

Modelagem Acústica por Diferenças Finitas utilizando OpenMP

Antônio José de Medeiros Filho¹, José Jeronimo Camata²

¹Grupo de Educação Tutorial do Curso de Engenharia Computacional – Universidade Federal de Juiz de Fora (UFJF)
CEP 36.036-330 – Juiz de Fora – MG – Brasil

²Departamento de Ciência da Computação – UFJF
Juiz de Fora – MG – Brasil

antonio.medeiros@engenharia.ufjf.br, camata@ice.ufjf.br

Resumo. A modelagem de propagação de ondas sísmicas têm sido objeto de estudo por décadas, principalmente pela indústria do petróleo e do gás. Neste trabalho implementa-se a propagação de ondas em um meio homogêneo bidimensional por diferenças finitas de quarta ordem. Paralelismo com OpenMP é empregado como estratégia de melhoria de desempenho. O programa é testado com diferentes casos de testes. O emprego do paralelismo apresenta uma boa escalabilidade.

Abstract. Seismic wave propagation has been the subject of study for decades, mostly by the oil and gas industry. In this work, wave propagation is implemented in a homogeneous two-dimensional medium using fourth order finite differences. OpenMP parallelism is employed as a performance improvement strategy. Different test cases are used to test our code. The use of parallelism presents a good scalability.

1. Introdução

A modelagem computacional é uma ferramenta muito utilizada na geofísica para estudos de sismologia. Este trabalho tem o objetivo de implementar um programa capaz de modelar a propagação de ondas sísmicas bidimensionais, considerando meios com diferentes campos de velocidades, usando diferenças finitas. Além disso, utiliza diretivas OpenMP para paralelizar os laços que demandam os principais recursos computacionais.

2. Modelagem Acústica Bidimensional

Considerando um domínio bidimensional Ω , a equação pode ser escrita na forma:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} + f(x, z, t) \quad (1)$$

onde u é o valor do campo de pressão, c a velocidade do meio e f é o termo fonte.

Dividimos então o domínio em n_x células na direção x e n_z células na direção z . Por simplicidade, n_x e n_z são escolhidos de tal forma que $\Delta x = \Delta z = h$. Fazendo uma

aproximação [DA SILVA and ANCELME 2015] por séries de Taylor de quarta ordem no espaço e segunda no tempo, obtém-se a forma discreta da equação 1:

$$\begin{aligned}
 u_{i,j}^{k+1} = & \frac{1}{12} \left(c \frac{\Delta t}{h} \right)^2 [-(u_{i-2,j}^k + u_{i,j-2}^k) + 16(u_{i-1,j}^k + u_{i,j-1}^k) \\
 & - 60u_{i,j}^k + 16(u_{i+1,j}^k + u_{i,j+1}^k) - (u_{i+2,j}^k + u_{i,j+2}^k)] + 2u_{i,j}^k - u_{i,j}^{k-1} \\
 & - (c \cdot \Delta t)^2 f(i, j, k)
 \end{aligned} \quad (2)$$

Para evitar a reflexão da onda nas fronteiras, este trabalho utiliza a condição de contorno não reflexiva de Reynolds [Reynolds 1978], junto as condições de camadas absorptivas proposta por Cerjan [Cerjan et al. 1985]. O principal custo computacional está relacionado ao cálculo do campo de pressão (Eq. 2) e do tratamento das condições de contorno. Por se tratar de um problema transiente, esses laços são calculados n_t vezes. Para reduzir esse custo, diretivas OpenMP foram inseridas nesses laços de modo a paralelizá-los, conforme pseudo-código abaixo:

```

#pragma omp parallel for private (...)
for (int j = zstart; j <= zend; j++){
    for (int i = xstart; i <= xend; i++){
        // ...
    }
}

```

3. Resultados

Três modelos de campos de velocidades foram utilizados nos testes. Dois modelos de velocidade sintéticos (veja Figura 1) e um modelo real da bacia de Marmousi [Versteeg 1994]. O domínio dos dois primeiros modelos são discretizados em uma malha cartesiana com 300×300 subdivisões nas direções x e z e 8000 passos do tempo são empregados na integração temporal. Já para o modelo Marmousi, usou-se uma discretização com 13600×2800 pontos e 16000 passos de tempo. Os testes foram executados em

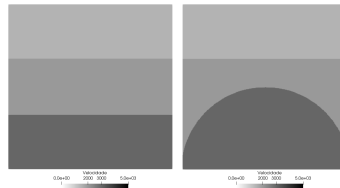


Figura 1. Modelos de planos paralelos, semi-circunferência, respectivamente.

uma máquina Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz, sistema operacional x86_64, 20.04.2-Ubuntu, compilando com o GCC versão 9.3.0. As flags de otimização usados foram `-march=native -ffast-math -O2 -fopenmp -ftree-vectorize`.

Nos três modelos considera-se uma fonte sísmica posicionada no topo do modelo, com 250m de profundidade. A Tabela 1 apresenta a média dos tempos de 10 execuções e o speedup para cada um dos modelos e para diferentes números de threads. A melhor eficiência paralela ocorreu para o modelo Marmousi com 88% e 67% ao executar com 2 e 4 threads, respectivamente. A utilização do hyper-threading com 8 núcleos virtuais não apresentou melhoria significativa.

<i>threads</i>	Planos Paralelos		Semicírculo		Marmousi	
	Tempo (s)	Speed Up	Tempo (s)	Speed Up	Tempo (s)	Speed Up
1	6,6808	1	6,6538	1	3869,4	1
2	4,2452	1,5737	4,3011	1,5470	2194,6	1,7632
4	3,6813	1,8148	3,73911	1,7795	1440,3	2,6866

Tabela 1. Tempo de execução para diferentes números de threads

Snapshots foram registrados em arquivos no formato VTI (VTK ImageData). Para contornar gargalos no pós-processamento do modelo Marmousi ocasionados pela alto requisito de memória, usou-se uma técnica de automação de pós-processamento através do módulo ParaView Python¹. A principal vantagem dessa técnica é a possibilidade de gerar as imagens a partir dos snapshots escritos em tempo de execução, sem a necessidade de aplicar o software de visualização. Além disso, há a possibilidade de identificar precocemente possíveis falhas. A Figura 2 mostra o campo de pressão do modelo Marmousi no instante final gerado por esse esquema.

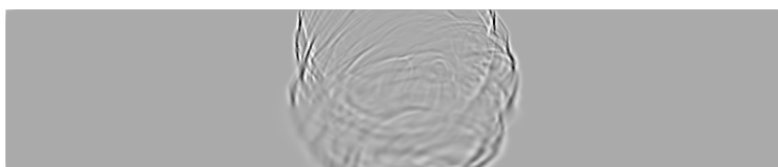


Figura 2. Campo de pressão no tempo final: Marmousi

4. Conclusão

Neste trabalho apresentamos uma implementação do problema propagação de ondas por diferenças finitas de quarta ordem. Como estratégia de melhoria de desempenho, optou-se na paralelização dos principais laços usando OpenMP. Os resultados demonstraram que as reflexões sísmicas observadas nas interfaces entre os campos de velocidades estão condizentes com a literatura. Além disso, houve uma redução significativa do tempo, principalmente para o maior modelo do estudo. Entretanto, uma análise de desempenho mais detalhada deve ser empregada em trabalho futuro para medir o custo de sincronização das threads openmp. Por fim, ressalta-se que a utilização de automação da geração das imagens sísmicas ParaView Python permitiu acelerar a análise dos resultados obtidos.

Referências

- Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M. (1985). A nonreflecting boundary condition for discrete acoustic and elastic wave equations. *Geophysics*, 50(4):705–708.
- DA SILVA, J. J. and ANCELME, R. L. (2015). Introdução ao método sísmico.
- Reynolds, A. C. (1978). Boundary conditions for the numerical solution of wave propagation problems. *Geophysics*, 43(6):1099–1110.
- Versteeg, R. (1994). The marmousi experience: Velocity model determination on a synthetic complex data set. *The Leading Edge*, 13(9):927–936.

¹<https://www.paraview.org/python/>