de execução e uso de memória em cada dimensão serão analisados. Ao final do trabalho esperamos compreender as razões por trás das diferenças observadas, aprimorar nossos conhecimentos em cálculo numérico e aplicar os conteúdos vistos durante a cadeira de Análise de desempenho.

2. MÉTODO DE ANÁLISE

As duas linguagens possuem abordagens diferentes em relação à execução do código. Julia utiliza compilação just-in-time (JIT): A primeira execução de um código em Julia pode ser mais lenta devido à necessidade de compilar o código antes de executá-lo. Em execuções subsequentes, o código já estará compilado, o que resulta em tempos de execução significativamente mais rápidos. Por sua vez, Python é uma linguagem interpretada: Cada vez que um script Python é executado, o interpretador lê e interpreta o código linha por linha. Isso pode resultar em tempos de execução mais lentos, especialmente para tarefas computacionalmente intensivas. Para garantir uma comparação justa, o tempo de compilação inicial do Julia será separado do tempo de execução. O foco da análise será o tempo de execução após a compilação inicial, assim como o uso de recursos pela aplicação durante sua execução, tais como memória utilizada.

3. JUSTIFICATIVA

A resolução de Equações Diferenciais Parciais já foi estudada pelos membros do grupo em outras disciplinas, através da abordagem analítica. No entanto, nenhum de nós explorou a dimensão computacional da solução, que é crucial para problemas complexos do mundo real. Enxergamos esse trabalho como uma oportunidade de estudar a equação sob outra visão, possibilitando abordar conteúdos relativos à cálculo numérico. Também é uma chance de se familiarizar com Julia, já que nenhum dos integrantes possui experiência com a linguagem.

4. DEFINIÇÃO DE MÉTRICAS

A análise de desempenho deste trabalho será fundamentada em duas métricas principais: o tempo de execução e o uso de memória. Para obter resultados robustos, faremos múltiplas medições para cada implementação. Depois, será calculada uma média para garantir robustez nos resultados. Julia, por utilizar a compilação just-in-time (JIT), terá o tempo inicial de compilação separado do tempo de execução real do código. Essa abordagem nos permitirá comparar o desempenho de Julia com o de Python, garantindo uma avaliação mais justa. Paralelamente, monitoraremos o uso de memória para avaliar a eficiência de alocação de ambas as linguagens, uma métrica crucial para entender o quão viável cada solução é para problemas de grande escala.

5. CRONOGRAMA PRELIMINAR

Atividade	Prazo Máximo
Escrita dos algoritmos de resolução da equação	05/09
Definição dos métodos de coleta	12/09
Coleta parcial dos dados	26/09
Analise dos dados parciais	04/10
Apresentação dos dados parciais	06/10
Coleta total dos dados	24/10
Análise dos dados	07/11
Escrita do relatório final	30/11