

DISSERTAÇÃO DE MESTRADO

**MODELAMENTO SÍSMICO ASSINTÓTICO
UTILIZANDO DIFERENÇAS FINITAS**

Matheus Fabiano Pila
Aluno

Prof. Dr. Lúcio Tunes dos Santos
Orientador

Prof. Dra. Amélia Novais
Co-orientadora

DMA – IMECC – UNICAMP
Março de 2005

Agradecimentos

Agradeço primeiramente ao meu orientador Prof. Lúcio Tunes dos Santos, com o qual trabalho junto desde o primeiro ano de graduação. Uma pessoa amiga e totalmente profissional, que ao longo destes anos me ajudou em tudo o que eu precisei. Outra pessoa muito importante na elaboração desta dissertação foi a Profa. Amélia Novais, que é extremamente prestativa e atenciosa, sempre se mostrando empolgada diante de qualquer desafio.

Não poderiam faltar agradecimentos à minha família, que acreditou no meu trabalho e sempre me incentivou a continuar meus estudos, e aos amigos, que sempre estão presentes tornando nossos dias menos difíceis.

Agradecimento especial a FAPESP, que financiou este projeto (processo 02/12337-5).

Sumário

Introdução	1
1 Métodos Assintóticos	3
1.1 Equação Iconal e Equação de Transporte	3
1.2 Raios Sísmicos	4
2 Métodos de Diferenças Finitas Usando Malha Fixa	7
2.1 Primeira Chegada do Tempo de Trânsito	8
2.2 Chegada Direta do Tempo de Trânsito	13
2.3 Métodos de Ordem Superior para o Cálculo da Primeira Chegada do Tempo de Trânsito	14
2.3.1 Esquemas RK-ENO de Ordem Dois	15
2.3.2 Esquemas RK-ENO de Ordem Três	17
2.3.3 Condição de Estabilidade	18
2.4 Implementação Computacional dos Esquemas RK-ENO Usando a Expansão DNO e Melhorias nos Métodos	20
2.4.1 Inicialização Refinada	22
2.4.2 Expansão Dinâmica DNO	25
2.4.3 Varredura Posterior (PS)	25
3 Método de Diferenças Finitas Adaptativo	29
3.1 Inicialização no Método Adaptativo	31
3.1.1 Aproximação por Taylor	32
3.1.2 Aproximação por Raios	34
3.1.3 Testes Computacionais	38

3.2	Interpolação no Refinamento	39
3.3	Experimentos Numéricos	42
3.3.1	Meio Homogêneo ($v(x, z) = v_0$)	44
3.3.2	Meio com Velocidade Afim em z ($v(x, z) = v_0 + gz$)	45
3.3.3	Meios Homogêneos Separados por Interface Sinclinal	47
3.3.4	Meios Homogêneos Separados por Interface Anticlinal	50
3.3.5	Meios Homogêneos Separados por Interface com Quina	50
3.3.6	Meio com Velocidade Exponencial em z ($v(x, z) = v_0 e^{gz}$)	55
3.3.7	Meio com Velocidade Não Linear em z	55
3.4	Comentários Gerais	60
4	Conclusão	63
	Referências Bibliográficas	65

Lista de Figuras

2.1	Domínio Discretizado	8
2.2	Avanço do Método no Domínio Discretizado	9
2.3	Raio Direto e Raio Refletido	10
2.4	Geometrias de Blocos de Nós na Malha	10
2.5	Modelo com um Campo de Velocidades Descontínuo	12
2.6	Primeira Chegada do Tempo de Trânsito (Vidale)	12
2.7	Chegada direta do Tempo de Trânsito (Mo & Harris)	14
2.8	Domínio Numérico e Domínio Analítico - Condição CFL	19
2.9	Erros Percentuais - Vidale e RK-ENO Ordem 2	21
2.10	Erros Percentuais - RK-ENO Ordem 2 Sem e Com o Refinamento	23
2.11	Erros Percentuais - RK-ENO de Ordem 2 e 3	24
2.12	Modelo com um Campo de Velocidades Descontínuo	27
2.13	Aproximação Após Correção Usando o Método PS	28
3.1	Função smmax	30
3.2	Erro na Inicialização - Meio Afim em z	39
3.3	Erro na Inicialização - Meio com Velocidade Exponencial em z	40
3.4	Erro na Inicialização - Meio com Velocidade Decaindo em z	40
3.5	Diferentes Interpolações	41
3.6	Modelo e Tempo de Trânsito para Testes	43
3.7	Erros nas Diferentes Interpolações	44
3.8	Tempo de Trânsito em um Meio Homogêneo	45
3.9	Erros Percentuais e Variação de Δz no Meio Homogêneo	46
3.10	Meio com Velocidade Afim em z e o Tempo de Trânsito Neste	48
3.11	Erros Percentuais e Variação de Δz no Meio com Velocidade Afim em z	49

3.12 Meio com um Refletor Sinclinal e o Tempo de Trânsito Neste	51
3.13 Erros Percentuais e Variação de Δz no Meio com um Refletor Sinclinal . .	52
3.14 Meio com um Refletor Antinclinal e o Tempo de Trânsito Neste	53
3.15 Erros Percentuais e Variação de Δz no Meio com um Refletor Anticlinal .	54
3.16 Meio com um Refletor com Quina e o Tempo de Trânsito Neste	56
3.17 Erros Percentuais e Variação de Δz no Meio com um Refletor com Quina .	57
3.18 Meio com Velocidade Exponencial em z e o Tempo de Trânsito Neste . . .	58
3.19 Erros Percentuais e Variação de Δz no Meio com Velocidade Exponencial em z	59
3.20 Meio com Velocidade Decaindo em z e o Tempo de Trânsito Neste	61
3.21 Erros Percentuais e Variação de Δz no Meio com Velocidade Decaindo em z	62

Introdução

O modelamento sísmico é uma técnica para simular a propagação de ondas no subsolo terrestre. O objetivo é predizer o que um conjunto de sensores obterá, dada uma estrutura para a sub-superfície. Esta técnica é uma ferramenta poderosa para interpretação sísmica e uma parte essencial nos algoritmos de inversão sísmica. Existem três categorias nas quais podemos classificar esses métodos: métodos diretos, métodos integrais e métodos assintóticos (Carcione et al. [4]).

Para resolver a equação da onda por *métodos diretos*, o modelo geológico é aproximado por uma malha numérica, isto é, o modelo é discretizado em um número finito de pontos e um método numérico é utilizado para encontrar uma aproximação para a solução da equação da onda completa. Podemos citar os métodos de Diferenças Finitas, Pseudo-Espectrais e Elementos Finitos. Os métodos diretos não têm restrições quanto a variabilidade do meio e podem ser muito precisos quando uma malha suficientemente fina for usada, obedecendo as condições para convergência.

Os *métodos integrais* utilizam representações integrais para o campo de ondas gerado por fontes pontuais (Princípio de Huygens). Estes métodos são mais restritos nas aplicações que os métodos diretos. No entanto, para geometrias específicas, os métodos integrais apresentam muita eficiência e fornecem resultados precisos. Podemos citar as representações integrais de Born e Kirchhoff.

Já os *métodos assintóticos* são frequentemente usados no modelamento sísmico. Esses métodos computam o campo de ondas por partes, decompondo a equação da onda em duas outras equações: *Iconal* e *Transporte*. Conforme vários textos da literatura, eles são os mais eficientes computacionalmente destas três categorias apresentadas. Uma de suas grandes vantagens é o tempo computacional, mesmo para casos em três dimensões. Podemos citar o Método dos Raios (características) e Diferenças Finitas, agora aplicadas às equações resultantes.

Neste trabalho, estudamos o modelamento sísmico assintótico utilizando os esquemas de diferenças finitas (DF) na equação iconal, para calcularmos o tempo de trânsito em uma malha fixa e posteriormente em uma adaptativa. No caso de esquemas usando malha fixa, o primeiro trabalho de maior importância nesta área foi Vidale [16], onde esquemas de DF são aplicados de forma engenhosa porém sem um bom embasamento teórico. Depois disso, vários trabalhos foram publicados sem uma relevante mudança na eficiência dos métodos. Um resultado importante foi apresentado em C. Belfi [1], pois os métodos de malha fixa, agora implementados com os esquemas Essencialmente Não Oscilatório (ENO), são aplicados de forma que exista um limitante para a inclinação das frentes de onda, devido à condição CFL. Este resultado seria a base para os algoritmos adaptativos. Em Kim & Cook [9], os métodos de malha fixa associados aos esquemas ENO e à inicialização refinada mostraram ser uma ótima ferramenta para o cálculo do tempo de trânsito, até mesmo em meios que contenham regiões de alto contraste, aplicando uma correção que usa os mesmos esquemas em outras direções (correção PS).

O maior problema dos métodos de malha fixa é o erro relativamente alto que aparece na região próxima à fonte, devido à alta curvatura da frente de onda nesta região. Os métodos adaptativos surgem como uma ótima opção para contornar este problema. Em Belfi & Symes [2], os métodos de DF associados aos esquemas ENO e Runge-Kutta (RK) conseguem contornar este problema, pois os espaçamentos da malha se ajustam conforme o erro satisfaz ou não uma tolerância fornecida pelo usuário. Por fim, em Qian & Symes [12], o método de DF adaptativo é exposto com todo o embasamento teórico necessário para seu bom funcionamento. Sua alta eficiência pode ter um alto custo, dependendo do alto contraste do meio fornecido.

Esta dissertação está organizada da seguinte forma. No Capítulo 1, apresentamos os métodos assintóticos, explicando como obter a equação iconal e as curvas características desta. No Capítulo 2 discutimos os métodos de malha fixa, onde apresentamos desde os esquemas mais simples até os mais sofisticados, para resolver numericamente a equação iconal. Também apresentamos neste capítulo as condições para um bom funcionamento e os experimentos computacionais. No Capítulo 3 apresentamos o método adaptativo, onde os esquemas do capítulo anterior são aplicados em uma malha que varia, conforme uma tolerância é satisfeita ou não em cada passo. Finalmente no Capítulo 4, apresentamos a conclusão desta dissertação, discutindo as vantagens e desvantagens de usarmos ou não uma malha que se adapte ao problema.

Capítulo 1

Métodos Assintóticos

1.1 Equação Iconal e Equação de Transporte

Seja G a *Função de Green* para a *Equação da Onda Acústica Homogênea* com densidade constante no espaço da frequência (Equação de Helmholtz). Então G satisfaz a seguinte equação

$$\left[\Delta + \frac{\omega^2}{v(\mathbf{x})^2} \right] G(\mathbf{x}, \mathbf{x}_s, \omega) = 0, \quad (1.1)$$

onde Δ é o operador laplaciano, \mathbf{x} é um ponto no domínio, \mathbf{x}_s é a posição da fonte, ω é a frequência e $v(\mathbf{x})$ é o campo de velocidades. Pode ser mostrado que, para $\omega \rightarrow \infty$, a solução tem o mesmo comportamento que a solução do problema para um meio homogêneo ($v(\mathbf{x}) = v_0$ constante), cuja expressão é conhecida (Bleistein [3]). Na verdade, queremos que

$$G(\mathbf{x}, \mathbf{x}_s, \omega) = A(\mathbf{x}, \mathbf{x}_s) e^{i\omega\tau(\mathbf{x}, \mathbf{x}_s)}, \quad (1.2)$$

onde A é a função que fornece a amplitude e τ é a função que fornece o tempo de trânsito.

De maneira a encontrar A e τ , tomemos como tentativa de solução para a função de Green, para um meio não homogêneo, uma série de potências de $(-i\omega)^{-1}$, ou seja,

$$G(\mathbf{x}, \mathbf{x}_s, \omega) = \sum_{k=0}^{\infty} (-i\omega)^{-k} A_k(\mathbf{x}, \mathbf{x}_s) e^{i\omega\tau(\mathbf{x}, \mathbf{x}_s)}, \quad (1.3)$$

e, portanto, $A = A_0$. Para obter as equações que determinam as amplitudes A_k e o tempo de trânsito τ , devemos substituir esta tentativa de solução na equação de Helmholtz e

agrupar os coeficientes, ou seja,

$$\sum_{k=0}^{\infty} \{ (-i\omega)^{2-k} [\|\nabla\tau\|^2 - v^{-2}] A_k - (-i\omega)^{1-k} [2\nabla A_k \cdot \nabla\tau + A_k \Delta\tau] + (-i\omega)^{-k} \Delta A_k \} e^{i\omega\tau} = 0. \quad (1.4)$$

Abrindo este somatório e atribuindo o valor zero para os coeficientes de cada potência de $(-i\omega)^{-1}$, obtemos, respectivamente, a equação iconal, a equação de transporte e a relação de recorrência:

$$\|\nabla\tau\|^2 = \frac{1}{v^2}, \quad (1.5)$$

$$2\nabla A_0 \cdot \nabla\tau + A_0 \Delta\tau = 0, \quad (1.6)$$

$$2\nabla A_{k+1} \cdot \nabla\tau + A_{k+1} \Delta\tau = \Delta A_k, \quad (1.7)$$

onde $\|\cdot\|$ denota a norma Euclidiana.

Assim, a nova solução que atribuímos para a função de Green pode ser construída primeiro resolvendo a equação iconal (determinando τ), e em seguida a equação de transporte (determinando A_0), para posteriormente determinarmos os A_k , com $k > 0$.

1.2 Raios Sísmicos

Uma maneira de encontrar a solução da equação iconal é aplicar o método das características. Este é um método geral que encontra soluções para equações diferenciais parciais de primeira ordem, que é o caso da equação iconal. A idéia é construir curvas $\mathbf{x}(\sigma)$ ao longo das quais a equação diferencial parcial é reduzida a uma equação diferencial ordinária (Carcione et al. [4]).

Para entendermos o conceito desta nova formulação no caso da equação iconal, vamos primeiramente considerar o caso mais simples, onde o meio é homogêneo. As superfícies de nível (considerando o caso tri-dimensional) da solução da equação iconal para este caso são esferas centrados na posição da fonte pontual \mathbf{x}_s , que correspondem às frentes de onda, com raio vt . Assim, escolhendo um ponto arbitrário \mathbf{x} em nosso domínio situado na frente de onda $\tau = t$, a linha que liga os pontos \mathbf{x}_s e \mathbf{x} é ortogonal à essa frente de onda e assim, na direção do gradiente $\nabla\tau(\mathbf{x}, \mathbf{x}_s)$. Portanto, podemos escrever esta linha

na forma parametrizada

$$\mathbf{x}(\sigma) = \mathbf{x}_s + \sigma \nabla \tau(\mathbf{x}, \mathbf{x}_s) = \mathbf{x}_s + \frac{\sigma}{v} \frac{\mathbf{x} - \mathbf{x}_s}{|\mathbf{x} - \mathbf{x}_s|}, \quad (1.8)$$

onde escalamos o parâmetro σ de forma que

$$\frac{d\mathbf{x}}{d\sigma} = \nabla \tau = \frac{1}{v} \frac{\mathbf{x} - \mathbf{x}_s}{|\mathbf{x} - \mathbf{x}_s|}. \quad (1.9)$$

Diferenciando o tempo de trânsito τ com relação a σ , obtemos

$$\frac{d\tau}{d\sigma} = \nabla \tau \cdot \frac{d\mathbf{x}}{d\sigma} = \|\nabla \tau\|^2 = \frac{1}{v^2}, \quad (1.10)$$

que é uma equação diferencial ordinária para o tempo de trânsito. Integrando com relação a σ encontramos $\tau = \sigma/v^2$. Assim, para o caso onde temos um meio homogêneo, é suficiente termos a família de linhas partindo da posição da fonte e a equação diferencial ordinária (1.10) ao longo destas linhas para resolver a equação iconal.

No caso geral, onde a velocidade do meio depende das variáveis espaciais, também queremos construir curvas $\mathbf{x}(\sigma)$ satisfazendo

$$\frac{d\mathbf{x}}{d\sigma} = \lambda \nabla \tau, \quad (1.11)$$

isto é, queremos que as curvas procuradas sejam ortogonais às frentes de onda, e assim estejam na direção de $\nabla \tau$. Na equação (1.11), λ é uma função das variáveis espaciais. Observemos que para encontrar estas curvas, precisamos conhecer τ . Para contornar este problema, diferenciamos (1.11) novamente com relação a σ ,

$$\begin{aligned} \frac{d\mathbf{x}^2}{d\sigma^2} &= \frac{d\lambda}{d\sigma} \nabla \tau + \lambda \nabla^2 \tau \frac{d\mathbf{x}}{d\sigma} \\ &= \frac{d\lambda}{d\sigma} \nabla \tau + \lambda^2 \nabla^2 \tau \nabla \tau \\ &= \frac{d\lambda}{d\sigma} \nabla \tau + \frac{\lambda^2}{2} \nabla (\|\nabla \tau\|^2) \\ &= \frac{d\lambda}{d\sigma} \nabla \tau + \frac{\lambda^2}{2} \nabla \left[\frac{1}{v^2} \right]. \end{aligned} \quad (1.12)$$

Este resultado nos fornece uma equação diferencial de segunda ordem para a curva $\mathbf{x}(\sigma)$.

Chamando $p = \nabla\tau$, temos

$$\frac{d\mathbf{x}}{d\sigma} = \lambda\mathbf{p}, \quad (1.13)$$

e assim

$$\frac{d\mathbf{x}^2}{d\sigma^2} = \frac{d\lambda}{d\sigma}\mathbf{p} + \lambda\frac{d\mathbf{p}}{d\sigma}. \quad (1.14)$$

Igualando este resultado ao obtido em (1.12) temos

$$\frac{d\mathbf{p}}{d\sigma} = \frac{\lambda}{2}\nabla\left[\frac{1}{v^2}\right]. \quad (1.15)$$

Portanto, as equações (1.13) e (1.15) mais a equação

$$\frac{d\tau}{d\sigma} = \nabla\tau \cdot \frac{d\mathbf{x}}{d\sigma} = \lambda\|\nabla\tau\|^2 = \lambda\frac{1}{v^2}, \quad (1.16)$$

formam um sistema que nos fornece as curvas características $\mathbf{x}(\sigma)$, ou em nosso caso, simplesmente raios. Este resultado é muito importante e vai ser usado em nossos métodos numéricos, visto que precisamos dizer ao algoritmo se ele está ou não usando pontos na malha de forma a respeitar a propagação da onda.

Capítulo 2

Métodos de Diferenças Finitas Usando Malha Fixa

Os métodos de Diferenças Finitas (DF) são bastante eficazes computacionalmente para resolver a equação iconal, a partir daí podemos construir as curvas de nível (no caso bidimensional) da função τ , que correspondem às frentes de onda que se propagam a partir da posição da fonte. Os métodos de DF para a equação iconal, porém, não costumam obter a solução correspondente ao problema de mais de uma frente de onda propagando. Os métodos aqui estudados (exceto o apresentado em Mo & Harris [10]) se referem ao problema que calcula a primeira onda vinda da fonte até um determinado ponto, chamada *primeira chegada do tempo de trânsito*.

Vamos estudar a equação iconal no espaço bidimensional, ou seja,

$$\left(\frac{\partial\tau}{\partial x}\right)^2 + \left(\frac{\partial\tau}{\partial z}\right)^2 = \frac{1}{v(\mathbf{x})^2}, \quad (2.1)$$

onde $\mathbf{x} = (x, z)$ com a fonte localizada na superfície ($z = 0$). Dado o domínio $\mathcal{D} = \{(x, z) \mid x \in [x_{\min}, x_{\max}] \text{ e } z \in [0, z_{\max}]\}$ onde queremos resolver esta equação, podemos discretizá-lo, criando assim uma malha com $J + 1$ nós na vertical e $I + 1$ nós na horizontal, correspondendo às direções z e x , respectivamente. Assim, $z_j = j\Delta z$ com $j = 0, 1, \dots, J$ e $x_i = x_{\min} + i\Delta x$ com $i = 0, 1, \dots, I$, onde $\Delta z = z_{\max}/J$ e $\Delta x = (x_{\max} - x_{\min})/I$. Dada a posição da fonte e o campo de velocidades nesta malha retangular, podemos dar início ao processo. Primeiramente, precisamos aproximar o valor da solução da equação iconal numa região próxima à fonte, representada pelos nós pretos na Figura 2.1, para poder-

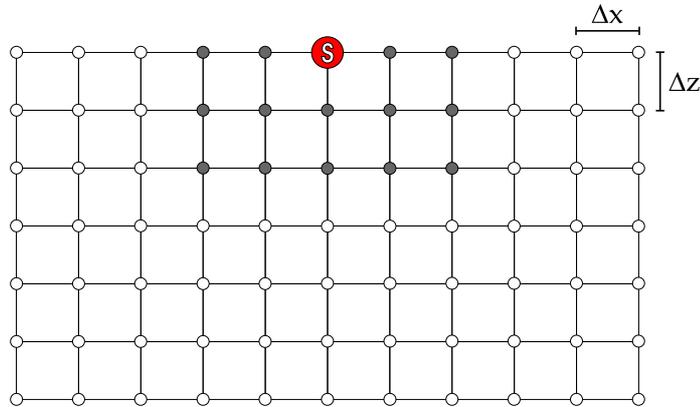


Figura 2.1: Representação de um domínio discretizado. Os nós pretos representam a malha inicializada ao redor da fonte (representada por S). Já os nós brancos representam os pontos onde devemos aproximar a solução utilizando DF.

mos aplicar um esquema numérico nesses. O tempo de trânsito é determinado integrando v^{-1} ao longo de um caminho que o torna mínimo. Os métodos estudados propõem usar estes caminhos como sendo segmentos de retas, o que corresponde a supor que a velocidade é constante nesta região, e é igual a velocidade na posição da fonte. Métodos mais avançados, conforme veremos adiante, usam uma inicialização refinada. Feito isso, podemos dar partida ao método e aplicar esquemas de DF nos pontos inicializados.

Esta é a idéia geral de como atuam os métodos de diferenças finitas. A seguir, temos dois métodos simples para resolver o problema. O objetivo de colocá-los na mesma seção é mostrar que pequenas mudanças nos métodos podem fornecer soluções bem diferentes.

2.1 Primeira Chegada do Tempo de Trânsito

O método que segue foi apresentado em Vidale [16], e funciona bem para um grande número de problemas. O espaçamento é igual tanto na vertical quanto na horizontal, isto é, $\Delta z = \Delta x = h$. Considerando a velocidade constante, calculamos analiticamente o tempo de trânsito numa região de dois nós ao redor da fonte (conforme Figura 2.1). Na Figura 2.2 temos a configuração da malha, dada uma região onde a solução já está calculada. Os nós pretos indicam os pontos da malha onde o tempo já foi calculado e os nós brancos os pontos onde o tempo será calculado. Podemos ver esta região formada pelos pontos calculados como um retângulo, e queremos que este retângulo se expanda

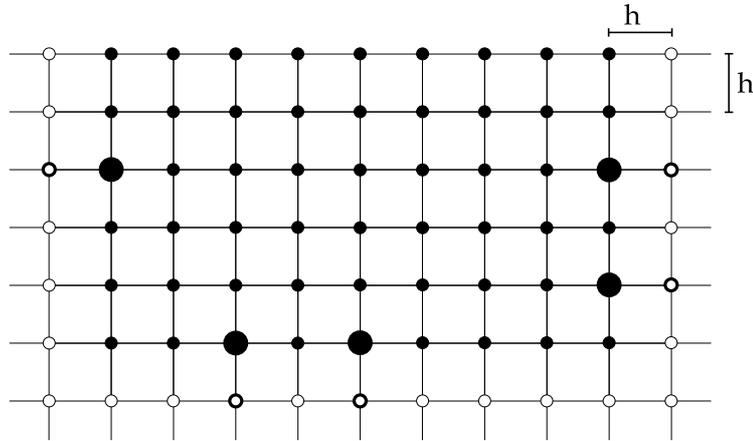


Figura 2.2: Os nós pretos representam pontos onde a aproximação já foi calculada, onde os maiores representam os mínimos locais em cada aresta. Os nós brancos estão sobre as próximas três arestas do retângulo em expansão ao redor da fonte, onde o tempo de trânsito deve ser aproximado.

até que o domínio fornecido seja preenchido. Assim, cada passo deste algoritmo consiste em calcularmos o tempo de trânsito seqüencialmente em cada uma das três arestas deste retângulo. Este modo de expansão é conhecido como DNO (“*Down'n'out*”). Como queremos que os tempos sejam determinados de forma *upwind* (entre esquema atrasado e avançado, o método escolhe aquele que respeita a propagação da onda), o cálculo dos nós brancos deve ser feito de forma que seus vizinhos do passo anterior estejam organizados de forma crescente. Ou ainda, se pensarmos de uma forma mais engenhosa, devemos dar início ao processo pelos mínimos locais da última aresta calculada, representados pelos pontos pretos maiores na borda do retângulo em expansão, e calcular o tempo de trânsito nos respectivos nós vizinhos mais próximos, representados pelos pontos brancos com contorno maior. Em seguida, os nós vizinhos são calculados de maneira consecutiva, até chegar nos máximos locais. Quando ocorrer de um máximo local ter duas maneiras de ser calculado (pela direita e pela esquerda), o valor assumido deve ser o tempo de trânsito menor entre os dois valores, pois queremos a frente de onda que chega primeiro.

Esta forma de resolução apresenta um problema. Como vamos sempre nos afastando da fonte, estamos considerando que $\partial\tau/\partial z > 0$ em cada aresta horizontal do retângulo em expansão ao redor da fonte (consideração análoga com a derivada em x ocorre nas arestas verticais do mesmo retângulo). Logo, pode acontecer de calcularmos um tempo de trânsito que necessita da resolução em passos posteriores. Como ilustrado na Figura 2.3, o ponto

G sente a chegada da frente de onda por dois caminhos diferentes. Caso a velocidade v_2 no meio de baixo seja maior que a velocidade v_1 do meio de cima, dependendo da posição do ponto G a onda que atingiu a descontinuidade chega primeiro neste local. Métodos corretivos, como o PS (“*post sweeping*”) apresentado mais adiante, tentam sanar este problema.

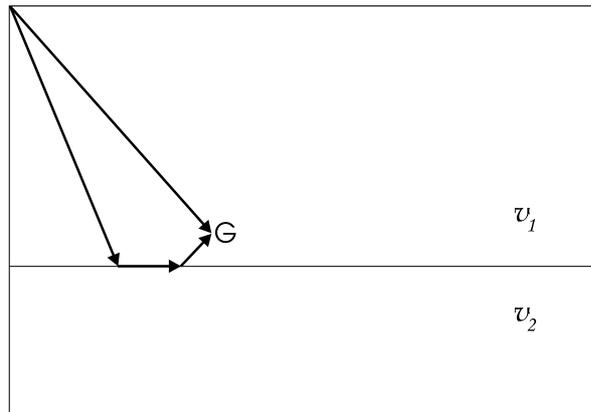


Figura 2.3: O raio que atinge a região de descontinuidade pode chegar antes que o raio direto.

Uma vez visto como o algoritmo procede, falta apresentarmos os esquemas usados nestes cálculos. Na Figura 2.4 temos as duas geometrias possíveis usadas no método. Como anteriormente, os nós pretos indicam onde o tempo de trânsito já foi determinado e os brancos onde os tempos serão calculados. A primeira geometria refere-se ao cálculo de tempos em e cujo vizinho do passo anterior é um mínimo local (m). Já a segunda geometria refere-se a todos os outros pontos.

Primeiramente, vamos discretizar as derivadas parciais no ponto m usando as diferenças

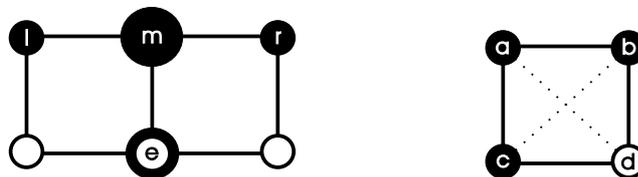


Figura 2.4: Duas geometrias de blocos de nós usados para calcular o tempo de trânsito.

centrada e avançada. Logo,

$$\left(\frac{t_{\mathbf{r}} - t_{\mathbf{l}}}{2h}\right)^2 + \left(\frac{t_{\mathbf{e}} - t_{\mathbf{m}}}{h}\right)^2 = \frac{1}{v_{\mathbf{m}}^2}. \quad (2.2)$$

Considerando que a onda está sempre se afastando da fonte, temos então que

$$t_{\mathbf{e}} = t_{\mathbf{m}} + h\sqrt{\frac{1}{v_{\mathbf{m}}^2} - \left(\frac{t_{\mathbf{r}} - t_{\mathbf{l}}}{2h}\right)^2}. \quad (2.3)$$

Para o caso da segunda geometria, vamos discretizar as derivadas parciais no ponto que está no meio do quadrado formado pelos quatro pontos. Assim, usando diferenças centradas nestes novos eixos ortogonais (representadas pelas linhas pontilhadas na Figura 2.4), obtemos

$$\left(\frac{t_{\mathbf{d}} - t_{\mathbf{a}}}{\sqrt{2}h}\right)^2 + \left(\frac{t_{\mathbf{c}} - t_{\mathbf{b}}}{\sqrt{2}h}\right)^2 = \frac{1}{v^2}, \quad (2.4)$$

ou ainda

$$t_{\mathbf{d}} = t_{\mathbf{a}} + \sqrt{2(h/v)^2 - (t_{\mathbf{c}} - t_{\mathbf{b}})^2}, \quad (2.5)$$

onde $v = (v_{\mathbf{a}} + v_{\mathbf{b}} + v_{\mathbf{c}} + v_{\mathbf{d}})/4$. O esquema é aplicado de forma que $t_{\mathbf{a}} < t_{\mathbf{b}}$, pois assim satisfaz o método *upwind* descrito anteriormente. Isto é válido pois a equação iconal não muda após sofrer rotações. Para vermos isto, basta fazer $u = (x+y)/\sqrt{2}$ e $v = (-x+y)/\sqrt{2}$ e assim

$$\left(\frac{\partial\tau}{\partial x}\right)^2 + \left(\frac{\partial\tau}{\partial z}\right)^2 = \left(\frac{\partial\bar{\tau}}{\partial u}\right)^2 + \left(\frac{\partial\bar{\tau}}{\partial v}\right)^2 = \frac{1}{v^2}, \quad (2.6)$$

onde $\tau(x, z) = \bar{\tau}(u, v)$.

Quando ocorrer de o argumento da raiz ser negativo, a alternativa proposta em Vidale [16] é substituir o valor do argumento por zero. Assim, passamos a ter que $t_{\mathbf{d}} = t_{\mathbf{a}}$, o que significa que estamos considerando que os pontos \mathbf{d} e \mathbf{a} estão na mesma frente de onda.

Para ilustrar o cálculo da primeira chegada do tempo de trânsito, aplicamos o método no modelo da Figura 2.5, que possui um campo de velocidades de 2000 m/s no domínio fornecido, exceto na caixa que ultrapassa 500 m na profundidade e 2500 m na superfície, onde a velocidade é 5000 m/s. A fonte está posicionada em $x = 2000$ m na superfície. A aproximação resultante está na Figura 2.6.

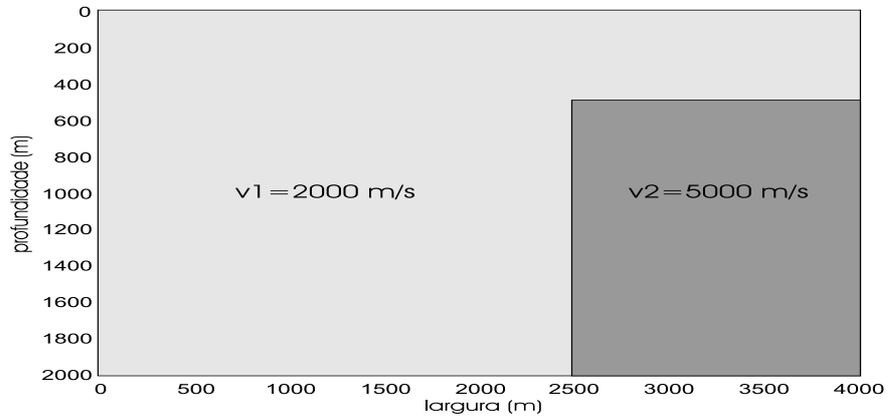


Figura 2.5: Modelo com um campo de velocidades descontínuo.

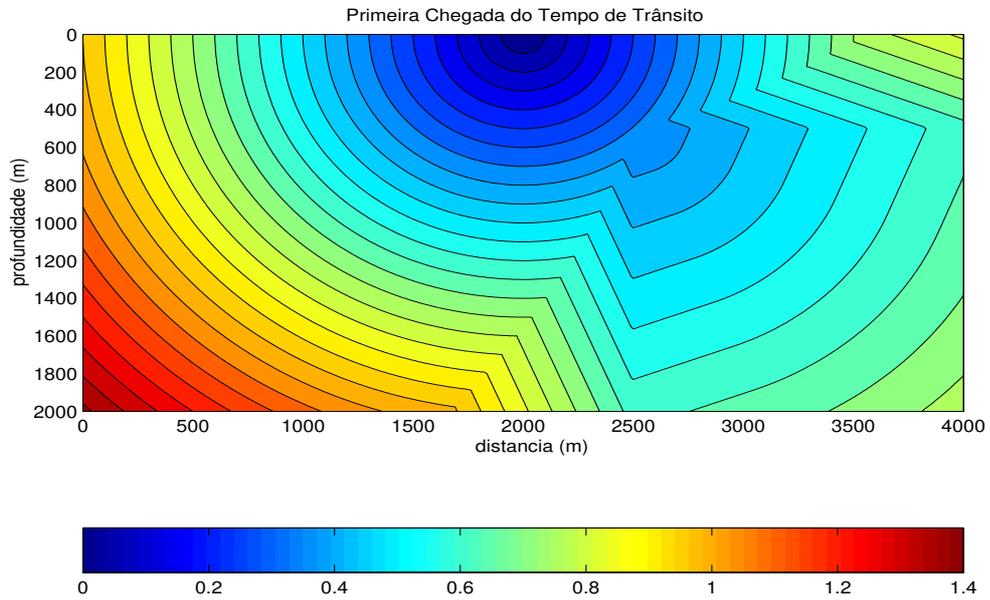


Figura 2.6: Primeira chegada do tempo de trânsito para o modelo apresentado na Figura 2.5, calculada usando o método apresentado por Vidale [16].

2.2 Chegada Direta do Tempo de Trânsito

Com pequenas mudanças, é possível criar um método que calcule a onda que propaga de maneira direta entre dois pontos, chamada *chegada direta do tempo de trânsito*. Este método foi apresentado em Mo & Harris [10] e é muito parecido com o exposto anteriormente.

A primeira mudança é quando um máximo local em uma das arestas do retângulo em expansão tem duas formas de ser calculado. Desta vez, o valor assumido é aquele que usa informação de pontos mais próximos à fonte. O motivo disto é porque estamos supondo que a onda direta está sempre se afastando da fonte. A outra mudança é quando o argumento da raiz for negativo. Como queremos a onda que propaga direto, devemos supor que a onda foi transmitida sobre a aresta do bloco (referente a primeira ou segunda geometria da Figura 2.4) onde está sendo calculado o tempo de trânsito. Assim, para o cálculo de mínimos locais, conforme a primeira geometria da Figura 2.4 passamos a ter

$$t_e = t_m + h/v_e . \quad (2.7)$$

Já no cálculo dos outros tempos, conforme o segundo bloco da mesma figura, passamos a ter

$$t_d = \min(t_b, t_c) + h/v_d . \quad (2.8)$$

No cálculo da chegada direta do tempo de trânsito, não temos problema ao estarmos sempre nos afastando da fonte. Diferentemente do problema anterior, aqui não nos interessam ondas refletidas, que podem chegar antes da onda direta. Na Figura 2.7, temos a aproximação obtida usando este método, para o mesmo campo de velocidades do último exemplo (ver Figura 2.5). A principal diferença entre esta aproximação e aquela obtida anteriormente (ver Figura 2.6) é que agora temos apenas as ondas diretas e não as refletidas, que haviam sido geradas por causa da descontinuidade do campo de velocidades do modelo.

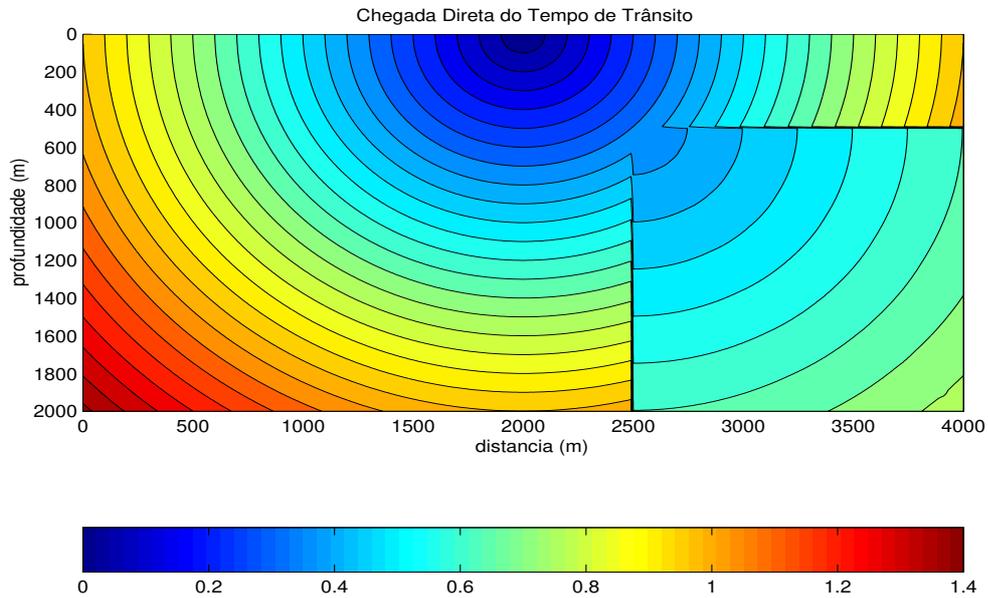


Figura 2.7: Chegada Direta do tempo de trânsito para o modelo apresentado na Figura 2.5, calculada usando o método apresentado por Mo & Harris [10].

2.3 Métodos de Ordem Superior para o Cálculo da Primeira Chegada do Tempo de Trânsito

Métodos como o apresentado na última seção não tornam possível uma modificação afim de aumentar sua eficácia. Mas quando evoluímos em apenas uma variável, passamos a ter uma equação diferencial nesta. Assim, considerando $\partial\tau/\partial z > 0$, podemos reescrever a equação iconal como uma equação de evolução em z , ou seja,

$$\frac{\partial\tau}{\partial z} = \sqrt{\frac{1}{v^2} - \left(\frac{\partial\tau}{\partial x}\right)^2} = H\left(v, \frac{\partial\tau}{\partial x}\right). \quad (2.9)$$

Usamos a notação H para o operador propositalmente, pois esta equação é uma equação Hamilton-Jacobi.

As soluções para este tipo de equação foram bastante discutidas na literatura. Uma das primeiras idéias é introduzir condições de entropia e soluções com viscosidade, devido à não unicidade das soluções (Crandall & Lions [5]). Outra classe importante de métodos numéricos para resolver (2.9) é a classe de esquemas monótonos discutida em

Crandall & Lions [6]. No entanto, apenas esquemas monótonos de primeira ordem são adequados, pois os de alta ordem apresentam oscilações que geralmente ocorrem na presença de derivadas descontínuas. Em nosso caso, as derivadas descontínuas aparecem frequentemente, principalmente quando o método passa por uma interface.

Os esquemas essencialmente não oscilatórios (ENO) foram implementados com grande sucesso para resolver o problema hiperbólico de leis de conservação, conforme apresentado por vários autores (Harten & Osher [7], Harten et al. [8], Shu & Osher [13] e [14]). A idéia chave do esquema ENO é usar uma interpolação local que escolhe a interpolação mais suave possível. Em Osher & Shu [11], foi discutida a proximidade dos problemas hiperbólicos de leis de conservação e as equações Hamilton-Jacobi, de onde concluiu-se que os esquemas ENO podem ser aplicados para encontrar aproximações de alta ordem nas equações Hamilton-Jacobi.

Seguindo esta mesma idéia, Belfi [1] construiu os esquemas ENO agora aplicados à equação iconal evoluindo em apenas uma variável. Assim, temos um método onde são aplicados esquemas Runge-Kutta (RK) na variável z e esquemas ENO na variável x . Neste método, a inicialização também considera a velocidade constante em uma região, e igual a velocidade na posição da fonte. A diferença é que ao invés de a inicialização ocorrer em um pequeno retângulo composto por nós ao redor da fonte, ela ocorre em toda uma região logo abaixo da superfície, em uma certa profundidade fornecida. A seguir, apresentamos como são construídos estes esquemas.

2.3.1 Esquemas RK-ENO de Ordem Dois

Aplicando o esquema Runge-Kutta de ordem dois na variável z da equação (2.9) temos

$$\tilde{\tau}_{i,j+1} = \tau_{i,j} + \Delta z H(v_{i,j}, \hat{D}_{i,j}), \quad (2.10)$$

$$\tau_{i,j+1} = \frac{1}{2}(\tau_{i,j} + \tilde{\tau}_{i,j+1} + \Delta z H(v_{i,j+1}, \hat{D}_{i,j+1})), \quad (2.11)$$

onde os índices i e j correspondem a x_i e z_j , respectivamente. Vejamos como é construído o operador $\hat{D}_{i,j}$, que é o esquema *upwind* ENO de ordem dois que aproxima $\partial\tau/\partial x$.

Definindo os operadores de diferenças avançado e atrasado,

$$D_1^\pm \tau = \pm \frac{\tau(x \pm \Delta x, z) - \tau(x, z)}{\Delta x}, \quad (2.12)$$

podemos escrever (usando a expansão de Taylor),

$$\frac{\partial \tau}{\partial x} = D_1^\pm \tau \mp \frac{\Delta x}{2} \frac{\partial^2 \tau}{\partial x^2} + \mathcal{O}(\Delta x^2). \quad (2.13)$$

Queremos que a aproximação da derivada primeira seja de ordem dois. Podemos então, aproximar a derivada segunda acima por esquemas até de ordem um. Vamos usar o esquema ENO de segunda ordem:

$$\frac{\partial^2 \tau}{\partial x^2} \approx M(D_1^- D_1^+ \tau, D_1^\pm D_1^\pm \tau), \quad (2.14)$$

onde,

$$M(\xi, \eta) = \begin{cases} 0, & \text{se } \xi \eta \leq 0 \\ \xi, & \text{se } |\xi| \leq |\eta| \text{ e } \xi \eta > 0 \\ \eta, & \text{se } |\xi| > |\eta| \text{ e } \xi \eta > 0 \end{cases}. \quad (2.15)$$

Logo, a aproximação ENO de segunda ordem é dada por

$$\frac{\partial \tau}{\partial x} \approx D_2^\pm \tau = D_1^\pm \tau \mp \frac{\Delta x}{2} M(D_1^- D_1^+ \tau, D_1^\pm D_1^\pm \tau). \quad (2.16)$$

Notemos que $D_1^- D_1^+ \tau$ e $D_1^\pm D_1^\pm \tau$ são os esquemas de DF centrados em (x, z) e $(x \pm \Delta x, z)$, respectivamente, e eles fornecem a curvatura nestes pontos. Logo, a função M escolhe a componente menos oscilatória para realizar a aproximação. Outra vantagem, é que caso estejamos em uma região onde há mudança muito brusca na velocidade do meio, a função M calcula a aproximação usando os pontos cujos tempos são menos oscilatórios, permitindo assim que o método *upwind* funcione de maneira mais eficaz. Para finalizar, vamos definir o esquema *upwind* ENO de segunda ordem que aproxima $\partial \tau / \partial x$

$$\frac{\partial \tau}{\partial x} \approx \hat{D} \tau = \max\{|\max\{D_2^- \tau, 0\}|, |\min\{D_2^+ \tau, 0\}|\}, \quad (2.17)$$

e assim, $\hat{D}_{i,j} = \hat{D} \tau(x_i, z_j)$.

2.3.2 Esquemas RK-ENO de Ordem Três

Analogamente ao que foi feito no caso anterior, vamos aplicar o esquema Runge-Kutta de ordem três na variável z . Logo,

$$\tilde{\tau}_{i,j+1} = \tau_{i,j} + \Delta z H(v_{i,j}, \check{D}_{i,j}), \quad (2.18)$$

$$\hat{\tau}_{i,j+1/2} = \frac{1}{4}(3\tau_{i,j} + \tilde{\tau}_{i,j+1} + \Delta z H(v_{i,j+1}, \check{D}_{i,j+1})), \quad (2.19)$$

$$\tau_{i,j+1} = \frac{1}{6}(2\tau_{i,j} + 4\hat{\tau}_{i,j+1/2} + 4\Delta z H(v_{i,j+1/2}, \check{D}_{i,j+1/2})). \quad (2.20)$$

Vejamos agora como fica o operador $\check{D}_{i,j}$, que é o esquema *upwind* ENO de terceira ordem. Da expansão de Taylor, temos

$$\frac{\partial \tau}{\partial x} = D_1^\pm \tau \mp \frac{\Delta x}{2} \frac{\partial^2 \tau}{\partial x^2} - \frac{\Delta x^2}{6} \frac{\partial^3 \tau}{\partial x^3} + \mathcal{O}(\Delta x^3). \quad (2.21)$$

Agora, queremos que a aproximação da derivada primeira seja de ordem três. Podemos então, aproximar a derivada segunda acima por esquemas de ordem dois, e a derivada terceira por esquemas de ordem um. Novamente, vamos usar os esquemas ENO para aproximar essas derivadas, ou seja,

$$\frac{\partial^2 \tau}{\partial x^2} \approx M(D_1^\pm D_1^\pm \tau - \Delta x D_1^\pm D_1^\pm D_1^\pm \tau, D_1^- D_1^+ \tau), \quad (2.22)$$

$$\frac{\partial^3 \tau}{\partial x^3} \approx M(D_1^\pm D_1^\pm D_1^\pm \tau, M(D_1^+ D_1^+ D_1^- \tau, D_1^+ D_1^- D_1^- \tau)). \quad (2.23)$$

Assim, a aproximação ENO de terceira ordem é dada por

$$\begin{aligned} \frac{\partial \tau}{\partial x} \approx D_3^\pm \tau &= D_1^\pm \tau \mp \frac{\Delta x}{2} M(D_1^\pm D_1^\pm \tau - \Delta x D_1^\pm D_1^\pm D_1^\pm \tau, D_1^- D_1^+ \tau) \\ &\quad - \frac{\Delta x^2}{6} M(D_1^\pm D_1^\pm D_1^\pm \tau, M(D_1^+ D_1^+ D_1^- \tau, D_1^+ D_1^- D_1^- \tau)). \end{aligned} \quad (2.24)$$

Concluindo, vamos definir o esquema *upwind* ENO de terceira ordem que aproxima $\partial \tau / \partial x$

$$\frac{\partial \tau}{\partial x} \approx \check{D} \tau = \max\{|\max\{D_3^- \tau, 0\}|, |\min\{D_3^+ \tau, 0\}|\}, \quad (2.25)$$

e assim, $\check{D}_{i,j} = \check{D} \tau(x_i, z_j)$.

2.3.3 Condição de Estabilidade

Podemos ver a estabilidade de um método da seguinte maneira: para um esquema de diferenças estável, pequenos erros na condição inicial causam pequenos erros na solução. Para obter condições de estabilidade para um esquema, podemos usar o critério de von Neumann e obtermos uma condição para o espaçamento na malha.

Este critério é usado apenas para problemas lineares. Em problemas não-lineares, uma condição de estabilidade teria que ser feita para cada equação diferente. Ou seja, teríamos que estudar condições para convergência separadamente para cada equação. Então, vamos ao menos garantir uma condição necessária para a estabilidade. Uma condição necessária para que qualquer método numérico seja estável é que o domínio de dependência numérico deve incluir o domínio de dependência físico, ao menos no limite $\Delta z, \Delta x \rightarrow 0$. Esta condição foi criada por Courant, Friedrichs e Lewy e é conhecida como condição CFL (Thomas [15]). O domínio de dependência numérica do ponto (x_i, z_{j+1}) em relação aos pontos que estão no j -ésimo nível é

$$N_j = \{(x, z_j) : |x - x_i| \leq \Delta x\} . \quad (2.26)$$

O intervalo pode ser maior, como no caso em que as derivadas atrasada ou avançada em x usam mais de dois pontos, mas tomamos este intervalo pois ele é o menor possível, englobando assim todos os esquemas.

Quanto ao domínio físico, usamos o resultado que diz que as curvas características da equação iconal são formadas pela família de curvas ortogonais às frentes de onda, denominadas raios (Bleistein [3]). Assim, dado um ponto sobre um raio, o tempo de trânsito neste ponto depende do tempo de trânsito em todos os pontos situados sobre o mesmo raio. Queremos então que o raio que passa pelo ponto (x_i, z_{j+1}) intercepte o domínio de dependência numérica (o segmento formado pelos três pontos escuros na Figura 2.8). Supondo que os raios nesta região discretizada são todos paralelos e aproximados por retas e sendo θ o ângulo que o vetor $\vec{p} = (\partial\tau/\partial x, \partial\tau/\partial z)$ faz com a vertical, devemos então impor que

$$\Delta z \tan \theta < \Delta x . \quad (2.27)$$

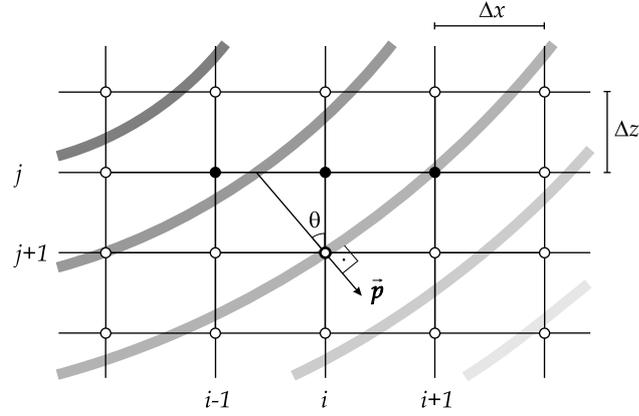


Figura 2.8: O raio que passa pelo ponto a ser calculado deve interceptar o domínio de dependência numérica.

O vetor \vec{p} é normal a frente de onda, e portanto, podemos escrever

$$\tan \theta = \frac{\partial \tau / \partial x}{\partial \tau / \partial z}. \quad (2.28)$$

Assim, a condição CFL torna-se

$$\Delta z_{\text{CFL}} < \frac{\Delta x \sqrt{(1/v)^2 - (\partial \tau / \partial x)^2}}{|\partial \tau / \partial x|} = \Delta x \frac{H(v, \partial \tau / \partial x)}{|\partial \tau / \partial x|}. \quad (2.29)$$

Como estamos expandindo na direção z apenas, pode ocorrer de termos raios fazendo ângulos muito próximos de 90° com a vertical e, assim, Δz tendendo a zero. A solução para este problema é fixar um ângulo θ_{max} , e assumir que nenhum raio faça um ângulo maior que este com a vertical, isto é, calcular a aproximação apenas numa região onde os raios fazem um ângulo de no máximo θ_{max} com a vertical, com $0 \leq \theta_{\text{max}} \leq \pi/2$. Uma maneira de fazer isto é reescrever a equação (2.9) na forma

$$\frac{\partial \tau}{\partial z} = \sqrt{\max \left\{ \frac{1}{v^2} - \left(\frac{\partial \tau}{\partial x} \right)^2, \frac{\cos^2 \theta_{\text{max}}}{v^2} \right\}}, \quad (2.30)$$

com a condição CFL escrita agora como

$$\Delta z < \frac{\Delta x}{\tan \theta_{\text{max}}}. \quad (2.31)$$

2.4 Implementação Computacional dos Esquemas RK-ENO Usando a Expansão DNO e Melhorias nos Métodos

Na seção anterior, vimos como escrever esquemas de alta ordem. Agora, vamos aplicá-los em um método que usa a expansão DNO, como apresentado em Kim & Cook [9], e compará-los com o método apresentado em Vidale [16].

A inicialização é a mesma que a discutida no início deste capítulo (ver Figura 2.1). O diferencial é que agora vamos aplicar os esquemas RK-ENO de ordem dois e três, tanto na aresta horizontal quanto nas verticais, do retângulo que se expande ao redor da fonte. Logo, devemos tomar o cuidado de neste método a condição CFL ser respeitada. Agora que estamos expandindo em três direções diferentes, não podemos simplesmente escrever a equação iconal na forma (2.30), para cada direção. A solução proposta em Kim & Cook [9] é a de que devemos satisfazer a condição CFL em cada passo da expansão ao redor da fonte, onde podemos dar sub-passos de modo a garantir que a restrição seja satisfeita.

Vamos discutir o algoritmo apenas para o caso da aresta horizontal. No cálculo de uma nova aresta, podemos estimar o valor de $\tan\theta$, usando a relação (2.27), em cada ponto (x_i, z_j) , aproximando as derivadas parciais usando DF. Em seguida, tomamos a maior tangente entre as encontradas, e pela condição CFL encontramos o valor Δz_{CFL} . Caso $\Delta z > \Delta z_{\text{CFL}}$, devemos impor um sub-passo com espaçamento Δz_{CFL} e calcular a aproximação nesta nova profundidade $z_j + \Delta z_{\text{CFL}}$, de modo a garantir que a condição CFL seja satisfeita. Devemos então calcular outros sub-passos com o espaçamento Δz_{CFL} , até que o espaçamento Δz seja coberto pela soma destes espaçamentos da condição CFL.

Em regiões de alta mudança no campo de velocidades, podemos ter raios com inclinações muito próximas a 90° , o que implica na realização de vários desses sub-passos para tentar garantir que a condição CFL seja satisfeita. Em Kim & Cook [9], este problema é simplesmente ignorado. O máximo que o autor comenta é que devemos ter um limitante para este sub-passo para evitarmos difusão. Logo, devemos fornecer sim um espaçamento mínimo Δz_{min} , mas quando ocorrer o caso $\Delta z_{\text{CFL}} < \Delta z_{\text{min}}$, simplesmente damos um sub-passo do tamanho limitado e juntamente com isto, trabalhamos com a equação na forma (2.30), onde o ângulo θ_{max} é claramente $\arctan(\Delta x/\Delta z_{\text{min}})$.

Na Figura 2.9, os gráficos da esquerda e da direita no topo representam os erros

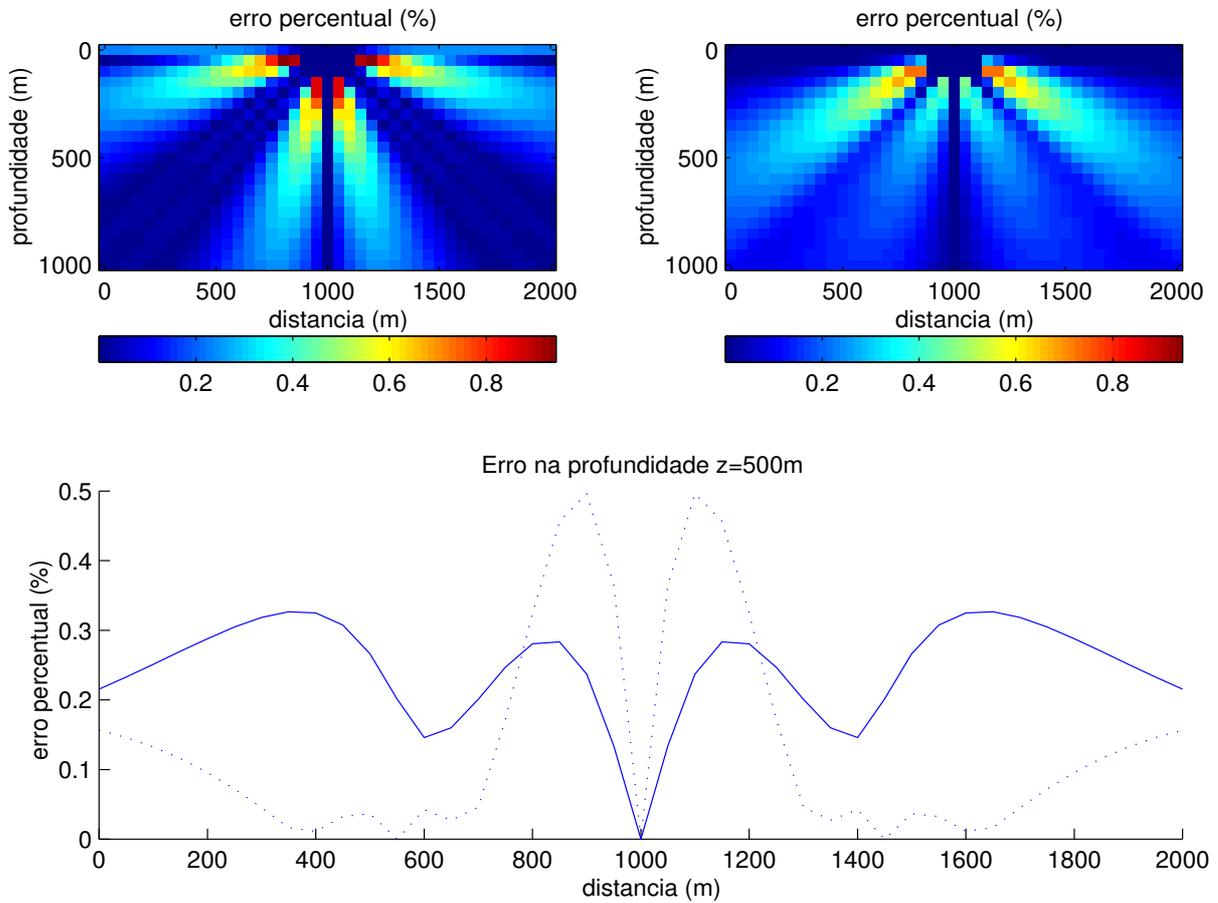


Figura 2.9: Na parte superior, temos a comparação entre os métodos: Vidale [16] (à esquerda) e RK-ENO de ordem dois (à direita). Na parte inferior, temos as curvas dos erros obtidos usando os métodos: Vidale [16] (linha pontilhada) e RK-ENO 2 (linha sólida), na profundidade $z = 500$ m.

relativos do método apresentado em Vidale [16] e do esquema RK-ENO de ordem dois, respectivamente. Na parte inferior, temos o erro relativo na profundidade de 500 m. O modelo é de um meio homogêneo com velocidade constante $v = 1000$ m/s, com a fonte localizada em $x = 1000$ m na superfície. A solução é conhecida e dada por

$$\tau(x, z) = \frac{\sqrt{(x - 1000)^2 + z^2}}{1000} . \quad (2.32)$$

O espaçamento é $\Delta x = \Delta z = 50$ m. A segunda aproximação apresenta resultados mais satisfatórios, visto que a curva do erro é menos oscilatória e atinge um máximo menor.

2.4.1 Inicialização Refinada

A inicialização do tempo de trânsito perto da fonte considerando uma pequena região de velocidade constante (conforme visto na Seção 2), não produz um resultado satisfatório. Como a frente de onda tem uma grande curvatura nesta região, o erro nos cálculos também é grande. Sendo assim, os esquemas de DF necessitam de uma malha mais refinada neste local. Na literatura, este método é conhecido como Refinamento Uniforme da Malha Localmente (LUMR) e está bem descrito em Kim & Cook [9].

O método de refinamento que foi usado nos testes que seguem não é o LUMR citado na literatura. Aqui, apenas inicializamos o problema refinando cada espaçamento Δz e Δx em outros oito, num raio de oito espaçamentos ao redor da fonte, aplicando nesta nova malha o esquema RK-ENO de ordem dois apresentado anteriormente. Este número de espaçamentos ao redor da fonte, assim como apresentado no método LUMR, é escolhido pelo usuário e não tem relação com o método.

Na Figura 2.10, temos no topo o gráfico do erro relativo usando o esquema RK-ENO de ordem dois sem e com o refinamento, respectivamente. O modelo é o mesmo do caso anterior, $v = 1000$ m/s, e o espaçamento é $\Delta x = \Delta z = 50$ m. A fonte está localizada em $x = 1000$ m na superfície. O erro é muito menor no segundo caso, onde o refinamento proporcionou uma inicialização muito melhor.

Usando a inicialização refinada, podemos testar o esquema RK-ENO de ordem três, pois este necessita de uma malha com um número maior de pontos ao redor da fonte. Na Figura 2.11, o modelo é de velocidade variável, $v(z) = z + 1000$ m/s, novamente com a fonte localizada em $x = 1000$ m na superfície. A solução também é conhecida e dada por

$$\tau(x, z) = \arccos \left[\frac{(x - 1000)^2 + v(z)^2 + v(0)^2}{2 v(z) v(0)} \right]. \quad (2.33)$$

O espaçamento $\Delta x = \Delta z = 25$ m. Observemos que num raio de 200 m ao redor da fonte (oito vezes o espaçamento), o erro é o mesmo em ambos os métodos, pois ambos usaram a mesma inicialização refinada. Conforme a distância em relação a fonte aumenta, o esquema de ordem três apresenta uma aproximação melhor.

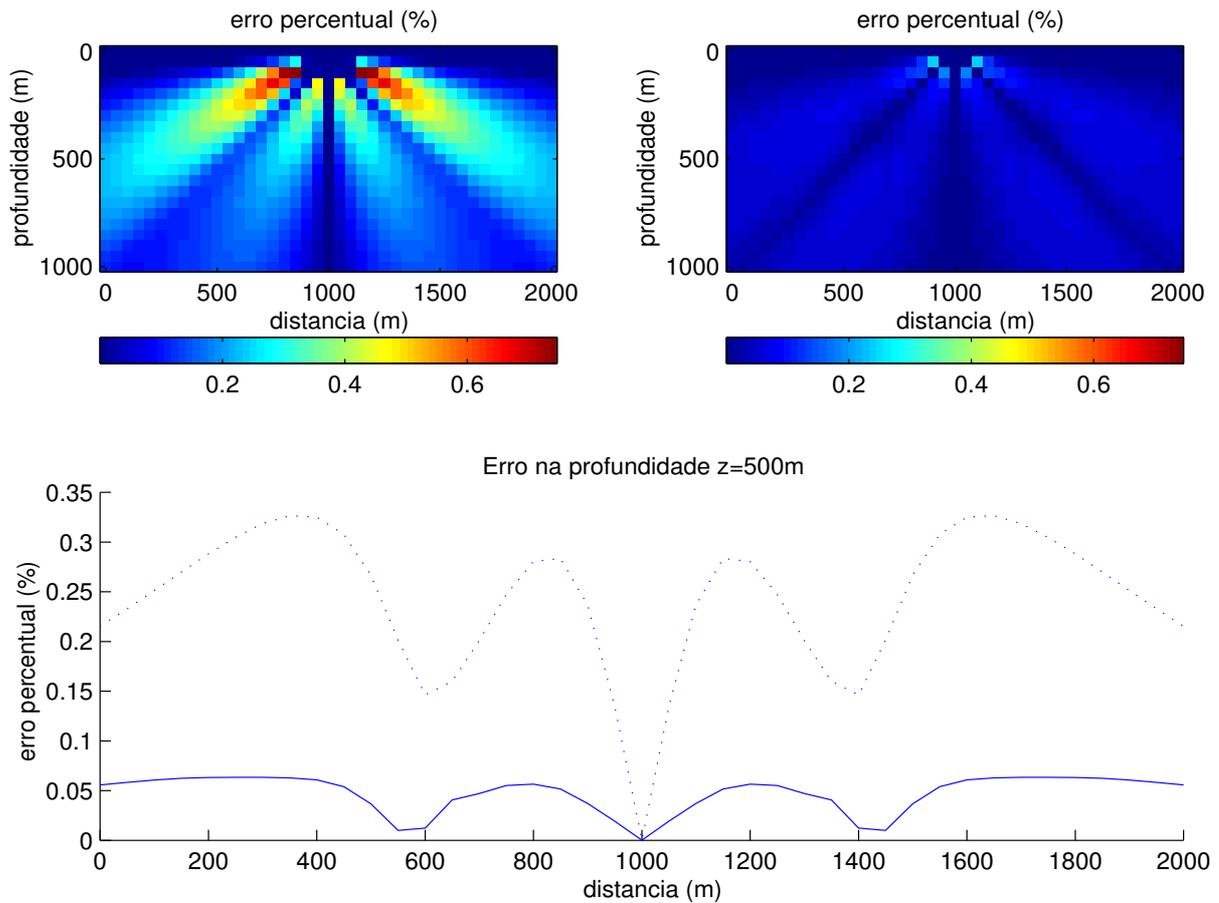


Figura 2.10: Na parte superior, temos o erro do esquema RK-ENO de ordem dois sem e com o refinamento, respectivamente. Na parte inferior, temos o erro na profundidade de 500 m usando o esquema RK-ENO de ordem dois sem o refinamento (linha pontilhada) e com o refinamento (linha sólida).

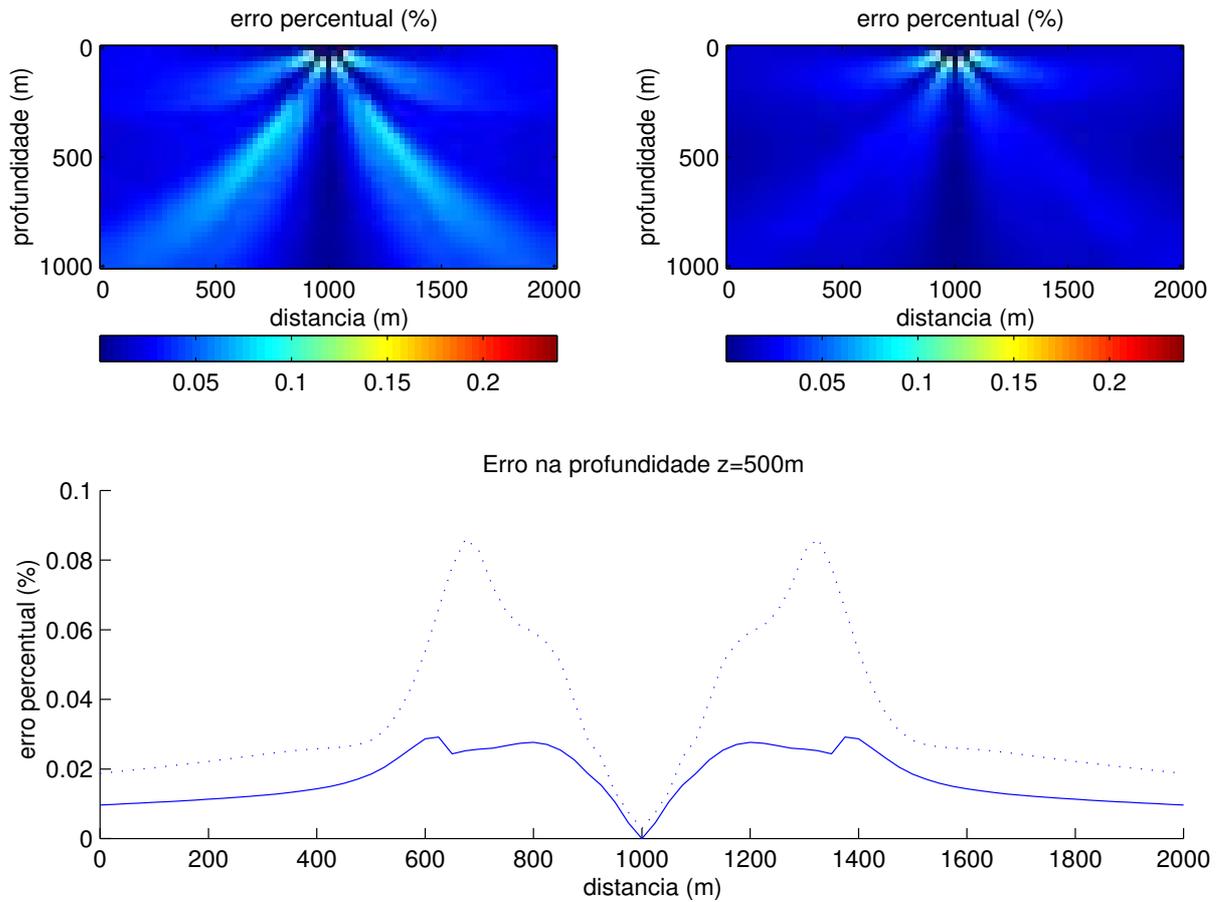


Figura 2.11: Na parte superior, temos o erro dos esquemas RK-ENO de ordem dois e três, respectivamente. Na parte inferior, temos o erro na profundidade de 500 m usando os esquemas RK-ENO de ordem dois (linha pontilhada) e três (linha sólida).

2.4.2 Expansão Dinâmica DNO

Conforme foi visto até o momento, a solução para o cálculo do tempo de trânsito marcha sequencialmente, calculando cada uma das três arestas do retângulo em expansão (expansão DNO), até acabar o domínio computacional. Mas dependendo do modelo de velocidades fornecido, os raios podem passar a curvarem-se para cima. Neste caso, mais passos na direção de z que nas direções horizontais são necessários, para termos um controle melhor do ângulo de propagação. Assim, a expansão DNO pode ser feita de forma dinâmica, onde a nova direção a ser calculada é encontrada checando o ângulo dos raios em cada aresta do retângulo em expansão (método descrito em Kim & Cook [9]).

Para fazer esta checagem, basta calcularmos as derivadas com respeito a z e a x nos dois nós que se encontram nos vértices inferiores (não os da superfície) do retângulo em expansão. Estas derivadas podem ser calculadas usando esquemas (atrasados) de primeira ordem. Se em ambos os nós, a derivada em z for menor que a em x (o que corresponde a dizer que o raio possui inclinação maior que $\pi/4$ em relação à vertical), então devemos andar na direção z e assim, expandir apenas a aresta horizontal do retângulo. Quando a derivada em z for maior que a em x em um desses dois nós (direito ou esquerdo), devemos andar na direção em que isso ocorrer e assim, calcular a respectiva nova aresta do retângulo. O processo se repete até acabar o domínio computacional.

A principal motivação da implementação deste algoritmo no método, é que assim menos sub-passos da condição CFL são realizados, ou seja, esta expansão dinâmica DNO tenta impedir que uma aresta do retângulo em expansão contenha raios com ângulos muito altos em relação à normal da respectiva aresta.

2.4.3 Varredura Posterior (PS)

Conforme comentado anteriormente, como nossa expansão DNO está sempre se afastando da fonte, pode ocorrer do cálculo de um tempo de trânsito necessitar da resolução em passos posteriores. Para corrigir isto, podemos tomar nossa aproximação, que foi obtida usando um esquema de DF, armazená-la em uma matriz e aplicar novamente o esquema, só que em outras direções.

Esta é a idéia básica do método PS (“*post sweeping*”) descrito em Kim & Cook [9]. Cada iteração deste consiste em aplicar um esquema de DF (RK-ENO de ordem dois ou três) em cada uma destas quatro diferentes direções: evolução em x para a direita,

evolução em x para a esquerda, evolução em z para baixo e evolução em z para cima. Tomemos, por exemplo, a primeira delas. Para inicializar a iteração, a primeira coluna da matriz armazenada da iteração anterior é assumida como condição inicial. Então, o esquema de DF é aplicado em toda a coluna, evoluindo para a direita, e o valor assumido como aproximação é simplesmente o menor entre este calculado (usando a coluna da esquerda) e o da matriz armazenada, da iteração anterior. Isto é feito até acabar o domínio computacional.

Após esta direção, as outras são aplicadas sucessivamente. Este ciclo de quatro direções se repete, até que alguma condição de convergência seja satisfeita, como por exemplo, que a norma da diferença entre duas iterações seja menor que uma tolerância ε .

A seguir, temos o exemplo de como o método PS corrige as superestimações no cálculo do tempo de trânsito. O modelo de velocidade (ver Figura 2.12) é constante com $v = 4500$ m/s, exceto num retângulo inserido nele, onde a velocidade é de $v = 2000$ m/s. A fonte está localizada em $x = 500$ m na superfície. Na Figura 2.13, a aproximação da esquerda relata tempos muito maiores na região logo abaixo do retângulo de baixa velocidade. Isto ocorre porque nesta região, vários sub-passos são requeridos para tentar satisfazer a condição CFL. O método, através de um custo computacional enorme, consegue usar informações que vinham, neste caso, da esquerda para a direita. No entanto, o método usa os tempos de trânsito baixos, originados pela onda lenta que se propagou na caixa de baixa velocidade inserida no meio.

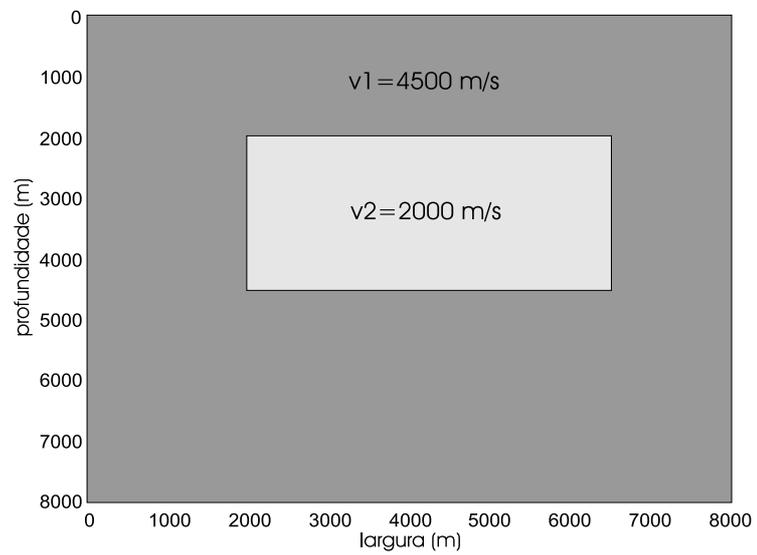


Figura 2.12: Modelo com um campo de velocidades descontínuo.

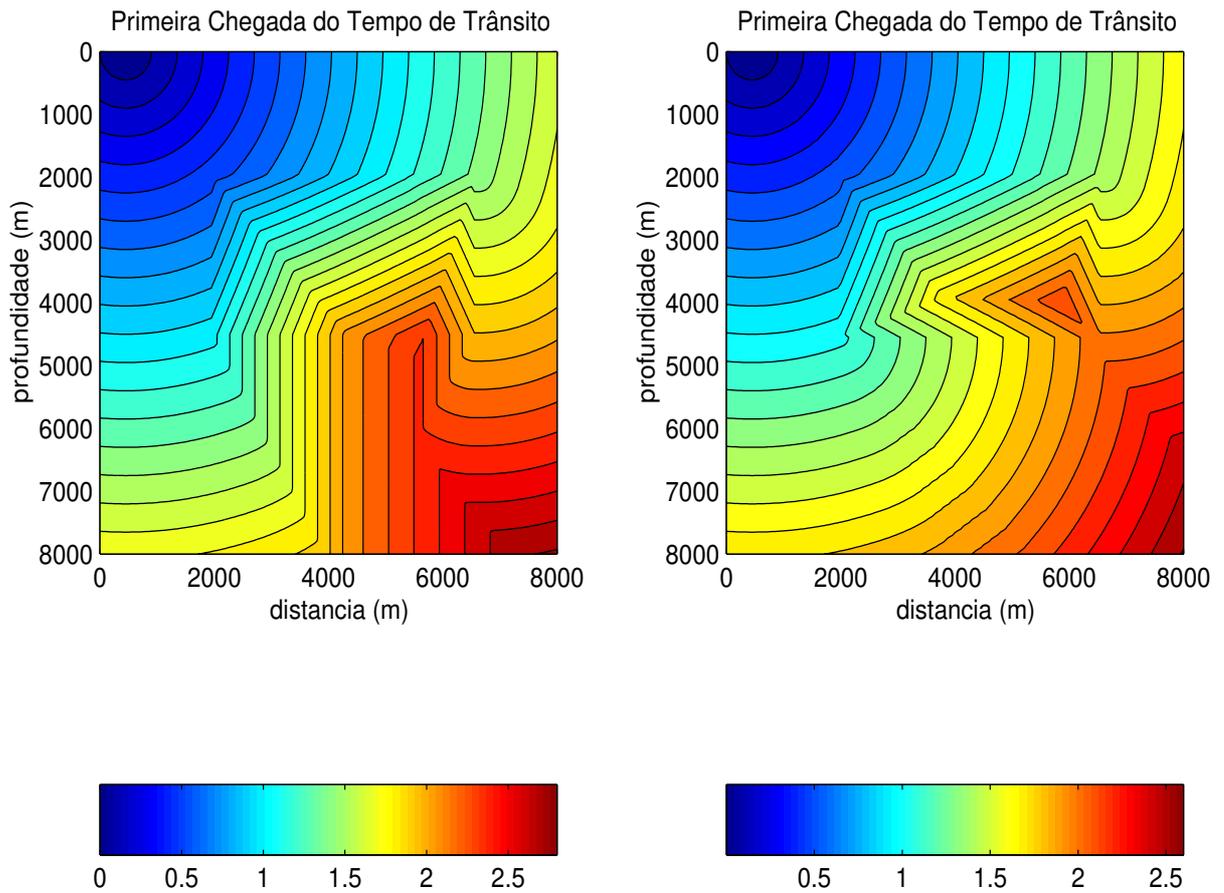


Figura 2.13: A esquerda, temos a aproximação encontrada usando o esquema RK-ENO de ordem dois com expansão dinâmica DNO. A direita, temos o resultado de uma iteração PS aplicada nesta aproximação.

Capítulo 3

Método de Diferenças Finitas Adaptativo

Os métodos de ordem superior da Seção 2.3 resolvem a equação iconal numericamente com bastante eficiência. No entanto, estes métodos não apresentam muita vantagem perto da fonte, onde a convergência ocorre de maneira bastante lenta. Mesmo usando inicializações refinadas, ficamos com parâmetros arbitrários, visto que os tamanhos dos espaçamentos na malha refinada e o tamanho dela própria são atribuídos pelo usuário, sem nenhum critério de seleção.

Uma maneira de corrigir este problema é usar uma malha adaptativa ao invés da fixa. A essência deste método é usar dois esquemas de ordens diferentes para o cálculo de um passo (descrito em Belfi & Symes [2]). Supondo que o esquema de ordem maior seja mais correto que o de menor ordem, o passo de ordem maior serve como solução exata naquele passo, e assim, conseguimos estimar o erro. Logo, a malha pode aumentar ou diminuir, afim de que uma tolerância para o erro seja permitida, logo, em locais onde o erro é grande, o método adaptativo permite que a malha se modifique a fim de melhorar a aproximação. Em nosso caso, temos uma equação diferencial parcial. Isto torna o cálculo mais complicado, pois é necessário ajustar a malha na variável que não está sendo evoluída, ao longo do passo em que ocorre a evolução.

O método de DF adaptativo aqui estudado não usa expansão DNO. A expansão e a inicialização são as mesmas que as apresentadas na Seção 2.3. Novamente, para tornar possível esta expansão (devido a condição CFL), devemos utilizar a forma dada em (2.30), lembrando que a condição CFL é $\Delta z < \Delta x / \tan \theta_{\max}$. A função \max é a mais simples

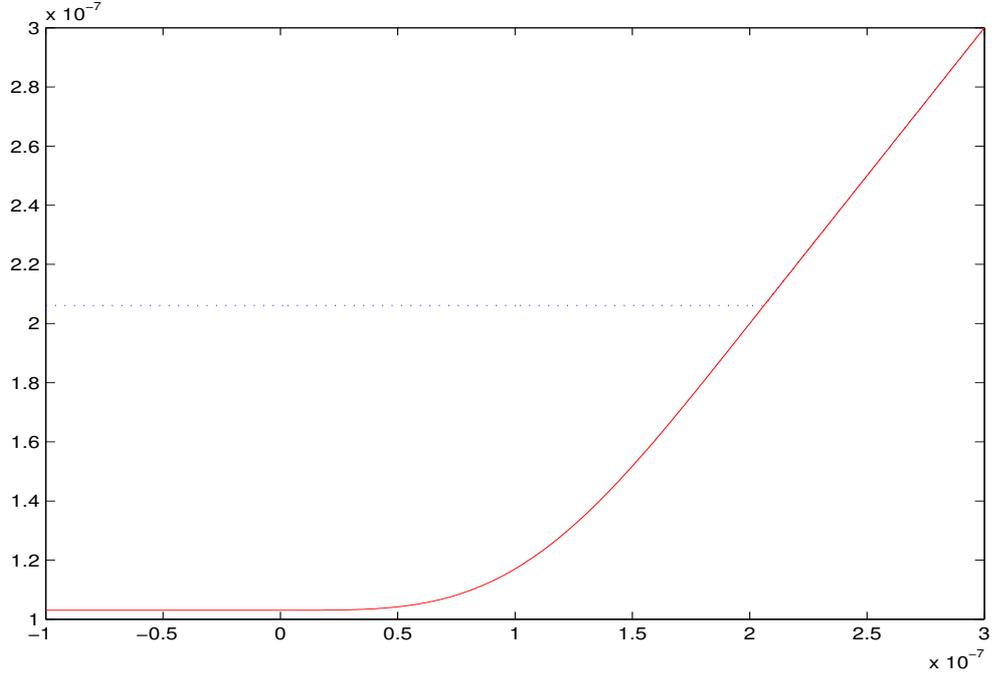


Figura 3.1: A curva sólida representa a função smmax , já a curva pontilhada representa a função max . Os argumentos usados são como os apresentados na equação (2.30).

para tornar possível esta restrição. No lugar desta, poderia estar qualquer outra função que desempenhasse o mesmo papel, mas que fosse diferenciável, por exemplo. Em Qian & Symes [12], no lugar da função max , foi usada uma função interpoladora dada por

$$\text{smmax}(a, b) = \begin{cases} \frac{1}{2}b, & \text{se } a < 0 \\ \frac{1}{2}b + 2\frac{a^4}{b^3} \left(1 - \frac{4a}{5b}\right), & \text{se } 0 \leq a < \frac{b}{2} \\ a + 2\frac{(a-b)^4}{b^3} \left(1 + \frac{4a-b}{5b}\right), & \text{se } \frac{b}{2} \leq a < b \\ a, & \text{se } a \geq b \end{cases}, \quad (3.1)$$

onde o ângulo máximo para este caso é um pouco maior que o ângulo θ_{max} . Na Figura 3.1, temos o gráfico desta função (curva sólida) e da função max (curva pontilhada), respectivamente, usando $\theta_{\text{max}} = 63^\circ$.

Para inicializar o algoritmo adaptativo, o usuário precisa fornecer o valor de uma tolerância ε para o erro e duas funções definidas positivas σ_1 e σ_2 de ε (com $\sigma_1(\varepsilon) < \sigma_2(\varepsilon)$)

para todo ε) para controlar o aumento e o refinamento dos espaçamentos. Os esquemas usados são os RK-ENO de ordem dois e três. Após calcular um novo passo usando ambos esquemas, podemos estimar o erro de truncamento calculando o máximo entre a diferença das duas aproximações para este passo, o qual denotamos por E . Caso $\sigma_1(\varepsilon) < E < \sigma_2(\varepsilon)$, damos continuidade e calculamos um novo passo. Quando $E < \sigma_1(\varepsilon)$, devemos aumentar os espaçamentos Δz e Δx . Similarmente, quando $E > \sigma_2(\varepsilon)$, devemos diminuir os espaçamentos Δz e Δx . A função σ_2 , pela própria construção do método, vale ε e a função σ_1 , que é usada para aumentar o espaçamento da malha, deve ter uma variação de pelo menos uma casa decimal em relação a σ_2 (Qian & Symes [12] usaram $\sigma_1(\varepsilon) = 0.1\varepsilon$). Um ponto muito importante, é que supomos como aproximação o cálculo feito usando o esquema de ordem três, ou seja, a aproximação calculada usando o esquema de ordem dois é descartada, sendo esta importante apenas para ter um controle no passo.

Estas mudanças nos espaçamentos devem ser feitas usando o mesmo fator em ambas as direções. Assim, a condição CFL para o problema continua sendo satisfeita. O fator usado em Qian & Symes [12] é dois, ou seja, os espaçamentos Δz e Δx são multiplicados ou divididos por dois. O ajuste usual para problemas envolvendo equações diferenciais ordinárias depende do erro de truncamento. Isto é impraticável em nosso caso, pois passaríamos a ter uma malha arbitrária em x , e assim seria necessária muita interpolação. Assim, este fator dois apresenta muita vantagem, pois mesmo quando for necessária a interpolação (quando devemos diminuir o espaçamento), apenas alguns nós são interpolados. Outro fato é que o tempo de trânsito tende a ficar cada vez mais suave conforme se afasta da fonte. Logo, a maioria dos ajustes na malha refere-se ao aumento dos espaçamentos, e sendo assim, pouca interpolação é necessária.

3.1 Inicialização no Método Adaptativo

Conforme vimos na Seção 2.4.1, podemos refinar a malha próxima a fonte de modo a diminuir o erro dos métodos de malha fixa. No caso do método adaptativo, este erro inicial depende da profundidade z_{ini} , onde iniciamos o algoritmo. Ou seja, ao aproximarmos a solução em z_{ini} considerando que o meio é homogêneo, o erro nesta aproximação deve ser menor que a tolerância ε fornecida ao algoritmo.

A seguir, apresentamos duas maneiras de fazer isto. Na primeira delas [2], usamos expansão de Taylor para obter uma condição sobre o valor de z_{ini} . Na segunda [12],

usamos as equações dos raios para impor estas condições. Por fim, apresentamos alguns testes computacionais afim de comparar estas duas inicializações.

3.1.1 Aproximação por Taylor

Vamos considerar coordenadas polares no plano $x = r \sin \theta$ e $z = r \cos \theta$, com $r \geq 0$, $-\pi/2 \leq \theta \leq \pi/2$ e denotar $\mathcal{T}(r, \theta) = \tau(x, z)$. Ao aproximarmos a solução próxima a fonte considerando que o meio é homogêneo, estamos usando a estimativa (fonte em $r = 0$)

$$\mathcal{T}(r, \theta) \simeq \frac{r}{v(0, 0)}. \quad (3.2)$$

A solução exata tem expansão de Taylor em torno de $r = 0$

$$\mathcal{T}(r, \theta) = r \frac{\partial \mathcal{T}}{\partial r}(0, \theta) + \frac{r^2}{2} \frac{\partial^2 \mathcal{T}}{\partial r^2}(0, \theta) + \mathcal{O}(r^3). \quad (3.3)$$

Precisamos investigar a equação iconal em coordenadas polares para estimar o valor da segunda derivada de \mathcal{T} em relação a r . Logo, escrevendo

$$\frac{\partial \mathcal{T}}{\partial r} = \frac{\partial \tau}{\partial x} \sin \theta + \frac{\partial \tau}{\partial z} \cos \theta, \quad (3.4)$$

$$\frac{\partial \mathcal{T}}{\partial \theta} = \frac{\partial \tau}{\partial x} r \cos \theta - \frac{\partial \tau}{\partial z} r \sin \theta, \quad (3.5)$$

obtemos a equação iconal em coordenadas polares

$$\left(\frac{\partial \tau}{\partial x} \right)^2 + \left(\frac{\partial \tau}{\partial z} \right)^2 = \left(\frac{\partial \mathcal{T}}{\partial r} \right)^2 + \frac{1}{r^2} \left(\frac{\partial \mathcal{T}}{\partial \theta} \right)^2 = s^2, \quad (3.6)$$

onde $s(r, \theta) = 1/v(x, z)$. Derivando esta equação com relação a r obtemos

$$2 \frac{\partial \mathcal{T}}{\partial r} \frac{\partial^2 \mathcal{T}}{\partial r^2} - \frac{2}{r^3} \left(\frac{\partial \mathcal{T}}{\partial \theta} \right) + \frac{2}{r^2} \frac{\partial \mathcal{T}}{\partial \theta} \frac{\partial^2 \mathcal{T}}{\partial \theta \partial r} = 2s \frac{\partial s}{\partial r}. \quad (3.7)$$

Supondo as derivadas primeiras de \mathcal{T} em $r = 0$ iguais às respectivas derivadas no caso homogêneo,

$$\frac{\partial \mathcal{T}}{\partial r}(0, \theta) = \frac{1}{v(0, 0)}, \quad \frac{\partial \mathcal{T}}{\partial \theta}(0, \theta) = 0, \quad (3.8)$$

podemos escrever da equação (3.7)

$$\frac{\partial^2 \mathcal{T}}{\partial r^2}(0, \theta) = \frac{\partial s}{\partial r}(0, \theta). \quad (3.9)$$

Assim, da diferença entre as equações(3.2) e (3.3), podemos escrever o erro cometido na inicialização na forma

$$E_{\text{ini}} \simeq \frac{r^2}{2} \left| \frac{\partial s}{\partial r}(0, \theta) \right|. \quad (3.10)$$

Como temos v em coordenadas cartesianas, vamos escrever a derivada de s neste sistema. Para isso, observemos que

$$s(x, z) = s(0, 0) + x \frac{\partial s}{\partial x}(\xi_1, \eta_1) + z \frac{\partial s}{\partial z}(\xi_2, \eta_2), \quad (3.11)$$

onde (ξ_1, η_1) e (ξ_2, η_2) pertencem ao domínio $D = \{(\xi, \eta) \mid |\xi| \leq |x|, 0 \leq \eta \leq z\}$. Mudando para coordenadas polares obtemos

$$s(r, \theta) = s(0, 0) + r \sin \theta \frac{\partial s}{\partial x}(\xi_1, \eta_1) + r \cos \theta \frac{\partial s}{\partial z}(\xi_2, \eta_2), \quad (3.12)$$

de onde,

$$\frac{\partial s}{\partial r}(0, \theta) = \sin \theta \frac{\partial s}{\partial x}(\xi_1, \eta_1) + \cos \theta \frac{\partial s}{\partial z}(\xi_2, \eta_2). \quad (3.13)$$

Olhando esta equação como um produto escalar e utilizando a desigualdade de Cauchy-Schwartz, obtemos

$$\begin{aligned} \left| \frac{\partial s}{\partial r}(0, \theta) \right| &\leq \sup\{\|\nabla s(\xi, \eta)\| \mid |\xi| \leq |x|, 0 \leq \eta \leq z\} \\ &\leq \sup\{\|\nabla s(\xi, \eta)\| \mid |\xi| \leq z_L \tan \theta_{\max}, 0 \leq \eta \leq z_L\} = S, \end{aligned} \quad (3.14)$$

onde usamos os fatos de estarmos calculando a aproximação apenas na abertura $x \leq z \tan \theta_{\max}$, e estarmos impondo um limite z_L para a profundidade de inicialização. Enfim, limitamos o erro por

$$\begin{aligned} E_{\text{ini}} &\simeq \frac{r^2}{2} \left| \frac{\partial s}{\partial r}(0, \theta) \right| \\ &\leq \frac{1}{2} \sqrt{x^2 + z^2} S \end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{2}(1 + \tan^2 \theta_{\max})z^2 S \\
&= \frac{1}{2} \frac{z^2 S}{\cos^2 \theta_{\max}} .
\end{aligned} \tag{3.15}$$

Por outro lado, queremos que E_{ini} seja limitado pela tolerância fornecida. Assim, obtemos a profundidade inicial dada pela restrição

$$z_{\text{ini}} \leq z_{\text{T}} = \sqrt{\frac{2\varepsilon}{S}} \cos \theta_{\max} , \tag{3.16}$$

e usamos $z_{\text{ini}} = \min(z_{\text{T}}, z_{\text{L}})$.

3.1.2 Aproximação por Raios

Supondo novamente que a fonte está na origem, vamos agora escrever as equações dos raios (Seção 1) no caso bidimensional. Escolhendo $\lambda = v^2$, pois assim σ é o próprio tempo t sobre um raio, temos

$$\frac{dx}{dt} = v^2 p_1 , \tag{3.17}$$

$$\frac{dz}{dt} = v^2 p_2 , \tag{3.18}$$

$$\frac{dp_1}{dt} = -\frac{1}{v} \frac{\partial v}{\partial x} , \tag{3.19}$$

$$\frac{dp_2}{dt} = -\frac{1}{v} \frac{\partial v}{\partial z} , \tag{3.20}$$

onde $p_1 = \partial\tau/\partial x$ e $p_2 = \partial\tau/\partial z$. Sejam θ e ψ os ângulos feitos com a vertical dos raios exato e aproximado (considerando o meio sendo homogêneo), respectivamente, em um determinado ponto. Vamos encontrar relações entre esses raios de modo a estimar o erro cometido na inicialização. Lembrando que

$$p_1 = \frac{\sin \theta}{v} \quad \text{e} \quad p_2 = \frac{\cos \theta}{v} , \tag{3.21}$$

e assim,

$$\frac{dx}{dt} = v \sin \theta \quad \text{e} \quad \frac{dz}{dt} = v \cos \theta . \tag{3.22}$$

Introduzindo coordenadas polares

$$x = r \sin \psi \quad \text{e} \quad z = r \cos \psi , \quad (3.23)$$

e diferenciando ambas com relação a t , obtemos o sistema

$$\begin{aligned} \frac{dx}{dt} &= \frac{dr}{dt} \sin \psi + r \cos \psi \frac{d\psi}{dt} \\ \frac{dz}{dt} &= \frac{dr}{dt} \cos \psi - r \sin \psi \frac{d\psi}{dt} . \end{aligned} \quad (3.24)$$

Usando as equações em (3.22) e (3.24), temos a solução

$$\frac{dr}{dt} = v \cos(\theta - \psi) , \quad (3.25)$$

$$\frac{d\psi}{dt} = \frac{v}{r} \sin(\theta - \psi) . \quad (3.26)$$

Agora podemos ter uma aproximação para o erro no tempo de trânsito. Vamos denotar por τ_0 a aproximação para o tempo verdadeiro τ quando usamos a velocidade constante v_0 igual à da origem. Como $\tau_0 = v_0^{-1}r$ e usando o resultado em (3.25), podemos escrever que

$$\frac{d\tau_0}{dt} = \frac{1}{v_0} \frac{dr}{dt} = \frac{v}{v_0} \cos(\theta - \psi) , \quad (3.27)$$

de onde,

$$\frac{d\tau_0}{dt} - \frac{d\tau}{dt} = \left(\frac{v}{v_0} - 1 \right) \cos(\theta - \psi) + \cos(\theta - \psi) - 1 , \quad (3.28)$$

e, assim,

$$\left| \frac{d\tau_0}{dt} - \frac{d\tau}{dt} \right| \leq \left| \frac{v}{v_0} - 1 \right| + |\cos(\theta - \psi) - 1| . \quad (3.29)$$

Queremos limitar a diferença entre os tempos τ_0 e τ , mas o que temos é uma relação entre a diferença de suas derivadas. Então, observando que

$$\left| \frac{d\tau_0}{dt} - \frac{d\tau}{dt} \right| \leq \varepsilon \Rightarrow |\tau_0 - \tau| \leq \varepsilon t , \quad (3.30)$$

vamos impor as restrições

$$\left| \frac{v}{v_0} - 1 \right| \leq \frac{\varepsilon}{2} \quad (3.31)$$

e

$$|\cos(\theta - \psi) - 1| \leq \frac{\varepsilon}{2}. \quad (3.32)$$

Agora precisamos encontrar relações que envolvam z nestas duas restrições. No caso da restrição em (3.31), o trabalho é mais fácil. Como anteriormente, observando que

$$v(x, z) = v(0, 0) + x \frac{\partial v}{\partial x}(\xi_1, \eta_1) + z \frac{\partial v}{\partial z}(\xi_2, \eta_2), \quad (3.33)$$

onde (ξ_1, η_1) e (ξ_2, η_2) pertencem ao domínio $D = \{(\xi, \eta) \mid |\xi| \leq |x|, 0 \leq \eta \leq z\}$, e usando a desigualdade de Cauchy-Schwartz, obtemos

$$\begin{aligned} \left| \frac{v - v_0}{v_0} \right| &\leq \frac{r}{v_0} \sup\{\|\nabla v(\xi, \eta)\| \mid |\xi| \leq |x|, 0 \leq \eta \leq z\} \\ &\leq \frac{z}{v_0 \cos \theta_{\max}} \sup\{\|\nabla v(\xi, \eta)\| \mid |\xi| \leq z_L \tan \theta_{\max}, 0 \leq \eta \leq z_L\} \\ &= \frac{zV}{v_0 \cos \theta_{\max}}, \end{aligned} \quad (3.34)$$

com $V = \sup\{\|\nabla v(\xi, \eta)\| \mid |\xi| \leq z_L \tan \theta_{\max}, 0 \leq \eta \leq z_L\}$. Logo, para valer a restrição (3.31), devemos escolher z_{ini} de modo que

$$z_{\text{ini}} \leq z_1 = \frac{v_0 \varepsilon \cos \theta_{\max}}{2V}. \quad (3.35)$$

Vamos trabalhar agora com a restrição (3.32). Para isso, precisamos de alguma relação envolvendo $\theta - \psi$. Vamos diferenciar a equação envolvendo p_1 em (3.21) com relação a t . Fazendo isto temos

$$\frac{dp_1}{dt} = \left(v \frac{d\theta}{dt} \cos \theta - \sin \theta \left[\frac{\partial v}{\partial x} \frac{dx}{dt} + \frac{\partial v}{\partial z} \frac{dz}{dt} \right] \right) v^{-2}. \quad (3.36)$$

Usando os resultados em (3.22) e simplificando, obtemos

$$\frac{d\theta}{dt} = -\cos \theta \frac{\partial v}{\partial x} + \sin \theta \frac{\partial v}{\partial z}. \quad (3.37)$$

Usando os resultados obtidos em (3.26) e (3.37), podemos escrever

$$\frac{d\theta}{dt} - \frac{d\psi}{dt} = a(t) - \frac{b(t)}{t}(\theta - \psi), \quad (3.38)$$

onde

$$a(t) = \frac{d\theta}{dt} \quad \text{e} \quad b(t) = \frac{vt \sin(\theta - \psi)}{r \theta - \psi}. \quad (3.39)$$

A equação (3.38) é resolvida usando o fator integrande. Logo,

$$\theta - \psi = \int_0^t a(\tau) \exp\left(-\int_\tau^t \frac{b(\sigma)}{\sigma} d\sigma\right) d\tau. \quad (3.40)$$

Este resultado está na forma implícita, visto que $b(t)$ contém a solução. No entanto, como $b(t) \geq 0$, limitamos a exponencial no integrando por 1. A função a é limitada por a_{\max} , o qual é igual ao supremo da norma do gradiente da velocidade v (ver equação (3.37)). Obtemos então a estimativa

$$|\theta - \psi| \leq a_{\max} t. \quad (3.41)$$

Antes de desenvolver a restrição (3.32), usando a desigualdade (3.41), vamos por fim escrever uma relação entre t e r . O tempo de trânsito é determinado integrando v^{-1} ao longo de um caminho s que o torna mínimo. Seja l um segmento de reta usado para aproximar s . Então, pelo princípio de Fermat temos

$$t = \int_s \frac{1}{v} d\sigma \leq \int_l \frac{1}{v} d\sigma \leq \int_l \frac{1}{v_{\min}} d\sigma = \frac{r}{v_{\min}}, \quad (3.42)$$

onde $r = \sqrt{x^2(t) + z^2(t)}$ e v_{\min} é a velocidade mínima sobre o segmento do raio. Coletando estes resultados temos que

$$\begin{aligned} |\cos(\theta - \psi) - 1| &= \left| -2 \sin^2 \frac{(\theta - \psi)}{2} \right| \\ &\leq \left| \frac{(\theta - \psi)^2}{2} \right| \\ &\leq \left| \frac{a_{\max}^2 r^2}{2v_{\min}^2} \right| \\ &\leq \left| \frac{V^2 z^2}{2v_{\min}^2 \cos^2 \theta_{\max}} \right|, \end{aligned} \quad (3.43)$$

onde usamos que $a_{\max} \leq V$ dentro da abertura. Enfim, para satisfazer a restrição (3.32),

devemos escolher z_{ini} de modo que

$$z_{\text{ini}} \leq z_2 = \frac{\sqrt{\varepsilon} v_{\text{min}} \cos \theta_{\text{max}}}{V}. \quad (3.44)$$

Sendo $z_{\text{R}} = \min(z_1, z_2)$ a escolha usando este método por raios, basta escolher $z_{\text{ini}} = \min(z_{\text{R}}, z_{\text{L}})$ para que a tolerância em (3.30) seja satisfeita.

3.1.3 Testes Computacionais

Testamos os dois tipos de inicialização desta seção em diferentes meios. Em todos os casos, o domínio usado foi $-1000 \text{ m} \leq x \leq 1000 \text{ m}$ e $0 \leq z \leq 1500 \text{ m}$, com $\theta_{\text{max}} = 63^\circ$ e a fonte localizada na origem. O espaçamento da malha é $\Delta x = 2000/2^7 \simeq 15.6 \text{ m}$, com Δz obtido pela condição CFL. Usamos como tolerância $\varepsilon = 10^{-4}$ e $z_{\text{L}} = 450 \text{ m}$. Em todos os gráficos desta seção, analisamos o erro absoluto apenas na região de abertura limitada por θ_{max} .

No caso do meio homogêneo temos que os valores de S e V são nulos para qualquer z_{L} , e assim $z_{\text{ini}} = z_{\text{L}}$. Para o meio com velocidade afim em z , conforme a Seção 3.3.2, obtivemos $z_{\text{T}} = 11.75 \text{ m}$ e $z_{\text{R}} = 0.075 \text{ m}$. Na Figura 3.2, temos o gráfico dos erros cometidos ao inicializarmos nestas profundidades. Outro caso testado é o meio com velocidade exponencial em z , conforme a Seção 3.3.6, onde obtivemos $z_{\text{T}} = 11.13 \text{ m}$ e $z_{\text{R}} = 0.057 \text{ m}$. Os gráficos dos erros cometidos estão na Figura 3.3. Já no caso do meio com velocidade decaindo em z , conforme a Seção 3.3.7, obtivemos $z_{\text{T}} = 9.1 \text{ m}$ e $z_{\text{R}} = 0.046 \text{ m}$. Os gráficos para este caso estão na Figura 3.4. O erro muito menor representado pela curva pontilhada nestas figuras é devido ao fato dele estar limitado por εt . Assim, o passo inicial z_{R} obtido usando a inicialização da Seção 3.1.2 subestima este, o que implica em maior tempo computacional para o método adaptativo. Como o passo inicial z_{T} produziu uma aproximação que respeitou a tolerância ε e, além disso, tem um valor muito maior que z_{R} , vamos adotá-lo como nosso z_{ini} nos experimentos numéricos do método adaptativo da Seção 3.3.

Em meios com velocidade variando continuamente, como os casos aqui testados, a escolha de z_{L} praticamente não faz diferença. Já no caso de meios homogêneos separados por uma interface, a escolha de z_{L} é muito importante. Para z_{L} acima do refletor, temos que $z_{\text{ini}} = z_{\text{L}}$. Mas para z_{L} abaixo do refletor, devido a variação brusca na velocidade, temos z_{ini} muito próximo da fonte, o que implica em um custo computacional que poderia

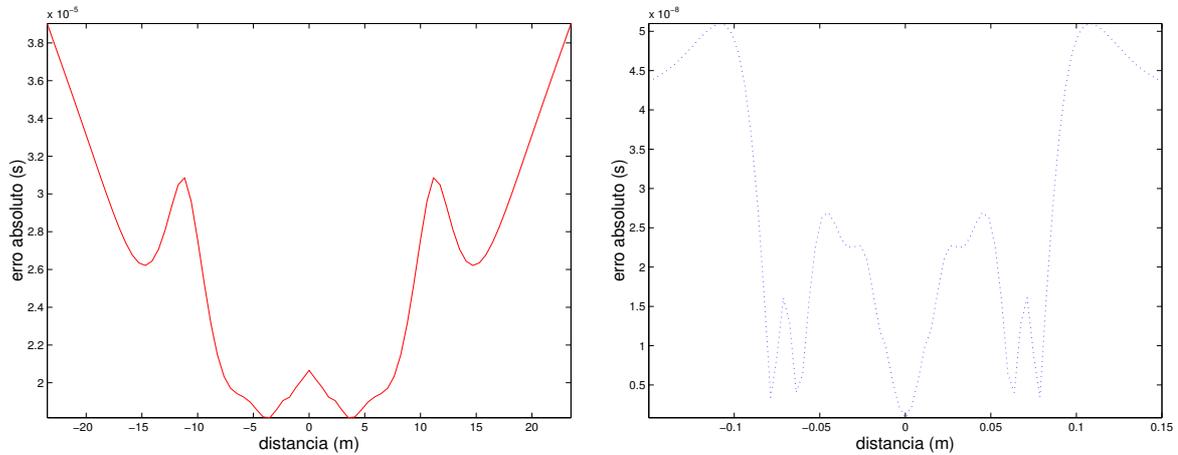


Figura 3.2: Erros cometidos nas profundidades de 11.75 m (curva sólida) e 0.075 m (curva pontilhada), obtidas usando as inicializações por Taylor e raios, respectivamente. Meio com velocidade afim em z .

ser evitado.

3.2 Interpolação no Refinamento

Conforme vimos no início desta seção, deve ocorrer uma interpolação quando o erro for maior que a tolerância fornecida. Em Qian & Symes [12], e nos outros textos que o antecedem, não foi apresentada a interpolação usada nem mesmo discutida de que maneira ela deve ser feita. Até este momento uma de nossas maiores preocupações foi apresentar um método que respeite a propagação da onda. Logo, a interpolação também deve ter esta propriedade, para que erros não sejam produzidos quando ocorrer o refinamento.

Em nosso algoritmo, usamos as splines cúbicas interpolantes. O que motivou esta escolha foram os fatos de a ordem deste método ser compatível com nosso algoritmo e o fato de interpolarmos os dados por partes. Este último fato é um diferencial em nosso algoritmo. Dada uma profundidade, se existir um máximo local no tempo de trânsito, não podemos simplesmente interpolar os dados, pois os novos pontos vizinhos ao máximo local devem ser calculados de maneira que apenas informações que influenciam fisicamente este ponto sejam usados. Logo, basta interpolarmos os dados de um máximo local ao outro. Para vermos isso, na Figura 3.5 temos um tempo de trânsito fictício em uma profundidade qualquer. Este comportamento ocorre quando duas frentes de onda se chocam, isto é, dois

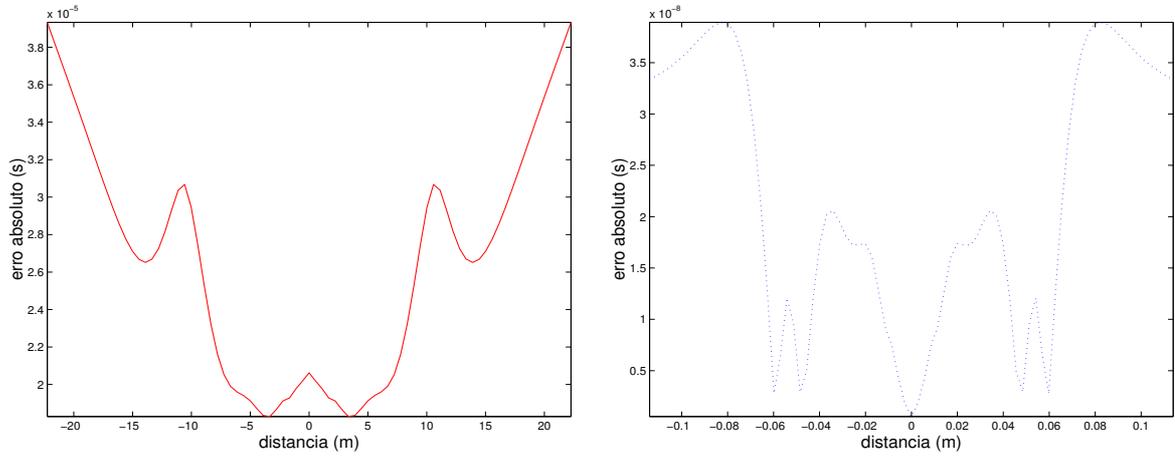


Figura 3.3: Erros cometidos nas profundidades de 11.13 m (curva sólida) e 0.057 m (curva pontilhada), obtidas usando as inicializações por Taylor e raios, respectivamente. Meio com velocidade exponencial em z .

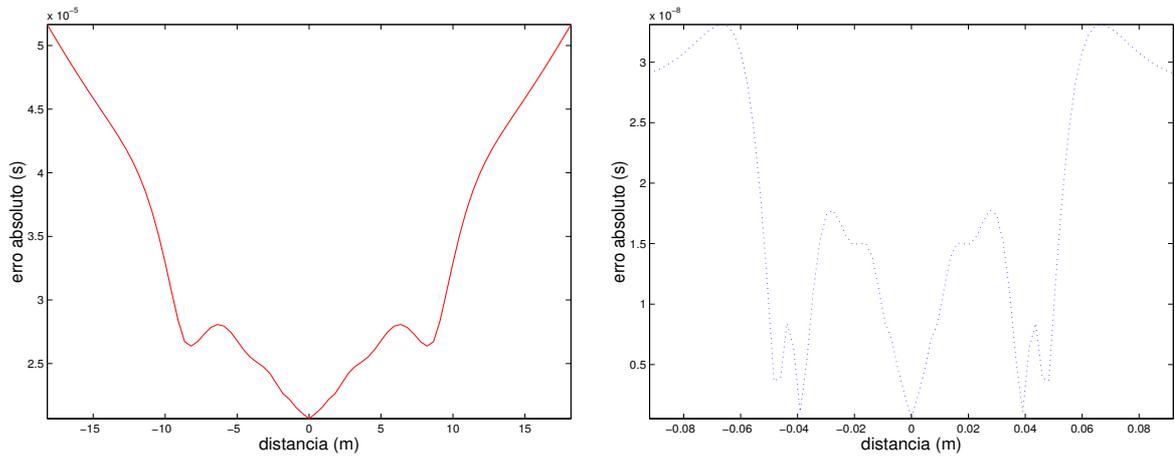


Figura 3.4: Erros cometidos nas profundidades de 9.1 m (curva sólida) e 0.046 m (curva pontilhada), obtidas usando as inicializações por Taylor e raios, respectivamente. Meio com velocidade decaindo em z .

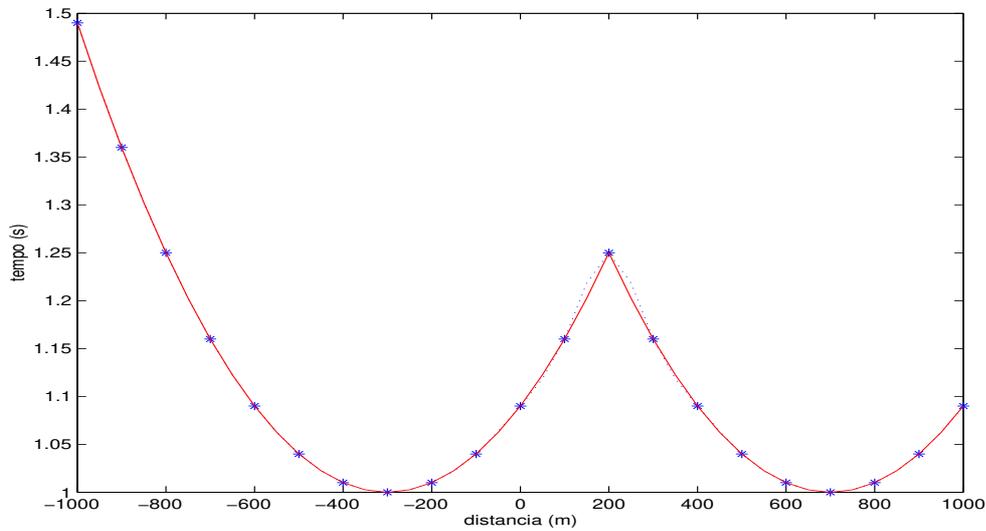


Figura 3.5: A curvas pontilhada e sólida representam os resultados da interpolação em todos os pontos e por partes, respectivamente.

raios chegam ao mesmo tempo em um dado ponto por dois caminhos diferentes. Os pontos representados pelo asterisco são os locais onde o tempo de trânsito está dado. Após um refinamento, a curva pontilhada representa o resultado de uma interpolação envolvendo todos os pontos, já a curva sólida representa o resultado de uma interpolação por partes. Comparando ambas as curvas, fica claro que a curva pontilhada superestimou o tempo de trânsito, pois o polinômio interpolador tentou suavizar uma região que não deve ser suavizada. Assim, ao interpolarmos separadamente nas duas regiões entre os máximos locais, o resultado produzido fica de acordo com toda a teoria até aqui discutida.

Para ilustrar a melhora deste tipo de interpolação, aplicamos o método adaptativo no modelo apresentado na Figura 3.6, usando os dois tipos de interpolação. Para visualizar a geometria das frentes de onda, temos na parte inferior da mesma figura, o tempo de trânsito encontrado usando o método de malha fixa RK-ENO de ordem 3 expandindo apenas em z . O domínio usado foi $-1000 \text{ m} \leq x \leq 1000 \text{ m}$ e $0 \leq z \leq 2000 \text{ m}$, com $\theta_{\max} = 63^\circ$ e a fonte localizada na origem. Iniciamos com o espaçamento $\Delta x = 2000/2^5 = 62.5 \text{ m}$ e usamos como tolerância $\varepsilon = 10^{-3}$, com limitantes para o erro $\sigma_1 = 0.1\varepsilon$ e $\sigma_2 = \varepsilon$. O passo inicial foi 450 m. Assumimos como solução o método RK-ENO de ordem dois com expansão dinâmica DNO em uma malha onde o espaçamento vale um oitavo do acima fornecido. Na Figura 3.7, as curvas pontilhada e sólida representam os erros na profundidade de 2000 m usando o método adaptativo sem e com a interpolação por

partes, respectivamente. O erro percentual sofreu uma diminuição de aproximadamente 30% no ponto onde ocorre o choque das ondas. Assim, como esperado teoricamente, a interpolação feita por partes é um fator muito importante na implementação do método adaptativo.

3.3 Experimentos Numéricos

Vamos analisar o comportamento do algoritmo do método de DF adaptativo e compará-lo com o método de malha fixa RK-ENO de ordem três que expande apenas na direção z , o qual nos referimos como *método de malha fixa* nesta seção. A intenção destes testes computacionais é ver quanto tempo é gasto computacionalmente para que os dois métodos obtenham uma aproximação com o erro da mesma ordem. Estes testes foram realizados utilizando o software MATLAB. Em cada seção é apresentado um modelo de velocidades diferente. Para visualizar a forma das frentes de onda usamos o método de malha fixa, visto que o método adaptativo não produz uma aproximação em uma malha regular e assim, teríamos que interpolar as aproximações de cada iteração para poder gerar a respectiva imagem.

Para o cálculo do erro percentual, usamos como solução a aproximação encontrada pelo algoritmo RK-ENO de ordem dois da Seção 2.4, implementado com a inicialização refinada e a expansão dinâmica DNO, usando um oitavo do espaçamento utilizado no experimento a seguir. Podemos fazer isto com total confiança, pois o algoritmo comporta-se muito bem em regiões de alto contraste, visto que ele executa sub-passos de tamanho Δz_{CFL} , e apresenta convergência para meios com solução conhecida, conforme vimos nos gráficos dessa mesma seção. No cálculo da aproximação na profundidade final de 1500 m, apresentamos também o tempo computacional gasto e o comportamento do método adaptativo, isto é, como variam os espaçamentos Δz e Δx ao longo das iterações. Para isso, basta observarmos os gráficos de Δz no decorrer das iterações, uma vez que Δx diminui sempre na mesma proporção.

Para todos os modelos, as aproximações foram obtidas no domínio $-1000 \text{ m} \leq x \leq 1000 \text{ m}$ e $0 \leq z \leq 1500 \text{ m}$, com $\theta_{\text{max}} = 63^\circ$ e a fonte localizada na origem. No método de malha fixa, o espaçamento foi de $\Delta x = 2000/2^7 \simeq 15.6 \text{ m}$, com Δz obtido pela condição CFL (Seção 2.3.3). No método adaptativo, iniciamos com o mesmo espaçamento e usamos como tolerância $\varepsilon = 10^{-4}$, com limitantes para o erro $\sigma_1 = 0.1\varepsilon$ e $\sigma_2 = \varepsilon$. O passo inicial

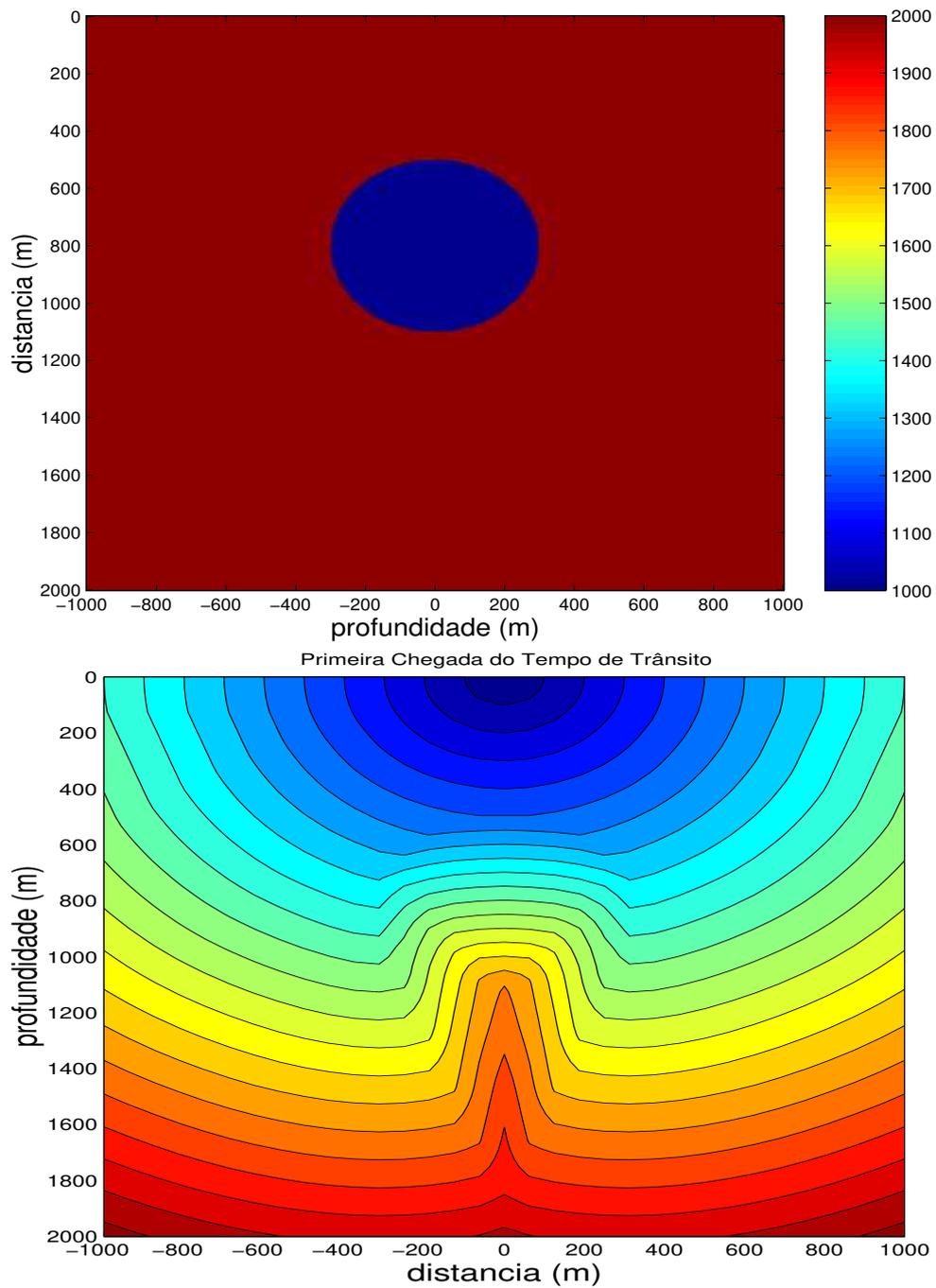


Figura 3.6: Na parte superior, temos um meio homogêneo contendo região com baixa velocidade. Na parte inferior, o respectivo tempo de trânsito encontrado usando o método de malha fixa expandindo apenas em z .

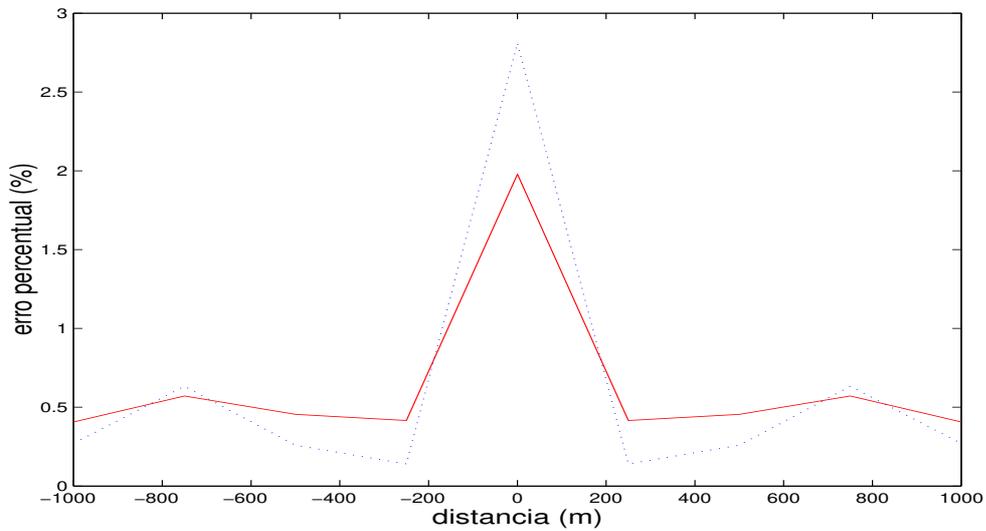


Figura 3.7: As curvas pontilhada e sólida representam o erro percentual usando o método adaptativo sem e com a interpolação por partes, respectivamente.

z_{ini} foi obtido pelas condições da Seção 3.1.1, onde usamos como limitante $z_L = 450$ m.

3.3.1 Meio Homogêneo ($v(x, z) = v_0$)

No caso de um meio homogêneo, temos que a solução da equação iconal são círculos centrados na origem (considerando esta como sendo a posição da fonte), ou seja,

$$\tau(x, z) = \frac{\sqrt{x^2 + z^2}}{v_0}. \quad (3.45)$$

Nos nossos testes tomamos $v_0 = 1000$ m/s. A Figura 3.8 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.9 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 2.7s para os método de malha fixa e adaptativo, respectivamente. Como o valor de z_{ini} neste caso foi o próprio z_L , o algoritmo começou os cálculos em uma profundidade distante da fonte e assim, bastante tempo computacional foi economizado. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. Estes picos que aparecem neste exemplo e também em alguns

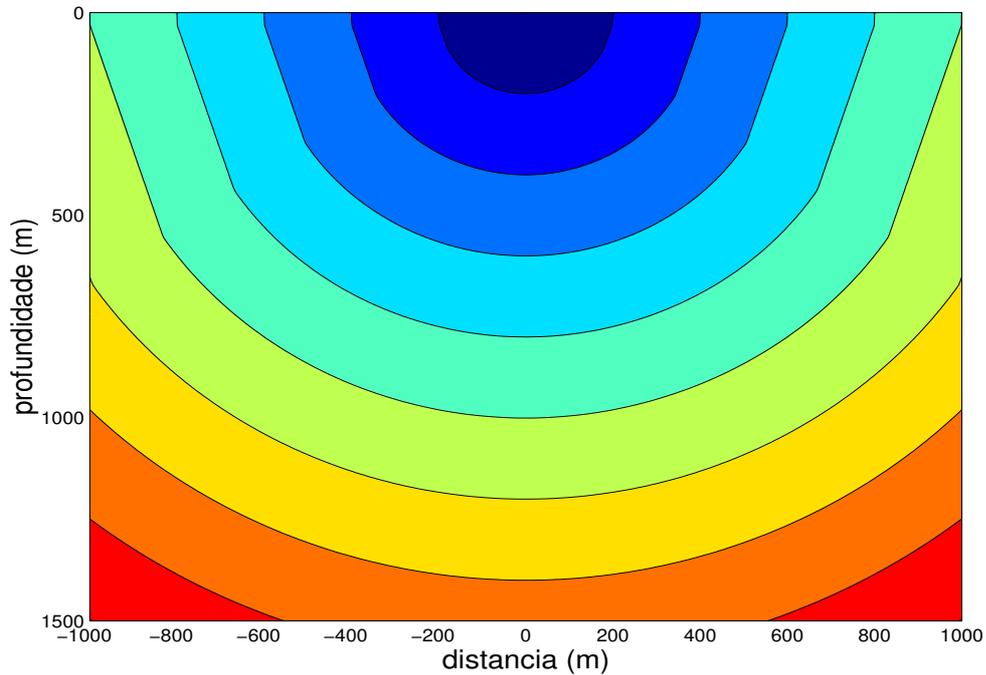


Figura 3.8: Aproximação encontrada usando o método de malha fixa expandindo apenas em z .

outros ocorrem porque o método encontra que o erro está muito baixo e então ele aumenta o espaçamento. Mas neste novo espaçamento o método encontra que o erro está alto, ou seja, na verdade ele não deveria ter aumentado o espaçamento, e assim, ocorre um novo refinamento. Uma medida para tentar evitar isso é aumentar a diferença entre os limitantes para o erro, para por exemplo $\sigma_1 = 0.05\varepsilon$ e $\sigma_2 = \varepsilon$. Isto funciona para este exemplo, mas tornaria o método ainda mais lento para os casos com interface.

3.3.2 Meio com Velocidade Afim em z ($v(x, z) = v_0 + gz$)

Neste caso, o modelo é tal que a velocidade é uma função afim da profundidade (veja Figura 3.10), conforme a equação

$$v(x, z) = 0.3z + 1000 . \quad (3.46)$$

A Figura 3.10 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.11 temos a comparação entre os erros percentuais

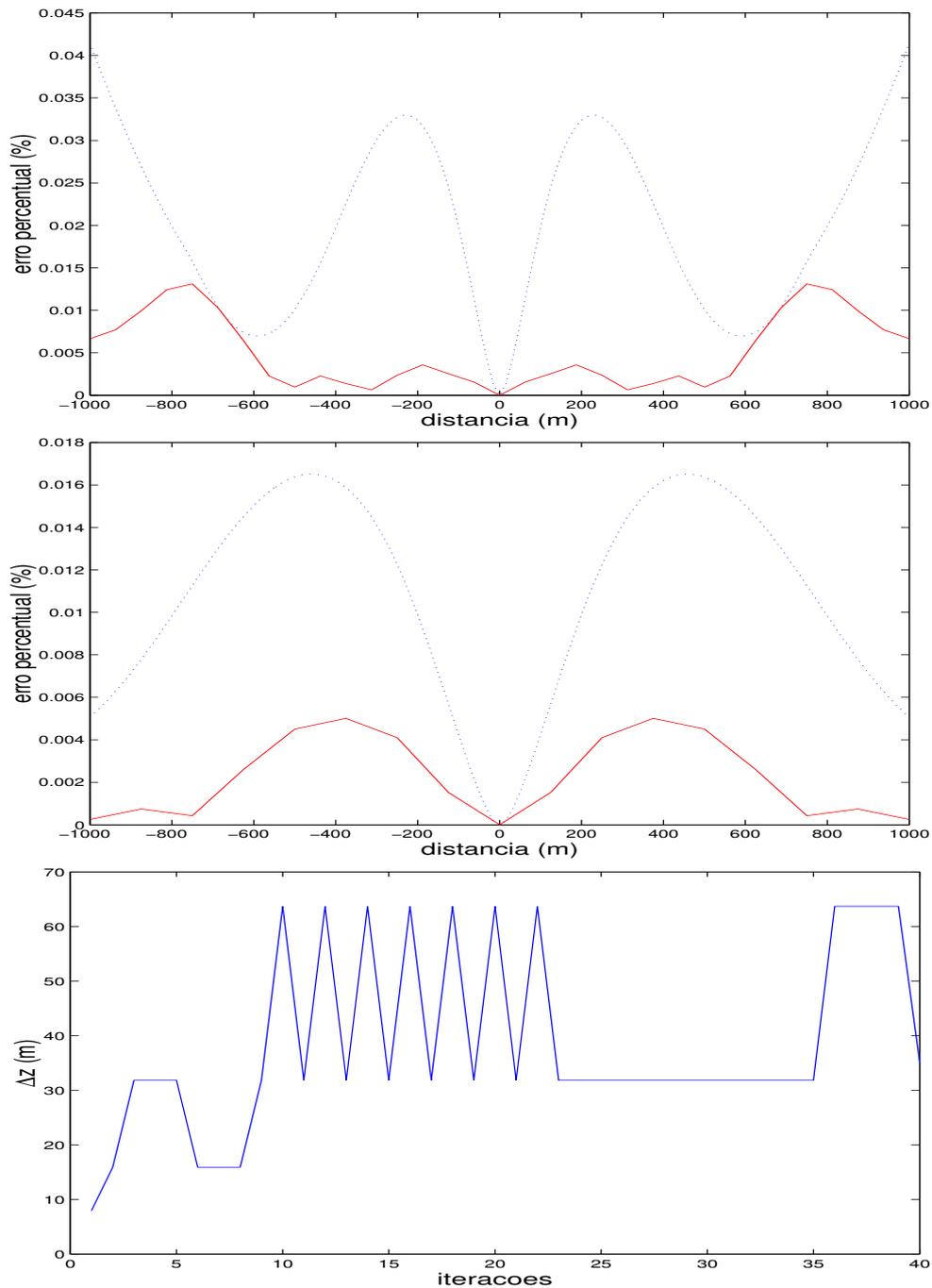


Figura 3.9: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Na profundidade de 750 m, ocorre um erro grande na região onde os raios fazem um ângulo maior que θ_{\max} com a vertical, pois as ondas são substituídas por outras artificiais. Este efeito está bastante visível no gráfico do tempo de trânsito da Figura 3.10. Na profundidade de 1500 m, já não há este problema, pois nesta região não há raios com tanta inclinação. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 17.8s para os métodos de malha fixa e adaptativo, respectivamente. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. Ocorreu um refinamento no início das iterações, mas conforme o processo se afastou da fonte, os espaçamentos aumentaram em algumas iterações, visto que a curvatura dos raios foi diminuindo. A tendência é que, ao sair desta região onde há raios com inclinação maior que θ_{\max} , ocorram apenas aumentos nos tamanhos dos espaçamentos.

3.3.3 Meios Homogêneos Separados por Interface Sinclinal

Aqui, temos que o modelo (veja Figura 3.12) é tal que a velocidade vale 1500 m/s acima e 2000 m/s abaixo do refletor descrito pela curva

$$z = 600 + 300e^{-x^2/10^5} . \quad (3.47)$$

A Figura 3.12 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.13 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Comparando com os casos anteriores, o erro percentual é maior nas duas profundidades estudadas. Isto ocorre porque agora temos um modelo com uma região de alto contraste. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 210s para os métodos de malha fixa e adaptativo, respectivamente. Notemos que o tempo computacional gasto usando o método adaptativo foi muito maior, devido à alta curvatura que a frente onda adquire ao passar pelo refletor. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. O valor de z_{ini} neste caso foi o próprio z_L , pois até esta profundidade não há variações no modelo de velocidades. Quando o refletor é alcançado, logo abaixo de 500 m, vários refinamentos são feitos pois a curvatura nesta

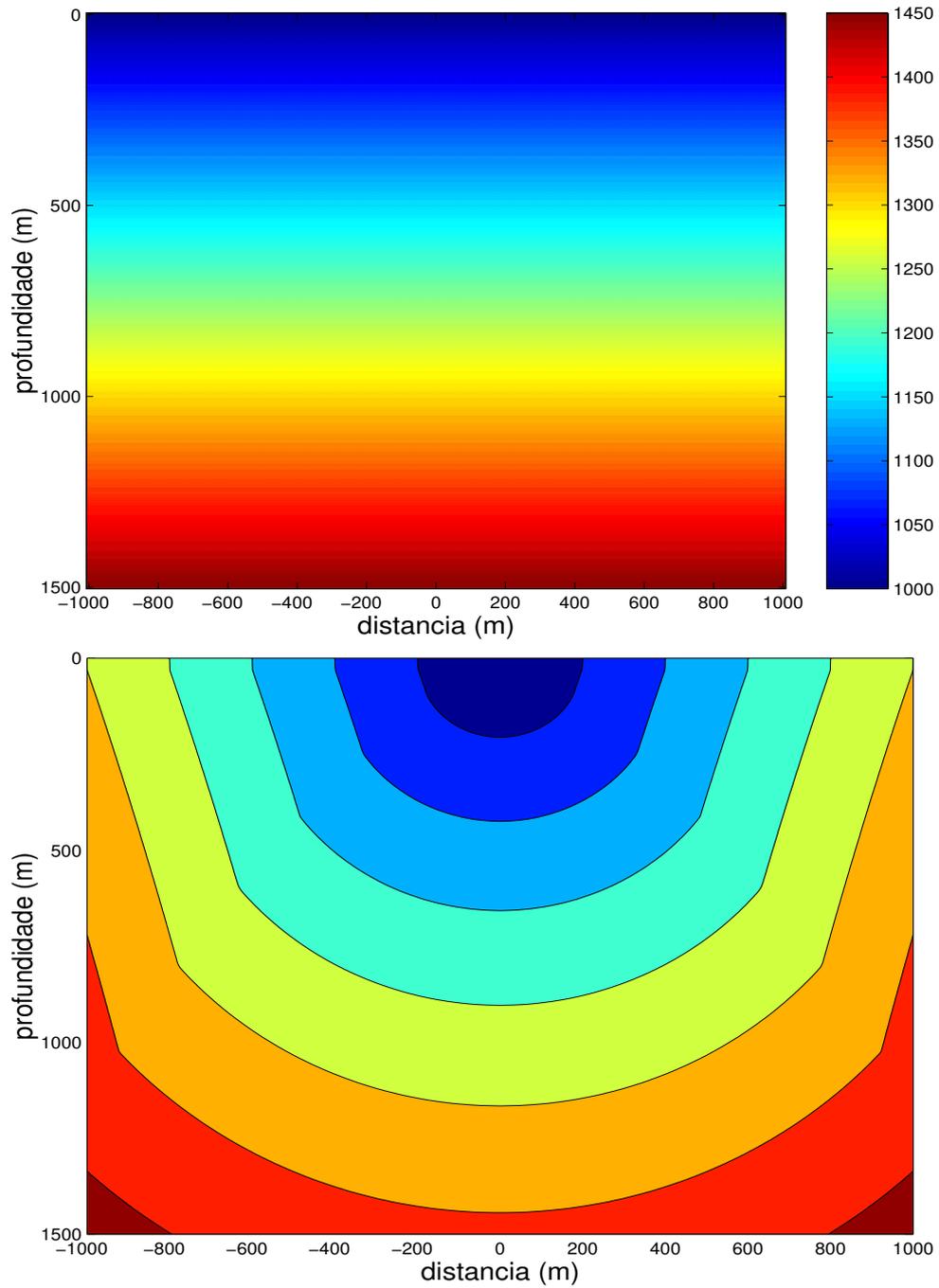


Figura 3.10: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

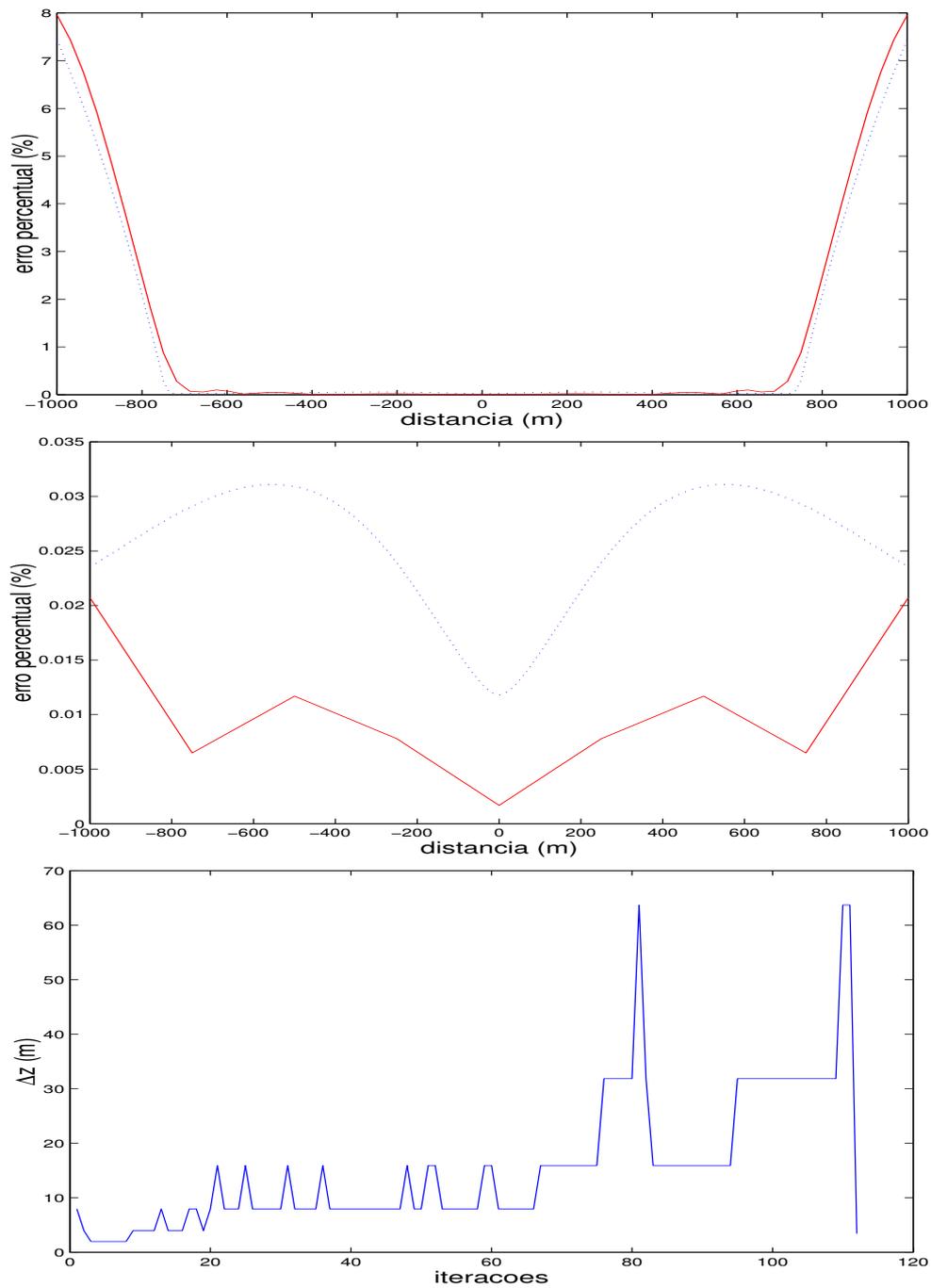


Figura 3.11: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

região muda bruscamente. Esses novos espaçamentos pequenos são mantidos até que a curvatura diminua, onde então voltam a ocorrer as expansões dos espaçamentos.

3.3.4 Meios Homogêneos Separados por Interface Anticlinal

Agora, temos que o modelo (veja Figura 3.14) é tal que a velocidade vale 1500 m/s acima e 2000 m/s abaixo do refletor descrito pela curva

$$z = 900 - 300e^{-x^2/10^5} . \quad (3.48)$$

A Figura 3.14 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.15 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Pelos mesmos motivos do último exemplo, o erro percentual é maior que o encontrado nos dois primeiros casos. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 500s para os método de malha fixa e adaptativo, respectivamente. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. Novamente, quando o refletor é alcançado, vários refinamentos são feitos pois a curvatura nesta região muda bruscamente. Este teste se comporta um pouco pior que o último, pois conforme podemos ver no gráfico com os tempos de trânsito na Figura 3.14, as frentes de onda possuem curvaturas maiores se comparadas ao exemplo anterior.

3.3.5 Meios Homogêneos Separados por Interface com Quina

Neste teste, o modelo contém um refletor com um ponto de difração (veja Figura 3.16). A velocidade vale 1500 m/s acima do refletor e 2000 m/s abaixo deste. A Figura 3.16 apresenta a aproximação encontrada usando o método de malha fixa. Na região logo abaixo da parte inclinada do refletor, o tempo de trânsito foi calculado de maneira totalmente errada. Isto ocorreu porque os métodos não “enxergam” raios com inclinação maior que θ_{\max} , e, assim, podemos ver uma outra linha logo abaixo desta reta inclinada do refletor, justamente até onde ocorreu a propagação de maneira correta. Nos dois primeiros gráficos da Figura 3.17 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva

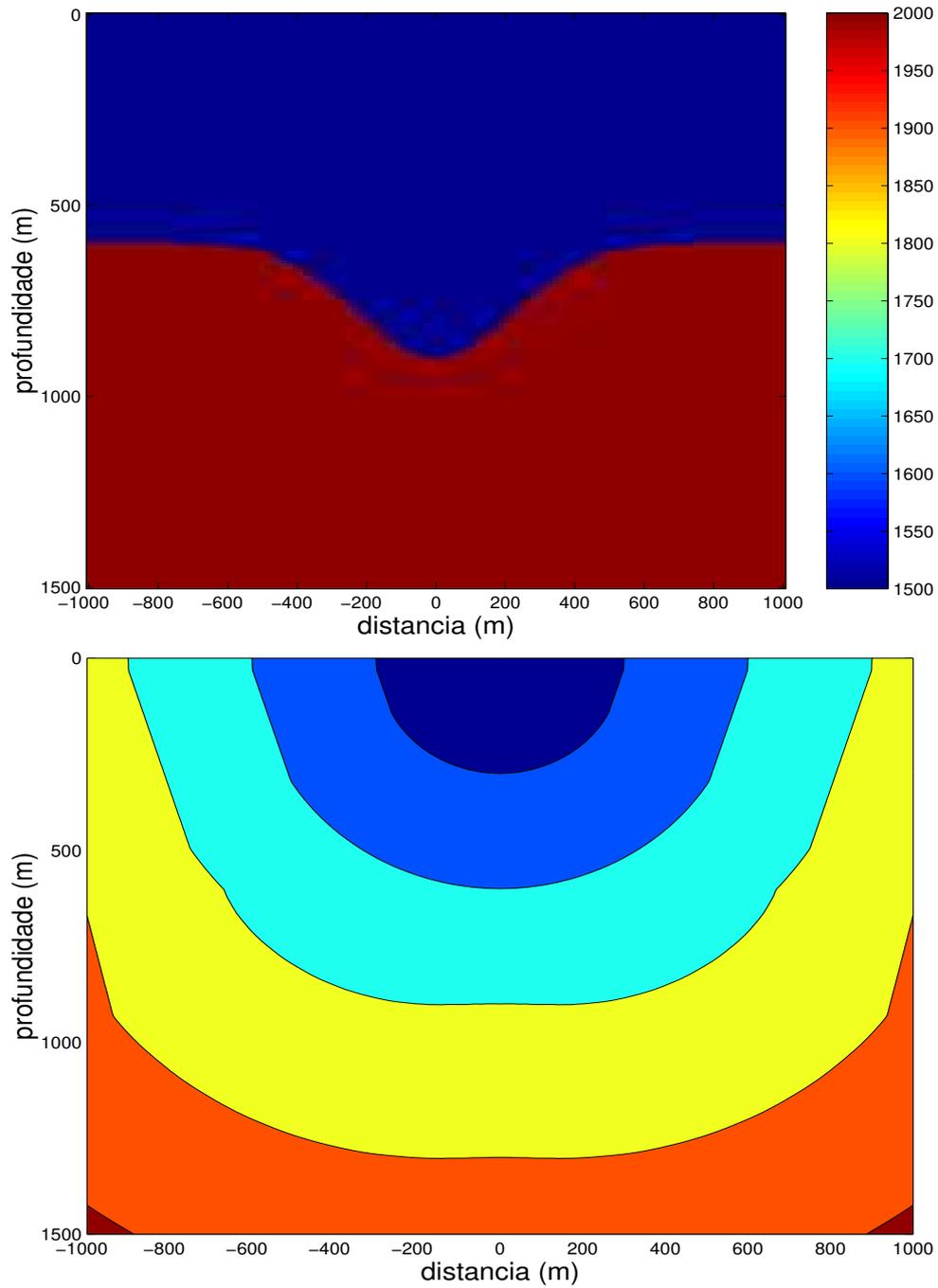


Figura 3.12: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

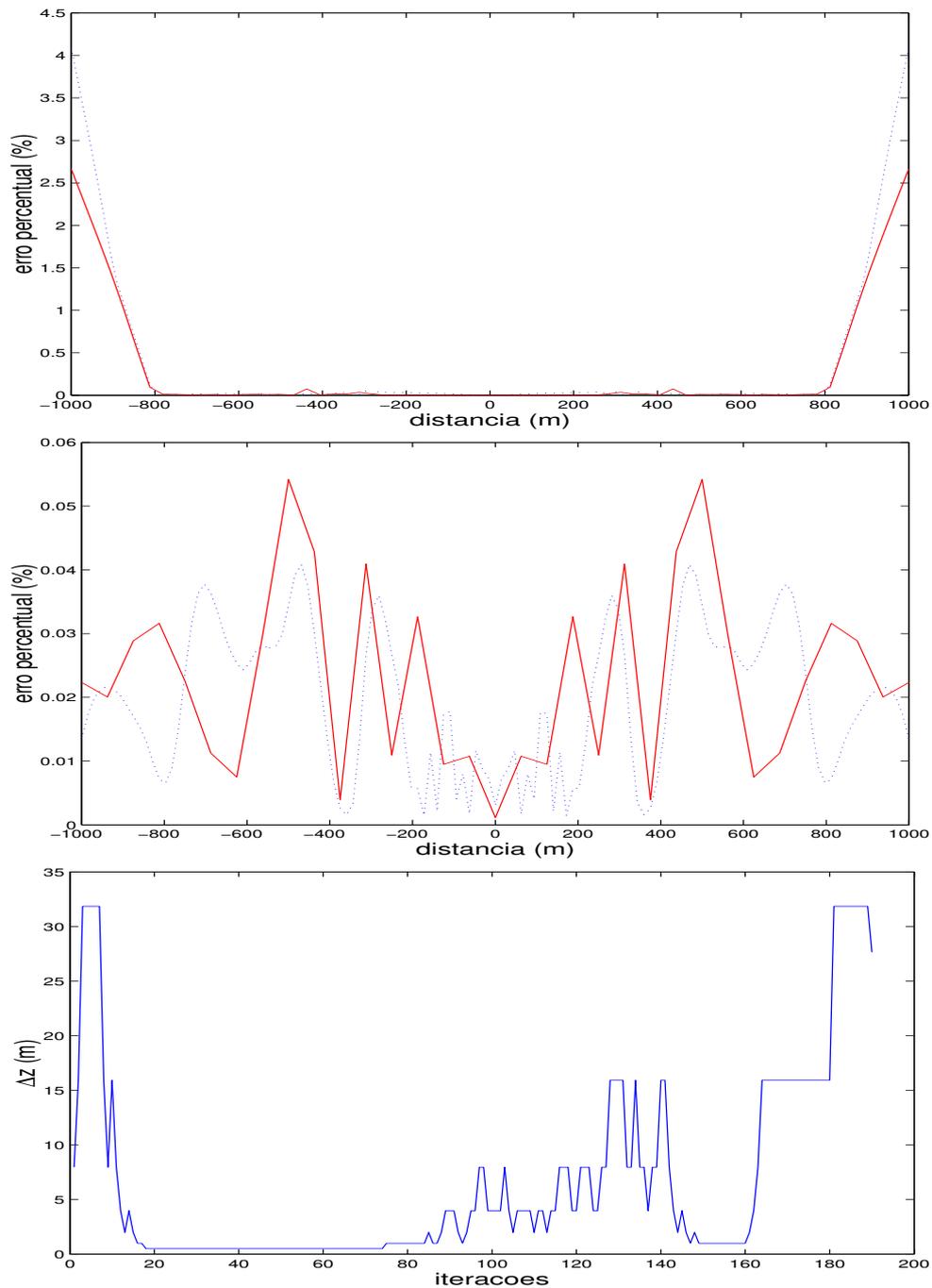


Figura 3.13: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

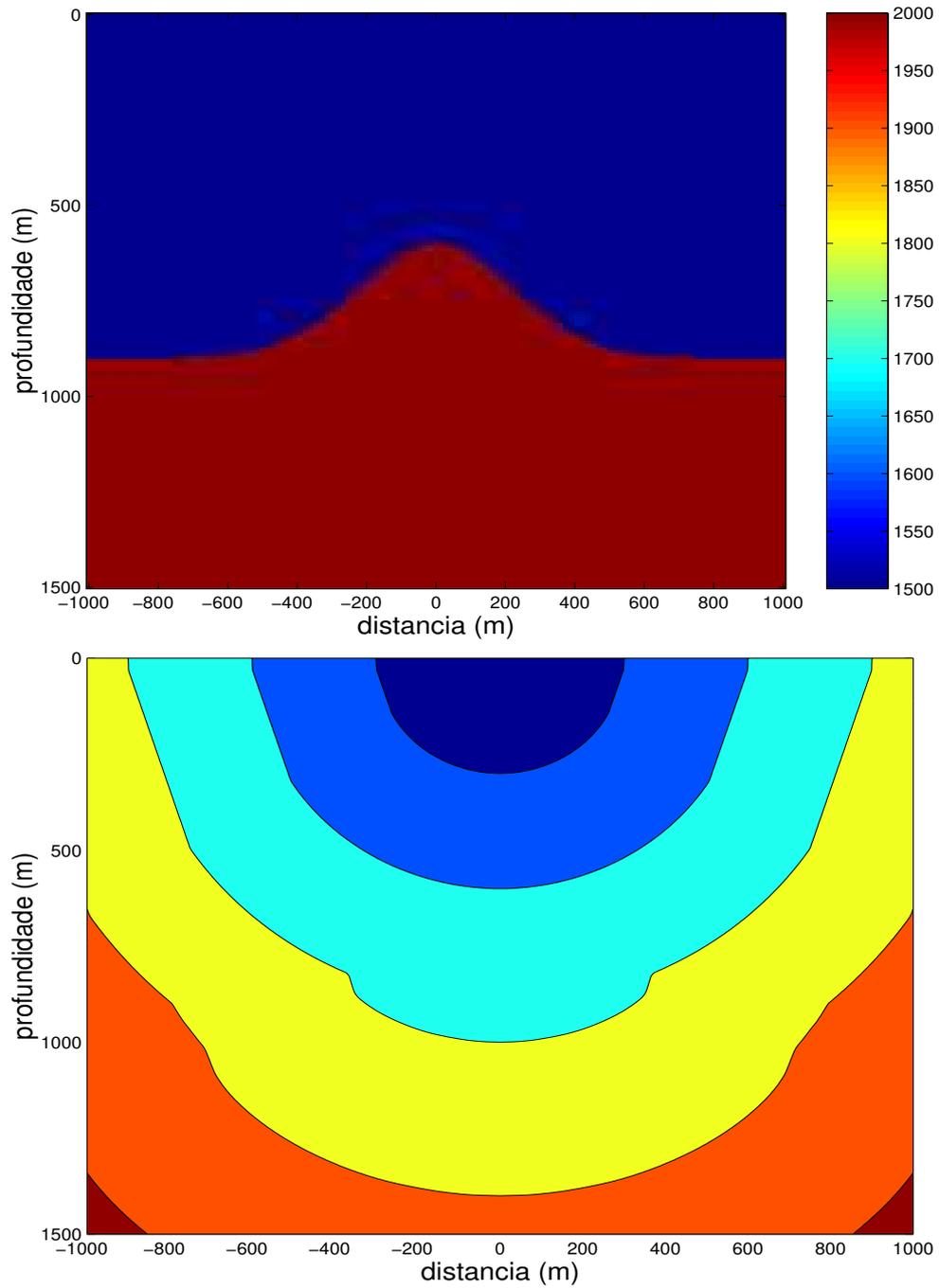


Figura 3.14: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

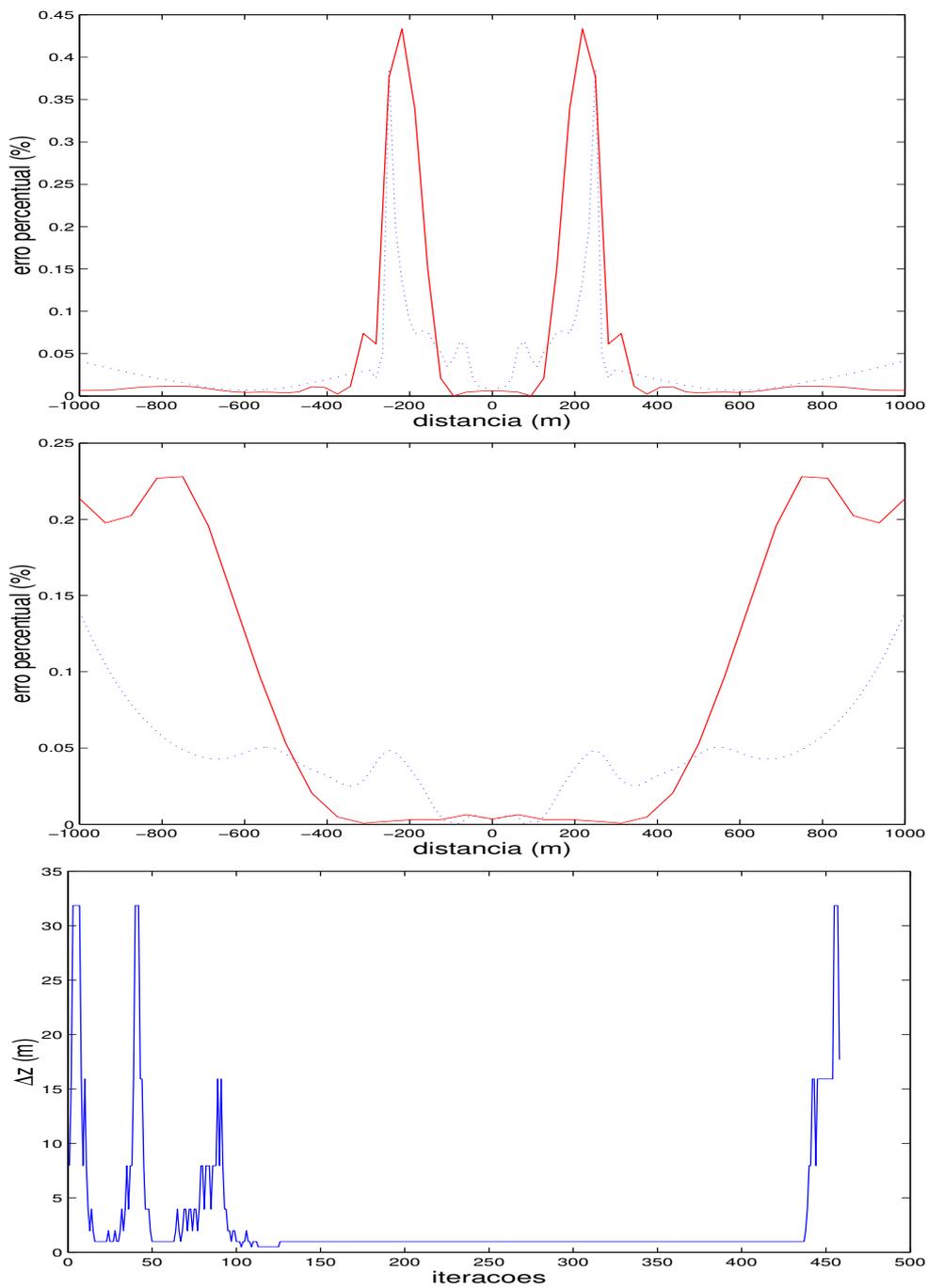


Figura 3.15: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

sólida), nas profundidades 750 m e 1500 m, respectivamente, onde podemos ver o resultado deste equívoco com bastante clareza. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 1200s para os método de malha fixa e adaptativo, respectivamente. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. Neste exemplo, os refinamentos na região de descontinuidade são muitos, e esses espaçamentos pequenos permanecem até que a região de subestimação do tempo de trânsito passa a sumir, já por volta de 1500 m.

3.3.6 Meio com Velocidade Exponencial em z ($v(x, z) = v_0 e^{gz}$)

Neste caso, o modelo é tal que a velocidade aumenta exponencialmente com a profundidade (veja Figura 3.18), conforme a equação

$$v(x, z) = 1000 e^{z/3000} . \quad (3.49)$$

A Figura 3.18 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.19 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Este caso comporta-se de maneira muito semelhante àquele onde a velocidade varia linearmente com z . O alto erro em $z = 750$ m é por causa do limitante θ_{\max} para o ângulo, depois o algoritmo encontra a aproximação com um erro percentual menor. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 13.8s para os método de malha fixa e adaptativo, respectivamente. Na mesma figura, temos a variação do espaçamento Δz no decorrer das iterações. O comportamento é análogo ao ocorrido no caso do meio que varia linearmente com z .

3.3.7 Meio com Velocidade Não Linear em z

Aqui, o modelo é tal que a velocidade decai não linearmente com a profundidade (veja Figura 3.20), conforme a equação

$$v(x, z) = \frac{1}{\sqrt{(1/v_0)^2 + 10^{-8}z}} , \quad (3.50)$$

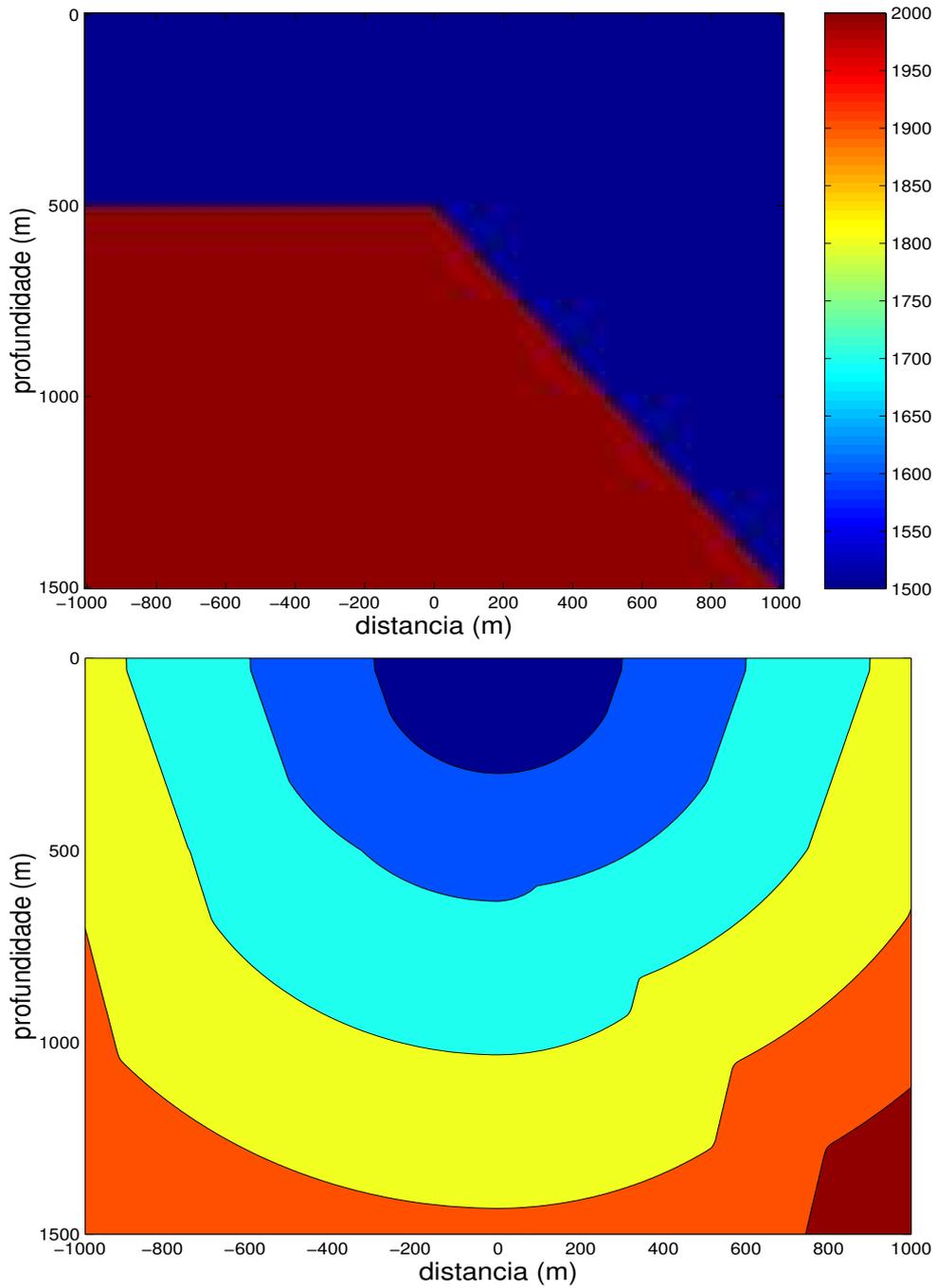


Figura 3.16: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

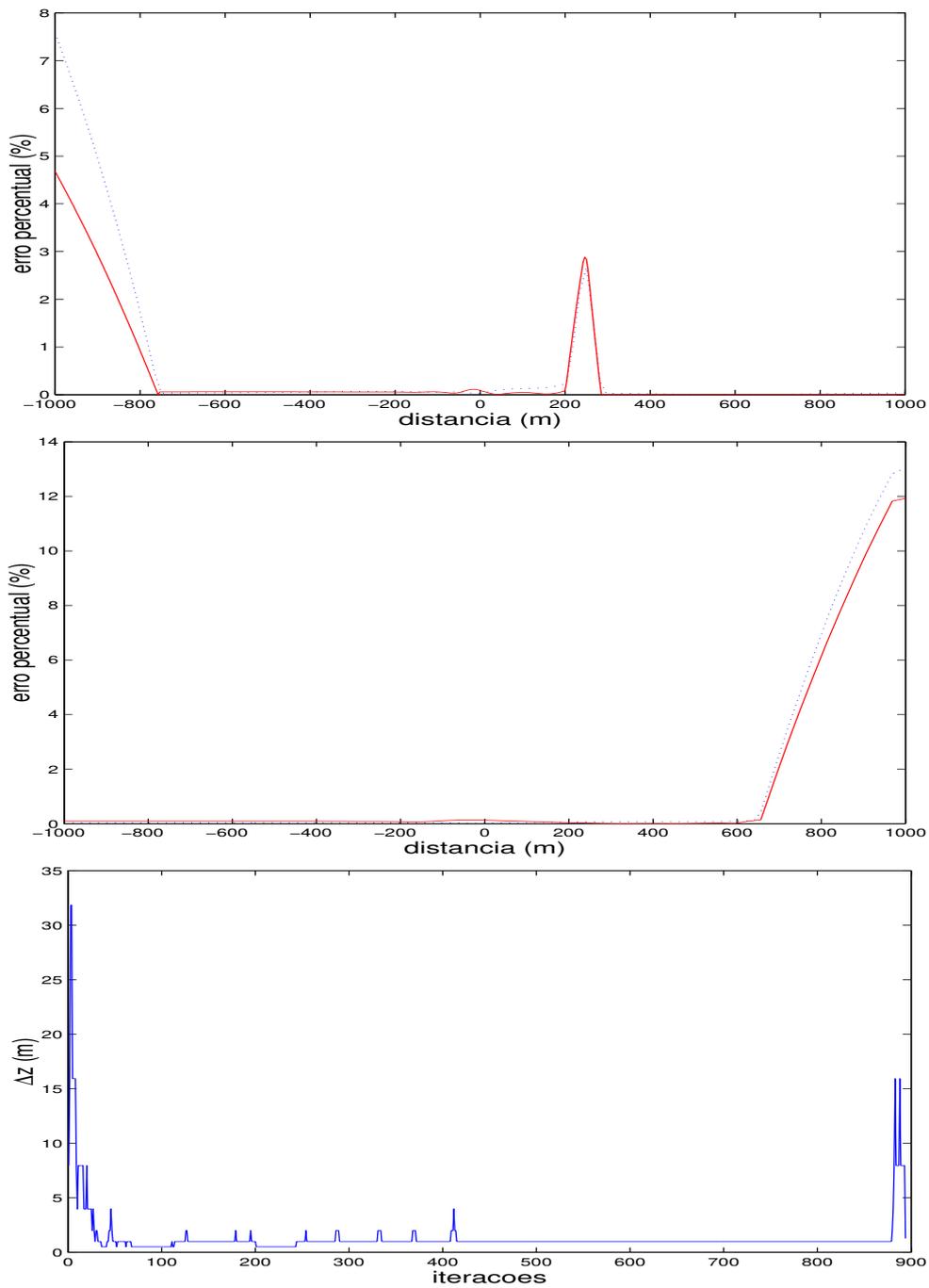


Figura 3.17: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

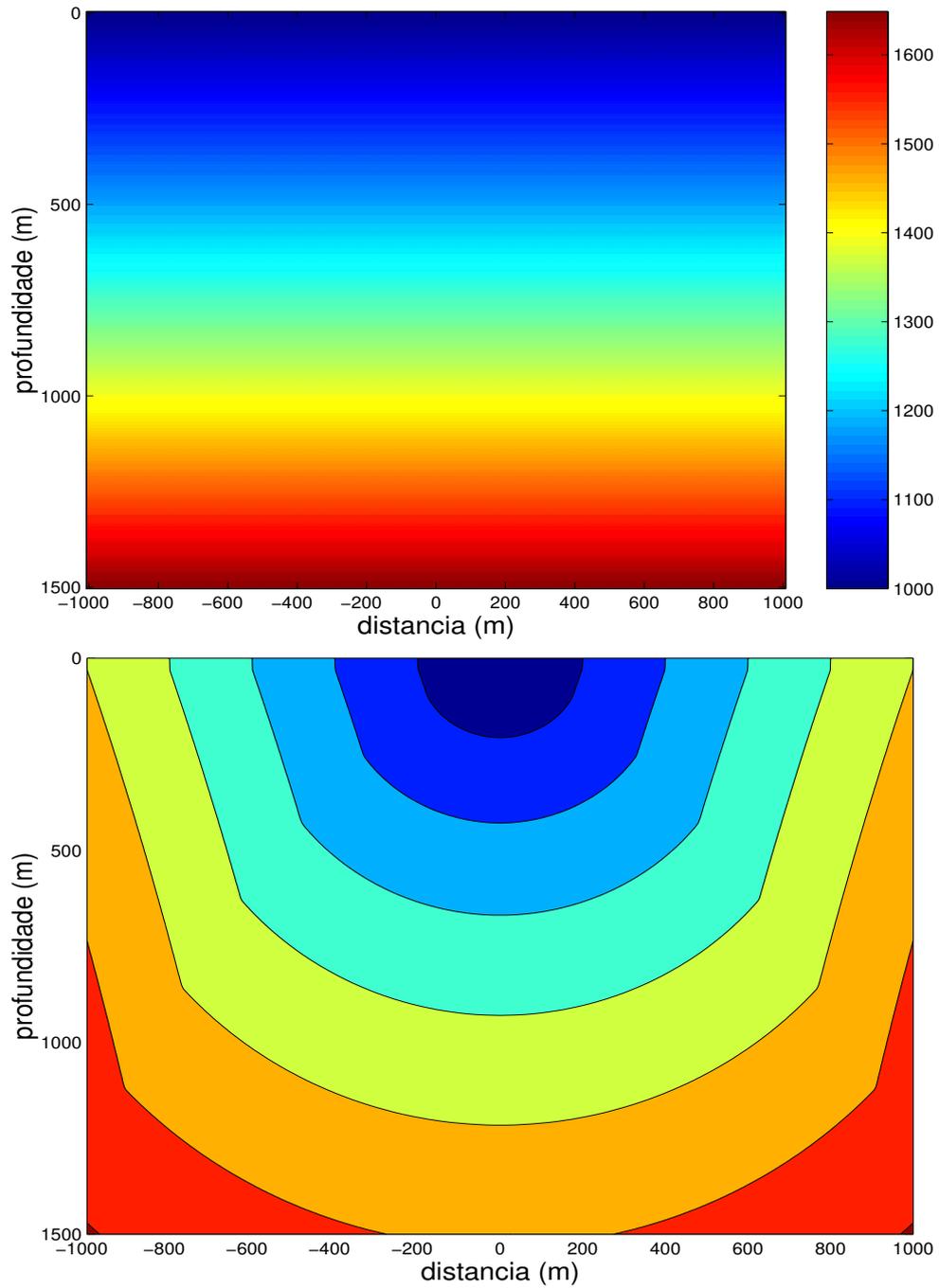


Figura 3.18: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

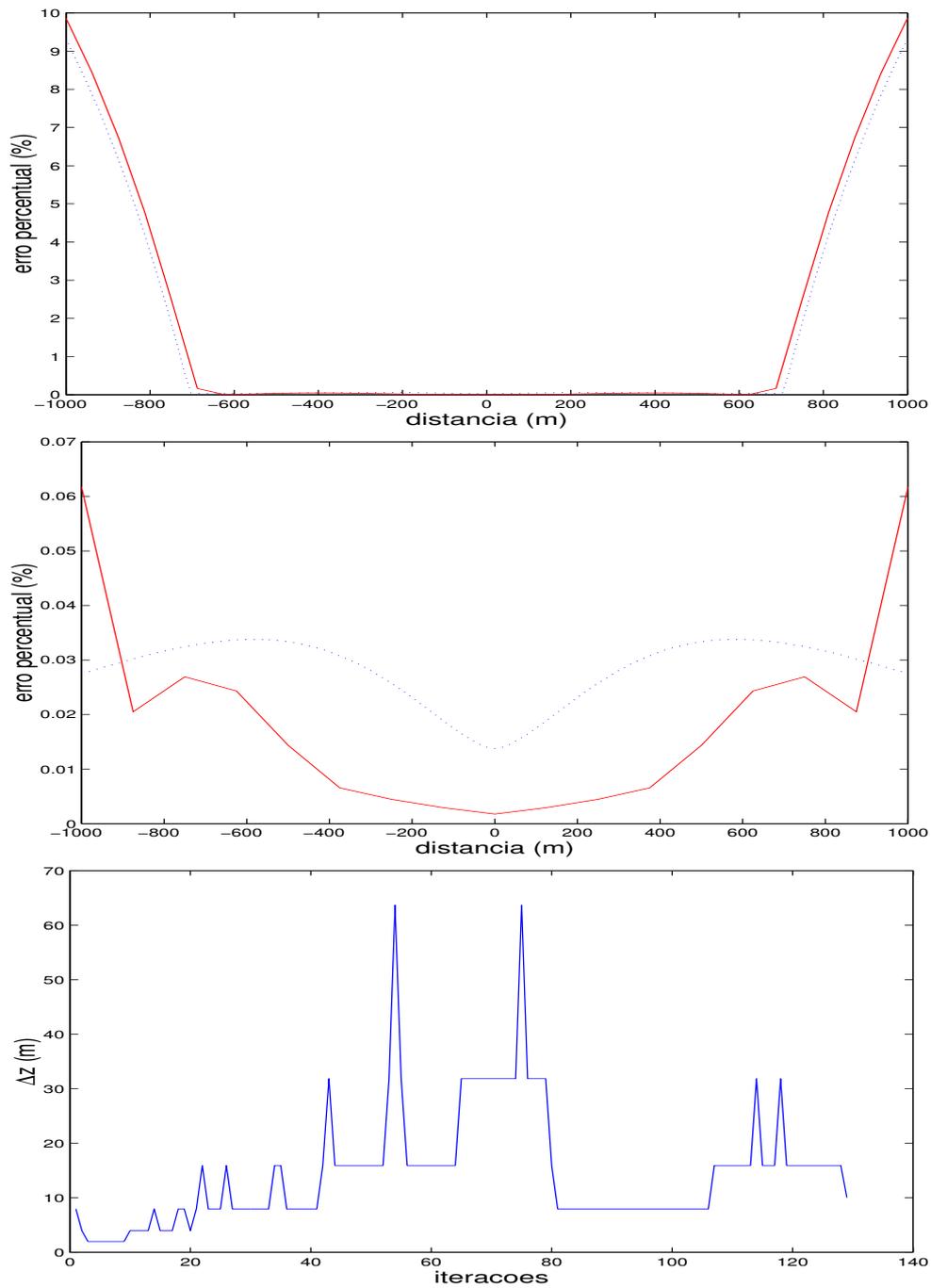


Figura 3.19: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

onde v_0 vale 1000 m/s. A Figura 3.20 apresenta a aproximação encontrada usando o método de malha fixa. Nos dois primeiros gráficos da Figura 3.21 temos a comparação entre os erros percentuais das aproximações encontradas utilizando o método de malha fixa (curva pontilhada) e o adaptativo (curva sólida), nas profundidades 750 m e 1500 m, respectivamente. Este erro percentual maior na aproximação encontrada usando o método adaptativo depende da nossa escolha para a tolerância ε , que para este caso deveria ser menor. O importante é observarmos que, devido às propriedades deste meio, os raios tendem a ficar com inclinação cada vez menor. Os tempos computacionais gastos para encontrar as aproximações na profundidade final foram de 22.7s e 29s para os métodos de malha fixa e adaptativo, respectivamente. Assim, a curvatura da frente de onda tende a ficar linear, e por isso, temos o aumento dos espaçamentos no decorrer das iterações logo após um refinamento próximo à inicialização.

3.4 Comentários Gerais

Estes experimentos numéricos nos mostram claramente as vantagens e desvantagens do método adaptativo. A primeira consideração importante é que quando temos um modelo de velocidades com descontinuidade, o método adaptativo torna-se muito caro computacionalmente. Em geral, nos casos onde a velocidade varia de maneira regular, o método adaptativo consegue uma aproximação com um tempo computacional menor que a aproximação encontrada pelo método de malha fixa.

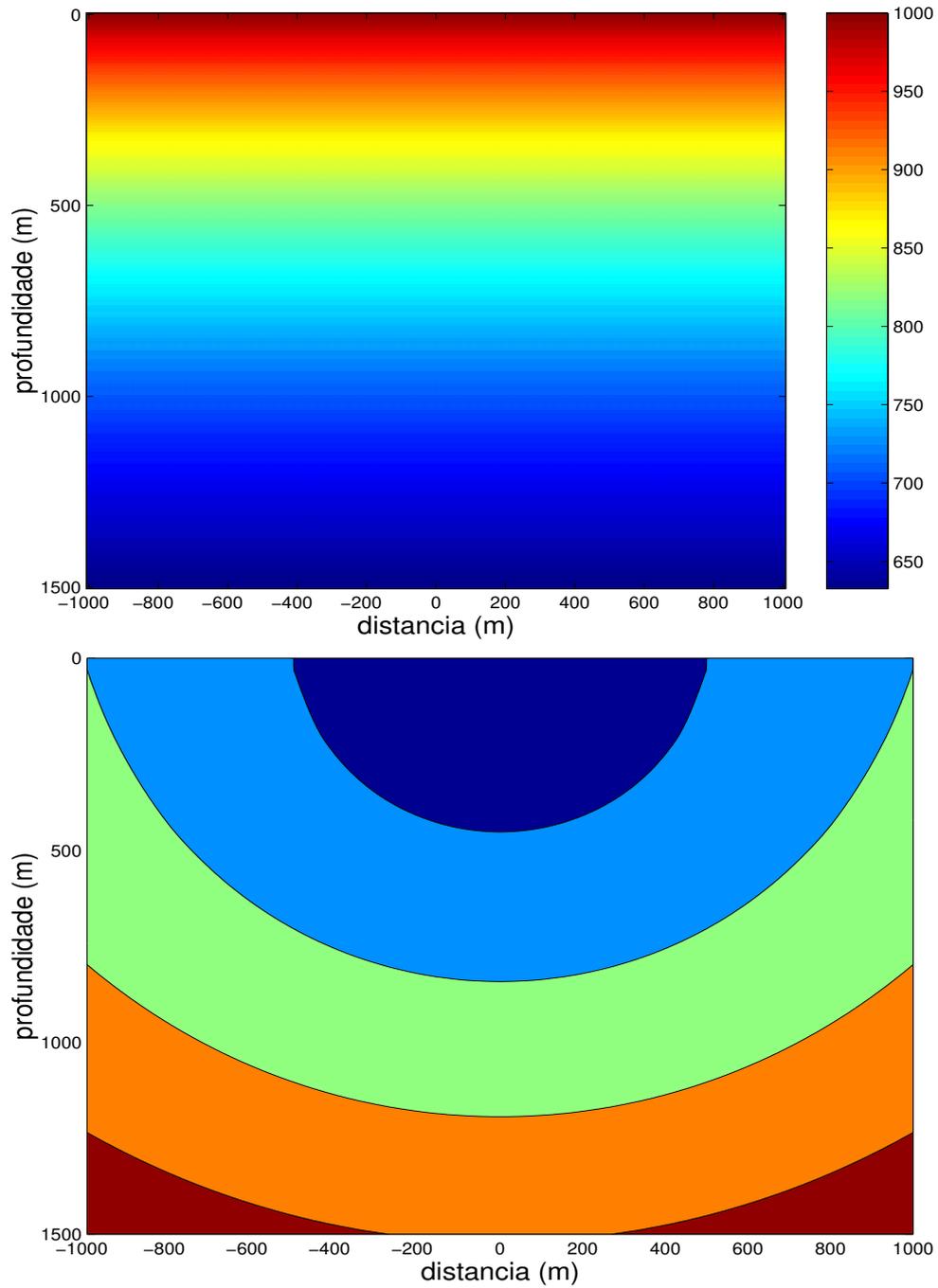


Figura 3.20: No parte superior, temos o modelo de velocidade. Abaixo, a aproximação encontrada usando o método de malha fixa expandindo apenas em z .

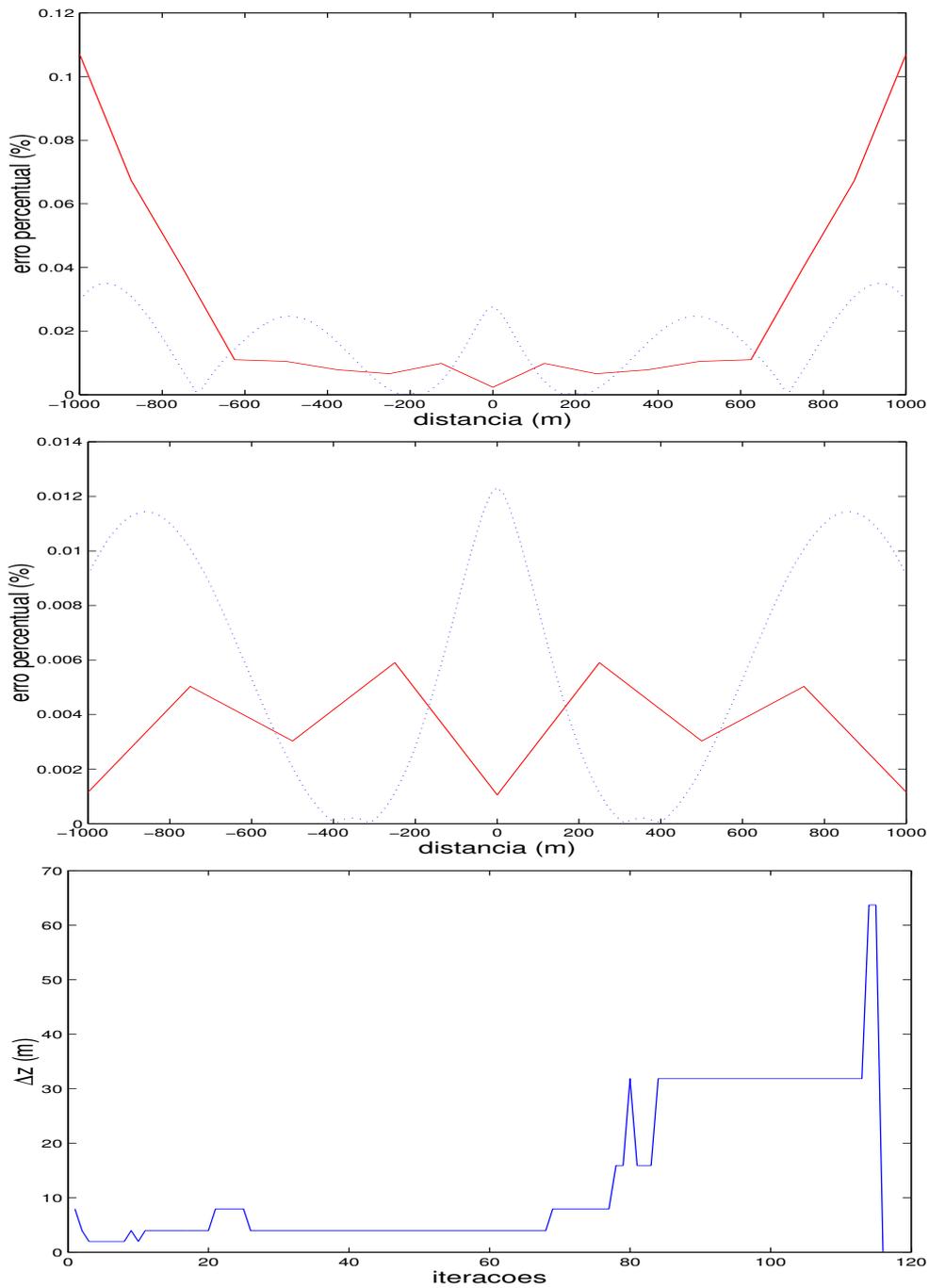


Figura 3.21: Os dois primeiros gráficos tratam dos erros percentuais usando o método de malha fixa (curva pontilhada) e o método adaptativo (curva sólida), nas profundidades de 750 m e 1500 m, respectivamente. O terceiro gráfico, apresenta a variação do espaçamento no decorrer das iterações do método adaptativo.

Capítulo 4

Conclusão

Neste trabalho, propusemos estudar o modelamento sísmico assintótico utilizando os esquemas de diferenças finitas na equação iconal. Desde o primeiro momento, percebemos as dificuldades de criar esquemas de maneira a produzir uma aproximação que tivesse um significado físico. Essas dificuldades surgem na inicialização, onde temos frentes de onda com alta curvatura, e no cálculo de forma a respeitar a propagação, que poderia fazer com que o esquema usasse informação errada no cálculo de uma aproximação.

No caso dos métodos usando malha fixa, conseguimos de certa forma contornar o problema da inicialização, visto que o erro nesta região pode ser diminuído após uma inicialização refinada. No entanto, o tamanho desta região, assim como o número de células a serem refinadas, são livres, não tendo qualquer relação com o problema. A grande vantagem dos métodos de malha fixa é que podemos usar a expansão DNO e assim, obter aproximações que exibem os raios que partem em todas as direções. Mais que isso, caso em alguma região o tempo de trânsito seja superestimado (ou subestimado), podemos calcular uma nova aproximação com base na antiga, escolhendo outras direções para aplicar o método de malha fixa, fazendo com que os nós na malha usem informações dos raios corretos (método corretivo PS).

O método adaptativo surge com a proposta de obter uma aproximação em uma certa profundidade, e não no domínio todo, o que já é um resultado menos rico em informações. A grande vantagem é que o erro pode ser controlado por uma tolerância ε . Para isso, o problema da inicialização é superado, já que a profundidade inicial pode ser tão pequena quanto necessária (obtida pelas condições da Seção 3.1), e podem ocorrer refinamentos ao longo do processo de expansão em z . Mas como o método usa expansão apenas na direção

z , pode ocorrer de o tempo de trânsito sofrer superestimação (ou subestimação). Como no exemplo da Figura 2.13, por mais refinamentos que ocorressem no método, o domínio de dependência numérica seria sempre o mesmo e, assim, a frente de onda continuaria se propagando de maneira errada. Outro fator importante é o tempo computacional, onde o método adaptativo é caro quando o aplicamos em um meio onde a velocidade varia bruscamente.

Portanto, o método adaptativo é muito vantajoso quando aplicado em meios onde a velocidade varia de forma regular, pois assim seu tempo computacional não é tão alto e podemos prever de forma segura o quão boa é a aproximação. Mas para meios mais complexos, métodos de malha fixa como o RK-ENO usando expansão DNO mostraram-se como uma opção bastante atraente, onde devido a usarem malha fixa, podem obter a aproximação em um conjunto de dados maior e ainda serem corrigidos caso haja a necessidade de um pós-processamento.

Um possível trabalho futuro neste problema seria o estudo de formas alternativas em suavizar o tempo de trânsito na interpolação, na tentativa de forçar o método adaptativo a não fazer vários refinamentos em locais de alta curvatura neste.

Referências Bibliográficas

- [1] C. BELFI, 1997, *Second and third order ENO methods for the eikonal equation*, Annual Report, The Rice Inversion Project.
- [2] C. BELFI & W. SYMES, 1998, *An adaptive ENO algorithm for the eikonal equation*, Annual Report, The Rice Inversion Project.
- [3] N. BLEISTEIN, 1984, *Mathematical methods for wave phenomena*, Academic Press Inc., New York.
- [4] J. M. CARCIONE, G. C. HERMAN & A. P. E. TEN KROODE, 2002, *Seismic modeling*, Geophysics, **67**, 1304-1325.
- [5] M. CRANDALL & P. LIONS, 1983, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., **227**, 1-42.
- [6] M. CRANDALL & P. LIONS, 1984, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comput., **43**, 1-19.
- [7] A. HARTEN & S. OSHER, 1987, *Uniformly high order accurate non-oscillatory schemes*, I, SIAM J. Numer. Anal., **24**, 279–309.
- [8] A. HARTEN, B. ENGQUIST, S. OSHER & S. CHAKRAVARTHY, 1987, *Uniformly high order accurate essentially non-oscillatory schemes*, III, J. Comput. Phys., **71**, 231–303.
- [9] S. KIM & R. COOK, 1999, *3-D travelttime computation using second order ENO scheme*, Geophysics, **64**, 1867–1876.
- [10] L. MO & J. M. HARRIS, 2002, *Finite-difference calculation of direct-arrival travel-times using the eikonal equation*, Geophysics, **67**, 1270–1274.

- [11] S. OSHER & C.-W. SHU, 1991, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Numer. Anal., **28**, 907–922.
- [12] J. QIAN & W. SYMES, 2002, *An adaptive finite-difference method for traveltimes and amplitudes*, Geophysics, **67**, 167–176.
- [13] C.-W. SHU & S. OSHER, 1988, *Efficient implementations of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., **77**, 439–471.
- [14] C.-W. SHU & S. OSHER, 1989, *Efficient implementations of essentially non-oscillatory shock-capturing schemes, II*, J. Comput. Phys., **83**, 32–78.
- [15] J. W. THOMAS, 1995, *Numerical Partial Differential Equations: Finite Difference Methods*, Springer.
- [16] J. VIDALE, 1988, *Finite-difference calculation of traveltimes*, Bull. Seis. Soc. Am., **78**, no. 6, 2062–2076.