

Métodos Multigrid Paralelos em Malhas Não Estruturadas Aplicados à Simulação de Problemas de Dinâmica de Fluidos Computacional e Transferência de Calor

Guilherme Galante, Rogério L. Rizzi
Centro de Ciências Exatas e Tecnológicas, UNIOESTE, Campus de Cascavel
Rua Universitária, 2069, 85801-110, Cascavel, PR, Brasil
gui.galante@gmail.com, rogerio@unioeste.br

Tiarajú A. Diverio
Instituto de Informática, UFRGS
CP 15064, 91501-970, Porto Alegre, RS, Brasil
diverio@inf.ufrgs.br

Resumo

Este trabalho apresenta dois métodos paralelos aplicados à solução dos problemas de dinâmica de fluidos e transferência de calor. Nos métodos propostos, a solução é obtida utilizando-se métodos multigrid paralelos por decomposição de domínio, mais especificamente pelos métodos aditivo de Schwarz e complemento de Schur. Através dos experimentos, pode-se notar que as implementações propostas são computacionalmente eficientes e escaláveis.

1. Introdução

Fenômenos naturais, tecnológicos e industriais podem ser modelados de modo acurado através de equações diferenciais parciais (EDPs), definidas sobre domínios contínuos que necessitam ser discretizados para serem resolvidos computacionalmente. Dependendo do esquema de discretização utilizado, pode-se gerar sistemas de equações lineares que são, de modo geral, esparsos e de grande porte. Levando em consideração essas características, o emprego de métodos iterativos é o mais apropriado para a resolução dos sistemas gerados, devido principalmente a sua potencialidade quanto à otimização de armazenamento e eficiência computacional.

Uma forma de incrementar ainda mais o desempenho dos métodos iterativos é empregar multigrid. Basicamente, os métodos multigrid constroem a solução utilizando uma

seqüência de malhas de diferentes refinamentos, onde se resolve o problema na malha mais refinada empregando as demais malhas como esquemas de correção.

Experimentos numéricos mostram que estes métodos são muito eficientes e podem ser aplicados com sucesso a uma ampla classe de problemas de computação científica [2]. A bibliografia sobre o assunto mostra que o método é bastante geral e sua eficiência não é restrita ao tipo da malha (estruturada, não-estruturada), discretização utilizada (elementos finitos, diferenças finitas, volumes finitos) ou do tipo do sistema de equações obtido da discretização (simétrico, não simétrico) [17], [5].

A solução de problemas de simulação, com alta qualidade numérica, pode requerer grande capacidade de processamento e de armazenamento, o que torna imprescindível o uso de ambientes computacionais de alto desempenho. Sob tais ambientes, simulações computacionais podem ser realizadas com um nível de detalhe que não seria viável em abordagens computacionais seqüenciais [12].

Existem, pelo menos, duas grandes abordagens para a resolução de sistemas de equações em paralelo. Em uma delas, chamada de decomposição de dados, gera-se um único sistema de equações para todo o domínio que é resolvido através de um método numérico paralelizado. A segunda abordagem consiste na utilização de métodos de decomposição de domínio (MDD). Os MDDs são baseados no particionamento do domínio computacional em subdomínios, de modo que a solução global do problema é obtida pela combinação apropriada das soluções obtidas em cada um dos subdomínios [6].

Neste trabalho, utiliza-se métodos de decomposição de

domínios como abordagem de resolução de sistemas de equações em paralelo. Esta escolha é baseada nos resultados obtidos em Galante *et al.* [4], onde essa abordagem mostrou melhores resultados que a paralelização dos métodos numéricos. Além disso, a literatura técnica mostra que esta é a melhor abordagem para a paralelização de problemas que envolvem a discretização de um domínio físico [13, 16].

Sob o escopo apresentado, foram desenvolvidas implementações de métodos multigrid paralelizados pela abordagem de decomposição de domínio para a resolução paralela dos sistemas de equações gerados pela discretização das equações da difusão do calor e da hidrodinâmica. Assim, o objetivo deste trabalho é mostrar todos os passos da construção destes métodos de solução, desde a geração das malhas até a resolução dos sistemas de equações utilizando métodos multigrid paralelos.

Este trabalho está dividido em quatro atividades apresentadas nas próximas seções deste trabalho e sintetizadas na Figura 1.

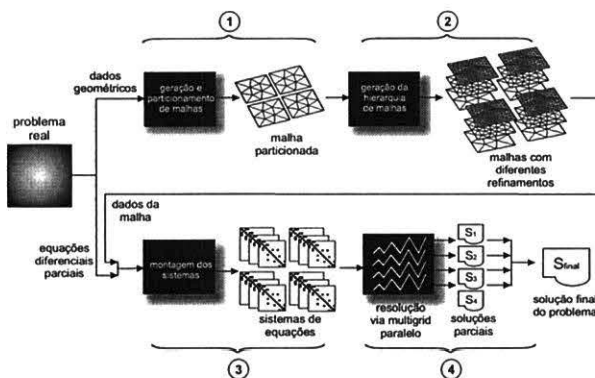


Figura 1. Visão geral das etapas de solução

2 Geração e Particionamento de Malhas

A geração e particionamento das malhas é a atividade que serve como base ao restante do trabalho. É nesta fase que as malhas são geradas e posteriormente particionadas por um software específico. Neste trabalho são utilizados dois pacotes para a geração de malhas não estruturadas 2D, o *Triangle* [15] e o *Easymesh* [9].

Para o particionamento das malhas emprega-se o pacote METIS. Como entrada de dados utiliza-se as informações sobre a conectividade dos triângulos geradas pelos softwares de geração de malhas. Como saída, tem-se uma lista das respectivas partições de cada triângulo. É importante ressaltar que nesta primeira fase apenas a malha mais grosseira é gerada, servindo como ponto de partida para a geração da hierarquia de malhas.

3 Criação da Hierarquia de Malhas

A criação da hierarquia de malhas é um passo importante na solução de problemas utilizando métodos multigrid. Existem alguns pacotes disponíveis que geram de modo automático os diversos níveis de malha, como é o caso do MGridGen/ParMGridGen. Estes pacotes partem de uma malha mais refinada e geram as demais malhas grosseiras através de algoritmos multível [8].

Embora esses pacotes gerem de forma eficiente a hierarquia de malhas necessárias optou-se por implementar um gerador próprio para as malhas. Dessa maneira implementou-se um módulo, chamado *MGTool*, capaz de gerar malhas com diferentes refinamentos. Essa escolha de projeto deve-se a dois principais motivos. O primeiro refere-se ao tipo especial de malha utilizado neste trabalho, as malhas não estruturadas ortogonais, não geradas por pacotes como o MGridGen/ParMGridGen. O segundo motivo deve-se as otimizações feitas nas malhas geradas pelo *MGTool*, que faz com que os mecanismos de transferência entre malhas não necessitem de comunicação entre processos, já que o particionamento da malha é mantido para todos os níveis de malha.

O *MGTool* tem como entrada de dados, os arquivos de saída dos geradores de malha e o arquivo de saída do METIS; e como saída produz os diversos níveis de malha desejado, como mostrado na Figura 2.

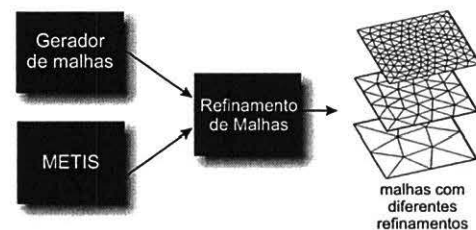


Figura 2. Relacionamento entre as ferramentas de geração e particionamento e o módulo de refinamento de malhas

Para o refinamento das malhas adotou-se o uma estratégia conhecida na literatura como *h-refinement*, caracterizada pela subdivisão dos elementos do domínio [7]. No módulo implementado, dada uma malha inicial, gera-se as demais malhas através da subdivisão sucessiva dos elementos em quatro subelementos.

Após a geração da hierarquia das malhas, deve-se relacionar cada nível com os níveis adjacentes. Nessa fase cria-se para cada elemento da malha uma relação dos triângulos para os quais deve receber informações, e para os quais deve enviar. Um exemplo é mostrado na Figura 3, onde o triângulo 1A é dividido em outros 4 triângulos, 2A, 2B, 2C e 2D. Na notação utilizada neste trabalho, o triângulo

1A é o “pai” dos triângulos 2A, 2B, 2C e 2D, e por sua vez estes triângulos são chamados de “filhos” do triângulo 1A.

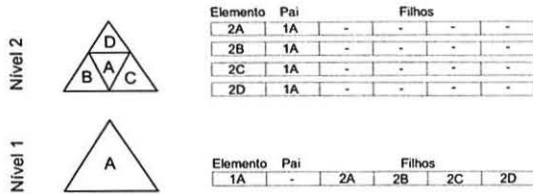


Figura 3. Exemplo de hierarquia de dois níveis de malha. Os números representam o nível e as letras identificam o triângulo na malha. No lado direito da figura, a tabela descreve o relacionamento entre os níveis adjacentes de malha.

Esta relação “pai-filho” é utilizada pelos procedimentos de interpolação e restrição presentes nos métodos multigrid, conforme descrito na Seção 5.1.

4 Montagem dos Sistemas de Equações

O *MGTool*, além de montar a hierarquia de malhas e as relações entre níveis adjacentes, é também responsável pela montagem das matrizes para cada nível de malha. Em particular, os sistemas são gerados a partir da discretização de duas EDPs: a EDP da difusão de calor e a EDP do cálculo do nível da superfície livre da hidrodinâmica do modelo UnHIDRA.

A difusão de calor é a movimentação de energia que ocorre devido à diferença entre temperaturas. O calor sempre flui das regiões de maior temperatura para regiões de temperatura inferior [10]. Assim, este estudo de caso consiste em determinar como o calor se propaga em um corpo, dado uma condição inicial e as condições de contorno.

No estudo de caso do modelo de hidrodinâmica, utiliza-se uma simplificação do modelo de hidrodinâmica do UnHIDRA, onde objetiva-se calcular a variação do nível da água. O modelo matemático completo empregado é chamado de equações *shallow water* (ESW) [12].

Dependendo do tipo de método de solução a ser empregado, existe um tratamento diferente para a malha. Quando se emprega o método aditivo de Schwarz, o módulo é responsável pela criação das áreas de sobreposição requeridas por este método. Caso utilize-se o método do complemento de Schur, o módulo deve renumerar os elementos de acordo com o especificado para o método.

A geração das matrizes é feita de forma distribuída, onde cada processo gera as matrizes de acordo com o subdomínio que lhe foi atribuído. Na geração das matrizes para o aditivo de Schwarz, gera-se a matriz estendida, ou seja, con-

siderando também as células de sobreposição. No caso do método do complemento de Schur, gera-se as submatrizes obedecendo os esquemas de numeração especial das células da malha. Para os dois casos gera-se também as estruturas utilizadas na comunicação entre processos.

5 Solução Paralela

Os sistemas de equações lineares estão entre os mais frequentes problemas que devem ser tratados pela computação científica. Por se tratarem geralmente de sistemas de grande porte, uma alternativa viável de se obter a solução em tempo útil é empregar métodos de resolução paralelizados, empregando *clusters* como ambiente computacional.

Nesta seção apresenta-se o método proposto para a resolução de sistemas de equações em paralelo. Uma visão geral do processo de resolução é mostrado na Figura 4.

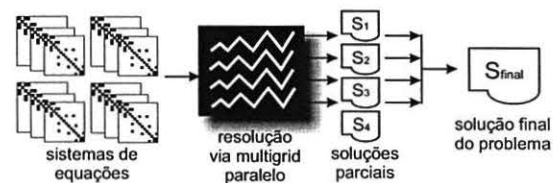


Figura 4. Visão geral da resolução dos sistemas de equações através do multigrid paralelo

Antes de abordar a solução paralela apresenta-se alguns aspectos importantes dos métodos multigrid e dos métodos de decomposição de domínio.

5.1 Métodos Multigrid

Multigrid são uma classe de métodos que resolvem eficientemente sistemas de equações através da aceleração da convergência de métodos iterativos. Basicamente, os métodos multigrid consideram uma seqüência de malhas M_1, M_2, \dots, M_n para a solução do sistema de equações.

Objetiva-se resolver o problema que é definido na malha mais refinada, empregando as demais malhas como esquemas de correção [18].

Os métodos multigrid baseiam-se em três elementos centrais: transferência de informações entre malhas, iterações aninhadas, e correção em malha grossa.

Para que se possa utilizar os diversos níveis de malha na solução de um determinado problema, define-se dois operadores para transferência de informações ente malhas. O primeiro operador transfere informações da malha grossa M_n para a malha mais fina M_{n-1} , sendo denominado operador de *interpolação*, definido por:

$$I_{M_n}^{M_{n-1}} : M_n \rightarrow M_{n-1}$$

O segundo operador, definido por:

$$I_{M_{n-1}}^{M_n} : M_{n-1} \rightarrow M_n$$

transfere informações da malha mais fina para a mais grossa, e é conhecido como operador de *restrição*.

A forma destes operadores pode variar com o tipo de elemento ou com o tipo de problema. Diversas formas de interpolação e restrição podem ser encontradas na literatura [2, 17]. O operador de restrição $I_{M_{fina}}^{M_{grossa}}$ utilizado neste trabalho é baseado no operador *full-weighted* descrito em Trottenberg *et al.* (2001) e Wesseling (1992). O operador de restrição é dado por:

$$x_i = \frac{1}{4}x_{filho_1(i)} + \frac{1}{4}x_{filho_2(i)} + \frac{1}{4}x_{filho_3(i)} + \frac{1}{4}x_{filho_4(i)}$$

O operador de interpolação $I_{M_{grossa}}^{M_{fina}}$ é definido por:

$$x_i = \frac{1}{4}(x_{pai(i)} + \sum_{j=1}^3 x_{pai(i_j)})$$

A partir da definição dos operadores de transferência, pode-se definir a estratégia de iterações aninhadas (*nested iteration*). O objetivo é encontrar uma melhor aproximação inicial para a solução através de iterações em malhas mais grossas. Como o sistema gerado a partir da malha mais grossa possui um número menor de incógnitas, o custo computacional de sua resolução é menor, em comparação à resolução do sistema gerado a partir da malha mais refinada. Logo, pode-se obter uma aproximação inicial para $Ax = b$ empregando as malhas mais grossas [14], como mostrado na Figura 5.

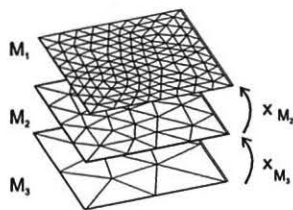


Figura 5. Iterações aninhadas

O processo é iniciado no nível mais grosseiro da malha (M_3), onde é encontrada uma solução inicial para x . Esta solução é então transferida para um nível acima através do operador de interpolação. Esse processo repete-se até que o nível mais refinado da malha (M_1) seja alcançado, e por consequência, uma boa aproximação para x .

Esta estratégia não garante que ao final, a solução em M_1 não contenha componentes de erro suaves (baixa frequência). O emprego da correção do erro proveniente das malhas grossas evita esta limitação [1]. Tendo iterado na malha fina até que os erros tenham sido removidos, itera-se a equação residual na malha grossa para obter uma aproximação do erro. Então retorna-se para a malha fina, onde a primeira aproximação obtida é corrigida.

Utilizando-se de combinações entre a estratégia de correção de malha grossa e o uso de iterações aninhadas, pode-se definir uma família de métodos multigrid [18][2]. Das distintas combinações é possível gerar diferentes algoritmos que são conhecidos na literatura técnica por ciclo V, ciclo W, ciclo V completo (FMV) e ciclo W completo. Uma síntese dessas diferentes abordagens podem ser encontradas em [17]. Neste trabalho é utilizado o algoritmo FMV.

A estratégia FMV inicia na malha mais grossa, para a obtenção de uma solução inicial com baixo custo computacional para os níveis superiores. Então, o número de níveis é incrementado em um. Este processo é repetido até que todos os níveis estejam envolvidos.

5.2 Métodos de Decomposição de Domínio

Os métodos de decomposição de domínio (MDD) designam um conjunto de técnicas e métodos matemáticos, numéricos e computacionais para resolver problemas em computadores paralelos [3]. Um MDD é caracterizado pela divisão do domínio computacional em diversos subdomínios. A solução global do problema é obtida através da combinação dos subproblemas que são resolvidos localmente. Abordagens paralelas via decomposição de domínio baseiam-se no fato de que cada processador pode fazer grande parte do trabalho de forma independente [13]. E, uma vez que os subdomínios podem ser tratados independentemente, tais métodos são atrativos para ambientes de memória distribuída.

Os MDDs podem ainda ser divididos em duas classes: os métodos de Schwarz, onde os subdomínios apresentam uma região de sobreposição, que pode variar de acordo com o tipo de aproximação empregada para resolver os modelos matemáticos já discretizados, e os métodos de Schur, onde os subdomínios não apresentam região de sobreposição. Neste trabalho utiliza-se um método com sobreposição, o método aditivo de Schwarz, e um método sem sobreposição, o método do complemento de Schur [6][3].

5.3 Resolução Paralela

O primeiro passo da resolução consiste na leitura dos arquivos de entrada contendo os sistemas de equações. Cada arquivo contém informações que pertencem a um sub-

domínio específico, e por consequência, a um processo. Após a leitura de seus respectivos arquivos de entrada, cada processo fica responsável por calcular independentemente a solução do sistema de equações referente ao seu conjunto de dados. Ao término da resolução, cada processo cria um arquivo de saída contendo a sua parte da solução. Estas diversas soluções obtidas podem então ser reunidas para formar a solução global do sistema.

Conforme apresentado na Seção 5.1, os métodos multigrid são baseados em três etapas centrais: transferência de informações entre malhas, iterações aninhadas e correção em malha grossa. Para a construção destas etapas, considera-se basicamente três elementos básicos que compreendem a *transferência de informações* (restrição e interpolação), o *cálculo do resíduo* da solução e a *resolução de sistemas de equações*.

Dessa forma, a obtenção do multigrid paralelo se dá através da execução de cada uma destas rotinas paralelizadas. As implementações foram feitas em linguagem C, utilizando a biblioteca de trocas de mensagens MPICH 1.2.7. A paralelização dos elementos apresentada ao longo do restante do capítulo. O esquema da solução proposta se encontra na Figura 6.

Na transferência de informações entre os níveis das malhas, cada processo é responsável por suas trocas locais. Note que da forma como os operadores foram definidos (seção 5.1), não são necessárias as contribuições recebidas de outros subdomínios, considerando apenas as contribuições locais. Esta decisão de desconsiderar os elementos de outros subdomínios evita a necessidade de comunicação entre processos.

O cálculo do resíduo é utilizado no multigrid no procedimento de correção em malha grossa. O cálculo do resíduo é dado por:

$$r = b - Ax'$$

onde r é o vetor de resíduo, b é o vetor dos termos independentes, A é a matriz de coeficientes e x' é uma aproximação da solução. Logo se tem duas operações a serem efetuadas, um produto matriz por vetor e uma subtração de vetores:

$$r = b - \underbrace{\underbrace{Ax'}_{\text{matriz-vetor}}}_{\text{subvetor}}$$

Portanto, a paralelização do cálculo do resíduo consiste em executar estas operações de tal maneira que cada processo trabalhe sob seu conjunto de dados. Na subtração de vetores são utilizados apenas dados locais, já na multiplicação matriz-vetor é necessário comunicação entre os processos para que troquem os dados necessários.

Para resolver os sistemas de equações de modo paralelo pode-se optar pela resolução dos sistemas pelo método aditivo de Schwarz ou pelo método do complemento de Schur.

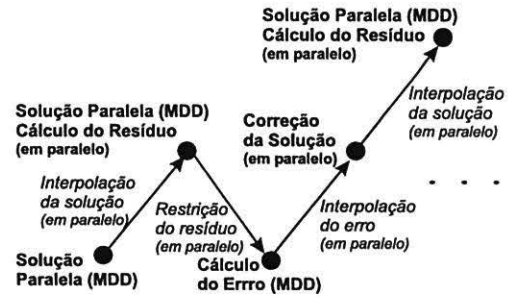


Figura 6. Solução paralela via multigrid paralelo

6 Estudos de Caso

6.1 Estudo de Caso 1: Transferência de Calor

Um problema clássico de aplicação de métodos numéricos é a transferência de calor em uma placa plana. O processo de transferência de calor em uma placa retangular ocorre pela troca de calor entre partículas do material de um ponto com mais energia para outro com menos. Esse processo é conhecido como condução do calor.

Considerando que todos os pontos da placa estejam a uma temperatura inicial T_0 e sendo esta temperatura diferente das temperaturas das bordas, o problema que se coloca é determinar a temperatura em qualquer ponto interno da placa em um dado instante de tempo. Neste experimento utilizou-se um quadrado unitário com temperaturas $T_1 = 10^\circ\text{C}$ e $T_2 = T_3 = T_4 = 0^\circ\text{C}$.

Para a realização dos testes empregando multigrid, utilizou-se 4 níveis de malha, com 1337, 5348, 21392 e 85568 triângulos, respectivamente. Nos demais testes, sem o emprego do multigrid, utilizou-se apenas a malha mais refinada, que é a malha onde procura-se a solução do problema. Na Figura 7 encontra-se a malha com 1337 elementos.

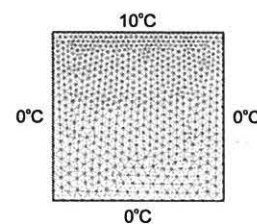


Figura 7. Malha inicial com 1337 triângulos

Na Figura 8, são mostrados alguns passos da resolução do problema utilizando 5 processos. Apresenta-se a solução

no primeiro passo de tempo, após 5 passos e após 20 passos de tempo.

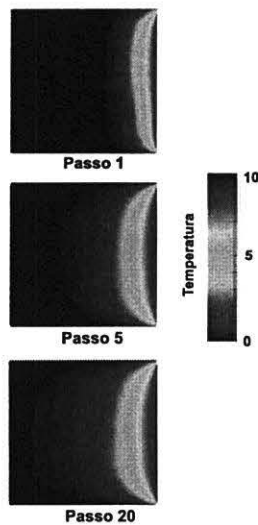


Figura 8. Passos da resolução do problema de transferência de calor

6.2 Estudo de Caso 2: Hidrodinâmica

Nos testes da hidrodinâmica utilizou-se como domínio computacional o Lago Guaíba. O Guaíba é um corpo de água doce que se situa entre o chamado delta do Jacuí e a Lagoa dos Patos. Banha toda a região metropolitana de Porto Alegre e tem uma área total de aproximadamente 468km^2 e profundidade média $4,0\text{m}$. A opção por utilizar o Lago Guaíba como estudo de caso deve-se ao fato de que existem alguns dados da geometria e de parâmetros físicos para definir as condições iniciais e de contorno do problema. Além disso, outros trabalhos do grupo, como por exemplo Rizzi [12], Picinin [11] e Martinotto [6], já utilizaram este domínio como estudo de caso, sendo possível o aproveitamento de dados para as simulações.

Neste estudo de caso utilizou-se 4 níveis de malha, com 2818, 11272, 45088 e 180352 triângulos, respectivamente. A malha mais grosseira é mostrada na Figura 9.

7 Resultados

Nesta seção apresenta-se os resultados obtidos com as paralelizações desenvolvidas neste trabalho. Inicialmente são apresentados os resultados obtidos com a resolução dos sistemas originados no problema de transferência de calor e em seguida são apresentados os resultados obtidos com a resolução dos sistemas da hidrodinâmica.

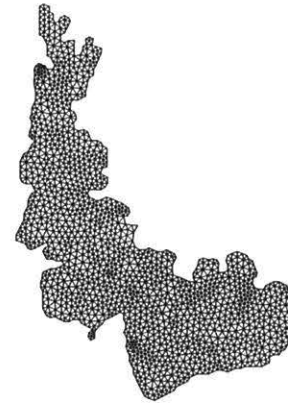


Figura 9. Guaíba: malha inicial com 2818 triângulos

7.1 Transferência de Calor

Os testes do problema de transferência de calor foram efetuados utilizando 19 nodos do cluster labtec do Instituto de Informática da UFRGS. Utilizou-se dois processos por nodo, de modo a aproveitar a característica dos nodos de serem biprocessados.

Nos testes de desempenho foram feitas 20 execuções de cada implementação e o tempo considerado foi a média aritmética dessas. Dos tempos coletados, o maior e o menor valores foram descartados. Resultados anômalos não foram considerados. Foram simulados 20 passos de tempo utilizando os dois diferentes MDDs.

Observando a Figura 10 pode-se comparar o tempo de execução e o speedup dos diversos métodos na resolução do problema de transferência de calor.

Utilizando o método Multigrid-aditivo os ganhos de desempenho foram representativos com o uso de até 22 processos, onde obteve-se 18,94 de *speedup*. A partir disso há ganhos de desempenho, porém menos significativos, tendo como ganho máximo de desempenho de 26,08 com 36 processos. Esse comportamento deve-se ao aumento da quantidade de comunicação e pela redução do processamento necessário para cada subdomínio, já que os domínios possuem tamanho reduzido quando o número de partições é alto. Além disso, com o aumento da quantidade de processos, cada subdomínio passa a ter mais vizinhos, o que causa o aumento das áreas de sobreposição, e por conseqüência o aumento dos sistemas locais.

Nos testes feitos com o método Multigrid-Schur o ponto de máxima eficiência alcançado pelo algoritmo foi obtido quando executado utilizando 20 processos, onde obteve-se ganhos de desempenho na ordem de 16,11 vezes, com 79% de eficiência. O ganho máximo de desempenho do algoritmo foi de 27,07 vezes, utilizando 36 processos, com 75%

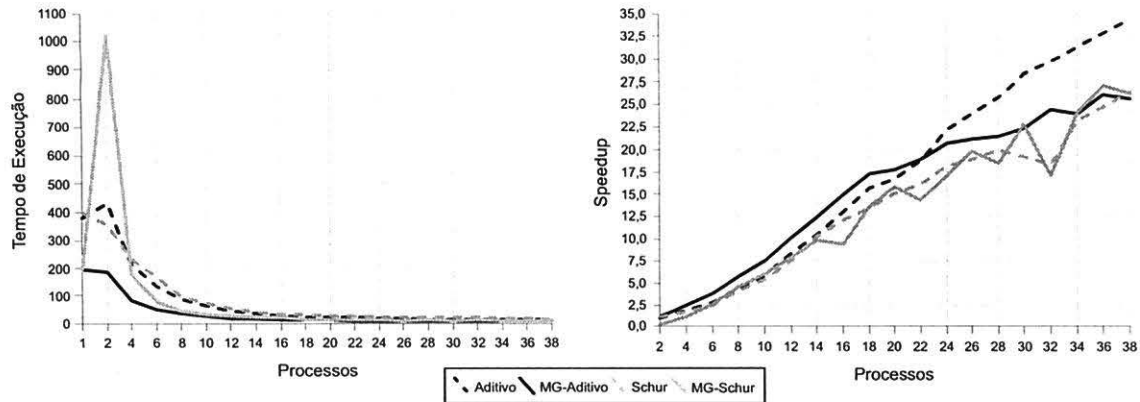


Figura 10. Transferência de Calor: Tempo de Execução e Speedup

de eficiência. No gráfico apresentado pode-se observar a presença de um pico na execução com 2 processos. Este pico é causado pela resolução dos sistemas de interface, que envolve o cálculo de aproximações de inversas que é bastante custoso.

A eficácia dos métodos multigrid na aceleração dos métodos iterativos pode ser confirmada. Em média, a implementação utilizando multigrid foi 2 vezes mais rápida que a implementação que não emprega multigrid.

Pode-se notar ainda que para este problema os tempos obtidos pelo método multigrid-aditivo são muito inferiores ao outro método comparado. Sendo assim, o método multigrid-aditivo mostrou-se a melhor opção para a resolução deste problema, dentre os métodos testados.

7.2 Hidrodinâmica

Os testes do problema de hidrodinâmica foram efetuados utilizando os 18 nodos do cluster Krusty, da Universidade Estadual do Oeste do Paraná.

A Figura 11 mostra, o tempo de execução e o *speedup* obtido na resolução do problema da hidrodinâmica, utilizando o método de solução que combina multigrid com os métodos aditivo de Schwarz e complemento de Schur, respectivamente. Da mesma forma como foi feito no problema da transferência de calor, compara-se os métodos com e sem o uso do multigrid.

Nesta aplicação, os resultados obtidos foram muito equivalentes, sendo qualquer um deles uma boa opção para a resolução do problema de hidrodinâmica. A escolha por um determinado método fica a cargo da qualidade numérica desejada, e das demais questões relacionadas à ele, como por exemplo geração do sistema de equações, que é mais complexa para os métodos de Schur do que para os métodos de Schwarz.

Na comparação entre os métodos com e sem o emprego de multigrid, pode-se notar, que para esta aplicação o ga-

nho foi menos significativo que o obtido no problema da transferência de calor. Em média, a implementação utilizando multigrid foi 40% mais rápida que a implementação que não emprega multigrid.

8 Conclusão

O principal objetivo deste trabalho foi o desenvolvimento e a avaliação de desempenho das soluções paralelas obtidas através da combinação de métodos multigrid e métodos de decomposição de domínio (MG+MDD). Mais especificamente combinou-se o método Full Multigrid V com dois métodos de decomposição de domínio: o método aditivo de Schwarz e o método do complemento de Schur.

Além dos métodos de solução, desenvolveu-se também toda uma estrutura de suporte para a resolução utilizando os métodos MG+MDD. Para as tarefas de geração da hierarquia de malhas e montagem dos sistemas de equações foi implementado um módulo chamado de *MG-Tool* responsável pela geração da hierarquia de malhas e pela montagem dos sistemas de equações. Através do *MG-Tool* foram geradas as malhas para os estudos de caso, bem como os sistemas de equações tanto para o método aditivo de Schwarz como para o método do complemento de Schur.

De acordo com os resultados observados, os métodos implementados mostraram-se altamente paralelizáveis, apresentando bons ganhos de desempenho. Pode-se observar que os métodos multigrid mostraram-se eficiente na aceleração dos métodos iterativos, já que métodos que utilizaram esta técnica apresentaram desempenho superior aos métodos que não utilizaram nenhum método de aceleração. Para o problema de transferência de calor, os métodos empregando multigrid convergiram em média duas vezes mais rápido. Já no problema de hidrodinâmica, os ganhos foram menos significativos, mostrando convergência em média 40% mais rápida.

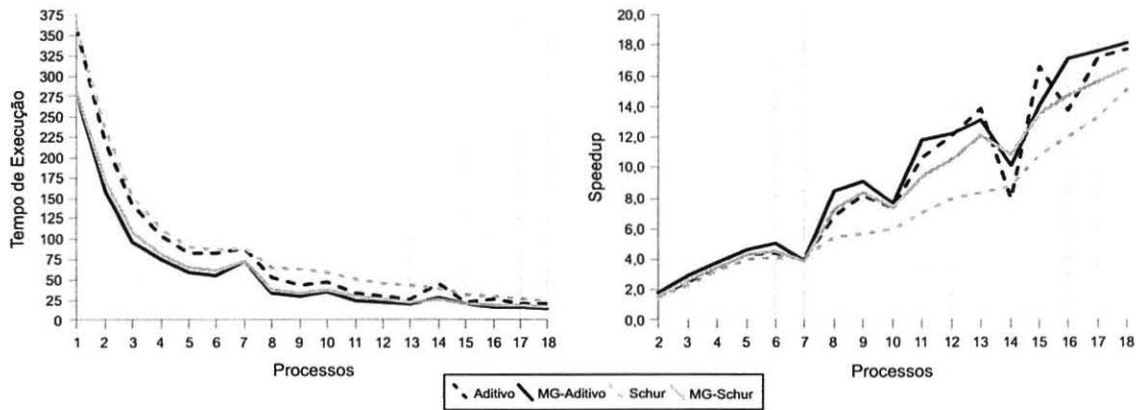


Figura 11. Hidrodinâmica: Tempo de Execução e Speedup

A avaliação do método multigrid-aditivo mostrou bons resultados na solução dos dois problemas tratados mostrando ser a melhor opção de solução entre os métodos implementados.

Como principal conclusão tem-se que a combinação de métodos multigrid e decomposição de domínios mostraram-se uma boa opção na solução de sistemas de equações provenientes da discretização de equações diferenciais parciais. Embora a solução por estes métodos possua algumas fases bastante complexas, como a geração da hierarquia de malhas e a geração dos sistemas de equações para os diversos níveis da malha, os ganhos de desempenho justificam estas questões.

Referências

- [1] M. L. Bittencourt. *Métodos Iterativos e Multigrid Adaptáveis em Malhas Não Estruturadas*. Tese (doutorado em engenharia mecânica), Faculdade de Engenharia Mecânica, UNICAMP, Campinas-SP, 1996.
- [2] W. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, 1987.
- [3] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–143. Cambridge University Press, 1994.
- [4] G. Galante, T. Divério, R. Rizzi, T. Martinotto, R. Dorneles, and D. Picinin. Comparação entre métodos de decomposição de domínio e decomposição de dados na solução de sistemas de equações. In *Anais do 5. Workshop em Sistemas Computacionais de Alto Desempenho*, pages 98–104, Foz do Iguaçu, 2004. Foz do Iguaçu, SBC.
- [5] R. Hornung and J. Trangenstein. Adaptive Mesh Refinement and Multilevel Iteration for Flow in Porous Media. *Journal of Computational Physics*, 136:522–545, 1997.
- [6] A. L. Martinotto. Resolução de Sistemas de Equações Lineares através de Métodos de Decomposição de Domínio. Dissertação de mestrado (mestrado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, 2004.
- [7] D. J. Mavriplis. Mesh generation and adaptivity for complex geometries and flows. In R. Peyret, editor, *Handbook of Computational Fluid Mechanics*. Academic, London, 1996.
- [8] I. Moulitsas and G. Karypis. Mgridgen/Parmgridgen Serial/Parallel Library for Generating Coarse Grids for Multigrid Methods. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, 2001.
- [9] B. Niceno. Easymesh: A two-dimensional quality mesh generator, 2005. <http://www.dinma.univ.trieste.it/nirftc/research/easymesh/>.
- [10] K. Paar, P. M. Athanas, and C. M. Edwards. Implementation of a finite difference method on a custom computing platform. In *Proc. High-Speed Computing, Digital Signal Processing, and Filtering Using Reconfigurable Logic*, pages 44–53, Oct. 1996.
- [11] D. Picinin. Paralelização de Métodos de Solução de Sistemas Lineares em Clusters de pcs com as Bibliotecas DECK, MPICH e Pthreads. Dissertação de mestrado (mestrado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, 2002.
- [12] R. L. Rizzi. *Modelo Computacional Paralelo para a Hidrodinâmica e para o Transporte de Massa Bidimensional e Tridimensional*. PhD thesis, Instituto de Informática, UFRGS, Porto Alegre-RS, 2002.
- [13] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
- [14] F. Sartoretto. *Appunti per le Lezioni di Calcolo Numerico*. Technical report, Università degli Studi di Venezia, Venezia, 2005.
- [15] J. R. Shewchuk. Lecture notes on delaunay mesh generation. Technical report, University of California at Berkeley, Berkeley, 1999.
- [16] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, 1996.
- [17] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, Oxford, UK, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.
- [18] P. Wesseling. *Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992.