

Otimização de Programas de escoamento de Fluidos em um Modelo Bidimensional*

Anna V. G. Marciano,[‡] Claudio Schepke, Roberta A. Spigolon, César F. C. Cristaldo

¹Laboratório de Estudos Avançados em Computação (LEA)
Universidade Federal do Pampa (UNIPAMPA) – Alegrete – RS – Brazil

{annamarciano, robertaspigolon}.aluno@unipampa.edu.br
{claudioschepke, cesarcristaldo}@unipampa.edu.br

Resumo. Este estudo aborda a otimização de programas de simulação computacional para reduzir o tempo de processamento e aumentar a eficiência. Como estudo de caso foi utilizado o método *Quick Scheme*. A análise do código resultou na eliminação de trechos ineficientes e variáveis desnecessárias, contribuindo para simulações mais rápidas, precisas e estáveis, com economia de recursos. A otimização mostrou-se viável, promovendo avanços na pesquisa.

1. Introdução

A simulação computacional desempenha um papel crucial em várias áreas, oferecendo percepções valiosas para a compreensão de fenômenos complexos. A aplicação prática dessas simulações é vital, especialmente em contextos como engenharia, onde a precisão e rapidez são essenciais. No entanto, a eficácia dessas simulações muitas vezes é limitada pelo tempo de processamento [Lucca 2022]. Ao enfrentar desafios significativos relacionadas a eficiência, busca-se por soluções inovadoras.

Este trabalho aborda a problemática do tempo elevado em programas de simulação bidimensional de escoamento de fluidos, destacando a importância de otimizar esses programas para reduzir o tempo de processamento e aumentar a precisão. Este artigo foca na otimização de programas de simulação por meio do método *Quick Scheme*, uma abordagem reconhecida por sua eficiência computacional [de Oliveira 2020].

2. Quick Scheme

O método de diferenças finitas *Quadratic Upstream Interpolation for Convective Kinetics*, conhecido como *Quick Scheme*, pode ser aplicado para a equação de convecção-difusão quase estacionária. O método *Quick Scheme* trata-se de uma técnica numérica reconhecida por sua eficiência computacional e habilidade em acelerar a convergência de soluções numéricas, servindo com muita precisão para “resolver problemas de convecção baseado em uma formulação conservadora do volume de controle integral” [Chichinelli 2022]. A Figura 1 apresenta o esquema de controle de volume único bem como a relação com os pontos discretos vizinhos. Este método tem como principal ponto a consideração de “interpolação quadrática ponderada de três pontos na face da célula” [Chichinelli 2022].

O *Quick Scheme* despertou particular interesse devido à sua capacidade de lidar com sistemas de equações diferenciais parciais associados a escoamentos de fluidos. A

*Trabalho parcialmente financiado pelo Edital CNPq: Projeto 135963/2023-0.

[‡]Bolsista PIBIC/CNPq 2023/2024.

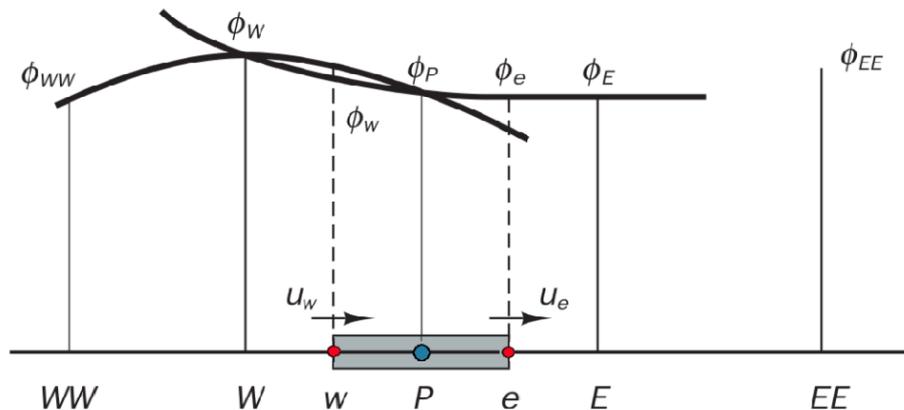


Figura 1. Esquema de um único volume de controle e seus respectivos vizinhos

curiosidade em compreender como esse método funciona em um contexto específico de simulação de fluidos em um modelo bidimensional motivou a equipe a explorar suas características e integrá-lo à metodologia de otimização. Ao deparar-se com desafios relacionados ao tempo de processamento significativo, a equipe viu no *Quick Scheme* uma oportunidade de não apenas superar essa limitação, mas também de entender como o método poderia ser adaptado de maneira eficaz ao programa em questão.

3. Metodologia

A otimização de programas de simulação demanda não apenas a implementação de técnicas eficazes, como o *Quick Scheme*, mas também exige um compromisso contínuo com a manutenção do algoritmo. Reconhece-se a importância de garantir que as melhorias realizadas no código-fonte fossem sustentáveis ao longo do tempo, adaptando-se às mudanças nas necessidades das simulações e nos recursos computacionais disponíveis.

Durante a fase de manutenção do algoritmo, foram implementadas diversas práticas para assegurar a eficiência e estabilidade do programa. Isso incluiu:

- a correção de trechos ineficientes;
- a eliminação de variáveis desnecessárias; e
- a reescrita de partes críticas do código.

O processo de manutenção também envolveu a adaptação do algoritmo para garantir compatibilidade contínua com as evoluções tecnológicas e requisitos específicos do projeto. Foi adotado uma abordagem proativa, realizando testes e validações regulares para certificar-se de que as alterações realizadas durante a manutenção não comprometem a precisão dos resultados ou a estabilidade do programa. Esse compromisso contínuo com a qualidade assegurou que o algoritmo otimizado não apenas atendesse aos objetivos imediatos, mas também permanecesse eficiente ao longo do tempo.

A manutenção do algoritmo não se restringiu apenas às questões técnicas. A documentação adequada, como comentários claros e explicativos incorporados ao código, foi uma prática essencial. Isso facilitou a compreensão do algoritmo por membros da equipe, garantindo uma comunicação eficaz sobre as melhorias implementadas.

Correção de Trechos de Código Ineficientes: Durante a análise minuciosa do código, identificaram-se trechos que apresentavam ineficiências, prejudicando o desempenho geral do programa. Essas ineficiências poderiam se manifestar em operações desnecessariamente complexas, laços mal otimizados ou uso inadequado de estruturas de dados. A correção desses trechos ineficientes envolveu uma abordagem cuidadosa para reescrever e otimizar o código, visando melhorar a eficiência computacional.

Eliminação de Variáveis Desnecessárias: Outro aspecto abordado durante a otimização foi a identificação e eliminação de variáveis desnecessárias. Essas variáveis, muitas vezes remanescentes de versões anteriores do código ou introduzidas durante o desenvolvimento, contribuem para o consumo excessivo de memória e podem impactar negativamente o desempenho geral. A equipe adotou uma abordagem rigorosa para avaliar a utilidade de cada variável, removendo aquelas que não desempenhavam um papel essencial no programa. Essa prática não apenas economizou recursos computacionais, mas também simplificou o código, tornando-o mais compreensível e fácil de manter.

Reescrita de Partes Críticas do Código: Certas partes do código foram identificadas como críticas para o desempenho global do programa. Podem ser algoritmos específicos, estruturas de controle de fluxo ou cálculos intensivos. A equipe optou por uma abordagem proativa, reescrevendo essas seções críticas utilizando algoritmos otimizados e técnicas de programação de alto desempenho. Essa reescrita não apenas melhorou a eficiência computacional, mas também contribuiu para a estabilidade do programa, reduzindo a probabilidade de erros e melhorando a confiabilidade dos resultados obtidos.

Definição de Comentários Claros e Explicativos Incorporados ao Código: Foram incorporados comentários claros e explicativos em áreas-chave do programa. Os comentários documentaram as decisões de otimização tomadas pela equipe e explicaram a lógica por trás de certas implementações. Isso facilitou a colaboração entre os membros da equipe e proporcionou uma base sólida para futuras modificações ou expansão do programa. Comentários bem elaborados são cruciais para garantir a transferência eficiente de conhecimento e a manutenção sustentável do código ao longo do tempo.

4. Resultados

Com a otimização parcial já implementada, constatou-se um aprimoramento no desempenho computacional, reduzindo o tempo de execução do programa em aproximadamente um minuto, levando em conta que o tempo de execução total já otimizado foi de 3 minutos, aproximadamente. Este ganho temporal vem sendo progressivamente otimizado e registrado em um arquivo de notas, modificado pelo próprio programa. Além disso, foram identificadas e corrigidas redundâncias, melhorando consideravelmente a legibilidade e a modularidade do código-fonte. Ademais, foram adotadas técnicas avançadas de otimização de desempenho, contribuindo para uma estrutura mais eficaz e facilitando drasticamente a manutenção do programa a longo prazo.

A implementação do *Quick Scheme*, embora tenha trazido benefícios significativos à otimização do programa de simulação de escoamento de fluidos, também apresentou desafios específicos durante o processo. É crucial destacar a transparência ao abordar potenciais problemas que a equipe enfrentou ao integrar essa técnica numérica. Alguns dos desafios possíveis incluem:

Convergência Numérica - Métodos numéricos, incluindo o *Quick Scheme*, po-

dem enfrentar desafios de convergência em determinados cenários. A sensibilidade às condições iniciais ou às propriedades do domínio pode afetar a estabilidade da solução. Dedicou-se esforços para ajustar parâmetros do *Quick Scheme* e implementar estratégias adicionais para garantir a convergência em uma ampla gama de condições.

Adaptabilidade a Domínios Complexos - O *Quick Scheme* pode se comportar de maneira variável ao lidar com domínios de simulação extremamente complexos ou não convencionais. É necessário realizar testes extensivos para avaliar a eficácia do *Quick Scheme* em diferentes cenários, adaptando-o conforme necessário ou explorando técnicas suplementares para lidar com complexidades específicas.

Impacto nas Métricas de Precisão - O uso do *Quick Scheme* pode impactar a precisão dos resultados obtidos em comparação com métodos tradicionais, especialmente em situações limite. Realizaram-se análises comparativas entre os resultados do *Quick Scheme* e métodos de referência, ajustando parâmetros ou aplicando correções para garantir a precisão adequada. A implementação do *Quick Scheme* no processo de otimização mostrou-se eficaz, proporcionando resultados mais rápidos e precisos nas simulações.

Sobrecusto Computacional Adicional - A implementação do *Quick Scheme* pode introduzir algum sobrecusto computacional adicional, dependendo da complexidade do código ou da estrutura do programa. Adotaram-se estratégias para mitigar qualquer sobrecarga, otimizando algoritmos ou distribuindo tarefas de maneira eficiente.

Compatibilidade com Atualizações Futuras - Mudanças no código base ou atualizações futuras do *Quick Scheme* podem introduzir incompatibilidades ou exigir ajustes substanciais. Está se fazendo o planejamento de medidas para garantir a compatibilidade contínua, como a implementação de boas práticas de versionamento ou a adoção de padrões que facilitem a integração de atualizações.

5. Considerações Finais

Ao adotar o método *Quick Scheme*, os pesquisadores não apenas buscaram uma solução técnica, mas também se aventuraram em uma abordagem inovadora para acelerar a convergência de soluções numéricas. A análise minuciosa do código-fonte resultou na eliminação de trechos ineficientes, na remoção de variáveis desnecessárias e na reescrita de seções críticas para aprimorar a eficiência computacional. Essas otimizações não apenas reduziram o tempo de processamento, mas também melhoraram a estabilidade do programa, proporcionando uma base sólida para avanços futuros.

Referências

- Chichinelli, S. C. (2022). Solução Numérica de um Escoamento Bidimensional Entre Placas Paralelas com Obstáculo. Dissertação de mestrado, Mestrado em Engenharia - Universidade Federal do Pampa, Campus Alegrete.
- de Oliveira, D. P. (2020). Fluid Flow Through Porous Media with the One Domain Approach: A Simple Model for Grains Drying. Dissertação de mestrado, Universidade Federal do Pampa.
- Lucca, N. (2022). Avaliação de estratégias de paralelismo em simulação de meios porosos. Dissertação de mestrado, Mestrado Profissional em Engenharia de Software - Universidade Federal do Pampa, Campus Alegrete.