

PARALELIZAÇÃO DO MÉTODO MULTIESCALA DE DIFERENÇAS FINITAS PARA ESCOAMENTOS EM MEIO POROSO

Alexandre S. Francisco* e Leonardo M. Sacramento

Escola de Engenharia Industrial Metalúrgica de Volta Redonda, Universidade Federal Fluminense, Volta Redonda, 27.255-125, RJ, Brasil

Palavras-chave: Método Multiescala, Função Base, Meios Porosos, OpenMP.

Resumo. Em busca de novos métodos numéricos que obtenham soluções acuradas em menor tempo de execução, desenvolveu-se um simulador com base na decomposição de domínio em malha grossa e geração de funções base em malha fina. Este simulador permite incorporar as informações de malhas finas, na resolução de problemas em malhas grossas. As funções base em malha fina são geradas usando discretização por diferenças finitas nos subdomínios, onde os sistemas de equações lineares são resolvidos pelo método do gradiente conjugado pré-condicionado. No presente trabalho, implementa-se uma paralelização do simulador usando a interface *OpenMP*, e obtém-se soluções em menor tempo de execução. Métricas de desempenho são determinadas com relação ao número de *thread*, considerando tanto o refinamento da malha computacional, quanto à heterogeneidade do meio. Comparação de campos de pressão e velocidade do simulador paralelo com o esquema tradicional de diferenças finitas é feita. O simulador paralelo mostra-se vantajoso em termos em malhas computacionais muito refinadas e em meios altamente heterogêneos.

*Endereço de e-mail: afrancisco@id.uff.br

PARALLELIZATION OF THE MULTISCALE FINITE DIFFERENCE METHOD FOR FLOWS IN POROUS MEDIA

Keywords: Multiscale Method, Basis Function, Porous Media, OpenMP.

Abstract. Seeking new numerical schemes that can achieve accurate solutions in shorter execution times, it was developed a solver based on coarse grid domain decomposition and fine grid basis functions. This solver allows to incorporate the fine grid information into a coarse grid problem. The fine grid basis functions are provided by discretizing the subdomains by a finite difference scheme and solving the resulting system of linear equations by the preconditioned conjugate gradient method. The present work performs the parallelization of the solver through the OpenMP interface to obtain solutions in shorter execution times. Performance metrics are calculated according to the number of threads, considering both the grid refinement and the heterogeneity of the medium. Comparisons of error for pressure and velocity fields between the solver and the traditional finite difference scheme are performed. The parallelized solver has demonstrated to be beneficial for more refined grids and highly heterogeneous media.

1 INTRODUÇÃO

O modelo matemático que descreve o escoamento em meios porosos pode ser dado pela equação diferencial parcial elíptica, conhecida como lei de Darcy. Diversas técnicas numéricas têm sido investigadas com objetivo de resolver o problema do escoamento em meios porosos, de maneira eficiente e acurada [2].

Uma técnica numérica tradicional é o método das diferenças finitas (FDM, do inglês *Finite Difference Method*). Sendo de fácil entendimento, ele é usado como referência em muitos cursos introdutórios de métodos numéricos. Sabe-se, entretanto, que a acurácia do FDM depende da malha ou nível de refinamento da discretização do domínio [4]. Em consequência, resultados acurados são obtidos com maiores custos computacionais.

Nos últimos anos, métodos multiescala vêm viabilizando a diminuição do esforço computacional quando se investigam fenômenos físicos complexos. Vale lembrar que os problemas, tais como a extração de petróleo em reservatórios e a despoluição de aquíferos ocorrem em escalas regionais (da ordem de quilômetros) e, muitas vezes, simulações acuradas exigem tamanhos de espaçamento de malha da ordem de metros. Daí a necessidade de se usar os recursos da computação paralela. Um procedimento iterativo que é bem adaptável para computação paralela é a decomposição de domínio, com a associação de cada subdomínio a um processador específico [5].

Um procedimento iterativo considerando duas escalas no problema é o *Multiscale Mixed Method* (MuMM). Esse método multiescala baseia-se na decomposição do domínio sem sobreposição, no qual o método dos elementos finitos mistos e híbridos é empregado para resolver problemas elípticos de segunda ordem. O uso de funções base reforça o potencial do método multiescala para a programação paralela [6].

Recentemente, desenvolveu-se o *Multiscale Finite Difference Method* (MsFDM) para resolver problemas elípticos de segunda ordem. Ele baseia-se na decomposição de domínio com sobreposição e emprega a discretização por diferenças finitas [7, 8, 10, 14]. A decomposição do domínio permite resolver iterativamente o problema global em malha grossa. Nesse método, conjuntos de funções base são construídos para obter soluções discretas de problemas locais. Esse método faz uso da escala múltipla e dos conjuntos de funções base para aproximar o problema global de forma eficiente.

Neste trabalho, emprega-se a computação paralela para aumentar a eficiência computacional do MsFDM. Na computação paralela têm-se vários dispositivos trabalhando em paralelo, cada um deles resolvendo o problema decomposto em subdomínios que se comunicam em sincronia. Com isto, reduz-se bastante o tempo de execução dos simuladores numéricos. A paralelização do MsFDM é implementada usando-se a interface OpenMP. Métricas de desempenho (speedup e eficiência) são investigadas com relação ao número de *thread*, considerando tanto refinamentos de malha computacional, quanto níveis de heterogeneidade distintos.

Os experimentos numéricos mostram que o MsFDM é um simulador competitivo quando comparado ao tradicional FDM, mesmo na sua versão serial. Consequentemente, indica-se a paralelização do MsFDM. O equilíbrio satisfatório entre acurácia e eficiência é alcançado com a escolha adequada da relação entre tamanhos de malha grossa e malha fina. O seu algoritmo paralelo mostra-se vantajoso em termos de tempo de execução em malhas computacionais muito refinadas e em meios altamente heterogêneos.

2 PROBLEMA GLOBAL

Considera-se uma equação diferencial parcial elíptica que reúne a equação da continuidade para escoamentos monofásicos e incompressíveis, e a lei de Darcy. Nesse problema, a velocidade u e a pressão p são resolvidas satisfazendo o seguinte sistema [12]:

$$\nabla \cdot \mathbf{u} = f \text{ em } \Omega, \quad (1)$$

$$\mathbf{u} = -\mathbf{K}(\mathbf{x})\nabla p \text{ em } \Omega, \quad (2)$$

$$p = p_b \text{ sobre } \Gamma_D \text{ e } \mathbf{u} \cdot \mathbf{n} = 0 \text{ sobre } \Gamma_N, \quad (3)$$

onde Ω é o domínio, Γ_D e Γ_N são fronteiras do domínio, $\mathbf{K}(\mathbf{x})$ é um tensor representando a permeabilidade dividida pela viscosidade, dependente da posição \mathbf{x} , e \mathbf{n} é um vetor unitário normal a Γ_N .

3 SIMULADOR MULTIESCALA

O simulador MsFDM é composto basicamente das seguintes etapas: decomposição do domínio com sobreposição, geração de funções base e um procedimento iterativo. Cada etapa permite que se tire uma vantagem em unidades de processamento paralelo com o objetivo de tornar a resolução mais eficiente [9]

3.1 Decomposição de domínio

Um procedimento comum para se resolver numericamente as equações elípticas de segunda ordem é dividir o domínio em vários subdomínios e resolver as equações nos subdomínios como problemas locais em diferentes processadores paralelos [3]. No presente trabalho, o domínio Ω é dividido em subdomínios Ω_j , $j = 1:m$, onde m é o número de subdomínios. Em cada subdomínio Ω_j , deve-se resolver o seguinte problema local:

$$\nabla \cdot (-\mathbf{K}_j(\mathbf{x})\nabla p_j) = f_j \text{ em } \Omega_j, \quad (4)$$

$$p_j = p_k \text{ sobre } \Gamma_{jk}, \quad (5)$$

onde p_j e $\mathbf{K}_j(\mathbf{x})$ são, respectivamente, a pressão e o tensor no subdomínio j , e Γ_{jk} é a interface do subdomínio Ω_j junto ao subdomínio adjacente Ω_k . Para superar dificuldades inerentes à decomposição de domínio, os subdomínios são sobrepostos conforme ilustrado na Figura 1 [13]. Observa-se na figura que subdomínios estão distribuídos numa malha grossa e pontos nodais estão distribuídos numa malha fina.

Os valores das funções na interface do subdomínio j são impostos com os mesmos valores nos pontos nodais no interior do subdomínio k adjacente, e vice-versa (ver a Figura 2).

3.2 Funções base

No desenvolvimento do simulador MsFDM, são geradas funções base locais para se obter a solução discreta de pressão em cada subdomínio, a partir de condições de contorno canônicas. A resolução do problema local configurado por uma condição de contorno canônica gera um conjunto de funções base [11].

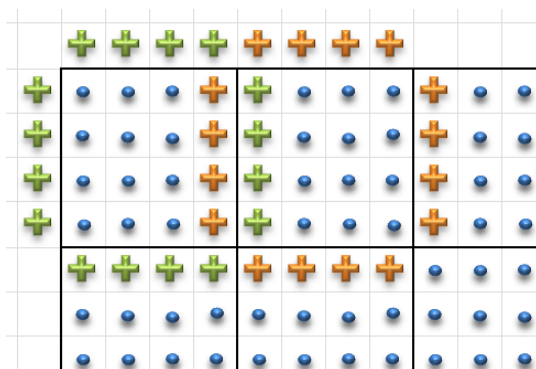


Figura 1: Sobreposição de subdomínios

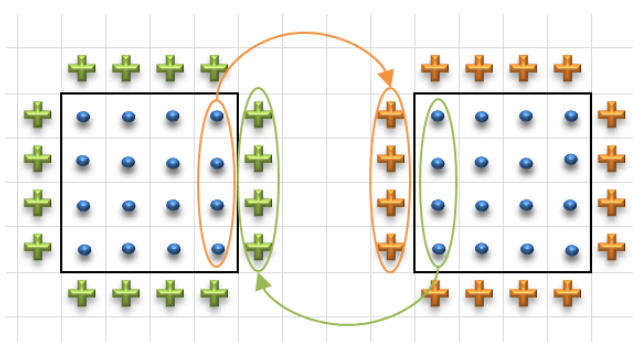


Figura 2: Valores no contorno de subdomínios adjacentes

Para subdomínios quadrangulares Ω_j , as funções base são geradas pela resolução do seguinte problema local:

$$\nabla \cdot (-\mathbf{K}_j(\mathbf{x})\nabla\psi_j^c) = 0, \quad c = 1, 2, \dots, 4N, \quad (6)$$

$$\psi_j^c = \begin{cases} 1, & \text{sobre } \Gamma_j^i, \quad i = c \\ 0, & \text{sobre } \Gamma_j^i, \quad i \neq c \end{cases} \quad i = 1, 2, \dots, 4N, \quad (7)$$

onde ψ representa as funções base, N é o número de pontos nodais em cada interface dos subdomínios, e Γ_j^i é a i -ésima partição de interface de um subdomínio. Como o número de pontos nodais é o mesmo em cada interface, tem-se $4N$ diferentes problemas locais, que geram $4N$ conjuntos de funções base para os subdomínios.

Por exemplo, para um subdomínio quadrangular Ω_j com $N = 2$, pode-se gerar 8 conjuntos de funções base. Na Figura 3, podem-se observar as diferentes condições de contorno canônicas para o subdomínio mencionado.

As funções base são geradas resolvendo-se as Eqs. (6)–(7) com discretização por diferenças finitas, e levando-se em conta o campo de permeabilidade do subdomínio. Cada condição de contorno canônica define um problema de valor de contorno do tipo Dirichlet, logo as funções base geradas de cada problema são funções linearmente independentes. A geração de funções base é o procedimento-chave que permite incorporar a informação do campo de permeabilidade em malha fina no procedimento iterativo em malha grossa.

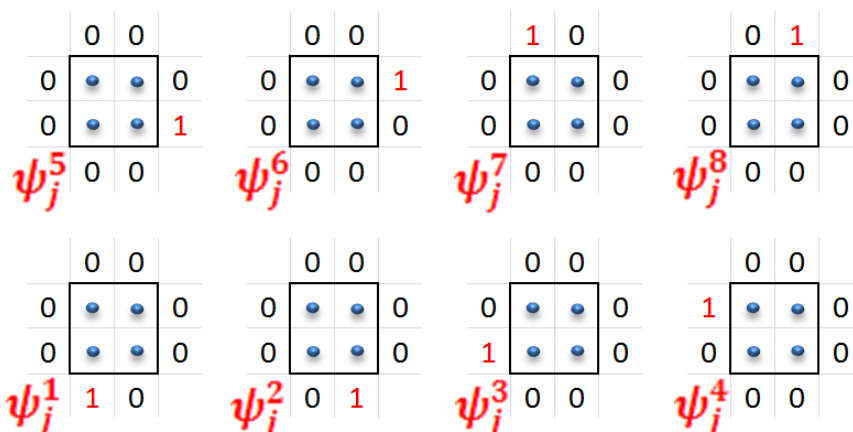


Figura 3: Condições de contorno canônicas do subdomínio

3.3 Procedimento iterativo

A resolução eficiente do problema global com decomposição de domínio requer um procedimento iterativo em malha grossa. Os problemas locais em cada subdomínio poderiam ser resolvidos usando o método das diferenças finitas. Entretanto, uma resolução direta usando diferenças finitas é computacionalmente pesada. Para evitar esse esforço, o MsFDM apresenta um procedimento iterativo no qual a solução atual p_j^ν num subdomínio Ω_j é aproximada pela seguinte combinação linear:

$$p_j^\nu \cong \sum_{c=1}^{4N} A_j^c \psi_j^c, \quad (8)$$

onde A_j^c são coeficientes definidos por

$$A_j^c = p_k^{\nu-1} |_{\Gamma_j^c}, \quad (9)$$

onde $p_k^{\nu-1}$ é a solução prévia no subdomínio k adjacente. A Figura 4 ilustra a condição de interface de um subdomínio dada pelos coeficientes A_j^1 , A_j^2 , A_j^3 , e A_j^4 .

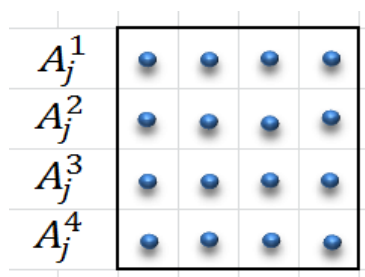


Figura 4: Condição de interface do subdomínio

Há uma vantagem em obter a solução de problemas locais como uma combinação linear: nela são feitas apenas operações de multiplicação e adição, que não demandam grandes esforços computacionais. Outra vantagem reside no fato de que durante o procedimento iterativo em malha grossa, os problemas locais necessitam ser resolvidos somente nos pontos nodais das interfaces de subdomínios. O que reduz bastante a quantidade de processamento de dados.

Para otimizar ou acelerar o processamento computacional e obter maior eficiência na resolução dos problemas, implementa-se um esquema simplificado nas interfaces com o objetivo de reduzir o número de funções bases. Para tal, usa-se uma escala intermediária \bar{H} entre a escala da malha fina h e a escala da malha grossa H , conforme mostrado na Figura 5. A escala intermediária é escolhida de tal forma que $r = H/\bar{H}$ seja um número inteiro. Dessa forma, define-se novas partições de interface de um subdomínio $\bar{\Gamma}_j^c, c = 1, 2, \dots, 4N/r$. A resolução do problema local configurado pelas novas condições de contorno canônicas gera um novo conjunto de funções base $\bar{\psi}$.

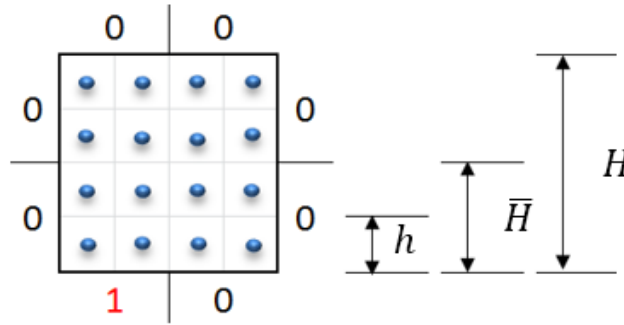


Figura 5: Escalas na interface de subdomínios

Assim, o procedimento iterativo para a solução discreta no subdomínio Ω_j será obtida com uma nova combinação linear simplificada:

$$p_j^\nu \cong \sum_{c=1}^{4N/r} \bar{A}_j^c \bar{\psi}_j^c, \quad (10)$$

onde \bar{A}_j^c são coeficientes definidos por valores médios na partição de interface $\bar{\Gamma}_j^c$,

$$\bar{A}_j^c = \frac{1}{r} \sum_{i=1}^r p_{k,i}^{\nu-1} |_{\bar{\Gamma}_j^c}. \quad (11)$$

Por exemplo, num subdomínio quadrangular Ω_j com $N = 4$ e $r = 2$ pode-se gerar 8 conjuntos de funções base. Na Figura 6, é ilustrada a relação entre a escala intermediária e um coeficiente da combinação linear do procedimento iterativo. Considerando os pontos nodais T e B do subdomínio adjacente Ω_k , o coeficiente da partição de interface é calculado por

$$\bar{A}_j^c = \frac{P_E^T + P_E^B}{2}. \quad (12)$$

4 IMPLEMENTAÇÃO DO OPENMP

O simulador MsFDM utiliza a decomposição do domínio para um problema em geometria *SLAB*. De acordo com a decomposição do domínio, têm-se diferentes arranjos para o simulador. Neste trabalho, três simuladores multiescala são investigados: MsFDM-Arranjo 1, MsFDM-Arranjo 2 e MsFDM-Arranjo 3. Na Figura 7 são mostrados esses arranjos e as respectivas configurações de funções base com $N = 1$.

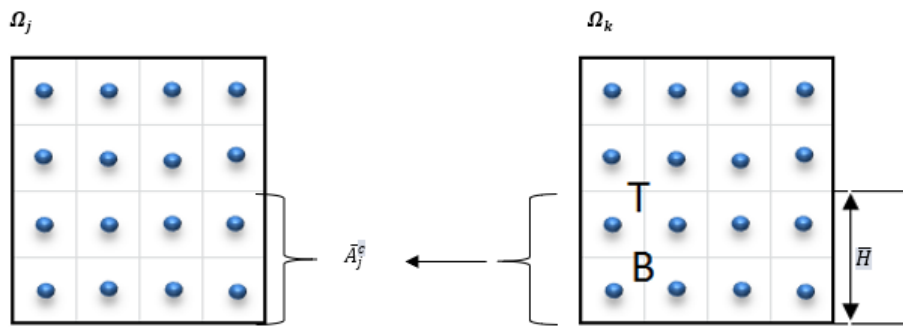


Figura 6: Coeficiente da partição de interface em escala intermediária

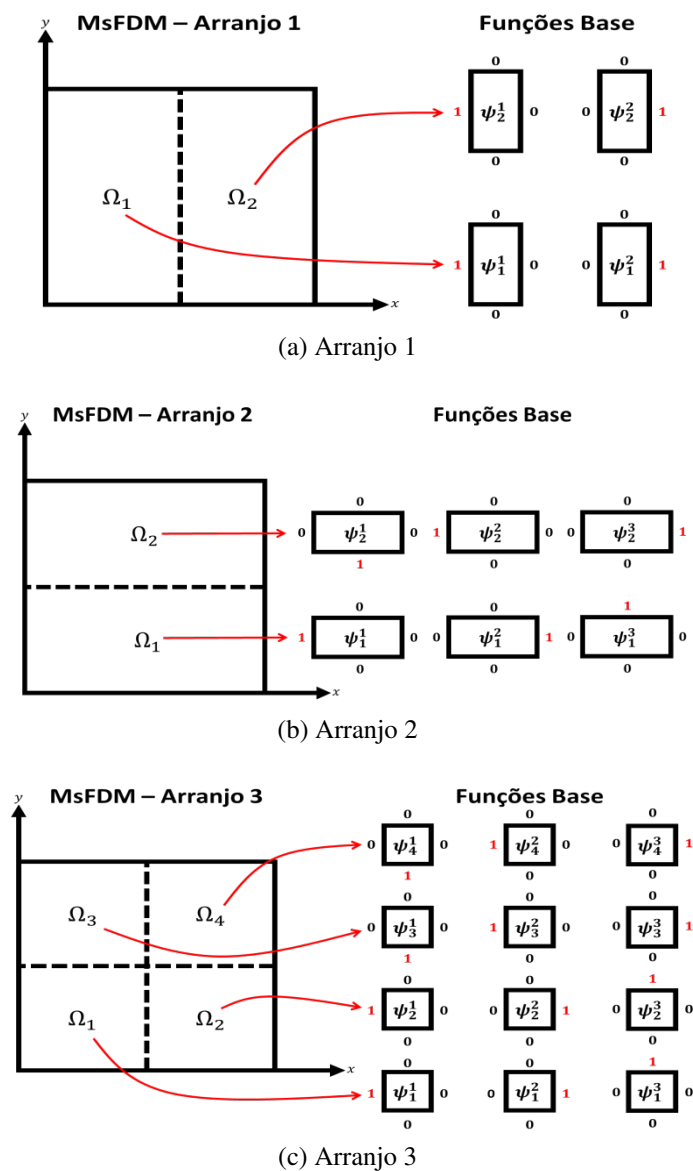


Figura 7: Decomposição de domínio no MsFDM

A paralelização dos simuladores multiescala é implantada apenas na definição das funções base, uma vez que essa etapa consome o maior tempo de execução do simulador. No algo-

ritmo paralelo, usa-se a diretiva *parallel sections*, de modo que cada seção seja processada por um *thread* independente. Na Figura 8, apresenta-se o algoritmo paralelo correspondente ao MsFDM-Arranjo 1, onde duas configurações de funções base são processadas para cada subdomínio. É importante ressaltar que, na geometria *SLAB*, não tem configuração de funções base correspondente à condição de contorno do tipo Neumann.

Nos experimentos numéricos deste trabalho, utiliza-se um computador com as seguintes especificações:

- processador: Intel Core I7-3770, 3,40 GHz, com 8 threads;
- memória: 8Gb RAM; e
- sistema operacional: Windows 10 Education.

```

/*+++++*/
/*SOLUTION*/
#pragma omp parallel sections private(SN, BFN)
{
    //<< BF - Left // Subdomain 0 >>
    #pragma omp section
    {
        SN = 0; BFN = 0;
        PCG1.Solver (BFN, SN);
    }

    //<< BF - Right // Subdomain 0 >>
    #pragma omp section
    {
        SN = 0; BFN = 1;
        PCG2.Solver (BFN, SN);
    }

    //<< BF - Left // Subdomain 1 >>
    #pragma omp section
    {
        SN = 1; BFN = 0;
        PCG3.Solver (BFN, SN);
    }

    //<< BF - Right // Subdomain 1 >>
    #pragma omp section
    {
        SN = 1; BFN = 1;
        PCG4.Solver (BFN, SN);
    }
}
/*+++++*/

```

Figura 8: Algoritmo paralelo do MsFDM-Arranjo 1

5 RESULTADOS DE SIMULAÇÃO

Os experimentos numéricos simulam o escoamento de um fluido monofásico e incompressível, e considera-se um domínio físico com dimensões de $128 \text{ m} \times 128 \text{ m}$, em geometria *SLAB*. As condições de contorno à esquerda e à direita são do tipo Dirichlet, com $p_b = 1,0 \text{ Pa}$ e

$p_b = 0,0$ Pa, respectivamente. As condições de contorno acima e abaixo são do tipo Neumann, com fronteiras impermeáveis. Não há termo fonte, $f = 0$. As malhas de discretização usadas são: 128×128 , 256×256 , 512×512 , e 1024×1024 . A convergência das simulações é alcançada com tolerância igual a 10^{-12} . Em todas as simulações, $\bar{H} = H$.

Nas simulações em meio homogêneo, o desempenho dos simuladores multiescala é comparado com o tradicional método de diferenças finitas (FDM), no qual o sistema resultante de equações lineares é resolvido pelo método dos gradientes conjugados pré-condicionado *SSOR*, com parâmetro de sobre-relaxação $\omega = 0,8$.

Nas simulações em meio heterogêneo, valores de permeabilidade são gerados através da relação:

$$k(\mathbf{x}) = k_0 \exp(\alpha \varepsilon(\mathbf{x})) \quad (13)$$

onde $\varepsilon(\mathbf{x})$ é um campo aleatório gaussiano [1], $k_0 = 1 \times 10^{-14}$ m² é a permeabilidade média, e α é a intensidade da heterogeneidade ligada à razão entre máxima e mínima permeabilidade, dada por

$$\delta = \frac{k_{max}}{k_{min}}. \quad (14)$$

Uma ilustração do campo de permeabilidade na malha 1024×1024 gerado com $\delta = 10^9$ é mostrado na Figura 9.

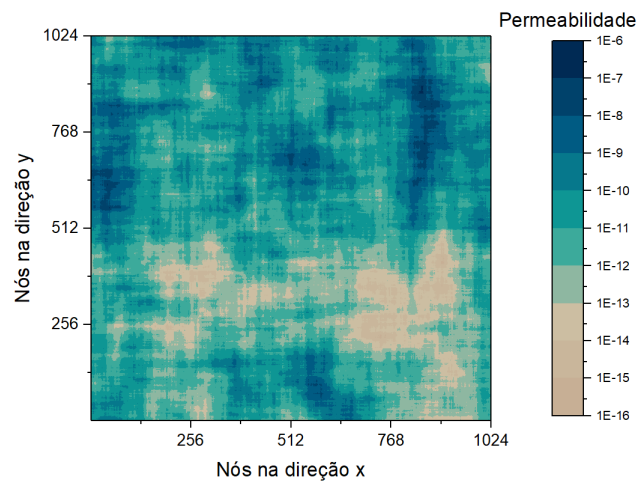


Figura 9: Campo de permeabilidade heterogêneo com $\delta = 10^9$

Para efeito de comparação de resultados, calcula-se o erro relativo global E para solução da pressão pela seguinte expressão:

$$E = \sqrt{\frac{\sum_{i,j} (p_{i,j} - p)^2}{\sum_{i,j} \max(p)^2}}, \quad (15)$$

onde $p_{i,j}$ é a solução do simulador multiescala, e p é a solução de referência do problema. Para problemas em meios homogêneos, a solução de referência é a solução exata do problema; para problemas em meios heterogêneos, a solução de referência é a solução numérica do simulador *Mixed Finite Element Method* (MFEM) [5].

5.1 Algoritmo serial

Inicialmente, as comparações entre os simuladores multiescala e FDM são realizadas com as versões seriais deles. Um estudo de refinamento de malha é conduzido considerando um meio homogêneo. A Figura 10 mostra os resultados do erro relativo global para os simuladores. Observa-se que as soluções numéricas são mais acuradas na medida em que a malha é refinada, ou seja, os simuladores apresentam convergência numérica. O FDM tem os resultados mais acurados, seguido pelo MsFDM-Arranjo 1.

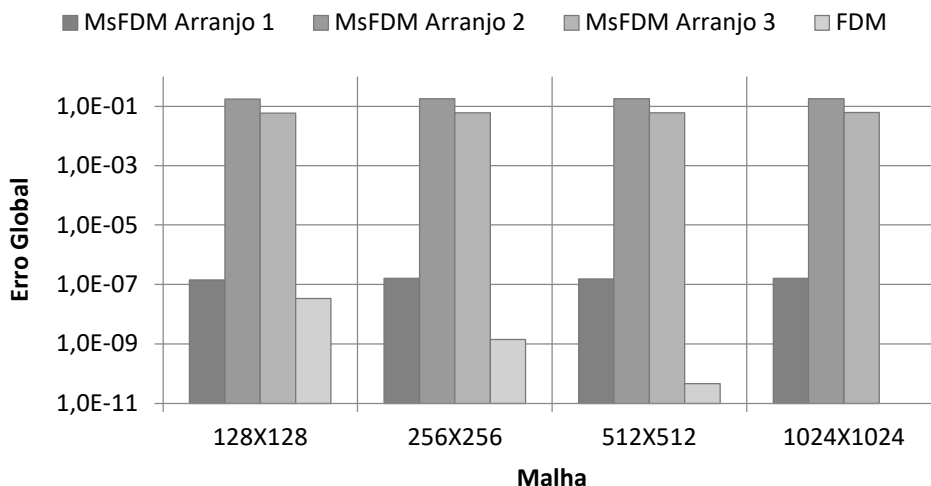


Figura 10: Erro relativo global de pressão para meio homogêneo

Considerando um meio altamente heterogêneo com $\delta = 10^9$, registra-se o tempo de execução dos simuladores. Pode-se observar na Figura 11 que o tempo de execução dos simuladores aumenta na medida em que a malha é refinada. O FDM apresenta os menores tempos de execução, muito embora a diferença dos tempos não seja significativa para os simuladores.

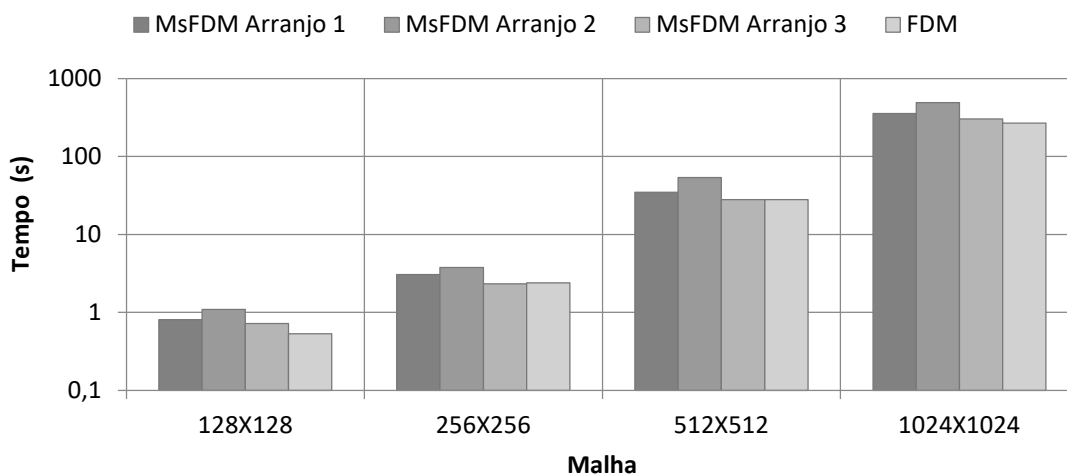


Figura 11: Tempo de execução para meio heterogêneo com $\delta = 10^9$

5.2 Algoritmo paralelo

A seguir, comparações entre os simuladores multiescala são realizadas com as versões paralelas deles. Essas versões paralelas implementadas em OpenMP são avaliadas através das métricas de desempenho tempo de execução, *speedup* e eficiência.

Na Tabela 1, observa-se o tempo de execução dos simuladores multiescala, considerando um meio homogêneo e uma malha 1024×1024 . Simula-se o mesmo problema com diferentes números de *thread*. Os tempos de execução dos simuladores multiescala diminuem com o aumento do número de *thread*. Nota-se que o MsFDM-Arranjo 1 apresenta o menor tempo de execução, que se estabiliza em torno de 101 segundos a partir de quatro *threads*.

Tabela 1: Tempo de execução da versão paralela para meio homogêneo

Thread	Tempo de execução do MsFDM		
	Arranjo 1	Arranjo 2	Arranjo 3
1	190,9	310,9	223,8
2	124,3	194,4	135,3
3	125,6	160,7	117,6
4	99,5	161,5	113,8
5	101,2	170,6	115,2
6	101,9	145,4	110,2
7	101,7	148,5	109,5
8	101,9	146,4	108,7

Outra forma de avaliar o desempenho dos simuladores multiescala é através do *speedup*. Sabe-se que quanto maior o *speedup*, melhor é o desempenho do simulador. Na Figura 12, apresenta-se o gráfico dos resultados de *speedup*, considerando um meio heterogêneo com $\delta = 10^9$ e uma malha 1024×1024 . Este gráfico mostra que há um leve aumento do *speedup* com o aumento do número de *thread*, que tende a se estabilizar, para todos os simuladores multiescala.

Finalmente, o desempenho dos simuladores multiescala é avaliado pela eficiência do processamento paralelo. Na Figura 13, podem-se comparar os resultados de eficiência considerando um meio heterogêneo com $\delta = 10^9$ e uma malha 1024×1024 . A eficiência dos simuladores multiescala decresce exponencialmente com o aumento do número de *thread*. Nota-se que não há uma superioridade em eficiência de um simulador multiescala sobre os outros à medida que o número de *thread* é aumentado.

Uma análise de escalabilidade é realizada para o MsFDM-Arranjo 3 considerando um meio homogêneo. Nota-se na Tabela 2 que o simulador multiescala não demonstra a capacidade de manter a eficiência à medida que o número de *thread* aumenta, nem mesmo quando a dimensão do problema também aumenta proporcionalmente. Portanto, pode-se afirmar que este simulador é não escalável. Resultados similares são obtidos para os outros simuladores multiescala.

Para uma avaliação visual dos resultados, na Figura 14 são apresentados os campos de pressão dos simuladores multiescala e do MFEM, considerando um meio heterogêneo com $\delta = 10^9$ e uma malha 1024×1024 . O MFEM é reconhecido por produzir campos de velocidades acurados para problemas de escoamento em meios altamente heterogêneos. Dos simuladores multiescala, o melhor resultado de campo de pressão é obtido com o MsFDM-Arranjo 1. Os outros simuladores multiescala apresentam um erro considerável ao longo de uma linha longitudinal ao escoamento.

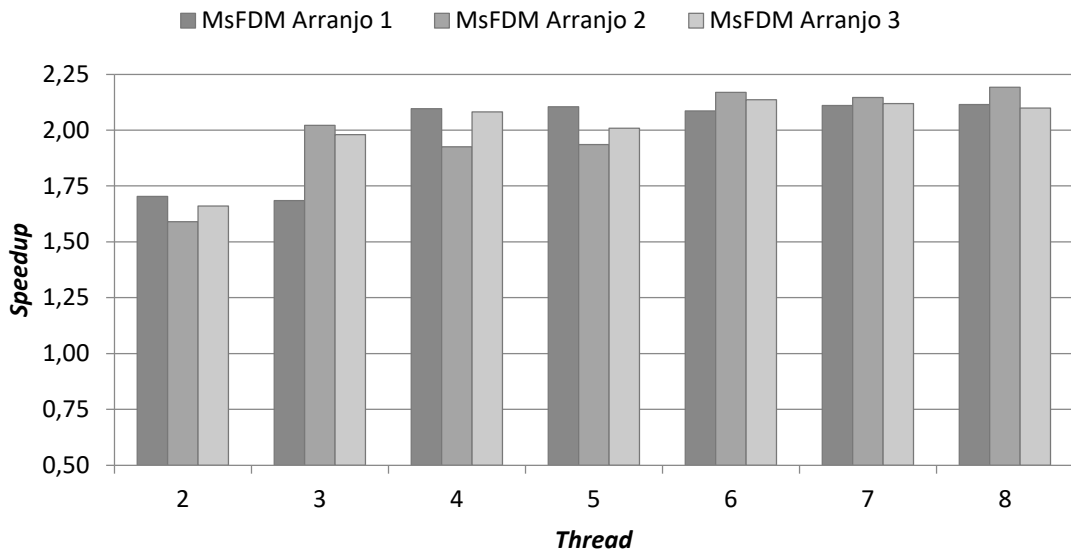


Figura 12: *Speedup* para meio heterogêneo com $\delta = 10^9$

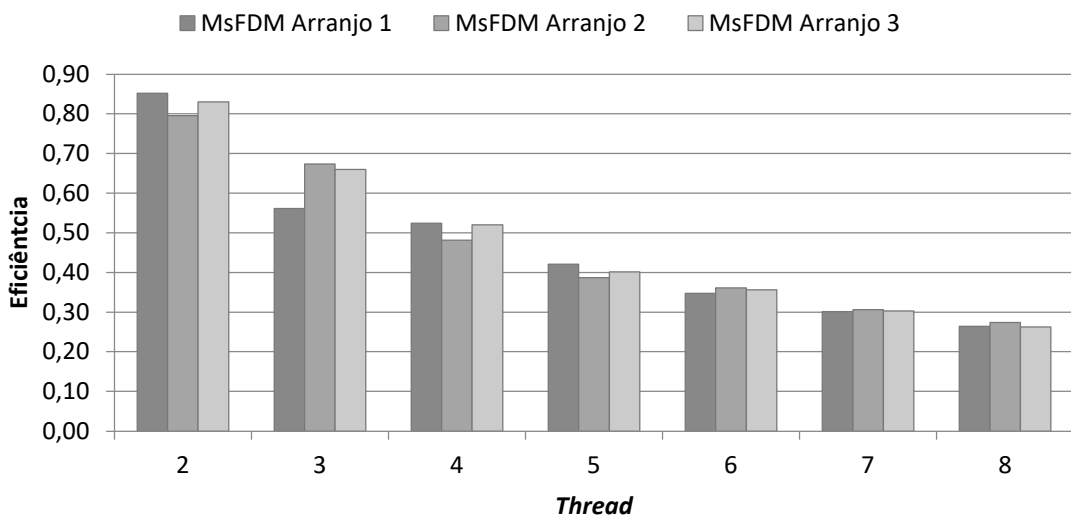


Figura 13: Eficiência para meio heterogêneo com $\delta = 10^9$

Tabela 2: Escalabilidade do MsFDM-Arranjo 3 para meio homogêneo

Malha	Eficiência		
	2 threads	4 threads	8 threads
512 × 512	0,64	0,42	0,21
1024 × 1024	0,72	0,48	0,24
2048 × 2048	0,72	0,47	0,24

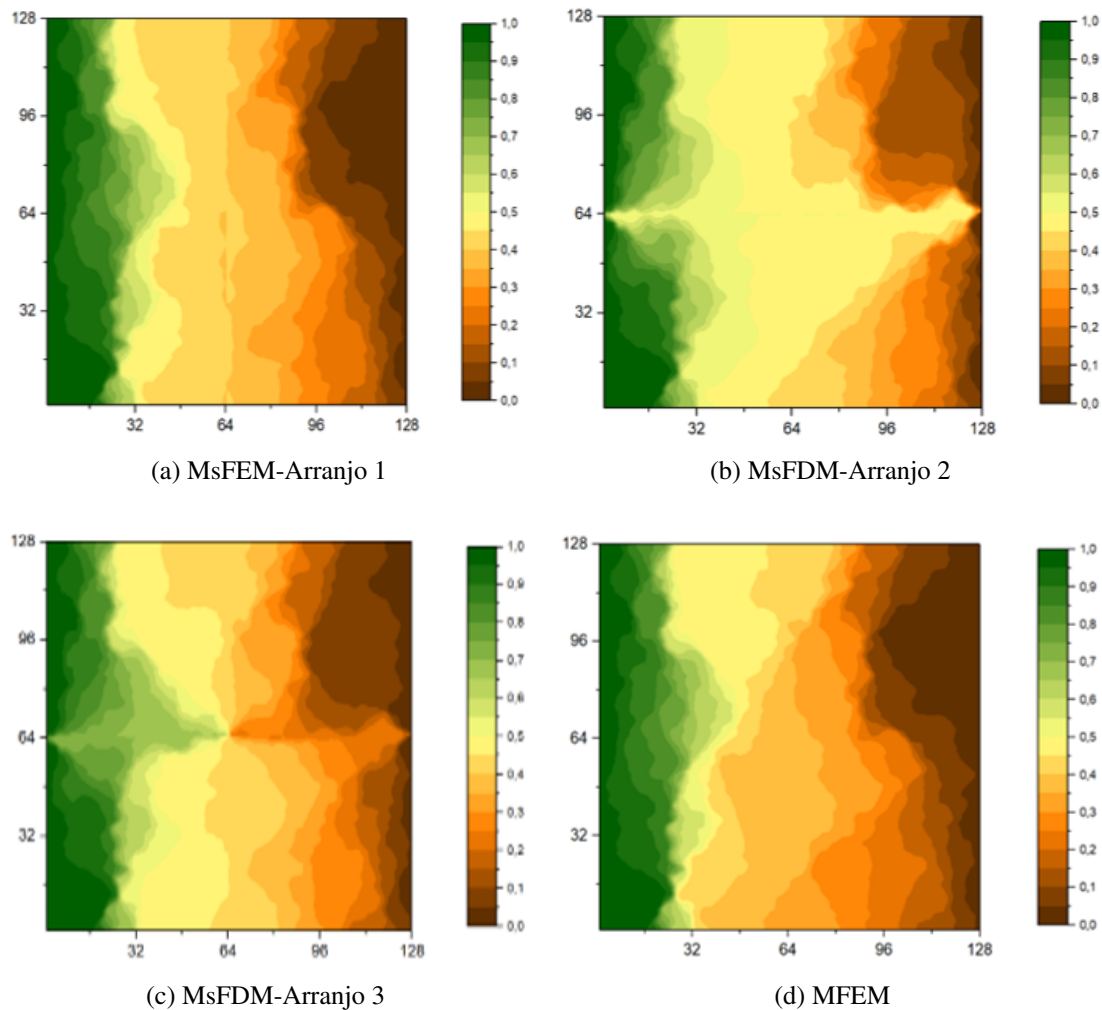


Figura 14: Campos de pressão para meio heterogêneo com $\delta = 10^9$

6 CONCLUSÕES

Uma das vantagens do *OpenMP* é que sua implementação em algoritmos é bastante simples, uma vez que, com apenas algumas linhas de comando, se consegue paralelizar o código naquelas funções que demandam um grande esforço computacional.

O MsFDM é um simulador computacionalmente comparável ao FDM, para problemas de escoamentos em meio poroso de geometria *SLAB*. Embora haja uma redução de acurácia com o MsFDM em relação ao FDM, há casos em que ele apresenta erros globais da ordem de 10^{-7} , ou seja, uma acurácia aceitável.

O MsFDM-Arranjo 1 quando comparado aos MsFDM-Arranjo 2 e 3, apresenta os menores valores de erro global para a pressão, sendo, portanto, o mais acurado dos simuladores multiescala. Embora não haja superioridade em speedup e eficiência de um arranjo sobre os outros, o MsFDM-Arranjo 1 apresenta os menores tempos de execução, tanto na versão serial quanto na paralela.

Portanto, pode-se afirmar que o simulador MsFDM-Arranjo 1 gera resultados numéricos que alcançam um equilíbrio adequado entre esforço computacional e acurácia.

Agradecimentos

Os autores agradecem a CAPES pela bolsa de estudo concedida a Leonardo Menezes Sacramento.

REFERÊNCIAS

- [1] M. Borges, F. Furtado, F. Pereira e H. Souto. Scaling analysis for the tracer flow problem in self-similar permeability fields. *Multiscale Modeling & Simulation*, 7(3):1130–1147, 2008.
- [2] J. A. Cuminato e M. Meneguette Jr. *Discretização de Equações Diferenciais Parciais: técnicas de diferenças finitas*. SBM, 1 edição, 2013.
- [3] C. N. Dawson, Q. Du e T. F. Dupont. A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Mathematics of Computation*, 57(195):63–71, 1991.
- [4] M. C. de Castro Cunha. *Métodos Numéricos*. Editora da Unicamp, 2 edição, 2000.
- [5] J. Douglas Jr, P. J. P. Leme, J. E. Roberts e J. Wang. A parallel iterative procedure applicable to the approximate solution of second order partial differential equations by mixed finite element methods. *Numerische Mathematik*, 65:95–108, 1993.
- [6] A. S. Francisco, V. Ginting, F. Pereira e J. Rigelo. Design and implementation of a multiscale mixed method based on a nonoverlapping domain decomposition procedure. *Mathematics and Computers in Simulation*, 99:125–138, 2013.
- [7] A. S. Francisco e L. M. Sacramento. Implementing intermediate interface scales into the msfdm procedure. *International Journal of Mechanical Engineering and Automation*, 2(9):419–423, 2015.
- [8] A. S. Francisco, J. J. Trujillo e L. M. Sacramento. A multiscale finite difference method for elliptic problems in heterogeneous porous media. *Revista de Engenharia da Universidade Católica de Petrópolis*, 9(1):21–30, 2015.
- [9] A. S. Francisco, J. J. Trujillo e L. M. Sacramento. A new multiscale simulation of water flows in heterogeneous porous media. In *Anais do 1st Pan-American Congress on Computation Mechanics*. Buenos Aires, Argentina, 2015.
- [10] A. S. Francisco, J. J. Trujillo e L. M. Sacramento. The advantage of multiscale basis functions over finite differences in a multiscale technique for porous media problems. *Cuadernos de Mecánica Computacional*, 14(1):106–114, 2016.
- [11] B. A. Ganis e I. Yotov. Implementation of a mortar mixed finite element method using a multiscale flux basis. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3989–3998, 2009.
- [12] G. D. Marsily. *Quantitative Hydrology*. Academic Press, 1986.
- [13] T. Min e D. Yang. Parallel finite difference schemes for heat equation based upon overlapping domain decomposition. *Applied Mathematics and Computation*, 186:1276–1292, 2007.
- [14] J. J. Trujillo. *Simulação numérica multiescala do escoamento de água em meios porosos heterogêneos*. Dissertação de Mestrado, Universidade Federal Fluminense, Volta Redonda - RJ, Brasil, 2014.