



# Migração reversa no tempo 2-D e 3-D: Estratégias para reduzir o custo computacional em ambiente paralelo

Adriano W. G. dos Santos\* e Reynam C. Pestana, CPGG/UFBA e INCT-GP/CNPQ

Copyright 2011, SBGf - Sociedade Brasileira de Geofísica.

This paper was prepared for presentation at the Twelfth International Congress of the Brazilian Geophysical Society, held in Rio de Janeiro, Brazil, August 15-18, 2011.

Contents of this paper were reviewed by the Technical Committee of the Twelfth International Congress of The Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of The Brazilian Geophysical Society is prohibited.

## Abstract

**In this paper, we discuss the computational costs of prestack reverse time migration, specially in a parallel environment, where the amount of storage required is high, due to the number of shots being processed simultaneously. This storage space is necessary to record the sections generated on modeling phase, which will be used later on the imaging condition. We show that it is possible to reduce the free space requirement, by saving only few sections spaced in time. The discarded sections can then be later restored with the modeling being held backwards in time. Another approach presented is the spatial decomposition of the section, resulting on a smaller amount of data processed at each computing node. The tests were done with 2-D and 3-D models, using the parallel interface MPI (Message Passing Interface). Results indicate that it is possible to drastically reduce the storage required, keeping the quality of the migrated section, however, with some increase in processing time.**

## Introdução

Durante alguns anos a migração reversa no tempo (RTM) permaneceu como tema de trabalhos predominantemente teóricos. Em 1983, quando o método foi descrito por Baysal et al. (1983), os computadores eram mais caros e tinham menos poder computacional, limitando a migração ao domínio pós-empilhamento.

Atualmente, empresas e instituições já podem contar com computadores de alto desempenho, como os *clusters* de computadores. Além disso, já é possível escrever programas para GPUs e FPGAs, mais rápidos que um processador convencional, diminuindo o tempo de execução de cada tiro, e, conseqüentemente, o tempo total. Portanto, hoje é possível realizar uma RTM pré-empilhamento em tempo hábil, o que a fez ganhar atenção renovada da comunidade de imageamento sísmico e da indústria.

Neste trabalho, utilizamos uma implementação voltada para um *cluster* de computadores, através da interface MPI. O processamento de cada tiro é delegado a um nó de computação diferente. Desta forma, vários tiros são processados simultaneamente, reduzindo o tempo de

execução de cada grupo de tiros ao de aproximadamente um único tiro.

Apesar de trazer grandes benefícios ao tempo total de execução, esta abordagem potencializa um problema presente na RTM: a necessidade de armazenar as seções geradas na fase de modelagem, que serão utilizadas na condição de imagem. O espaço exigido será multiplicado pelo número de processos simultâneos.

Este trabalho apresenta soluções implementadas a fim de amenizar este problema. Duas abordagens principais foram testadas. A primeira consiste em salvar apenas uma parte das seções, restaurando aquelas que foram descartadas através da modelagem realizada de forma reversa no tempo, durante a fase de migração. A segunda abordagem consiste em decompor as seções espacialmente, assim, cada tiro será dividido entre várias unidades de processamento, reduzindo a quantidade de dados presente em cada uma delas.

Os testes foram realizados no cluster Aguiá, do CPGG-UFBA, utilizando até 120 nós idênticos. Cada nó consiste em um processador Intel Xeon E5260 2.40GHz, com cache de 12Mb.

## Migração reversa no tempo (RTM)

Para realizar a migração reversa no tempo de um dado sísmico é preciso antes realizar a modelagem do seu correspondente modelo de velocidades, desde o tempo  $t = 0$  até o tempo  $t = t_{max}$ , armazenando o resultado em  $s(x, y, z, t)$ , para cada ponto do modelo. Em seguida, o dado sísmico é injetado no modelo de velocidades e é propagado desde o tempo  $t = t_{max}$  até o tempo  $t = 0$ , com o resultado armazenado em  $r(x, y, z, t)$ . A imagem final  $I$  é construída segundo a equação (1). A correlação destes campos de onda faz com que apenas os sinais presentes tanto na modelagem quanto na migração apareçam na seção migrada, tornando visíveis os refletores.

$$I(x, y, z) = \sum_{t=0}^{t_{max}} s(x, y, z, t) r(x, y, z, t) \quad (1)$$

A migração reversa no tempo utiliza a equação completa da onda, mostrada em (2), por isso, é mais genérica do que qualquer outro método de migração baseado na equação da onda unidirecional, pois permite a propagação da onda em qualquer direção sem limitação de mergulho. Além disso, pode contemplar quaisquer variações de velocidade e evitar problemas com ondas evanescentes (Baysal et al., 1983).

$$\frac{1}{c(x, y, z)^2} \frac{\partial^2 P(x, y, z, t)}{\partial t^2} = \nabla^2 P(x, y, z, t) \quad (2)$$

onde  $\nabla^2$  é o operador Laplaciano, dado por  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ ,  $P$  é o campo de onda,  $c$  é a velocidade de propagação,  $x$ ,  $y$  e  $z$  são as coordenadas espaciais e  $t$  é o tempo.

Primeiro, aproximamos a derivada segunda no tempo por:

$$\frac{\partial^2 P}{\partial t^2} \equiv \frac{P(t + \Delta t) - 2P(t) + P(t - \Delta t)}{\Delta t^2} \quad (3)$$

Então, o campo de onda no tempo  $t + \Delta t$  pode ser aproximado por:

$$P(t + \Delta t) = 2P(t) - P(t - \Delta t) + \Delta t^2 c^2 \nabla^2 P(t) \quad (4)$$

O cálculo do operador Laplaciano ( $\nabla^2$ ), pode ser feito de muitas formas. Neste trabalho foram testados dois métodos de extrapolação do campo de onda: diferenças finitas (Alford et al., 1974) e o método de expansão rápida (REM) (Pestana and Stoffa, 2010).

#### Diferenças Finitas

O método de diferenças finitas substitui as operações diferenciais da equação por aproximações utilizando truncamento da série de Taylor. Quanto maior for o operador de diferenças finitas, mais termos da série estarão presentes na aproximação e mais preciso será o resultado, porém mais cálculos serão necessários.

A derivada segunda em  $x$  é aproximada por:

$$\frac{\partial^2 P}{\partial x^2} = w_0 P_0 + \sum_{n=1}^{n/2} w_n [P(x + n\Delta x) + P(x - n\Delta x)] \quad (5)$$

onde  $w_n$  são os coeficientes do operador de diferenças finitas.

Utilizando um operador de segunda ordem, por exemplo, obtemos:

$$\frac{\partial^2 P(x, y, z, t)}{\partial x^2} \approx \frac{P(x + \Delta x, y, z, t) - 2P(x, y, z, t) + P(x - \Delta x, y, z, t)}{(\Delta x)^2} \quad (6)$$

O uso do esquema de diferenças finitas requer que uma determinada condição de estabilidade seja respeitada. Neste trabalho, utilizamos a relação dada por Lines et al. (1999) para determinar um intervalo máximo de amostragem no tempo. A equação (7) mostra esta relação para o caso 2-D.

$$\Delta t_{max} < \frac{2 \sqrt{\frac{2}{1/\Delta x^2 + 1/\Delta z^2}}}{c_{max} \sqrt{\mu}} \quad (7)$$

onde  $\mu$  é a soma dos valores absolutos dos coeficientes do operador de diferenças finitas, ou seja:

$$\mu = \sum_{-n/2}^{n/2} |w_x| + |w_z| \quad (8)$$

#### Método de expansão rápida (REM)

Para permitir uma melhor aproximação da derivada segunda no tempo, além de prover uma base para uma

solução recursiva que permite o avanço no tempo em grandes intervalos, utilizamos o método de expansão rápida (Pestana and Stoffa, 2010).

Na implementação deste método, as derivadas espaciais são calculadas usando transformadas de Fourier. Este método usa a solução exata da equação da onda, dada por:

$$P(x, y, z, t + \Delta t) + P(x, y, z, t - \Delta t) = 2 \cos(L\Delta t) P(x, y, z, t) \quad (9)$$

onde  $\bar{L}^2 = c^2(x, y, z) \nabla^2$ .

A solução da equação (9) se reduz a expansão da função cosseno. Segundo a abordagem apresentada por Tal-Ezer et al. (1987) a função cosseno pode ser expandida da seguinte forma:

$$\cos(L\Delta t) = \sum_{k=0}^{\infty} C_{2k} J_{2k}(\Delta t R) Q_{2k} \left( \frac{iL}{R} \right) \quad (10)$$

onde  $C_{2k} = 1$  para  $k = 0$  e  $C_{2k} = 2$  para  $k > 0$ . O valor  $J_{2k}$  representa as funções de Bessel de ordem  $2k$ , e  $Q_{2k}(w)$  são os polinômios modificados de Chebyshev, que são calculados segundo a relação:

$$Q_{k+2}(w) = 2(1 + 2w^2) Q_k(w) - Q_{k-2}(w) \quad (11)$$

que é iniciada por  $Q_0(w) = 1$  e  $Q_2(w) = 1 + 2w^2$ , onde  $w = iL/R$ .

O termo  $R$  é um escalar associado ao maior autovalor de  $-\bar{L}^2$ . Para o caso 2-D, o valor de  $R$  é dado por  $R = \pi c \sqrt{(1/\Delta x^2) + (1/\Delta z^2)}$ . Em geral, o valor de  $c$  deve ser substituído por  $c_{max}$ , o maior valor de velocidade do modelo.

O somatório da expressão (10) converge exponencialmente para  $k > \Delta t R$ . Portanto, a aproximação do cosseno pode ser obtida com boa precisão a partir de valores de  $k$  maiores que  $\Delta t R$ , dispensando a reamostragem do dado sísmico, como é feito no método de diferenças finitas.

#### Soluções Implementadas

Os programas foram implementados no sistema mestre/escravo, a partir da interface MPI. O papel do mestre é ler o dado de entrada, designar tiros aos nós escravos e reunir os resultados parciais, gerando a imagem final. O papel dos escravos é receber tiros do mestre, processá-los e, ao fim, enviar os resultados parciais para o mestre.

As Figuras 1 e 2 mostram o tempo de processamento dos modelos Marmousi (Versteeg, 1994) e SEG/EAGE 3-D (Aminzadeh et al., 1996), respectivamente, à medida em que o número de nós de processamento é aumentado. É possível perceber uma queda progressiva no tempo total de execução.

Apesar de diminuir notavelmente o tempo total de processamento, este tipo de implementação potencializa um problema presente na RTM: o armazenamento temporário das seções geradas durante a fase de modelagem. As Figuras 3 e 4 mostram o espaço médio necessário para armazenar todas as seções de um tiro, para os dados em questão.

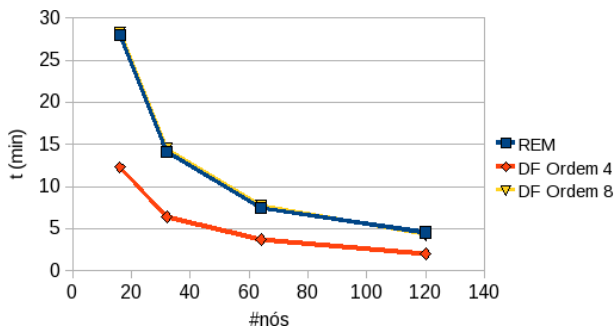


Figure 1: Tempo de processamento do modelo Marmousi

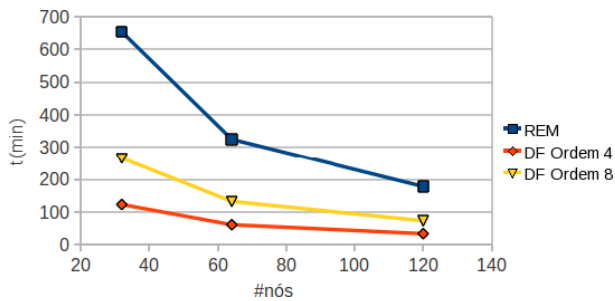


Figure 2: Tempo de processamento de 118 tiros do modelo SEG/EAGE 3-D

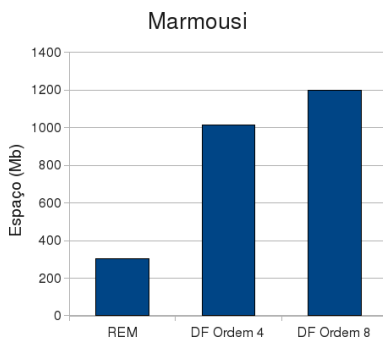


Figure 3: Espaço médio de armazenamento das seções geradas na modelagem de um tiro do modelo Marmousi

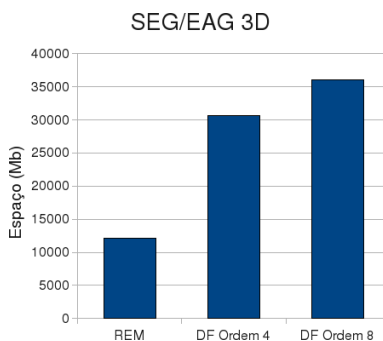


Figure 4: Espaço médio de armazenamento das seções geradas na modelagem de um tiro do modelo SEG/EAGE 3-D

Vale ressaltar que estes valores são multiplicados pelo número de processos simultâneos. Por exemplo, quando utilizamos 120 processos, o espaço necessário para guardar temporariamente as seções do dado SEG/EAGE 3-D supera 4 Tb. Este quadro é ainda pior se considerarmos o efeito negativo que as sucessivas operações de leitura/escrita no disco rígido teriam sobre o desempenho do programa.

Outro aspecto importante notado no gráfico é a variação de acordo com o método de extrapolação utilizado. Isto se deve a condição de estabilidade utilizado no método de diferenças finitas (equação (7)), que limita o intervalo de amostragem no tempo a um valor máximo. Quando o valor original superar este máximo, o número de iterações no tempo terá de ser aumentado.

*Checkpointing*

Uma das formas de reduzir esta demanda é salvar apenas algumas seções espaçadas no tempo, chamadas *checkpoints*. As seções descartadas podem ser recuperadas durante a fase de migração através da modelagem reversa no tempo, partindo dos dois últimos tempos, utilizando a equação (4). Esta técnica, chamada de *checkpointing*, é mostrada por Symes (2007).

A Figura 5 mostra a queda do armazenamento médio de cada tiro para o dado Marmousi, em função do intervalo entre *checkpoints*. É possível observar uma redução drástica do espaço necessário, seguindo o comportamento de uma função do tipo  $f(g) = k/g$ , onde  $k$  é uma constante. Neste contexto,  $k$  é o tamanho médio ocupado por todos as seções da modelagem de um tiro e  $g$  é o intervalo entre os *checkpoints*.

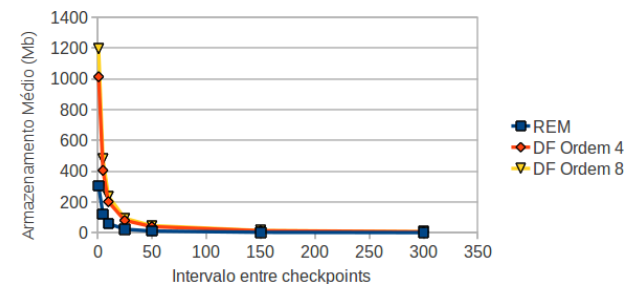


Figure 5: Variação do espaço de armazenamento em função do intervalo entre *checkpoints* para o dado Marmousi

Entretanto, existe um custo adicional no tempo de execução da migração, devido ao cálculo feito para recuperar as seções descartadas. A Figura 6 ilustra bem este efeito.

Além disso, o intervalo entre *checkpoints* escolhido terá efeito sobre a qualidade do resultado final. Podemos perceber isso mais claramente na Figura 7, que mostra o resultado do processamento do dado Marmousi com intervalos de 1, 50 e 300. É possível ver em destaque como os refletores foram melhor imageados quando se utilizou um intervalo menor.

Isto deve-se ao fato da função atenuante, aplicada às bordas do modelo, enfraquecer o campo de onda nesta região. Assim, não é possível recuperar as seções

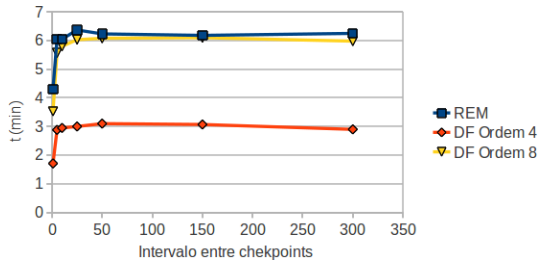


Figure 6: Variação do tempo de execução em função do intervalo entre checkpoints para o dado Marmousi

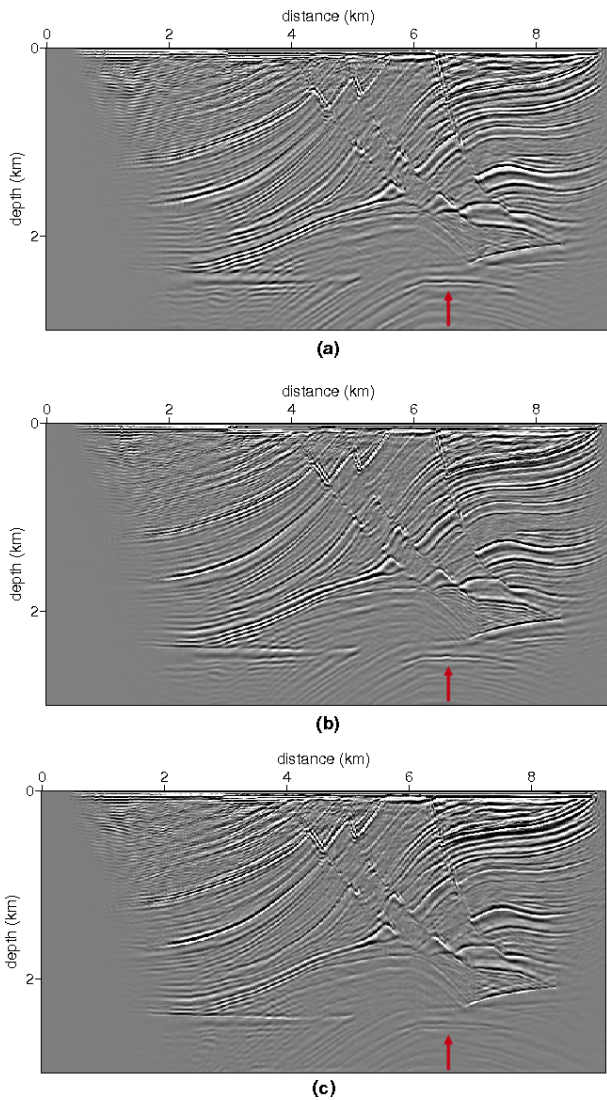


Figure 7: Resultado da migração do dado Marmousi utilizando o REM. Intervalos entre checkpoints de (a) 1, (b) 50 e (c) 300

descartadas perfeitamente, pois parte da informação foi perdida. Os checkpoints servirão como um ponto de restauração, possibilitando uma reprodução mais fiel das seções descartadas. A Figura 8a mostra o campo de onda em um instante de tempo  $t$  durante a modelagem. As Figuras 8b e 8c mostram o campo recuperado no mesmo instante  $t$ , sem e com o uso de checkpoints, respectivamente.

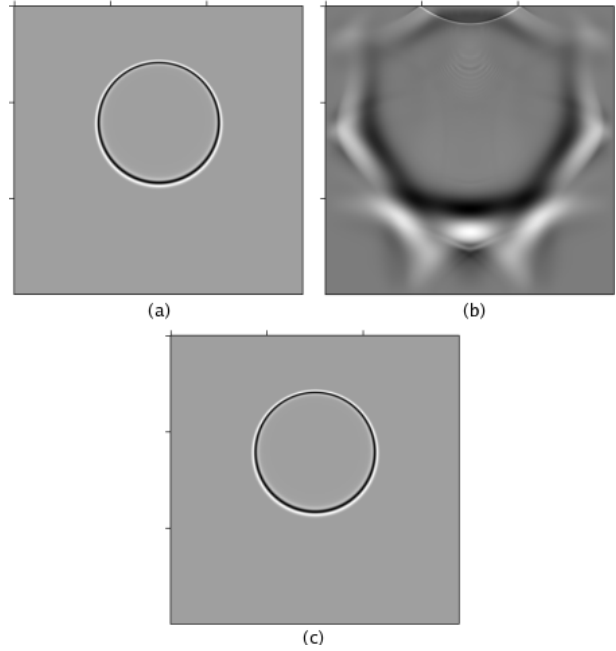


Figure 8: Campos de onda em um instante  $t$ . (a) Campo original, (b) Campo recuperado sem checkpoints e (c) Campo recuperado com checkpoints

Neste trabalho, foi implementada uma nova abordagem, proposta por Clapp (2009). A ideia principal é não mais aplicar uma função atenuante à região de borda, mas sim preenchê-la com valores aleatórios, de forma que não haverá correlação entre o campo de onda da fonte e dos receptores. Com esta técnica é possível recuperar as seções descartadas com boa precisão, já que não há perda de informação, dispensando o uso de checkpoints. As Figuras 9a e 9b mostram o campo de onda nos instantes 0.6s e 1.52s, enquanto as Figuras 9c e 9d mostram os mesmos instantes durante a modelagem reversa, sem o uso de checkpoints.

A Figura 10a mostra o resultado da migração reversa no tempo aplicada ao dado Sigsbee2a, com o uso da borda aleatória e sem checkpoints. Nota-se que os refletores na parte superior do modelo não foram tão bem imageados como quando usamos a borda atenuante e checkpoints (Figura 10b). No entanto, se compararmos com o resultado da Figura 10c, quando usamos a borda atenuante sem checkpoints, fica claro que houve uma melhora. Espera-se que um cálculo mais aprimorado dos valores aleatórios na borda possa melhorar ainda mais o resultado.

*Decomposição espacial*

Outra forma de tratar o problema é dividir um tiro em mais de um processo, assim, a quantidade de dados em cada

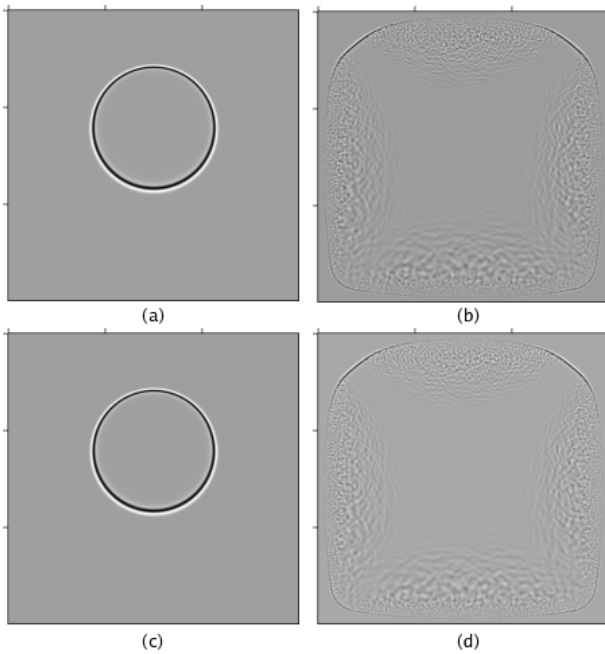


Figure 9: Campos de onda nos instantes: (a) 0.6s, (b) 1.52s, (c) 0.6s durante a modelagem reversa e (d) 1.52 durante a modelagem reversa.

nó de computação será menor, possibilitando salvar um maior número de seções.

Neste trabalho, a decomposição espacial foi testada apenas em modelos 3-D. A Figura 11 ilustra graficamente a decomposição de um cubo de dados entre 4 processos, que devem trocar informações entre si para que a extrapolação do campo seja realizada.

Por ocupar mais de um nó de computação com cada tiro, esta técnica diminuirá o número de processamentos realizados simultaneamente. Por exemplo, se utilizarmos 4 processos para cada tiro, com 120 nós será possível processar 30 tiros ao mesmo tempo. Entretanto, espera-se que o tempo de execução de um único tiro seja reduzido, compensando este fato.

A Figura 12 mostra o tempo de execução à medida em que o tamanho do grupo de processamento de cada tiro aumenta. Nota-se um aumento no tempo, devido ao custo adicional das mensagens trocadas entre os membros deste grupo. A Figura 13 mostra o tempo de processamento de um único tiro. Como era esperado, houve uma diminuição à medida que usamos mais processos.

**Conclusões**

Neste trabalho, implementamos duas técnicas capazes de reduzir o espaço de armazenamento temporário exigido pela migração reversa no tempo pré-empilhamento: *checkpointing* e decomposição espacial. Foi possível observar que as técnicas conseguiram redução significativa na demanda por espaço temporário. Porém, é preciso levar em conta os efeitos sobre o tempo de execução e a qualidade do resultado. O melhor equilíbrio entre esses fatores dependerá do dado a ser migrado e do poder computacional disponível.

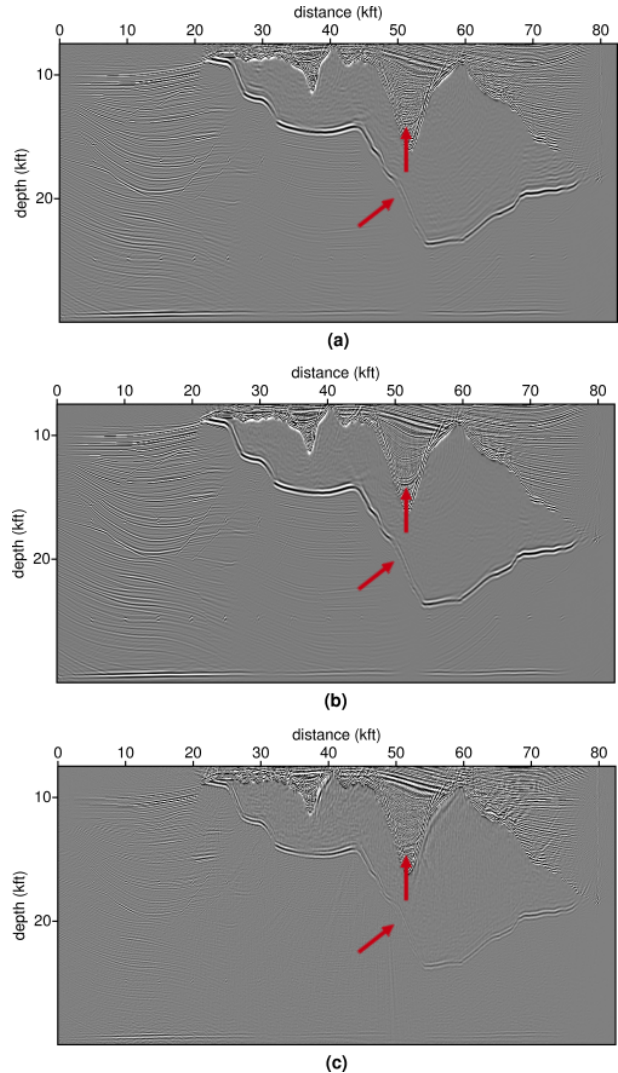


Figure 10: Resultado da RTM do dado Sigsbee2a: (a) Apenas com borda aleatória e sem *checkpoints*, (b) Com borda atenuante e *checkpoints*, (c) Com borda atenuante e sem *checkpoints*

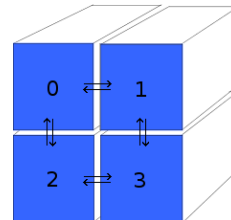


Figure 11: Esquema que ilustra o funcionamento da decomposição de um modelo 3-D em duas dimensões.

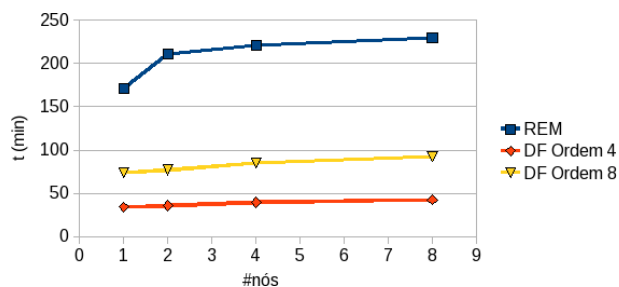


Figure 12: Tempo total de processamento de 118 tiros do modelo SEG/EAGE 3-D utilizando a decomposição espacial em uma dimensão

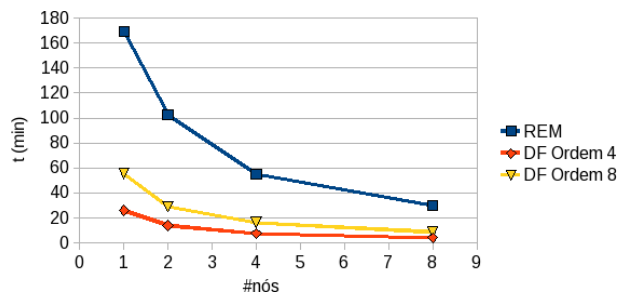


Figure 13: Tempo de processamento de um tiro do modelo SEG/EAGE 3-D utilizando a decomposição espacial em uma dimensão

Os resultados obtidos podem ser considerados satisfatórios. Com a técnica de *checkpointing*, obtemos boas seções com um intervalo entre *checkpoints* de 50 e uma redução de 95,86% no espaço de armazenamento. Com a decomposição espacial, a redução foi de até 87,5%. O aumento médio no tempo de execução para os resultados citados foi de, respectivamente, 66,08% e 27,67%.

## References

- Alford, R., K. Kelly, and D. Boore, 1974, Accuracy of finite-difference modeling of the acoustic wave equation: *Geophysics*, **39**, 834–842.
- Aminzadeh, F., N. Burkhard, J. Long, T. Kuntz, and P. Duclos, 1996, Three dimensional seg/eage models - an update: *The Leading Edge*, **15**, 131–134.
- Baysal, E., D. D. Kosloff, and J. W. C. Sherwood, 1983, Reverse time migration: *Geophysics*, **48**, 1514–1524.
- Clapp, R. G., 2009, Reverse time migration with random boundaries: *SEG Technical Program Expanded Abstracts*, **28**, 2809–2813.
- Lines, R. L., R. Slawinski, and P. Bording, 1999, A recipe for stability of finite-difference wave-equation computations: *Geophysics*, **64**, 967–969.
- Pestana, R. C. and P. L. Stoffa, 2010, Time evolution of the wave equation using rapid expansion method: *Geophysics*, **75**, 121–131.
- Symes, W. W., 2007, Reverse time migration with optimal checkpointing: *Geophysics*, **72**, SM213–SM221.
- Tal-Ezer, H., D. Kosloff, and Z. Koren, 1987, An accurate scheme for forward seismic modeling: *Geophysical Prospecting*, **35**, 479–490.
- Versteeg, R., 1994, The Marmousi experience: Velocity model determination on a synthetic complex data set:

*The Leading Edge*, **13**, 927–936.

## Agradecimentos

Gostaríamos de agradecer ao CPGG-UFBA que disponibilizou os recursos computacionais para a realização dos testes e ao INCT-GP/CNPQ pelo apoio financeiro.