

# Implementação Híbrida MPI/OpenMP do Método HOPMOC na Resolução da Equação de Convecção-Difusão

**Frederico Luís Cabral, Carla Osthoff**

Laboratório Nacional de Computação Científica, LNCC  
Petropolis, RJ

E-mail: fcabral@lncc.br, osthoff@lncc.br

**Mauricio Kischinhevsky**

UFF - Instituto de Computação  
Niterói, RJ

E-mail: kisch@ic.uff.br

**Diego N. Brandão**

CEFET-RJ, Colegiado de Informática  
Unidade de Nova Iguaçu

Nova Iguaçu, RJ

E-mail: dbrandao@ic.uff.br

**Resumo:** *A utilização da computação paralela na resolução de certos problemas descritos por equações diferenciais parciais permite um ganho significativo no tempo de computação. Este trabalho apresenta uma implementação paralela do método HOPMOC em um ambiente de máquinas multicore. O método HOPMOC utiliza conceitos do método das características modificado associado com método Hopscotch. Tal construção lhe fornece características ideais para abordagens em computação paralela em ambientes de memória distribuída. O presente trabalho utiliza uma abordagem híbrida para o HOPMOC: o MPI é utilizado para comunicação no ambiente distribuído, enquanto o OpenMP permite o paralelismo no ambiente de memória compartilhada de cada nó do cluster. Resultados preliminares demonstram ganhos significativos de eficiência da implementação híbrida apresentada quando comparado com uma versão sequencial do HOPMOC.*

**Palavras-chave:** *Computação Paralela, Equação de Convecção-Difusão, Método HOPMOC.*

## 1 Introdução

Máquinas multicore são o cerne dos sistemas computacionais dos dias de hoje. Tais arquiteturas fornecem um grande poder computacional para resolução de problemas de grande porte das áreas de engenharia, física, e etc. Assim, o desenvolvimento de métodos que sejam robustos e eficientes para tais arquiteturas torna-se necessário.

O método HOPMOC foi proposto como um método para resolução de problemas de convecção-difusão com dominância convectiva para ser executado em ambientes compostos por máquinas paralelas [7]. Ele permite uma decomposição espacial do domínio do problema, onde cada subproblema pode ser resolvido independentemente. Além disso ele minimiza a troca de informações entre regiões vizinhas, reduzindo assim a troca de mensagens entre os nós do ambiente distribuído. Outra característica é que seu custo por instante de tempo é conhecido *a priori*, além de usar uma estratégia baseada na busca por informação ao longo da linha característica para cada instante de tempo.

Técnicas de decomposição de operadores, como as empregadas pelo HOPMOC, foram estudadas durante as últimas décadas com o propósito de reduzir o custo computacional dos métodos para solucionar problemas multidimensionais evolutivos modelados por equações diferenciais parciais [13, 1, 3]. Hoje com o avanço das tecnologias de paralelismo em ambientes de memória compartilhada e distribuída, o foco da pesquisa tem sido no desenvolvimento de estratégias que explorem ao máximo as capacidades computacionais desses ambientes, com uma atenção especial para o consumo de memória e o custo de troca de mensagens.

No contexto de computação distribuída, a biblioteca *Message Passing Interface* (MPI) tem sido utilizada por muitos algoritmos para alcançar a escalabilidade em um grande número de processadores. Todavia, ela não consegue utilizar o paralelismo do ambiente de múltiplos núcleos do processador de forma eficiente [9]. Alternativamente, OpenMP é um paradigma de programação paralela que permite o paralelismo mais interno no contexto dos núcleos do processador. A associação dos modelos paralelos implementados pelo OpenMP e pelo MPI permite a construção de um programa híbrido que é capaz de atingir dois níveis de paralelismo. Essa abordagem reduz o *overhead* de comunicação gerado pelo MPI ao custo de introduzir o *overhead* do OpenMP para criação de cada thread. Contudo, as abordagens híbridas dadas por meio do modelo MPI/OpenMP apresentaram um ganho significativo de eficiência conforme pode ser observado em [10, 5].

Este trabalho apresenta uma abordagem híbrida com MPI/OpenMP para o método HOPMOC aplicado na resolução da equação de convecção-difusão. Uma versão sequencial deste método foi apresentada para o mesmo problema em [8]. Já sua análise teórica com as condições de estabilidade foi apresentada em [11]. Uma primeira abordagem em paralelo utilizando o MPI foi apresentada em [2], todavia não foi utilizado um ambiente de múltiplos núcleos. Além disso a técnica de decomposição de domínio utilizada aqui reduz o custo de comunicação quando comparada com do trabalho citado.

Este trabalho é organizado da seguinte maneira. A seção 2 apresenta a discretização da equação de convecção-difusão por meio do método HOPMOC e discute algumas características do método. A seção 3 descreve a implementação paralela por meio do modelo MPI-OpenMP. O ambiente computacional e as simulações realizadas são apresentados na seção 4. Por fim, as conclusões e desdobramentos do trabalho são apresentados na seção 5.

## 2 Descrição do Método HOPMOC

Considere a equação de convecção-difusão unidimensional:

$$u_t + vu_x = du_{xx} \tag{1}$$

com condições iniciais e de contorno adequadas, onde a velocidade  $v$  é constante e positiva,  $d$  é uma constante positiva de difusividade,  $0 \leq x \leq 1$  e  $0 \leq t \leq T$ .

Considere um esquema convencional para discretização em diferenças finitas deste problema, onde  $\Delta t = t_{n+2} - t_n$ ,  $\delta t = \frac{\Delta t}{2} = t_{n+1} - t_n$ ,  $\bar{u}_i^{n+1} = u(\bar{x}_i^{n+1})$ , sendo este o valor da variável no instante de tempo anterior do semi-passo obtido no pé da linha característica, essa linha originada por  $x_i^{n+2}$  e  $\Delta x = x_{i+1} - x_i = \frac{1}{\kappa+1}$ , onde  $\kappa$  é ímpar. A mesma linha característica permite obter  $\bar{u}_i^n$  no semi-passo de tempo anterior.

Seja  $u_i^{n+2}$ , onde  $n$  é ímpar, uma aproximação numérica para  $u$  em  $(x_i, t_{n+2})$ , e usando um operador diferencial  $L$ ,  $L_h u_i^n = d \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x^2}$ , os semi-passos consecutivos do HOPMOC podem ser descritos por:

$$\begin{aligned} \bar{u}_i^{n+1} &= \bar{u}_i^n + \delta t \left[ \theta_i^n L_h \bar{u}_i^n + \theta_i^{n+1} L_h \bar{u}_i^{n+1} \right], \\ u_i^{n+2} &= \bar{u}_i^{n+1} + \delta t \left[ \theta_i^n L_h \bar{u}_i^{n+1} + \theta_i^{n+1} L_h u_i^{n+2} \right], \\ \text{para } \theta_i^n &= \begin{cases} 1, & \text{se } n+i \text{ é ímpar,} \\ 0, & \text{se } n+i \text{ é par,} \end{cases} \end{aligned} \tag{2}$$

e o valor  $\bar{u}_i^n$  é obtido por meio de um processo de interpolação. Os valores  $\bar{x}_i^{n+1}$  e  $\bar{x}_i^{n+1}$  são obtidos por  $\bar{x}_i^{n+1} = x_i - v \delta t$  e  $\bar{x}_i^n = x_i - 2v \delta t$ .

A análise de convergência do HOPMOC foi apresentada em [11], nele são descritas as condições de consistência e estabilidade no caso da discretização do problema da equação de convecção-difusão. O teorema de Lax garante que se o problema de valor inicial é bem posto no sentido de Hadamard, as condições de estabilidade e consistência implicam na convergência do método numérico [12]. Assim, o método HOPMOC é incondicionalmente estável para Eq. 1. De acordo com Kischinhevsky [8], este resultado pode ser estendido para dimensões mais altas do problema. A Figura 1 ilustra esquematicamente o método HOPMOC para um problema unidimensional.

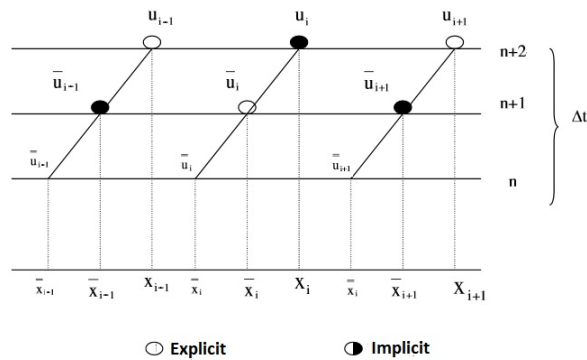


Figura 1: Método HOPMOC.

No caso bidimensional, os dois passos do método grupo HOPMOC com quatro pontos são descritos pela Eq. 3.

$$\begin{aligned}
 \bar{u}_{i,j}^{n+1} &= \bar{u}_{i,j}^n + \delta t \left[ \theta_{i,j}^n L_h \bar{u}_{i,j}^n + \theta_{i,j}^{n+1} L_h \bar{u}_{i,j}^{n+1} \right], \\
 \bar{u}_{i,j}^{n+2} &= \bar{u}_{i,j}^{n+1} + \delta t \left[ \theta_{i,j}^{n+1} L_h \bar{u}_{i,j}^{n+1} + \theta_{i,j}^{n+2} L_h \bar{u}_{i,j}^{n+2} \right], \\
 \text{para } \theta_{i,j}^n &= \begin{cases} 1, & \text{se } [i+1]/2 + [j+1]/2 + n \text{ é ímpar,} \\ 0, & \text{se } [i+1]/2 + [j+1]/2 + n \text{ é par} \end{cases} \quad (3)
 \end{aligned}$$

O HOPMOC não requer que um sistema linear seja resolvido; assim, sua paralelização é simples de ser realizada, pois o desacoplamento espacial permite que as variáveis sejam subdivididas em processadores diferentes, sem que dependam de variáveis armazenadas em outros processadores. Uma outra vantagem a ser observada refere-se ao custo computacional do HOPMOC que é  $O(N)$  por instante de tempo.

### 3 Computação Paralela

A biblioteca mais utilizada para comunicação entre computadores em um ambiente distribuído é a *Message Passing Interface* (MPI). Todavia, esta biblioteca não consegue obter o máximo de performance em ambientes de memória compartilhada, caso do paralelismo em processadores de múltiplos núcleos. Nestes ambientes é utilizada a biblioteca OpenMp, a qual permite a paralelização entre os núcleos de um processador. É um padrão industrial para programação em ambientes de memória compartilhada. Ela permite uma programação por meio de *multithreads* extraindo ao máximo das características do ambiente de memória compartilhada. Com as políticas de compartilhamento de dados entre as *threads*, ela elimina o *overhead* de comunicação dos dados. Um laço do programa pode ser facilmente paralelizado por meio de uma chamada de

uma subrotina da biblioteca de *threads* do OpenMP, simplesmente inserindo algumas diretivas de compilação. Assim, um modelo híbrido do MPI/OpenMP pode aproveitar os benefícios de ambos os modelos. Um programa híbrido do MPI/OpenMP contém códigos das diretivas do OpenMP dentro dos processos MPI.

### Versão Paralela do Método HOPMOC

O método HOPMOC é ideal para o paralelismo em ambientes de memória distribuída, pois a comunicação entre os subdomínios que são geometricamente contíguos é realizada uma única vez a cada instante de tempo. Essa característica permite que o problema seja resolvido de forma independente em cada subdomínio. Cabral [2] apresentou uma versão em paralelo do método HOPMOC e do grupo HOPMOC. Ambas as versões utilizavam o MPI em um ambiente de memória distribuída.

O procedimento empregado pelo HOPMOC em cada instante de tempo pode ser dividido em dois estágios, que podem ser avaliados sequencialmente, além de serem independentes de referências as variáveis correspondentes a pontos distantes; ou seja, não há acoplamento de longo alcance. Assim, os subdomínios são desacoplados, o custo computacional escalável é esperado, isto é, o *speedup* cresce quase linearmente com o número de processadores, uma característica desejável para programas paralelos.

Os passos da abordagem paralela do HOPMOC, com balanço de carga computacional, consistem em: primeiramente, o domínio do problema deve ser dividido em subdomínios de mesmo tamanho e que tenham interface de contato, com uma faixa de replicação, para que as informações de interfaces superpostas sejam replicadas entre subdomínios vizinhos a cada passo de tempo. A união de todos os subdomínios constitui uma decomposição do domínio, com redundância monocamada nas interfaces entre os subdomínios. Para que se tenha um resultado eficiente, deve-se buscar que o número de subdomínios seja igual ao de nós processadores do cluster. A Figura 2 representa a subdivisão descrita para o caso de quatro subdomínios sobrepostos.

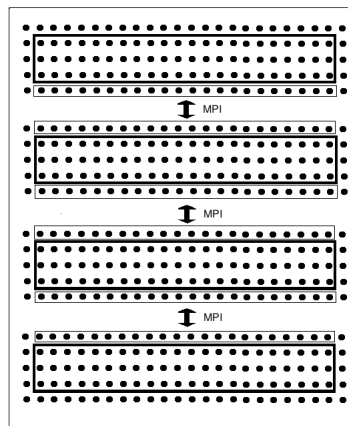


Figura 2: Divisão do domínio em quatro subdomínios sobrepostos.

Este esquema de decomposição do domínio reduz a quantidade de mensagens trocadas, uma vez que cada processo MPI reside em um nó do cluster. Desta forma as várias *threads* dentro de um mesmo nó se comunicam através de memória compartilhada. Na abordagem utilizada a quantidade máxima de troca de mensagens é de duas mensagens para cada subdomínio.

O esqueleto de código mostrado a seguir ilustra a implementação híbrida MPI/OpenMP do HOPMOC, de acordo com o esquema na figura 3, onde um semi passo do HOPMOC é ilustrado com dois processos que se comunicam via MPI.

```

while (time <= FinalTime){

    #pragma omp parallel
    {
        // Calcula a informa\c{c}\~ao no p\`e da caracter\`istica
        // Calcula a informa\c{c}\~ao com operador expl\`icito
    }

    // Envios e recebimentos via MPI da informa\c{c}\~ao na
    // fronteira do subdom\`inio

    #pragma omp parallel
    {
        // Calcula a informa\c{c}\~ao com operador impl\`icito
    }
    // Envios e recebimentos via MPI da informa\c{c}\~ao na
    // fronteira do subdom\`inio

    time += DeltaTime/2
}

```

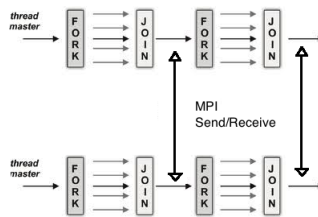


Figura 3: Ambiente Híbrido MPI/OpenMP.

## 4 Simulação Numérica e Análise de Desempenho

Neste trabalho os experimentos computacionais foram realizados simulando a Eq.4 bidimensional.

$$u_t + v\{u_x + u_y\} = d\{u_{xx} + u_{yy}\} \tag{4}$$

com condição inicial

$$u(x, y, 0) = e^{\{-D*((x-xo)^2+(y-yo)^2)\}} \tag{5}$$

onde  $D$  é o coeficiente de difusão. Os experimentos mostrados aqui foram executados com  $D = 10^{-2}$ ,  $xo = yo = 0.1$ ,  $vx = vy = 2,5$  e condição de contorno de Dirichlet  $u(X, t) = 0$  para  $X$  em toda fronteira e  $t > 0$ .

Esta equação tem papel fundamental na descrição dos modelos dos mais variados fenômenos físicos. Algumas aplicações podem ser encontradas na modelagem de transporte de poluentes em rios, nos modelos que descrevem o movimento de aerossóis na atmosfera, etc.

### Ambiente Computacional e Experimentos Numéricos

Os resultados numéricos apresentados foram obtidos em um cluster de máquinas multicore com sistema operacional Linux com um canal de comunicação *infiniband*. O cluster era composto

por 4 nós, sendo cada nó formado por dois processadores Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz (16 cores por nó) com memória de 64GB DDR3 DIMMs.

A figura 4 apresenta a simulação do método HOPMOC para a Eq.4. As figuras demonstram o método representado o fenômeno representado pela Eq.4.

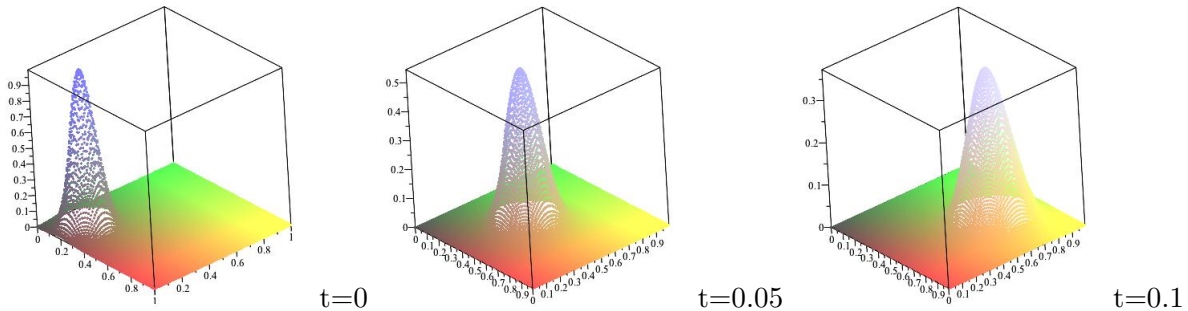


Figura 4: Simulação da equação de convecção-difusão com o método HOPMOC para diferentes instantes de tempo.

A Tabela 1 apresenta os resultados obtidos na simulação de um domínio contendo 4096x4096 pontos. Cada tarefa de trabalho MPI corresponde a um processo, dentro do qual existe um bloco de trabalho OpenMP, ou seja, várias *threads*, sendo que cada uma é executada por exatamente um core (núcleo). Assim, no início da execução são criados vários processos MPI, um para cada nó do cluster, conforme a figura 2. Estes processos duram durante toda a simulação, mas para cada semi-passo do HOPMOC as *threads* OpenMP são criadas e destruídas, de acordo com o modelo *fork-join* e apenas a thread master executa os envios e recebimentos do MPI, entre os nós do cluster. É importante ressaltar que os testes com 1, 2, 4, 8 e 16 *threads* rodaram em um único nó, enquanto que os testes com 32, 48 e 64 *threads* usaram 2, 3 e 4 nós, respectivamente.

Simulation Results		
<i>threads</i>	Runtime(in seconds)	<i>speedup</i>
1	3498,3	1.0
2	2015,7	1,74
4	1044,8	3,35
8	542,7	6,45
16	372,9	9,38
32	185,6	18,85
48	125,6	27,85
64	94,1	37,18

Tabela 1: MPI/OpenMP results

## 5 Conclusões e Trabalhos Futuros

Este artigo apresentou uma resolução para o problema de convecção-difusão utilizando uma implementação híbrida MPI/OpenMP do método HOPMOC em um cluster de máquinas multi-core. Os resultados obtidos indicam um ganho significativo de desempenho com essa abordagem. A característica intrinsecamente paralela do HOPMOC indica que abordagens utilizando cluster de máquinas paralelas de placas gráficas (GPUs) deverão ser avaliadas. Para tanto, uma nova abordagem híbrida MPI/OpenCL ou MPI/OpenACC será desenvolvida. Novos esquemas de subdivisão do domínio serão estudados de forma a avaliar o impacto do custo de troca de mensagens no método. Além disso, os resultados apresentados em [11] demonstram que o uso da interpolação para o cálculo da informação referente ao “pé” da linha característica reduz a

precisão do método. Técnicas baseadas na diminuição da variação total do fluxo (*TVD- Total Variation Diminishing*) são frequentemente utilizadas na redução do erro referente a interpolação do termo convectivo no instante de tempo precedente. Uma associação desses esquemas com o HOPMOC deverá ser avaliada, visando a redução do erro inserido pela fase de interpolação.

## Agradecimentos

Os autores agradecem à Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelo apoio financeiro ao projeto e ao SINAPAD por disponibilizar o ambiente computacional para as simulações apresentadas.

## Referências

- [1] Boonkkamp, J. H. M.T.J., Verwer, J. G.: On the odd-even hopscotch scheme for the numerical integration of time-dependent partial differential equations. *Appl. Num. Math.*, 3 (1987), 183-193.
- [2] Cabral, F.L.: HOPMOC methods to solve convection-diffusion equations and its parallel implementation (in Portuguese). Master Thesis, Instituto de Computação/Universidade Federal Fluminense, Brasil (2001).
- [3] Hansen, J., Matthey, T., Sorevik, T.: A Parallel Split Operator Method for the Time Dependent Schrodinger Equation. *Lecture Notes in Computer Science: Recent Advances in Parallel Virtual Machine and Message Passing Interface*. 2840, 503-510 (2003).
- [4] Harten, A.: On a Class of High Resolution Total-Variation-Stable Finite-Difference Schemes. *SIAM J. Numer. Anal.*, 21 (1984), 1-23.
- [5] Jin, H., Jespersen, D., Mehrotra, P., Biswas, R., Huang, L., Chapman, B.: High performance computing using MPI and OpenMP on multi-core parallel systems. *Parallel Computing*, (2011), 562-575.
- [6] Kirk, D.B., Hwu, W.W.: "Programming Massively Parallel Processors: A Hands-on Approach". Elsevier Inc. (2010).
- [7] Kischinhevsky, M.: An operator splitting for optimal message-passing computation of parabolic equation with hyperbolic dominance. SIAM Annual Meeting, Missouri (1996)
- [8] Kischinhevsky, M.: A spatially decoupled alternating direction procedure for convection-diffusion equations. Proceedings of the XXth CILAMCE-Iberian Latin American Congress on Numerical Methods in Engeneearing (1999)
- [9] Li, D., Zhou, Z., Wang, Q.: A hybrid MPI/OpenMP model based on DDM for large-scale partial differential equations. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 1839- -1843, Liverpool (2012).
- [10] Mininni, P.D., Rosenberg, D., Reddy, R., Poquet, A.: A hybrid MPI/OpenMP scheme for scalable parallel pseudospectral computations for fluid turbulence. *Parallel Computing*, (2011) 316-326.
- [11] Oliveira, S., Gonzaga, S.L., Kischinhevsky, M.: Convergence analysis of the HOPMOC method. *International Journal of Computer Mathematics*, 86, (2009) 1375-1393.
- [12] Richtmyer, R.D., Morton, K.W. : "Difference Methods for Initial-Value Problems". Interscience. New York (1967)
- [13] Yanenko, N. N.: "The Method of Fractional Steps". Springer-Verlag, New York (1970).