

Paralelização do método Diamond Difference aplicado a problemas de transporte de nêutrons em meios com elevada razão de espalhamento e geometria X,Y

Parallelization of the Diamond Difference method applied to neutron transport problems in media with high scattering ratio and X, Y geometry

Iram Bárbaro Rivas Ortiz¹, Joel Sánchez Domínguez², Dany Sánchez Domínguez³, Ricardo Carvalho de Barros⁴

RESUMO

Em problemas de transporte de nêutrons com meios materiais de elevada razão de espalhamento, os esquemas iterativos usados convergem lentamente à solução correta. Na literatura são reportadas várias técnicas para acelerar a convergência como a aceleração de Chebychev, reequilíbrio da malha e técnicas aceleração sintética. Neste trabalho, apresentamos a paralelização com OpenMP, como uma técnica de aceleração, usando o método numérico Diamond Difference (DD) com iteração da fonte de espalhamento. Para a paralelização do código usamos a técnica de inversão de ciclos e isolamos as secções críticas fora dos ciclos em cada iteração. Os resultados gerados pelo método convergem a solução de referência para grades espaciais finas e requerem elevados tempos de processamento no caso serial em problemas de alta razão de espalhamento. Para grades grossas, o método é instável obtendo-se valores negativos do fluxo escalar. A implementação paralela do método diminui os tempos de processamento. Melhores tempos de processamento e maiores valores de speedup são reportados com o aumento de número de CPUs usados na execução da versão paralela.

Palavras-chave: Paralelização. Problema de Transporte de Nêutrons. Aproximação em Ordenadas Discretas. Geometria X,Y.

ABSTRACT

In neutron transport problems with high scattering material media, the iterative schemes used converge slowly to the correct solution. Various techniques to accelerate convergence are reported in the literature, such as Chebychev acceleration, mesh rebalancing and synthetic acceleration techniques. In this work, we present the parallelization with OpenMP, as an acceleration technique, using the numerical method Diamond Difference (DD) with iteration of the scattering source. For the parallelization of the code we use the technique of inversion of cycles and isolate the critical sections outside the cycles in each iteration. The results generated by the method converge the reference solution for fine spatial grids and require high processing times in the serial case in high scattering ratio problems. For thick grids, the method is unstable, obtaining negative scalar flow values. The parallel implementation of the method decreases the processing times. Better processing times and higher speedup values are reported with the increase in the number of CPUs used in the execution of the parallel version.

Keywords: Parallelization. Neutron Transport Problem. Discrete Ordinate Approximation. X,Y Geometry.

¹ Doutor em Modelagem Computacional pelo IPRJ/UERJ

Instituto Politécnico da Universidade do Estado do Rio de Janeiro.

<https://orcid.org/0000-0001-6917-0102>

E-mail: ibrivasortiz92@gmail.com

² Doutor em Modelagem Computacional pelo IPRJ/UERJ

Instituto Politécnico da Universidade do Estado do Rio de Janeiro.

<https://orcid.org/0000-0003-4330-8593>

³ Doutor em Modelagem Computacional pelo IPRJ/UERJ

Universidade Estadual de Santa Cruz, Ilhéus Bahia.

<https://orcid.org/0000-0002-0640-2970>

⁴ Doutorado em Engenharia Nuclear pela Universidade de Michigan

Universidade do Estado do Rio de Janeiro.

<https://orcid.org/0000-0003-0979-0021>

1. INTRODUÇÃO

Os esquemas numéricos usados em problemas de transporte de partículas pouco difusivos convergem lentamente, necessitando de um elevado número de iterações para atingir resultados precisos. Nessas circunstâncias, nêutrons que já fizeram colisões dentro de um determinado grupo de energia provavelmente farão uma contribuição substancial para a distribuição do fluxo escalar. Numerosas técnicas têm sido usadas para a aceleração das iterações na fonte de espalhamento. Alguns deles, como a aceleração de Chebychev, são adaptações específicas de técnicas de iteração de matriz mais gerais (Adams e Larsen, 2002). Outros como o reequilíbrio da malha faz uso do fato de que a solução convergente deve obedecer à equação de conservação de nêutrons (Ozgener, 2012; Jarrett et. al., 2016). Com esta condição imposta na solução não convergente, é possível obter um procedimento iterativo acelerado que geralmente converge mais rapidamente para a solução correta. Na aceleração sintética, a teoria de difusão invariável é usada como um mecanismo para acelerar a solução numérica (Zika e Adams, 2000; Longoni et. al., 2004; Wang et. al., 2014, Willert et. al., 2014; Picca et. al., 2016). A maioria dos métodos de aceleração podem se tornar instáveis no sentido de que a solução não converge à solução correta no caso de que as malhas não sejam o suficientemente finas.

Neste trabalho, usamos a paralelização do esquema de iteração da fonte no método Diamond Difference (DD) como uma técnica de aceleração para problemas com elevada razão de espalhamento. A continuação, apresentamos as equações SN de balanço espacial do método DD. Na seção 3 e 4 escrevemos as equações auxiliares e deduzimos o esquema de varredura usado. Por último, apresentamos os resultados obtidos para um problema modelo de blindagem associado a um meio material com elevada razão de espalhamento.

2. EQUAÇÕES S_N DE BALANÇO ESPACIAL

Começamos com as equações de transporte de nêutrons em aproximação de ordenadas discretas (S_N), geometria retangular, monoenergéticas, com espalhamento isotrópico aplicadas na célula de discretização espacial $\Omega_{i,j}$ figura 1:

$$(1) \quad \mu_m \frac{\partial}{\partial x} \psi_{m,i,j}(x, y) + \eta_m \frac{\partial}{\partial y} \psi_{m,i,j}(x, y) + \sigma_{t,i,j} \psi_{m,i,j}(x, y) = \frac{\sigma_{s0,i,j}}{4} \sum_{n=1}^N \psi_{n,i,j}(x, y) \omega_n + Q_{i,j}$$

onde $m = 1:N, i = 1:I, j = 1:J, (x, y) \in \Omega_{i,j}$. $\psi_{m,i,j}(x, y)$ representa o fluxo angular de nêutrons que viajam na direção (μ_m, η_m) , onde $\mu_m = \cos \theta$ e $\eta_m = \sqrt{1 - \mu_m^2} \sin \varphi$, $0 \leq \theta \leq \pi, 0 \leq \varphi \leq 2\pi$. $\sigma_{t,i,j}$, $\sigma_{s0,i,j}$ e $Q_{i,j}$ são a sessão de choque macroscópica total, de espalhamento isotrópico e fonte externa, e são assumidas constantes no interior da célula de discretização espacial $\Omega_{i,j}$. N representa a ordem da quadratura que aproxima o termo da fonte por espalhamento no lado direito da equação 1 e $M = N(N + 2)/2$ é o número de direções de discretização angular.

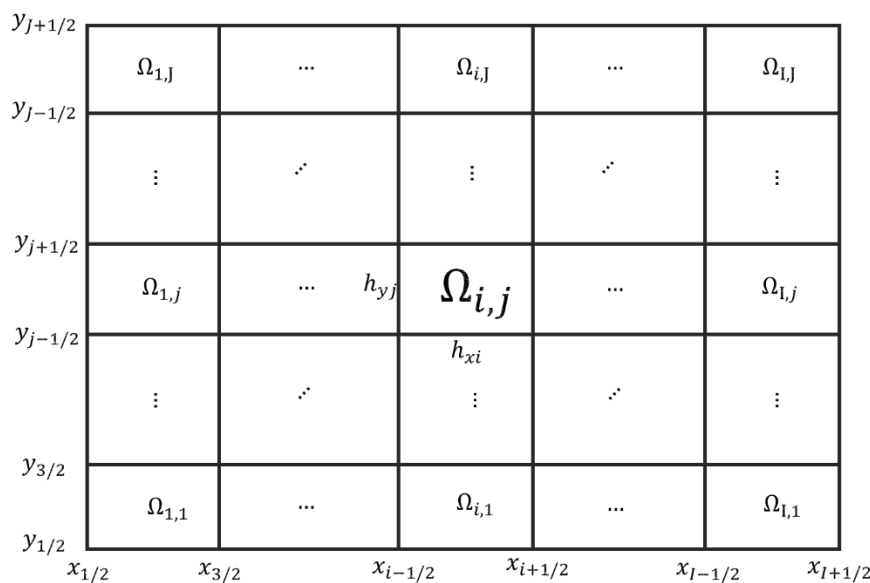


Figura 1. Discretização do domínio de trabalho em $I \times J$ células espaciais

Para obter as equações S_N de balanço espacial de ordem zero aplicamos o operador

$$\frac{(2l+1)(2r+1)}{h_{xi} h_{yj}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} P_l \left[\frac{2(x-x_i)}{h_{xi}} \right] P_r \left[\frac{2(y-y_j)}{h_{yj}} \right] \cdot dx dy \quad (2)$$

usando $l = r = 0$. P_l e P_r representam os polinômios de Legendre de grau l e r , respectivamente. h_{xi} e h_{yj} são a largura e espessura da célula de discretização espacial $\Omega_{i,j}$. x_i e y_j foram definidos como os pontos médios por x e por y nos respectivos nodos. Podemos escrever as equações S_N de balanço espacial como

$$\frac{\mu_m}{\sigma_{t,i,j}h_{xi}}(\hat{\psi}_{m,i+1/2,j} - \hat{\psi}_{m,i-1/2,j}) + \frac{\eta_m}{\sigma_{t,i,j}h_{yj}}(\tilde{\psi}_{m,i,j+1/2} - \tilde{\psi}_{m,i,j-1/2}) + \bar{\psi}_{m,i,j} = \frac{c_{i,j}}{4} \sum_{n=1}^N \bar{\psi}_{n,i,j} \omega_n + \frac{Q_{i,j}}{\sigma_{t,i,j}} \quad (3)$$

onde definimos os fluxos angulares médios nas direções y e x como

$$\hat{\psi}_{m,i\pm 1/2,j} = \hat{\psi}_{m,i,j}(x_{i\pm 1/2}) = \frac{1}{h_{yj}} \int_{y_{j-1/2}}^{y_{j+1/2}} \psi_{m,i,j}(x_{i\pm 1/2}, y) dy \quad (4)$$

$$\tilde{\psi}_{m,i,j\pm 1/2} = \tilde{\psi}_{m,i,j}(y_{j\pm 1/2}) = \frac{1}{h_{xi}} \int_{x_{i-1/2}}^{x_{i+1/2}} \psi_{m,i,j}(x, y_{j\pm 1/2}) dx \quad (5)$$

e o fluxo angular médio na célula de discretização espacial $\Omega_{i,j}$ como

$$\bar{\psi}_{m,i,j} = \frac{1}{h_{xi}h_{yj}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \psi_{m,i,j}(x, y) dx dy \quad (6)$$

Na equação (3), $c_{i,j} = \sigma_{s0,i,j}/\sigma_{t,i,j}$. As equações S_N de balanço espacial constituem um sistema de M equações lineares algébricas com $5M$ incógnitas em cada célula de discretização espacial. Sabendo que as condições de contorno aos lados esquerdo, direito, inferior e superior fornecem $M/2$ equações cada, reduzimos a quantidade de incógnitas do problema. Isto é, $(5M - 2M = 3M)$. Logo, temos como resultado M equações e $3M$ incógnitas. As $2M$ equações faltantes são fornecidas pelo esquema numérico usado. Neste trabalho vamos usar o método Diamond Difference (DD) como esquema numérico modelo (Lewis e Miller, 1984).

3. EQUAÇÕES AUXILIARES E ESQUEMA DE VARREDURA

Para que seja possível a solução única das equações de balanço espacial S_N utilizamos as equações auxiliares do método DD que escrevemos na forma:

$$\bar{\psi}_{m,i,j} = \frac{\hat{\psi}_{m,i+1/2,j} + \hat{\psi}_{m,i-1/2,j}}{2} \quad (7)$$

$$\bar{\psi}_{m,i,j} = \frac{\tilde{\psi}_{m,i,j+1/2} + \tilde{\psi}_{m,i,j-1/2}}{2} \quad (8)$$

Na figura 2 mostramos os varridos usados no esquema iterativo SI (Source Iteration) para resolver numericamente as equações de balanço espacial SN junto com as equações auxiliares do método DD e condições de contorno prescritas (Lewis e Miller, 1984). O esquema de varredura consta de quatro direções de varrido: 1. Varrido do Sudoeste para nordeste (A), 2. Varrido do Sudeste para noroeste (B), 3. Varrido do Nordeste para sudoeste (C), e 4. Varrido do Noroeste para sudeste (D). A continuação, derivaremos as equações de varredura para o caso (A).

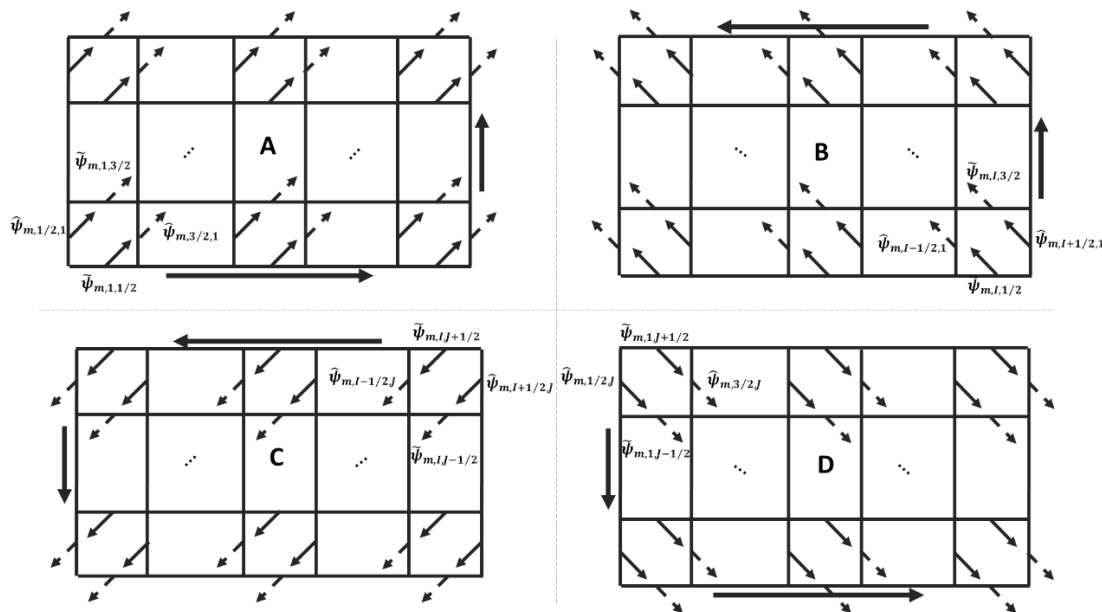


Figura 2. Esquemas de varredura

Primeiramente, explicitamos os fluxos angulares emergentes das equações 7 e 8. Substituímos o resultado nas equações SN e resolvemos para o fluxo angular médio na célula de discretização espacial $\Omega_{i,j}$. Podemos escrever o resultado da forma:

$$\hat{\psi}_{m,i+1/2,j} = 2\bar{\psi}_{m,i,j} - \hat{\psi}_{m,i-1/2,j} \quad (9)$$

$$\tilde{\psi}_{m,i,j+1/2} = 2\bar{\psi}_{m,i,j} - \tilde{\psi}_{m,i,j-1/2} \quad (10)$$

$$\bar{\psi}_{m,i,j} = \frac{\frac{2\mu_m}{\sigma_{t,i,j}h_{xi}}\hat{\psi}_{m,i-1/2,j} + \frac{2\eta_m}{\sigma_{t,i,j}h_{yj}}\hat{\psi}_{m,i,j-1/2} + \frac{c_{i,j}}{4}\sum_{n=1}^N\bar{\psi}_{n,i,j}\omega_n + \frac{Q_{i,j}}{\sigma_{t,i,j}}}{1 + \frac{2\mu_m}{\sigma_{t,i,j}h_{xi}} + \frac{2\eta_m}{\sigma_{t,i,j}h_{yj}}}$$

(11)

O algoritmo do esquema iterativo SI consta de quatro passos:

1. Com o fluxo angular incidente e a estimativa inicial ou da iteração anterior do fluxo angular médio em cada nodo, atualizamos o fluxo angular médio usando a equação 11 no caso A,
2. Calculamos o fluxo emergente a partir do fluxo angular médio atual e os fluxos angulares incidentes usando as equações 9 e 10 no caso A em cada célula de discretização espacial,
3. Percorremos pelas quatro direções de varredura aplicando os passos (1) e (2), e usando a lei da continuidade nas células adjacentes,
4. Atualizamos o termo de fonte de espalhamento e fazemos o mesmo procedimento até satisfazer um critério de parada prescrito.

Estes quatro passos devem ser executados para cada uma das direções de varredura (A, B, C e D, vide Figura 2) ajustando em cada caso os fluxos incidentes e emergentes.

4. PARALELIZAÇÃO DO ESQUEMA ITERATIVO

Para distribuir o processamento entres os núcleos neste trabalho é utilizada a interface de programação de aplicativos OpenMP (Open Multi-Processing) que possibilita a criação de programas paralelos em máquinas de memória compartilhada através da implementação automática e otimizada de um conjunto de threads. O OpenMP não é uma linguagem de programação, ele representa um padrão que define como os compiladores devem gerar códigos paralelos através da incorporação, nos programas sequenciais, de diretivas que indicam como o trabalho será dividido entre os processadores (Chapman, 2008).

Para a paralelização do esquema numérico SI primeiramente precisamos aplicar a técnica de inversão de ciclo no algoritmo descrito anteriormente (Pacheco, 2011). No caso clássico, as direções de varredura são percorridas fazendo um ciclo primário por todas as

células de discretização espacial e um ciclo secundário pelas direções dos nêutrons. No ciclo primário têm dependências entre nodos adjacentes, mas no ciclo secundário, não há dependências entre direções. A estratégia de inversão de ciclos consiste em paralelizar as direções pelas que viajam os nêutrons, isto é, distribuir equitativamente as direções de discretização angular entre os processadores disponíveis. Portanto, nosso código é baseado em um ciclo primário donde se percorrem todas as direções pelas quais viajam os nêutrons, e depois um ciclo secundário percorrendo todas as células de discretização espacial. A paralelização é feita no ciclo primário onde cada grupo de direções é tratada independentemente nos processadores disponíveis.

Por outro lado, no código serial geralmente o cálculo do fluxo escalar, requerido para a atualização do termo de fonte de espalhamento, é feito dentro do ciclo primário. Neste caso, identificamos uma sessão crítica onde o tempo de execução é prejudicado pois em cada atualização é preciso a comunicação entre processadores. Para evitar problemas de rendimento em nosso código, fazemos um ciclo serial adicional fora do ciclo primário para atualização do termo de fonte de espalhamento dentro de cada iteração.

3. RESULTADOS E DISCUSSÃO

O problema modelo corresponde a um cálculo de blindagem para um meio material com alta razão de espalhamento (Barros, 1990). A geometria do problema é mostrada na figura 3. O problema consta de dois zonas. Na zona 1 as sessões macroscópicas total e de espalhamento isotrópico são 0.80 cm^{-1} e 0.00 cm^{-1} , respetivamente. Na zona 2 temos 1.00 cm^{-1} e 0.95 cm^{-1} , respetivamente

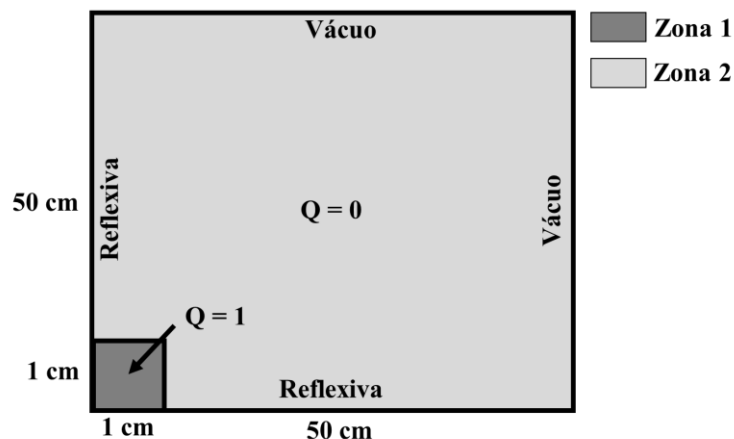


Figura 3. Geometria do problema modelo

Na Tabela 1 mostramos a estimação da fuga de nêutrons pela fronteira da blindagem para diferentes grades de discretização espacial. Devido às condições de simetria do problema, a fuga de nêutrons pela fronteira superior é igual à fuga de nêutrons pela fronteira direita. Os cálculos foram realizados utilizando a quadratura LQn para ordem 6 (Lewis e Miller, 1986), e o critério de parada das iterações foi tal que a norma máxima do desvio relativo do fluxo escalar médio da célula espacial não excedesse a 1×10^{-5} . A solução de referência considerada neste problema foi a reportada por (Barros, 1990) no caso do método SGF-CN para uma grade espacial de nodos. Na medida que a grade espacial é mais fina os resultados das fugas de nêutrons pelos contornos direito e superior gerados pelo método DD tendem a um valor fixo. O método é instável para os casos de grades espaciais 100×100 e 200×200 onde são reportados valores da fuga de nêutrons negativos. Na figura 4 mostramos a distribuição do fluxo escalar de nêutrons para o caso de uma grade espacial de 200×200 . Os valores em azul escuro representam os fluxos escalares não-físicos com valores negativos gerados em esse caso.

Tabela 1. Fuga de nêutrons gerada pelo método DD para várias grades de discretização espacial.

Método	Grade espacial	Fuga de nêutrons (n/cm ²)
DD	100x100	-1.782e-08
	200x200	-6.737e-10
	500x500	5.526e-09
	1000x1000	5.911e-09
SGF-CN	500x500	5.998e-09

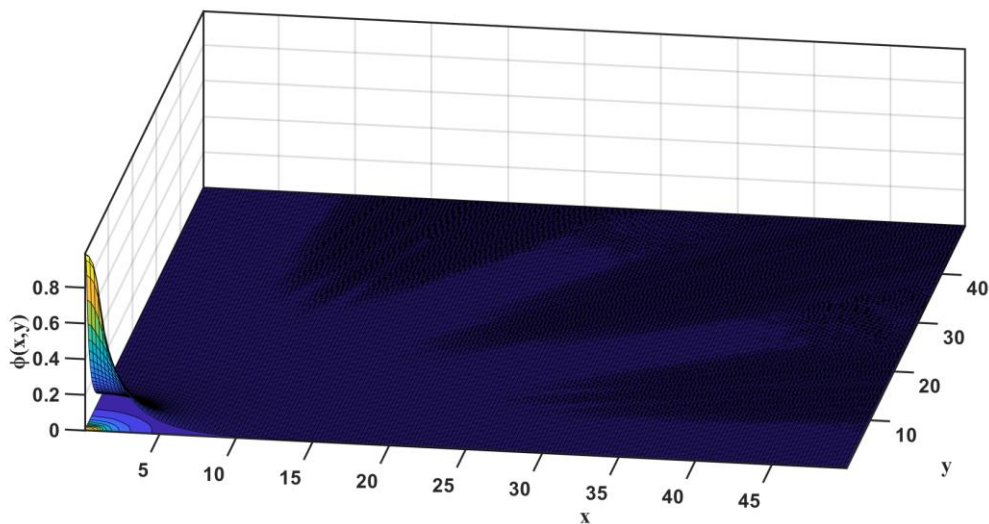


Figura 4. Fluxo escalar de nêutrons para uma grade de discretização espacial 200 x 200.

Para os testes das versões serial e paralela foi usado um computador com o sistema operativo (SO) Ubuntu 16.04 de 64 bits com dois processadores XEON E5-2420 Hexa Core da Intel, totalizando 12 núcleos físicos e 12 núcleos hyper-threading, disponibilizando 24 núcleos com velocidade de 1.90 GHz; e 16 GB de memória RAM. Na Tabela 2 mostramos os resultados dos tempos de execução variando a quantidade de processadores usados para executar a versão paralela. Na tabela 2 reportamos as métricas de programas paralelos (speedup e eficiência) nos experimentos numéricos realizados. Para os testes foi usada uma grade de discretização espacial de 1000 x 1000 nodos e uma quadratura LQn de ordem 6. Na figura 5 mostramos o speedup do código em função do número de processos. De forma geral ao usarmos um maior número de processos conseguimos tempo de processamento menores. O menor tempo de execução foi obtido ao usar os 24 processadores disponíveis conseguindo um speedup de 7.53 em comparação com a simulação serial. O speedup apresenta comportamento sublinear, com valores próximo do comportamento linear para poucos processos, e um forte afastamento do comportamento linear ao aumentar o número de processos. Ao analisarmos a eficiência, nota-se que entre 2 e 12 processos os valores de eficiência são satisfatórios com resultados entre 0,5 e 0,8. Entretanto, para 24 processos se observa uma queda pronunciada na eficiência.

Tabela 2. Tempo de execução e aceleração dos resultados do problema modelo para o caso de uma grade de discretização espacial de 1000 x 1000 nodos.

CPUs	Tempo em CPUs	SpeedUp	Eficiência
1	1929	-	-
2	1269	1.52	0.76
4	689	2.80	0.70
6	579	3.33	0.56
8	436	4.42	0.55
12	305	6.32	0.53
24	256	7.53	0.31

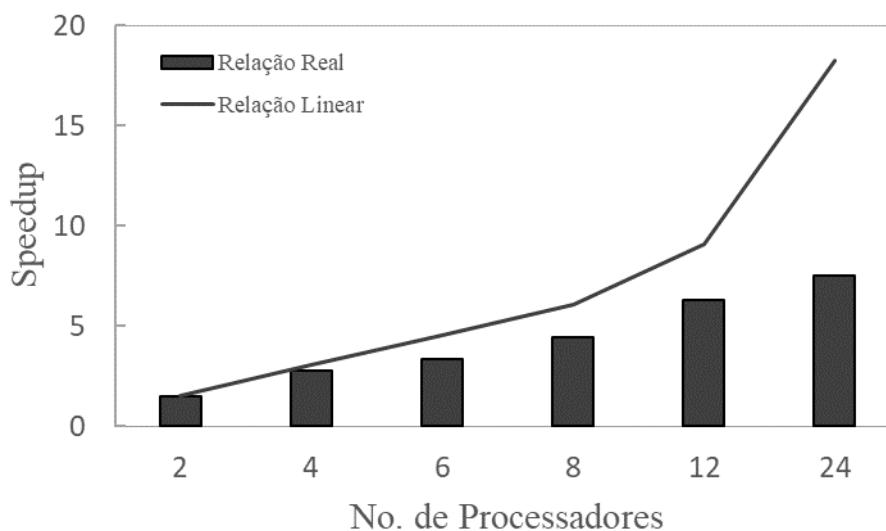


Figura 5. Speedup v.s. numero de processadores.

5. CONCLUSÕES

Neste trabalho mostramos os resultados da paralelização com OpenMP do método DD, como técnica de aceleração, para um problema de cálculo de blindagem em um meio material com elevada razão de espalhamento. Conforme esperado, os resultados gerados pelo método convergem para um valor fixo em grades espaciais finas e para os casos de malhas grossas o método apresenta um comportamento instável. O menor tempo de execução foi de 256 s atingindo a maior aceleração ao usar os 24 CPUs, o máximo número disponível na máquina paralela disponível. Em relação as métricas paralelas de desempenho (speedup e eficiência) nota-se um deterioro dos valores ao aumentar o número de processos, o que sugere que melhorias podem ser introduzidas no código paralelo.

Os desdobramentos futuros deste trabalho, envolvem o aprimoramento do código paralelo usando OpenMP, e a utilização de um paradigma paralelo híbrido que combine MPI (Message Passing Interface) com OpenMP e permita utilizar máquinas paralelas de memória distribuída.

REFERÊNCIAS

- ADAMS, M.L., LARSEN, E.W. Fast iterative methods for discrete-ordinates particle transport calculations. **Progress in Nuclear Energy**, 40, 3-159, 2002
- BARROS, R.C. **A spectral Nodal method for the solution of discrete ordinates problems in one and two dimensional cartesian geometry**. 1990 Tese (Doutorado em Engenharia Nuclear) – University of Michigan, USA
- CHAPMAN, B.; JOST, G.; VAN DER PAS, R. **Using OpenMP portable shared memory parallel programming**. MIT Press. 2008.
- JARRET, M.; KOCHUMAS, B.; ZHU, A.; DOWNAR, T. Analysis of stabilization techniques for CMFD acceleration of neutron transport problems. **Nuclear Science and Engineering**, 184, 208-227, 2016
- LEWIS, E.E.; MILLER, J.W. **Computational methods of neutron transport**. Jhon Wiley & Sons, New York, 1 ed., 1984
- LONGONI, G.; HAGHIGHAT, A.; SJODEN, G. A new synthetic acceleration technique based on the simplified even-parity SN equations. **Transport Theory and Statistical Physics**, 33, 347-360, 2004
- OZGENER, B. An assessment of acceleration techniques in scattering source iterations. **Fusion Science and Technology**, 61, 308-313, 2012
- PACHECO, P.S. **An introduction to parallel programming**. Morgan Kaufmann, Burlington, MA 01803, USA, 2011
- PICCA, P.; FURFARO, R.; GANAPOL, D. Application of non-linear extrapolations for the convergence acceleration of source iteration. **Journal of Computational and Theoretical Transport**, 0, 1-17, 2016
- WANG, Y., ZHANG, H., MATINEAU, R.C. Diffusion acceleration schemes for self-adjoint angular flux formulation with void treatment. **Nuclear Science and Engineering**, 176, 201-225, 2014
- WILLERT, J., PARK, H., KNOLL, D.A. A comparison of acceleration methods for solving the neutron transport k-eigenvalue problem. **Journal of Computational Physics**, 274, 681-694, 2014
- ZIKA, M.R., ADAMS, M.L. Transport synthetic acceleration with opposing reflecting boundary conditions. **Nuclear Science and Engineering**, 134, 159-170, 2000