

CÁLCULO DOS TEMPOS DE TRÂNSITO POR DIFERENÇAS FINITAS: UMA COMPARAÇÃO

Reynam Pestana¹ & Anderson Pimentel²

Muitos algoritmos de modelamento e imageamento sísmico requerem um mapa de tempo de trânsito distribuído em uma rede regular. Tradicionalmente, as funções de tempo de trânsito e de amplitude são obtidas pelo traçado de raios dinâmico, que calcula os tempos de trânsito e as amplitudes ao longo dos raios e interpola ao longo de uma rede regular. Outro eficiente meio de obter o mapa de tempo de trânsito é resolver a equação eiconal diretamente usando um esquema de diferenças finitas. Recentemente, muitos algoritmos de diferenças finitas foram desenvolvidos para calcular estes tempos de trânsito numa rede regular para um modelo de velocidade arbitrário. Neste trabalho, foram selecionados quatro algoritmos, testando-os em diferentes campos de velocidade e comparando os seus resultados com o método de traçado de raios padrão. Foram feitas comparações com os tempos de processamento destes algoritmos. Com estes testes, conclui-se que alguns deles trabalham bem em diferentes campos de velocidade com eficiência e estabilidade e outros não. Uma aplicação potencial destes algoritmos pode ser em procedimentos de migração Kirchhoff em profundidade antes do empilhamento e em tomografia de tempos de trânsito, porque com este novo procedimento pode-se obter o tempo de trânsito com boa acurácia e eficiência numa rede regular em modelos com grande contrastes de velocidade.

Palavras-chave: Tempo de trânsito; Diferenças finitas; Equação eiconal.

COMPUTATION OF TRAVELTIMES USING FINITE-DIFFERENCE: A COMPARISON-
Many seismic modeling and imaging algorithms require a travelttime map on a regular grid. Traditionally, traveltimes and amplitude functions are obtained by dynamic ray tracing. This involves computing the traveltimes and amplitudes along rays, and then interpolating them into a regular grid. Another efficient way to obtain this travelttime map is to solve the eikonal equation directly using a finite-difference scheme. More recently, several finite-difference schemes have been developed to calculate traveltimes for an arbitrary velocity model directly on a regular grid. We have selected four such algorithms and evaluated them using different velocity models and compared their results with a dynamic ray tracing method. We have also compared them in terms of computing time. With these tests we find that some of these algorithms work well in different types of slowness models and they are efficient and stable. A potential application of these algorithms could be in Kirchhoff prestack depth migration and in travelttime tomographic inversion. With this new procedure we can obtain the travelttime on a regular grid on models with strong contrast of velocity even with good accuracy and efficiency.

Key words: *Traveltimes; Finite-difference; Eikonal equation.*

¹Departamento de Geofísica Nuclear, Instituto de Física / PPPG
Universidade Federal da Bahia, Rua Barão de Geremoabo, 128, Campus Universitário da Federação
Salvador - BA., CEP 40.210-350, E-mail: reynam@pppg.ufba.br

²Programa de Pesquisa de Pós-graduação em Geofísica - PPPG
Universidade Federal da Bahia, Rua Barão de Geremoabo, 128, Campus Universitário da Federação
Salvador - BA., CEP 40.210-350, E-mail: anderson@pppg.ufba.br

INTRODUÇÃO

O cálculo dos tempos de trânsito é de grande importância sobre vários aspectos no processamento sísmico. Por exemplo, os métodos de migração e modelagem sísmica do tipo Kirchhoff requerem o cálculo da função de Green, que por sua vez depende do tempo de percurso entre as posições de registro na superfície e os pontos em profundidade no modelo de velocidade. O método mais usual de se calcular os tempos de trânsito é o traçado de raios (Červený et al., 1977) e uma vez que os tempos são geralmente requeridos sobre uma malha regular, os valores dos tempos ao longo dos raios são interpolados sobre a malha. Entretanto, para muitas funções de velocidade, o caminho do raio pode ser muito complicado, por exemplo, os raios podem se cruzar ou então não penetram em zonas cegas ("shadow zones") e assim o processo de interpolação pode se tornar muito difícil e até mesmo demorado computacionalmente. De maneira a resolver esses problemas para o cálculo dos tempos, vários autores tem recentemente introduzido métodos para calcular os tempos de trânsito diretamente sobre uma malha regular, evitando-se assim o processo de interpolação. Esses métodos calculam somente os tempos de chegada dos primeiros eventos a partir da solução da equação eiconal por diferenças finitas.

Reshef & Kosloff (1986) foram os primeiros a usarem a solução da equação eiconal por diferenças finitas. Eles resolveram a equação eiconal através do processo de integração de Runge-Kutta de quarta ordem. A extrapolação é feita de forma planar na direção Z , tendo como condição inicial os tempos na superfície do modelo ($Z=0$). Este procedimento apresenta bons resultados apenas em modelos homogêneos e falha em modelos com fortes contrastes de velocidade. A ineficiência do método se deve principalmente a extrapolação planar, que não simula corretamente a propagação de ondas circulares e também não considera as ondas refratadas.

Como a geometria de extrapolação representa um papel fundamental na eficiência dos algoritmos, Vidale (1988), propôs a utilização de uma geometria de extrapolação retangular, para melhor representar a propagação de frentes de ondas circulares. A computação dos tempos localmente utiliza uma interpolação linear, conseqüentemente aproxima a frente de onda circular numa frente de onda plana. No processo de determinação dos tempos uma extrapolação com geometria retangular é utilizada e os pontos de tempo mínimo, identificados ao longo da frente de onda, são usados para

extrapolar os tempos futuros. O algoritmo proposto por Vidale (1988) não funciona corretamente em fortes contrastes de velocidade, pois o seu algoritmo viola o princípio da causalidade.

Na intenção de fazer um algoritmo causal para qualquer modelo de velocidade Qin et al. (1992) basearam-se no procedimento de Vidale (1988) e propuseram que a extrapolação dos tempos de trânsito fosse feita obedecendo uma geometria circular em vez de retangular. Essa geometria de extrapolação se encaixa melhor com a propagação das frentes de onda circulares e conseqüentemente não viola o princípio da causalidade.

Van Trier & Symes (1991) introduziram outro esquema de diferenças finitas para resolver a equação eiconal. No método de Van Trier & Symes (1991) a equação eiconal é fatorada de acordo com a direção principal de propagação e a nova equação obtida é diferenciável conseguindo-se assim uma equação para as derivadas dos tempos ao invés dos tempos de trânsito. Essa equação determinada é uma equação de conservação de fluxo que aparece muito freqüentemente na mecânica dos fluidos e eles propuseram a utilização do método chamado "upwind finite-difference" para sua solução. Neste método a depender da direção de fluxo um esquema direto ou reverso de aproximações por diferenças finitas do operador diferencial é usado.

Uma vez que a equação é fatorada na sua direção principal de propagação, então não se faz necessário manter a causalidade durante a expansão das frentes de ondas na malha. Uma grande vantagem desse método é que ele pode ser colocado na filosofia do processamento vetorial com mais eficiência do que o método de Vidale (1988). Entretanto, os resultados de vários experimentos mostraram que ambos os algoritmos encontram sérios problemas de estabilidade, quando aplicados a modelos de velocidade com grandes discontinuidades.

Podvin & Lecomte (1991) conseguiram melhorias bastante significativas para o método de Vidale (1988). Eles aplicaram o princípio de Huygens conjuntamente com a aproximação por diferenças finitas de tal maneira que o método fornece os tempos de trânsito com uma boa precisão, mesmo na presença de fortes contrastes de velocidade com forma arbitrária. Eles também introduziram o cálculo da extrapolação reversa, permitindo assim obter os tempos dos eventos que se propagam na direção da fonte.

Zhang (1991) também apresentou um método que combina as vantagens dos algoritmos existentes. Ele usa uma representação simplificada do esquema de diferenças finitas

de Podvin & Lecomte (1991) e consegue que o seu algoritmo funcione bem em fortes contrastes de velocidade. E além disso, escolheu trabalhar no sistema de coordenadas polares objetivando tornar o seu algoritmo completamente vetorizável.

Mais recentemente, Schneider et al. (1992) apresentam uma nova metodologia para a obtenção dos tempos de chegada que usa a técnica de propagação dinâmica. O método baseia-se no princípio de Fermat e usa técnica de cálculo simples e um esquema de mapeamento sistemático para calcular os tempos de chegada dos primeiros eventos em uma malha regular uniforme, a partir de uma distribuição discreta do campo de velocidade. O método consegue obter os tempos em casos em que o campo de velocidade apresente grandes contrastes e distribuição de velocidade com descontinuidades. Esse método consegue obter os tempos das ondas transmitidas, difratadas ou mesmo ondas refratadas.

Esses e vários outros métodos estão sendo utilizados com o objetivo de obter os tempos de chegada em uma malha regular para as mais diversas aplicações na sísmica, tais como: inversão sísmica tomográfica, migração de ponto de tiro comum, migração com inversão de velocidade e o próprio traçado de raios a partir dos tempos fornecidos por esses algoritmos. Também alguns trabalhos usam os tempos, a partir da solução da equação eiconal por diferenças finitas para obter a amplitude em cada ponto da malha.

A nossa principal intenção com este trabalho é mostrar alguns dos mais importantes métodos de cálculo dos tempos de chegada dos primeiros eventos em uma malha regular através da solução da equação eiconal por diferenças finitas. Também discutir as principais características dos métodos aqui abordados e comparar os tempos obtidos em modelos de velocidade relativamente complexos com os tempos fornecidos pelo método convencional de traçado de raios (programa SEIS88, Červený & Pšenčík, (1988)). E assim apresentar as possíveis alternativas de se obter os tempos de chegada para as diversas aplicações no processamento de dados sísmicos.

CÁLCULO DO TEMPO DE TRÂNSITO POR DIFERENÇAS FINITAS

Os algoritmos que serão abordados neste trabalho utilizam a aproximação por diferenças finitas da equação eiconal na extrapolação do tempos de trânsito.

A equação eiconal é uma equação diferencial parcial, não linear, não-homogênea, de primeira ordem e de segundo grau. A sua expressão em coordenadas cartesianas bidimensionais é

$$\left(\frac{\partial\tau(x,z)}{\partial x}\right)^2 + \left(\frac{\partial\tau(x,z)}{\partial z}\right)^2 = S(x,z)^2, \quad (1)$$

onde,

$\tau(x,z)$ = tempo de trânsito,

$S(x,z)$ = vagarosidade sísmica,

e a sua representação em coordenadas polares é

$$\left(\frac{\partial\tau(r,\theta)}{\partial r}\right)^2 + \frac{1}{r^2}\left(\frac{\partial\tau(r,\theta)}{\partial\theta}\right)^2 = S(r,\theta)^2. \quad (2)$$

Ao invés de fornecer os tempos de trânsito ao longo do caminho do raio como fazem os métodos convencionais, por exemplo o "shooting method", a equação eiconal fornece os tempos de trânsito ao longo das frentes de onda. As frentes de onda penetram em zonas de sombra e também propagam ondas refratadas ou fenômenos de interferência.

O método das diferenças finitas discretiza o modelo, representando uma aproximação física do modelo real. O modelo real é substituído por um modelo virtual onde cada espaço é preenchido por materiais com velocidade constante, enquanto que na discretização cada cela da rede tem uma velocidade. É evidente que a acurácia desta aproximação depende essencialmente do tamanho de cada cela.

Duas etapas são necessárias para construir o esquema computacional das diferenças finitas. Na primeira etapa deve-se definir o esquema de propagação local, já que o tempo de trânsito num dado ponto da rede depende dos tempos vizinhos e também do valor local de vagarosidade. Numa segunda etapa, deve-se definir as direções de propagação.

A maioria dos esquemas de computação local parte do princípio da linearidade do tempo de trânsito. Esta linearidade ocasiona o uso de um procedimento de interpolação linear do tempo de trânsito, que aproxima a frente de onda curva numa frente de onda plana local. Esta aproximação requer uma rede muito fina para manter a acurácia do algoritmo. Outros esquemas de computação usam uma interpolação não linear do tempo, que simula uma

frente de onda circular com mais acurácia e com uma rede mais espaçada.

As direções de propagação definem de que modo é feita a extrapolação do tempo de trânsito no esquema de diferenças finitas. Este esquema de extrapolação deve obedecer duas condições principais:

- O princípio da causalidade deve ser respeitado. Os tempos devem ser computados de maneira que cada etapa da extrapolação deve propagar ondas do passado para o futuro.
- Nenhum dos eventos em potencial podem ser ignorados. Todos os modos de propagação, tais como ondas transmitidas, difratadas e refratadas devem ser tratados.

O esquema de extrapolação do tempo de trânsito deve, de uma maneira geral, tentar simular a geometria de propagação da frente de onda no modelo. Este procedimento, se bem feito, minimiza o número de chamadas da função cálculo de tempo de trânsito local.

Os objetivos principais dos algoritmos que trabalham com a técnica das diferenças finitas são manter a estabilidade numérica, melhorar a acurácia e tornar o procedimento mais eficiente. A evolução histórica destes algoritmos mostra como se desenvolveu a melhoria destes parâmetros. A seguir iremos fazer uma breve discussão dos métodos de diferenças finitas, mostrando as principais etapas de cada método.

MÉTODOS TESTADOS

Método de Schneider et al. (1992)

O cálculo do tempo de trânsito avança uniformemente ao longo da malha e em cada estágio do mapeamento, apenas o tempo de trânsito atual é usado no cálculo do tempo de trânsito futuro. O autor propôs dois métodos de cálculo. Um é o chamado mapeamento de força bruta, que usa uma ordem de mapeamento independente do modelo com um esquema de aproximação de ondas planas e uma versão mais acurada que simula frentes de onda circulares. Já o outro método se constitui num mapeamento que depende do modelo e é similar ao esquema proposto por Vidale (1988).

O mapeamento de força bruta começa com o cálculo do tempo de trânsito para todos os pontos da primeira coluna a partir da fonte usando apenas a distância e a vagariedade em cada cela, como mostra a Fig. 1. Estes pontos estão marcados com um círculo e a fonte está marcada com um asterisco (*).

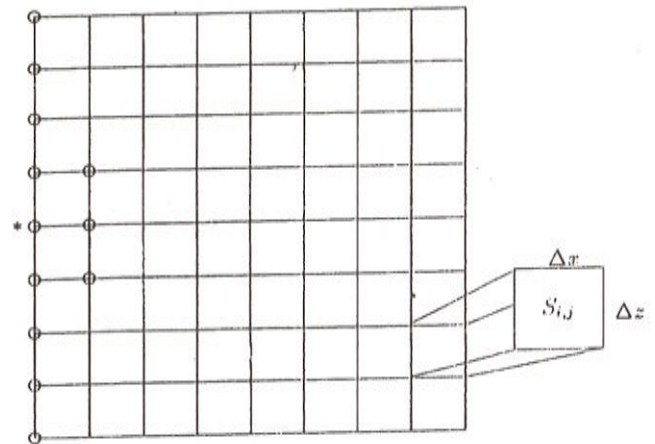


Figura 1 - A disposição da rede para o mapeamento de força bruta (Schneider et al., 1992). O asterisco indica a posição da fonte. Os círculos indicam os pontos onde os tempos de trânsito são computados analiticamente e usados como condição inicial.

Figure 1 - The disposition of the grid for the brute force mapping (Schneider et al., 1992). The star denotes the source position. The circles denote the points where the traveltimes are analytically computed and used as initial condition.

A Fig. 2 mostra três colunas localizadas no modelo da Fig. 1. O mapeamento começa na parte superior esquerda, onde τ_1 e τ_2 são conhecidos e τ é o tempo de trânsito que se quer conhecer, usando a configuração representada na Fig. 2. Após ser calculado o tempo de trânsito τ , move-se a configuração uma cela para baixo e calcula-se o tempo no canto inferior direito de cada cela. Este procedimento é adotado nas três configurações representadas na Fig. 2 para o cálculo dos tempos na segunda coluna e dos quatro tempos calculados em cada nó, guarda-se o de menor valor. Com estes tempos de trânsito da segunda coluna, obtém-se pelo mesmo procedimento os tempos da terceira coluna e assim por diante, até terminar o cálculo de todos os pontos da rede.

No cálculo do tempo de trânsito em cada cela utiliza-se uma aproximação de onda plana que considera que o tempo de trânsito varia linearmente com a distância em qualquer direção. A Fig. 3 mostra como é feito o cálculo em cada cela. Os tempos τ_1 e τ_2 são conhecidos e estão localizados nas posições (x, z_1) e (x, z_2) , respectivamente. Na posição (x, z_0) , a energia passa com um tempo τ_0 e τ é o tempo que se quer calcular, na posição $(x + \Delta x, z_2)$.

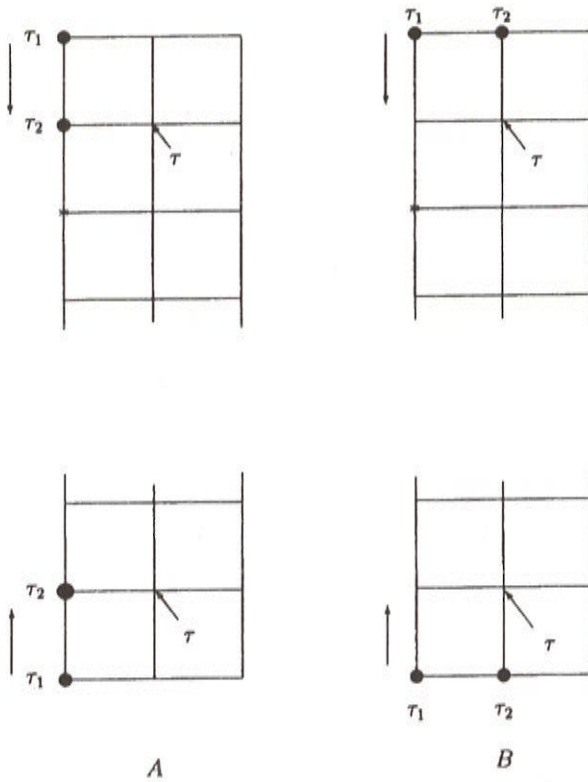
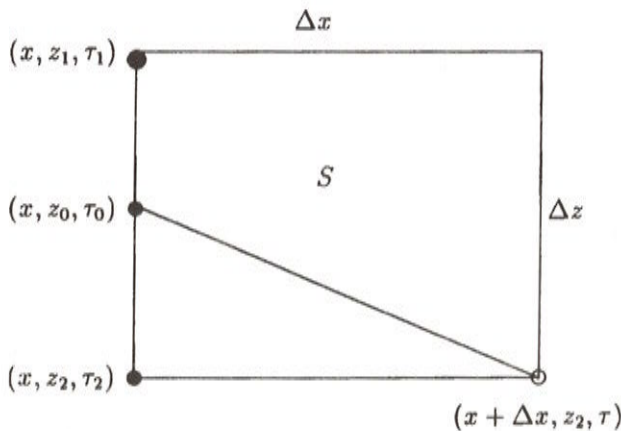


Figura 2 - As três primeiras colunas da Fig. 1 são mostradas em A e também em B. A parte central destas colunas foram cortadas. Os quatro tipos de configurações são usados para computar o tempo τ , na segunda coluna, usando os tempos τ_1 e τ_2 conhecidos. A posição da fonte é denotada pelo asterisco.

Figure 2 - The three left-most columns of the grid are shown in A and repeated again in B. The centers of these columns have been "cut away" for clarity. The four configuration types that are used to compute a new τ , on column 2, from a pair of known times τ_1 and τ_2 , on columns 1 and 2 are illustrated. A star denotes the source point.



Pode-se escrever uma interpolação linear em z_0 como:

$$\tau_0 = \left[\frac{(\tau_2 - \tau_1)}{\Delta z} \right] (z_0 - z_1) + \tau_1. \quad (3)$$

O tempo de trânsito τ em $(x+\Delta x, z_2)$ é obtido minimizando

$$\tau = \tau_0 + S \left[(z_2 - z_0)^2 + \Delta x^2 \right]^{\frac{1}{2}} \quad (4)$$

com respeito a z_0 . Diferenciando com respeito a z_0 obtém-se

$$\frac{d\tau}{dz_0} = \frac{(\tau_2 - \tau_1)}{\Delta z} - S(z_2 - z_0) \left[(z_2 - z_0)^2 + \Delta x^2 \right]^{-\frac{1}{2}}, \quad (5)$$

que é igualada a zero e resolvida diretamente para z_0 . Substituindo na expressão 4 o valor z_0 , obtém-se:

$$\tau = \tau_2 + \left[S^2 \Delta x^2 - \frac{\Delta x^2}{\Delta z^2} (\tau_2 - \tau_1)^2 \right]^{\frac{1}{2}} \quad (6)$$

que é o tempo de trânsito mínimo τ em termos dos tempos conhecidos τ_1 e τ_2 . Entretanto, frentes de onda são circulares e não planares. Para manter a acurácia com a Eq. (6), uma rede fina é necessária, especialmente perto da fonte, onde a curvatura da frente de onda é maior.

Schneider et al. (1992) também propuseram uma interpolação não linear do tempo de trânsito, que modela a frente de onda de forma circular. Considerando-se um meio bidimensional, com velocidade constante, o que implica que a frente de onda gerada por um ponto é circular. Então,

$$\tau^2 = S_a^2 (x_a^2 + z^2). \quad (7)$$

Figura 3 - Esquema para cálculo dos tempos com o algoritmo de Schneider et al. (1992). Os tempos τ_1 e τ_2 são conhecidos, a energia sísmica passa por (x, z_0) no tempo τ_0 e o tempo τ é o objetivo do cálculo. A vgarosidade S desta cela é constante.

Figure 3 - The procedure for computation of times with the Schneider's algorithm. The times τ_1 and τ_2 are known and the seismic energy passes for (x, z_0) in time τ_0 and the time τ is the objective. The slowness S of this cell is constant.

Na Eq. (7), z é a profundidade e x_a é a distância lateral da fonte pontual. Se x_a permanece constante, τ_2 está relacionado linearmente com z_2 . Conhecendo-se os tempos de trânsito τ_1 e τ_2 localizados nos pontos (x_a, z_1) e (x_a, z_2) respectivamente. Estes pontos definem uma linha paralela ao eixo z . A Eq. (7) pode ser reescrita para estes dois pontos como:

$$w = \frac{(\tau_2^2 - \tau_1^2)}{(z_2^2 - z_1^2)} = S_a^2 \tag{8}$$

Usando esta interpolação, o tempo τ_0 no ponto (x_a, z_0) ao longo desta linha vertical satisfaz:

$$\tau_0^2 = w(z_0^2 - z_1^2) + \tau_1^2 \tag{9}$$

Quando a vagarosidade do meio é variável, a Eq. (8) torna-se uma interpolação não linear com relação ao quadrado da distância z_0 . Considera-se que na Fig. 3 x, z_1, z_2, τ_1 e τ_2 são conhecidos e a origem do sistema de coordenadas coincide com a fonte pontual real. Então com a Eq. (9), que interpola $\tau_0(z_0)$, minimiza-se a Eq. (4) para obter o tempo mínimo τ num meio onde as frentes de onda são curvas. Obtém-se então

$$\frac{d\tau}{dz_0} = z_0 \frac{\omega}{\tau_0} - S(z_2 - z_0) \left[(z_2 - z_0)^2 + \Delta x^2 \right]^{\frac{1}{2}} \tag{10}$$

que deve ser resolvida para z_0 . Quando τ_1 e τ_2 são tempos associados a um meio com vagarosidade realmente constante, o problema de minimização se torna exato.

Método de Reshef & Kosloff (1986)

Reescrevendo a equação eiconal (Eq. (1)), chega-se a

$$\frac{\partial \tau(x, z)}{\partial z} = \sqrt{S(x, z)^2 - \left(\frac{\partial \tau(x, z)}{\partial x} \right)^2} \tag{11}$$

A derivada $\partial \tau / \partial x$ pode ser calculada usando-se a aproximação por diferenças finitas de primeira ordem, de acordo com a equação a seguir, usando os valores de $\tau(x, z)$ num dado horizonte, como mostra a Fig. 4.

$$\frac{\partial \tau}{\partial x} = \frac{\tau(x + \Delta x, z) - \tau(x - \Delta x, z)}{2\Delta x} \tag{12}$$

onde Δx e Δz são os incrementos na direção x e z , respectivamente.

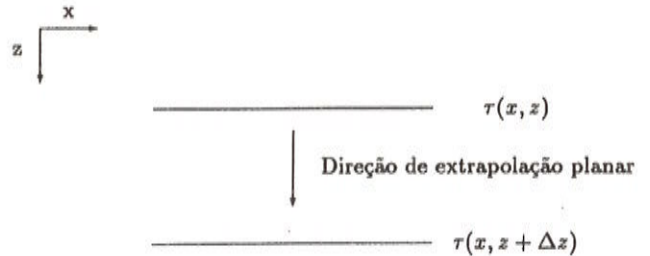


Figura 4 - Mostra a maneira como se faz a extrapolação dos tempos com o algoritmo de Reshef & Kosloff (1986).

Figure 4 - Shows the time extrapolation procedure of the algorithm proposed by Reshef & Kosloff (1986).

A extrapolação para baixo (eixo z) do tempo de trânsito em cada horizonte é feita a partir do método de Runge-Kutta de quarta ordem. As equações abaixo são utilizadas no método de Reshef & Kosloff (1986). Nota-se que a extrapolação é feita em quatro etapas, como mostra as equações abaixo.

$$\tau(x, z + \Delta z) = \tau(x, z) + \frac{\Delta z}{6} (k_1 + 2k_2 + 2k_3 + k_4), \tag{13}$$

onde

$$\begin{aligned} k_1 &= F'(x, z, \tau) \\ k_2 &= F' \left(x, z + \frac{\Delta z}{2}, \tau + \frac{1}{2} k_1 \right) \\ k_3 &= F' \left(x, z + \frac{\Delta z}{2}, \tau + \frac{1}{2} k_2 \right) \\ k_4 &= F' \left(x, z + \Delta z, \tau + \frac{1}{2} k_3 \right) \end{aligned}$$

onde, F' é uma função conhecida e representa a derivada da função $\tau(x, z)$ com relação a z calculada nos pontos indicados.

O cálculo de $\partial\tau/\partial z$ poderia também ter sido feito utilizando o método de diferenças finitas, entretanto, fornecem resultados com uma menor precisão numérica comparando com o cálculo feito por Runge-Kutta, adotado no artigo de Reshef & Kosloff (1986). Para inicializar o algoritmo, os valores de $\tau(x, z = 0)$ na superfície precisam ser especificados.

Método de Zhang (1991)

Zhang (1991) desenvolveu um método de cálculo de tempo de trânsito que combina as vantagens destes algoritmos. Ele usou a representação simplificada por diferenças finitas como proposta por Podvin & Lecomte (1991). Trabalhou em coordenadas polares para tornar o algoritmo vetorizável. Esta escolha também facilita a extrapolação reversa. A seguir descreveremos como é feito o cálculo do tempo de trânsito no algoritmo proposto por Zhang (1991).

A condição inicial é

$$\tau(\theta, r) = S(\theta, r)r, \tag{14}$$

para $r \leq r_0$, onde r_0 define um círculo centrado na fonte. Começando com um círculo, o algoritmo extrapola o tempo de trânsito nas direções radiais. Como indicado na Fig. 5, usa-se $\tau_{i,j}$ para denominar o tempo de trânsito em cada ponto da malha (θ_i, r_j) . Então, no passo j de extrapolação do tempo de trânsito, conhecendo-se os tempos de trânsito $\tau_{i,j}$ para todos os i determina-se os tempos $\tau_{i,j+1}$ para todos os i .

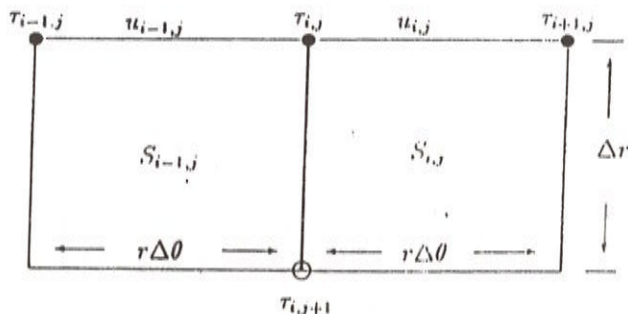


Figura 5 - A malha e as notações utilizadas no método proposto por Zhang (1991) e também utilizado no método de Kosloff (1991).

Figure 5 - The grid and the notation used in the Zhang's (1991) method and in the Kosloff's (1991) method.

De acordo com a lei de Snell a componente tangencial do gradiente do tempo de trânsito é contínua quando atravessa uma interface. O autor definiu $u_{i,j}$ como sendo a componente tangencial do gradiente do tempo de trânsito. Aproximando-se a frente de onda numa célula como uma frente plana local, tem-se

$$u_{i,j} = \frac{1}{r_j} \frac{\partial\tau}{\partial\theta} = \frac{\tau_{i+1,j} - \tau_{i,j}}{r_j \Delta\theta}. \tag{15}$$

Como mostra a Fig. 5, $\tau_{i,j}$ é o tempo de trânsito no ponto (θ, r) . Na etapa j da extrapolação do tempo de trânsito se conhece o tempo $\tau(i,j)$ para todo i e obtém-se o tempo $\tau_{i,j+1}$. A componente ortogonal do gradiente do tempo de trânsito pode ser achada a partir da equação eiconal

$$w_{i,j} = \frac{\partial\tau}{\partial r} = \sqrt{s_{i,j}^2 - u_{i,j}^2}. \tag{16}$$

Estas duas componentes determinam a direção de propagação da onda plana local. Definindo

$$\hat{w}_{i,j} = \frac{\tau_{i,j+1} - \tau_{i,j}}{\Delta r} \tag{17}$$

e usando os princípios de Podvin & Lecomte (1991) pode-se calcular $\hat{w}_{i,j}$ para quatro casos:

- Se $u_{i-1,j} \geq 0$ e $u_{i,j} \leq 0$ então $\hat{w}_{i,j} = \min(w_{i-1,j}, w_{i,j}, s_{i-1,j}, s_{i,j})$
- Se $u_{i-1,j} \geq 0$ e $u_{i,j} \geq 0$ então $\hat{w}_{i,j} = \min(w_{i-1,j}, s_{i,j})$
- Se $u_{i-1,j} \leq 0$ e $u_{i,j} \leq 0$ então $\hat{w}_{i,j} = \min(w_{i,j}, s_{i-1,j})$
- Se $u_{i-1,j} \leq 0$ e $u_{i,j} \geq 0$ então $\hat{w}_{i,j} = \min(s_{i-1,j}, s_{i,j})$

A presença da vagarosidade nestas fórmulas indica que possíveis refrações ao longo de uma curva de θ constante são levadas em consideração. A função *min* garante que será computado o tempo mínimo. Resolvendo a equação para $\tau_{i,j+1}$, obtemos:

$$\tau_{i,j+1} = \tau_{i,j} + \hat{w}_{i,j} \Delta r \text{ para todo } i \tag{18}$$

que completa um passo da extrapolação.

Após ser feita a extrapolação dos tempos de trânsito em coordenadas polares, usa-se a interpolação bilinear para encontrar os tempos de trânsito em coordenadas cartesianas, obtendo-se assim os tempos de trânsito no domínio $x-z$.

Método de Kosloff (1991) em coordenadas polares

Este autor resolve a equação eiconal em coordenadas polares usando os métodos de Runge-Kutta de quarta ordem e das diferenças finitas. Como condição inicial para o processo de expansão da frente de onda, temos que

$$\tau(\theta, r) = S(\theta, r) r, \quad (19)$$

para $r \leq r_0$ onde r_0 define um círculo centrado na posição da fonte, dentro do qual a vagariedade é constante. A partir deste círculo o campo de tempos é extrapolado na direção radial. Usamos no nosso algoritmo $\tau_{i,j}$ para indicar o tempo na posição (θ_i, r_j) .

O passo seguinte no processo de extrapolação é determinar a componente tangencial do campo de tempo $\partial\tau/\partial\theta$ para r constante. Utilizando-se uma aproximação de onda plana para a frente de onda dentro de uma célula, então temos a Eq. (15) que calcula $u = \partial\tau/\partial\theta$.

No seu algoritmo, Kosloff utiliza alguns critérios para determinar $u_{i,j}$ que posteriormente é usado para extrapolar $\tau(\theta, r)$. Inicialmente calcula-se u^1 e u^2 definidos como:

$$u^1 = \frac{\tau_{i+1,j} - \tau_{i,j}}{\Delta\theta}, \quad (20)$$

$$u^2 = \frac{\tau_{i+2,j} - \tau_{i+1,j}}{\Delta\theta}, \quad (21)$$

e então, usa-se os seguintes critérios para determinar $u_{i,j}$:

Se $u^1 \geq 0$ e $u^2 \geq 0$ então $u_{i,j} = u^1$

Se $u^1 \leq 0$ e $u^2 \leq 0$ então $u_{i,j} = u^2$

Se u^1 e u^2 não satisfazem os casos acima então

$$u_{i,j} = \frac{1}{2}(u^1 + u^2). \quad (22)$$

Já a componente ortogonal do gradiente do tempo de trânsito $\partial\tau/\partial r$ pode ser calculada a partir da Eq. (17) e os tempos de chegada $\tau(\theta, r)$ são obtidos a partir da integração numérica de $\partial\tau/\partial r$ usando o método de Runge-Kutta de quarta ordem, que foi apresentado anteriormente. Em seguida os tempos são obtidos no domínio $x - z$ através do uso do método de interpolação bilinear.

RESULTADOS NUMÉRICOS

Os algoritmos implementados neste trabalho foram: o método de mapeamento de força bruta com interpolação não linear proposto por Schneider et al. (1992), a extrapolação do tempo de trânsito pelo método de Runge-Kutta de quarta ordem proposto por Reshef & Kosloff (1986) e a sua variação para coordenadas polares proposta por Kosloff (1991) e o método de aproximação da equação eiconal por diferenças finitas proposto por Zhang (1991).

Os modelos aqui testados são bidimensionais, com uma rede que apresenta um espaçamento regular $\Delta x = \Delta z = 10m$. Como os algoritmos apresentados são soluções da equação eiconal, os modelos geológicos apresentados aqui têm como variável apenas a velocidade de propagação das ondas sísmicas.

As curvas de contorno de mesmo tempo dos quatro algoritmos testados representam as frentes de onda e mostram como se comporta a propagação da energia no meio. Os modelos foram escolhidos visando criar condições para a propagação de diversos eventos, tais como: difrações, refrações e transmissões e também para estudar os erros dos algoritmos testados sobre diversas condições.

Modelo 1 - Modelo Homogêneo

Este modelo é constituído por uma camada isotrópica, homogênea e com velocidade constante ($v = 4000$ m/s) numa rede de $1000m \times 1000m$, com a fonte localizada na posição (500m, 10m). Este modelo foi escolhido para estudar como se comporta a função de erro dos algoritmos de diferenças finitas ao longo de toda a rede. Esta função de erro foi obtida a partir da diferença entre o tempo de trânsito calculado pelo algoritmo e a solução analítica.

A Fig. 7 mostra uma comparação das frentes de onda obtidas pelos quatro algoritmos testados com as frentes de onda obtidas pela solução analítica. A Fig. 7a mostra a comparação entre as frentes de onda obtidas pela solução analítica e as frentes obtidas pelo algoritmo proposto por Schneider et al. (1992). Na Fig. 6a, nota-se que a função de erro é maior nas direções em que o ângulo tomado em relação ao eixo x varia entre 60 a 90 graus. As tonalidades mais fortes indicam erros maiores e as mais fracas erros menores. Verifica-se que a solução do algoritmo de Schneider se aproxima da solução exata nas colunas próximas à coluna em que se localiza a fonte e também nas primeiras linhas do modelo. De uma maneira geral o erro cresce com a distância em relação à fonte.

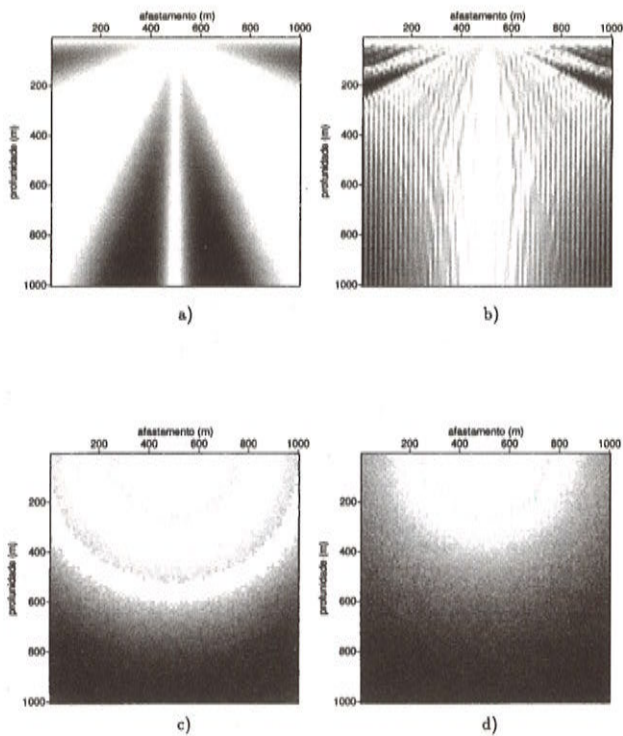


Figura 6 - Mostra a função erro no modelo 1 para todos os algoritmos testados. a) Schneider et al. (1992): erro máximo absoluto, $e = 3,16 \times 10^{-6}s$, b) Reshef & Kosloff (1986) ($e = 1,33 \times 10^{-4}s$), c) Zhang (1991) ($e = 2,38 \times 10^{-7}s$) e d) Kosloff (1991) em coordenadas polares ($e = 2,08 \times 10^{-7}s$).

Figure 6 - Shows the error function obtained in the model 1 using all the algorithms. a) Schneider: absolute error, $e = 3.16 \times 10^{-6}s$, b) Reshef & Kosloff (1986) ($e = 1.33 \times 10^{-4}s$), c) Zhang (1991) ($e = 2.38 \times 10^{-7}s$) and d) Kosloff (1991) in polar coordinates ($e = 2.08 \times 10^{-7}s$).

A Fig. 7b representa as frentes de onda obtidas pelo algoritmo de Reshef & Kosloff (1986) em coordenadas cartesianas. Nota-se que o erro é grande na região próxima às bordas do modelo e é ocasionado pela extrapolação planar na direção do eixo z e também pela forma com é aproximado $\partial\tau/\partial x$ nas bordas do modelo. Nestas regiões a extrapolação planar não modela com acurácia as frentes de onda circulares. A Fig. 6b indica justamente isto. O erro é maior nas bordas do modelo, é menor no centro e nas primeiras linhas e aumenta com o aumento da distância em relação à fonte.

As Figs. 7c e 7d representam a comparação entre os algoritmos que trabalham em coordenadas polares e a solução analítica. Nota-se que a solução exata parece coincidir com a solução obtida pelos dois algoritmos. Este resultado

se deve principalmente à extrapolação do tempo de trânsito feita em coordenadas polares, já que neste sistema de coordenadas a computação é feita radialmente, coincidindo exatamente com a propagação de frentes de onda circulares. Analisando as Figs. 6b e 6c, a função de erro apresenta um padrão circular e cresce com a distância em relação à fonte.

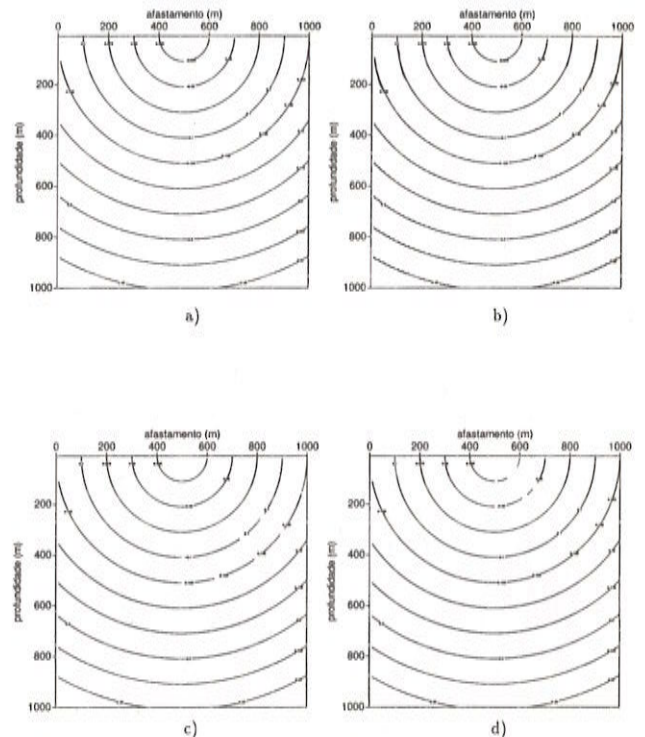


Figura 7 - Comparação entre as frentes de onda obtidas pelos algoritmos testados no modelo 1 e solução analítica. O intervalo entre curvas é de 0,025 s. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) e d) Kosloff (1991).

Figure 7 - Comparison between the wavefronts obtained with the algorithms in the model 1 and the analytic solution. The interval between the curve is 0.025 s. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) and d) Kosloff (1991).

Neste modelo com velocidade constante, os algoritmos que trabalham em coordenadas polares apresentam maior acurácia do que os algoritmos que trabalham em coordenadas cartesianas. A função de erro de uma maneira geral aumenta com a distância em relação a fonte em todos os algoritmos aqui testados. Analisando

o tempo de CPU (Tab. 3), o algoritmo de Reshef & Kosloff (1986) é o mais rápido e o do Schneider et al. (1991) o mais lento.

Modelo 2 - Duas Camadas Homogêneas

O modelo é constituído por duas camadas homogêneas e isotrópicas. A camada superior tem uma velocidade de propagação das ondas sísmicas de $V_1 = 1000$ m/s e a camada inferior tem a velocidade de propagação de $V_2 = 4000$ m/s. O modelo tem uma rede de 1000 m x 1000 m, com uma interface plana e horizontal que separa as duas camadas e que está localizada na posição $z = 500$ m (Fig. 8). A fonte está localizada na posição (500 m, 10 m). Este modelo apresenta um contraste forte de velocidade e foi utilizado para testar a acurácia destes quatro algoritmos nestas condições.

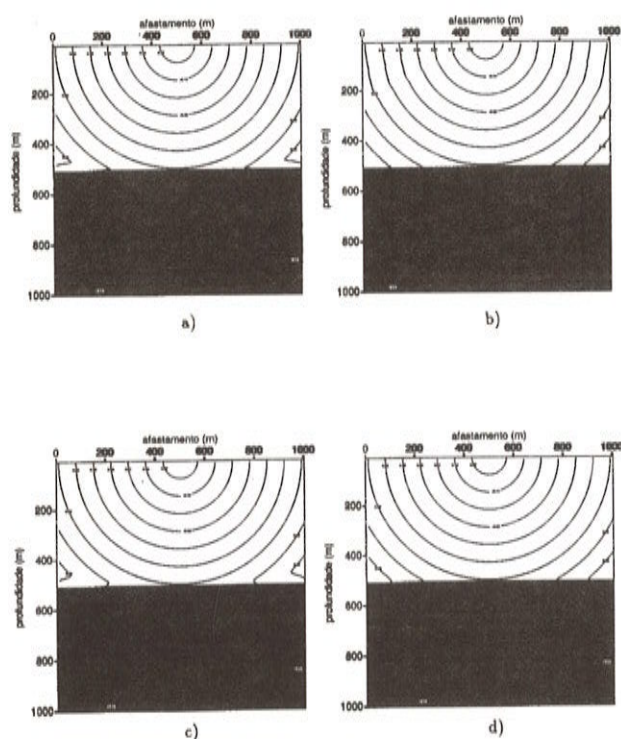


Figura 8 - Mostra as curvas de tempo para o modelo 2 pelos quatro algoritmos testados. São dez curvas com intervalo de 0,07 s entre cada uma. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) e d) Kosloff (1991).

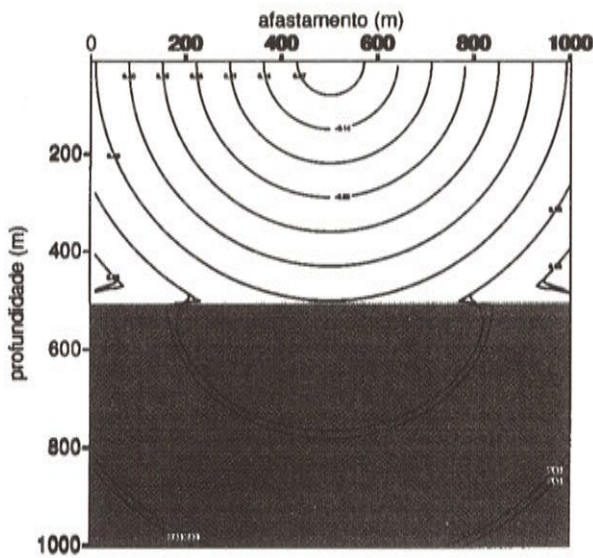
Figure 8 - Shows the time curves obtained with the four algorithms tested. There are ten curves with interval of 0.07 s. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) and d) Kosloff (1991).

A Fig. 8a mostra as curvas de tempo obtidas pelo algoritmo proposto por Schneider et al. (1992). Nota-se que este algoritmo apresenta um bom desempenho nestas condições de contraste de velocidade. As “head waves” são obtidas com uma boa acurácia. A Fig. 8b apresenta os resultados obtidos pelo algoritmo de Reshef & Kosloff (1986) em coordenadas cartesianas. Nota-se que este algoritmo não calcula ondas refratadas nem apresenta o avanço das frentes de onda, quando estas chegam na camada de maior velocidade.

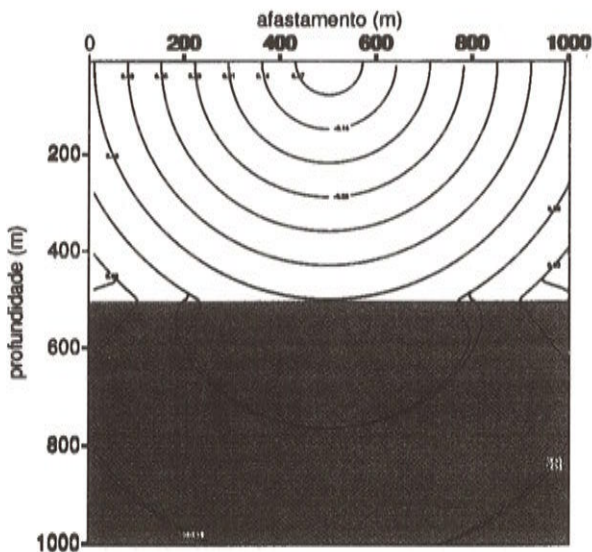
A Fig. 8c mostra as frentes de onda obtidas pelo algoritmo proposto por Zhang (1991). Nota-se que o algoritmo apresenta um bom desempenho nestas condições de contraste de velocidade, gerando “head waves”. A Fig. 9a é uma comparação entre os algoritmos de Schneider et al. (1992) e de Zhang (1991). Percebe-se que o avanço das frentes de onda obtidas pelo algoritmo de Schneider et al. (1992) quando estas atingem a camada de maior velocidade ocorre mais abruptamente do que nos resultados do algoritmo de Zhang (1991). Este efeito ocorre devido à transformação do campo de velocidade de coordenadas cartesianas para polares, que é feito usando uma interpolação bilinear. Esta interpolação faz com que o contraste de velocidade se transforme num gradiente de velocidade, resultando nas feições encontradas.

A Fig. 8d mostra o desempenho do algoritmo de Kosloff (1991) em coordenadas polares. Nota-se que o avanço das frentes de onda devido ao contraste de velocidade ocorre um pouco abaixo da interface. A Fig. 9b é uma comparação feita entre o algoritmo de Zhang (1991) e o de Kosloff (1991) em coordenadas polares. Verifica-se que o avanço das frentes de onda ocorre mais próximo da interface nos resultados obtidos pelo algoritmo de Zhang (1991). Isto se deve ao fato que o algoritmo de Kosloff (1991) usa o parâmetro dr fixo, o que não ocorre no algoritmo de Zhang (1991), sendo que neste último, o parâmetro dr é calculado de acordo com o modelo de velocidade, resultando numa otimização deste parâmetro.

A Fig. 10 mostra uma comparação entre os tempos de trânsito obtidos pelos quatro algoritmos testados e os tempos obtidos usando o programa SEIS88 (“shooting method”). Primeiramente nota-se que a solução obtida usando o algoritmo de Schneider et al. (1992) é praticamente igual à solução obtida pelo “shooting method”. A solução obtida pelo algoritmo de Reshef & Kosloff (1986) apresenta um erro grande nas bordas do modelo. Os tempos obtidos pelo algoritmo de Zhang (1991) e os obtidos pelo



a)



b)

Figura 9 - Comparação entre as curvas obtidas no modelo 2. a) Comparação entre as curvas de tempo obtidas usando o algoritmo de Zhang (1991) e de Schneider et al. (1992), b) O mesmo para os algoritmos de Zhang (1991) e Kosloff (1991).

Figure 9 - Comparison between the time curves obtained in the model 2. a) Comparison between the time curves obtained using the Zhang's and Schneider's algorithms, b) The same for the Zhang's and Kosloff's algorithm in polar coordinates.

algoritmo de Kosloff (1991) são um pouco menores do que os que foram obtidos pelo SEIS88 e diferenciam-se entre si com mais intensidade nas bordas do modelo. Na Tab. 1, onde é feita uma comparação entre os tempos obtidos pelos 4 algoritmos com o tempo calculado analiticamente obtidos na posição (500m,800m), verifica-se que o programa que utiliza o algoritmo de Schneider et al. (1992) é o que mais se aproxima da solução exata. Os programas que utilizam os algoritmos de Zhang (1991) e de Kosloff (1991) apresentam tempos menores do que o calculado analiticamente. Este fato também pode ser explicado por erros na transformação do campo de velocidades para coordenadas polares.

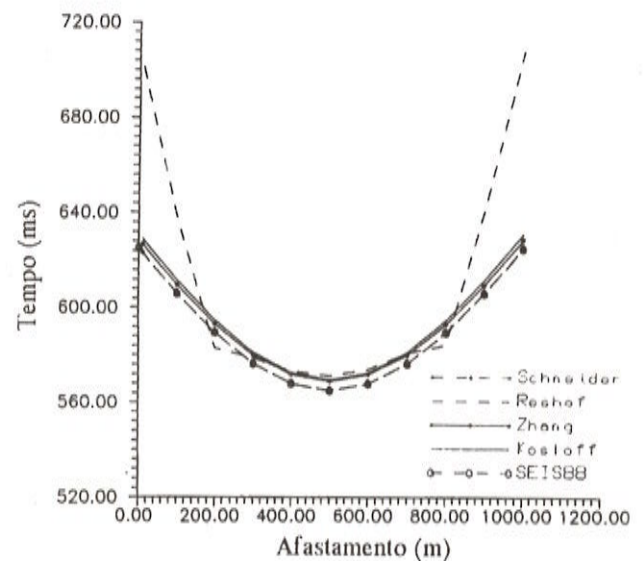


Figura 10 - Gráfico mostrando a comparação entre os tempos obtidos pelos quatro algoritmos e os obtidos pelo programa SEIS88 (Červený & Pšencík) com o modelo 2. Os receptores estão a 800 m abaixo da fonte, com espaçamento de 10 m.

Figure 10 - Graph showing the comparison between the times obtained with the four algorithms and the times obtained with the SEIS88's program (Červený & Pšencík) in the model 2. The receivers are 800 m below the source with spacing of 10 m.

Analisando de uma maneira geral os resultados obtidos neste modelo com contraste forte de velocidade o método proposto por Schneider et al. (1992) é o que apresenta os melhores resultados, porém é mais lento computacionalmente (Tab. 3).

Algoritmo	Tempo (ms)	Erro absoluto (ms)
Schneider	564,999	0,001
Resheff	571,250	6,250
Zhang	569,016	4,016
Kosloff	569,442	4,442
Scis88	565,010	0,010

Tabela 1 - Tabela com os tempos de trânsito obtidos no modelo 2 para os algoritmos testados obtidos 800 m abaixo da fonte e o seu respectivo erro absoluto. O erro foi obtido através da diferença entre o tempo vertical (calculado analiticamente) e o fornecido pelo respectivo algoritmo.

Table 1 - Table with the travel-times obtained in the model 2 for the algorithms tested, picked in 800 m below the source and the respective absolute error. The error is the difference between the vertical time (analytically calculated) and the algorithm computation time.

Modelo 3 - Modelo de Cinco Camadas Homogêneas

Este modelo é constituído por cinco camadas, homogêneas e isotrópicas e com velocidade de propagação das ondas sísmicas de $V_1 = 500$ m/s, $V_2 = 1000$ m/s, $V_3 = 1500$ m/s, $V_4 = 2000$ m/s e $V_5 = 2500$ m/s para cada camada. A fonte está localizada na posição (500 m, 10m) e a rede tem dimensões de 1000 m x 1000 m. Este modelo foi escolhido com o intuito de verificar como se comporta a propagação do erro ocasionada pela presença de várias interfaces. Na Fig. 11 são apresentadas as curvas obtidas pelos diversos algoritmos testados. Nota-se que a última curva de tempo no algoritmo de Schneider et al. (1992) está localizada um pouco mais abaixo do que os algoritmos que trabalham em coordenadas polares. Verifica-se também que o algoritmo de Reshef & Kosloff (1986) (Fig. 11b) apresenta os maiores erros.

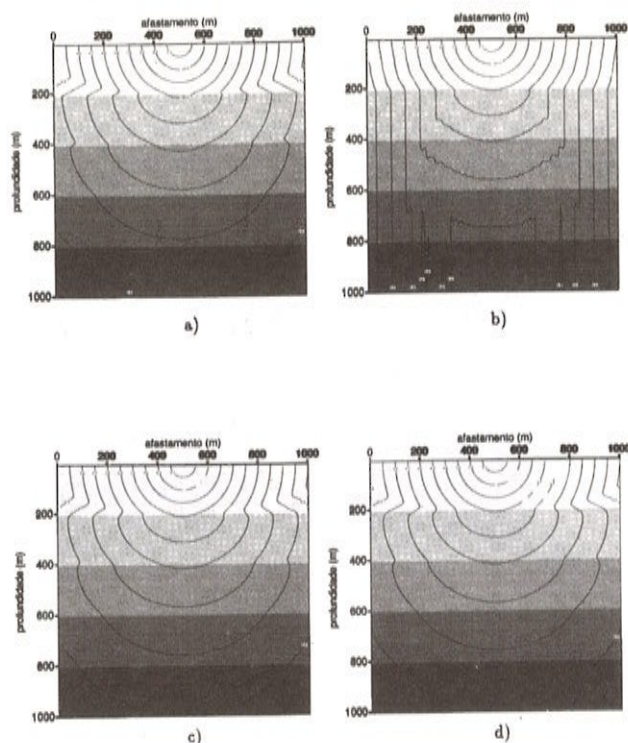


Figura 11 - Mostra as curvas de tempo obtidas com os algoritmos usando o modelo 3. As curvas têm intervalo de 0,1 s entre elas. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) e d) Kosloff (1991). No fundo de cada figura está o campo de velocidade. Neste modelo a velocidade cresce com profundidade.

Figure 11 - Shows the time curves obtained with the presented algorithms in model 3. The interval curves are of 0.15 s. a) Schneider et al. (1992), b) Reshef & Kosloff (1986), c) Zhang (1991) and d) Kosloff (1991). The velocity model is in the background of each figure. In this model, velocity increases with depth.

Já na Fig. 12 são mostrados os tempos de trânsito obtidos na posição $z = 900$ m para todos os algoritmos. Nota-se que a diferença entre os tempos obtidos pelos algoritmos que trabalham em coordenadas polares e os tempos obtidos pelo SEIS88 é maior do que a diferença de tempo entre o SEIS88 e o algoritmo de Schneider et al. (1992). Os resultados obtidos pelo algoritmo de Schneider coincidem visualmente com os resultados obtidos com o programa SEIS88. Os erros que ocorrem nos algoritmos que fazem a

extrapolação do tempo de trânsito em coordenadas polares em modelos acamados é devido a interpolação usada para se fazer as transformações de coordenadas. Na Tab. 2, que é uma comparação entre os tempos de trânsito obtidos na posição (500 m, 900 m) pelos diversos algoritmos e a solução analítica para incidência normal, nota-se que o resultado obtido pelo algoritmo de Schneider et al. (1992) é muito próximo da solução analítica. Todos os outros algoritmos apresentam erros.

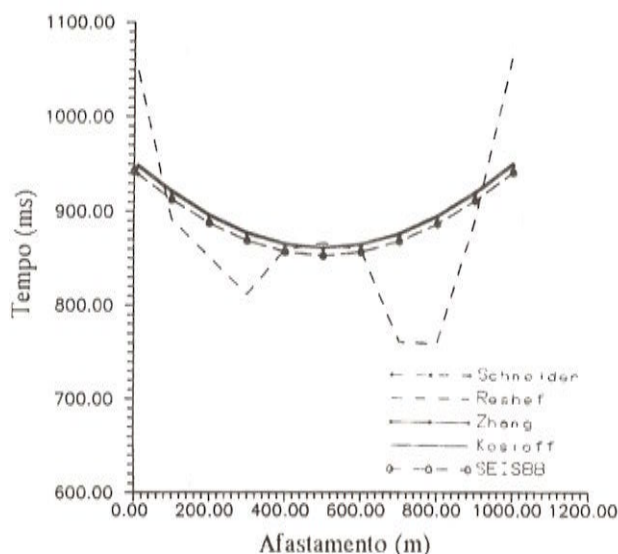


Figura 12 - Gráfico mostrando os tempos obtidos pelos quatro algoritmos e pelo programa SEIS88 com o modelo 3. Os receptores estão a 900 m abaixo da fonte, com espaçamento de 10 m.

Figure 12 - Graph showing the times obtained with the four algorithms and with the SEIS88's program in the model 3. The receivers are 800 m below the source and with spacing of 10 m.

Com relação ao tempo de computação gasto pelos programas, verifica-se que o algoritmo de Reshef & Kosloff (1986) é o mais rápido e o algoritmo de Schneider et al. (1992) apresenta os melhores resultados numéricos, entretanto o seu tempo de computação é alto com relação aos outros algoritmos (Tab. 3).

Algoritmo	Tempo (ms)	Erro absoluto (ms)
Schneider	853,333	0,000
Resheff	866,666	13,333
Zhang	861,111	7,778
Kosloff	862,042	8,709
Seis88	853,350	0,017

Tabela 2 - Tabela com os tempos de trânsito vertical obtidos no modelo 3 para os algoritmos testados e o seu respectivo erro absoluto. Os tempos foram obtidos a 900 m abaixo da fonte.

Table 2 - Table with the travel times obtained in the model 3 for the algorithms tested and their absolute errors. The times are picked in 900 m below the source.

Modelo 4 - Modelo com Zona de Baixa Velocidade

Este modelo é constituído por 2 camadas e um quadrado com baixa velocidade. A primeira camada tem uma velocidade de propagação das ondas sísmicas de $V_1 = 3000$ m/s. A segunda camada tem velocidade de propagação das ondas sísmicas de $V_2 = 6000$ m/s e apresenta uma interface inclinada, que começa na posição (0 m, 1000 m) e termina na posição (2000 m, 2000 m). O quadrado no meio do modelo tem velocidade ($V_3 = 1000$ m/s). O modelo tem uma rede de 2000 m x 2000 m. Este modelo foi escolhido com a intensão de testar os algoritmos de tempo de trânsito em regiões onde ocorrem cáusticos e zonas de sombra.

Na Fig. 13a mostra-se as curvas de tempo obtidas usando o algoritmo de Schneider et al. (1992). Verifica-se primeiramente o bom desempenho deste algoritmo em contrastes fortes de velocidade. As ondas refratadas aparecem devido à presença da camada de mais alta velocidade e estão localizadas na parte inferior do modelo. Nota-se que no quadrado de baixa velocidade aparecem ondas refratadas em todas as suas arestas. Estes eventos ocasionam o fechamento das frentes de onda. No lado direito inferior ao quadrado, nota-se uma convergência das frentes de onda e consequentemente uma convergência dos raios associados às frentes. Isto caracteriza uma região de cáustico, onde há concentração dos raios.

Modelos	Malha (m)	Δx e Δz (m)	Algoritmos - Tempo de CPU (s)			
			Schneider	Resheff x-z	Zhang r- θ	Kosloff r- θ
Modelo 1	1000 × 1000	10	1,40	0,39	0,69	1,22
Modelo 2	1000 × 1000	10	1,44	0,38	1,03	1,18
Modelo 3	1000 × 1000	10	1,41	0,38	0,92	1,19
Modelo 4	2000 × 2000	10	9,50	2,23	10,24	7,01

Tabela 3 - Tabela com os tempos de computação de cada algoritmo testado para os modelos usados. Os tempos foram estimados em uma estação de trabalho RISC / IBM modelo 340.

Table 3 - Table that shows the computing times for each of the tested algorithm in the models. The times are evaluated in workstation RISC / IBM model 340.

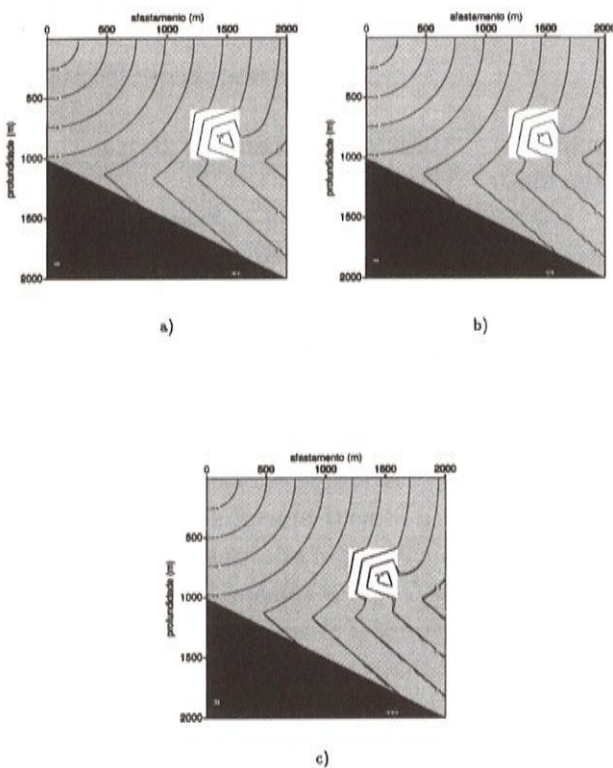


Figura 13 - As frentes de onda para o modelo 4. O intervalo entre as curvas é de 0,08 s. No fundo da figura está o campo de velocidade. Este modelo apresenta uma região quadrada de baixa velocidade. a) Schneider et al. (1992), b) Zhang (1991) e c) Comparação entre os resultados dos algoritmos de Schneider et al. (1992) e Zhang (1991).

Figure 13 - Wavefronts for the model 4. The interval between the curves is 0.08 s. The velocity model is the background.

A Fig. 13b mostra as frentes de onda obtidas usando o algoritmo de Zhang (1991). As frentes de onda apresentam as mesmas características descritas no parágrafo anterior. Observa-se que o algoritmo de Zhang (1991) apresenta uma certa instabilidade na região próxima ao quadrado de baixa velocidade, como mostra a Fig. 13c. Esta instabilidade pode ser explicada também por erros numéricos relacionados aos processos de transformação de coordenadas.

Os resultados numéricos obtidos com os programas baseados nos algoritmos de Kosloff (1991) e Reshef & Kosloff (1986) não foram mostrados. O tempo de computação do algoritmo proposto por Schneider et al. (1992) é um pouco menor que o do algoritmo proposto por Zhang (1991), como mostra a Tab. 3. Conclui-se portanto que o algoritmo proposto por Schneider et al. (1992) é o mais estável e o mais eficiente neste modelo.

Analisando-se os resultados numéricos obtidos nos 4 modelos testados, conclui-se que o programa baseado no algoritmo de Schneider et al. (1992) tem a maior acurácia. Os algoritmos de Zhang (1991) e de Kosloff (1991) apresentam bons resultados, com pequenos erros numéricos. O algoritmo de Reshef & Kosloff (1986) apresenta os piores resultados para todos os tipos de modelo.

This model presents a square region with low velocity. a) Schneider et al. (1992), b) Zhang (1991) and c) comparison between the results of Schneider's and Zhang's algorithms.

CONCLUSÃO

Neste trabalho apresentamos alguns dos principais algoritmos existentes na literatura para calcular os tempos de trânsito em uma malha regular que usam esquemas de diferenças finitas. Entre os vários algoritmos para cálculo dos tempos de chegada por nós estudados, três dos algoritmos aqui testados e comparados com o método de traçado de raios mostraram resultados satisfatórios, mesmo em modelos de velocidade com grandes contrastes. Os testes mostram que esses algoritmos podem ser usados com bastante eficiência em situações em que o método convencional de traçado de raios não apresenta bons resultados. Desde a introdução deste tipo de cálculo por Vidale (1988), diferentes aplicações vêm sendo propostas na literatura geofísica. Aplicações, tais como migração e inversão sísmica tomográfica em meios com campo arbitrário de velocidade. A partir dos resultados encontrados, os algoritmos de Schneider et al. (1992), de Zhang (1991) e Kosloff (1991) seguramente podem ser usados em técnicas de migração em profundidade do tipo Kirchhoff em meios complexos, pois mostraram-se estáveis e eficientes do ponto de vista computacional nos diversos exemplos testados.

AGRADECIMENTOS

Os autores agradecem à Petrobras pelo apoio no desenvolvimento deste trabalho. Também agradecem ao geofísico Manoel Gontijo dos Santos, pelos resultados obtidos usando o programa SEIS88 e ao geofísico Eduardo Farias pelo programa para calcular os tempos de chegada através do algoritmo de Schneider et al. (1992). Reynam Pestana agradece ao CNPq pelo suporte, através de bolsa de pesquisa, processo 300.312/88-1 (RM).

REFERÊNCIAS

- ČERVENÝ, V., MOLOTKOV, I.A. & PŠENČÍK, I. - 1977 - Ray method in seismology. Univerzita Karlova, Praha.
- ČERVENÝ, V. & PŠENČÍK, I. - 1988 - Programa de modelamento usando a técnica de traçado de raios (SEIS88). Programa de domínio público.
- KOSLOFF, D. - 1991 - Solução da equação eiconal em coordenadas polares por diferenças finitas e Runge-Kutta. Comunicação pessoal
- PODVIN, P. & LECOMTE, I. - 1991 - Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools. *Geophys. J. Int.*, **105**: 271-284.
- QIN, F., LUO, Y., OLSEN, K.B., WENYING, C. & SCHUSTER, T. - 1992 - Finite-difference solution of the eikonal equation along expanding wavefronts. *Geophysics*, **57**: 478-487.
- RESHEF, M. & KOSLOFF, D. - 1986 - Migration of Common Shot Gathers. *Geophysics*, **51**: 324-331.
- SCHNEIDER, W.A., RANZINGER, K.A., BALCH, A.H. & KRUSE, C. - 1992 - A dynamic programming approach to first arrival traveltimes in media with arbitrarily distributed velocities. *Geophysics*, **57**: 39-50.
- VAN TRIER, J. & SYMES, W. - 1991 - Upwind finite-difference calculation of traveltimes. *Geophysics*, **56**: 812-821.
- VIDALE, J. E. - 1988 - Finite-difference calculation of Traveltimes. *Bull. Seis. Soc. Am.*, **78**: 2062-2076.
- ZHANG, L. - 1991 - Finite difference calculation of Green's function. *Stanford Exploration Project*, **72**: 153-170.

Submetido em: 15/01/95

Revisado pelo(s) autor(es): 25/04/96

Aceito em: 30/05/96

COMPUTATION OF TRAVELTIMES USING FINITE-DIFFERENCE: A COMPARISON

Ray tracing techniques have been used as a conventional procedure in forward modeling to inversion and migration and in other seismic modeling applications. These applications require a travelttime map on a regular grid. However, when velocity models are complicated, the raypath connecting the source to all receivers may be difficult to determine, even if a dense ray fan is used, which can be computationally expensive. Another efficient way to obtain this travelttime map is to solve the eikonal equation directly using a finite-difference scheme. More recently, several finite-difference schemes have been developed. The methods solve the eikonal equation for first arrival time. Rather than solving for travelttime along a ray path, the eikonal equation solution gives the travelttimes along the wavefront. Wavefronts will penetrate shadow zones and also account for head waves and interference phenomena in waveguides; hence the eikonal equation solution should be more

robust than that from the ray tracing method. In this work we selected four such algorithms and evaluated them using different velocity models and compared their results with the dynamic ray tracing method. We have also compared them in terms of computing time. With these tests we find that Schneider et al. (1992), Zhang (1991), and Kosloff's (1991) algorithms work well in different types of slowness models and they are efficient and stable and also are not time expensive. The comparison of the results of all tested models showed that Scheineder's algorithm has the best accuracy. Zhang and Kosloff's algorithms also showed good results, but with small numerical error. A potential application of these algorithms could be in Kirchhoff prestack depth migration and travelttime tomographic inversion. With this new procedure we can obtain the travelttime on a regular grid on models with strong contrast of velocity even with good accuracy and efficiency.

PUBLICAÇÃO DE INFORMAÇÕES INSTITUCIONAIS

Para a publicação de informações sobre atividades de Geofísica em sua instituição, na seção ANÁLISES/RELATÓRIOS da Revista Brasileira de Geofísica, favor submeter cópia do artigo em papel e em disquete, contendo títulos e resumos em português ou espanhol, e em inglês. O artigo não poderá ultrapassar quatro páginas da Revista incluindo ilustrações e referências.