Geofísica Computacional Aplicada
Equação da Onda Sísmica

Carlos H. S. Barbosa[1]
José Luis Drummond Alves

31 de Julho de 2023

Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

[1] clshenry@lamce.coppe.ufrj.br

» UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

CONTATOS:

- José Luis Drummond Alves
- Contato: jalves@lamce.coppe.ufrj.br

- Carlos Henrique dos Santos Barbosa
- Contato: clshenry@lamce.coppe.ufrj.br

- Bruno Souza Silva
- Contato: brunosi@lamce.coppe.ufrj.br

PRINCIPAIS REFERÊNCIAS:

- LeVeque, R. J. Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. Society for Industrial and Applied Mathematics, 2007.
- Barbosa, C. H. S. Advanced Computational Strategies for Reverse Time Migration (Doctoral dissertation, Federal University of Rio de Janeiro), 2023.

REFERÊNCIAS COMPLEMENTARES:

- Chapman, C. Fundamentals of seismic wave propagation. Cambridge university press, 2004.
- Bartolo, L. D. Propagação de ondas aplicadas ao mapeamento geológico: formulação acústica. Revista Brasileira de Ensino de Física, 43, 2021.

» UM POUCO MAIS SOBRE DISCRETIZAÇÃO COM DIFERENÇAS FINITAS

- Diferenças finitas implícita e explícita

» DISCRETIZAÇÃO DA EQUAÇÃO DA ONDA SÍSMICA

- Formulação da equação da onda acústica
- Discretização das derivadas temporal e espaciais
- Equação da onda acústica discretizada

» IMPLEMENTAÇÃO COMPUTACIONAL

» ATIVIDADES PRÁTICAS

- Atividade 1: o sinal sísmico
- Atividade 2: discretização espacial de alta ordem
- Atividade 3: o sismograma - aquisição de spread fixo

UM POUCO MAIS SOBRE DISCRETIZAÇÃO COM DIFERENÇAS FINITAS

Diferenças Finitas Implícita e Explícita

Diferenças Finitas Implícito e Explícito

Esquema de Diferenças Finitas Explícito

$$y_{n+1} = x_n + f(y_n)$$

- O esquema de diferenças finitas é dito explícito se pode ser calculado iterativamente a partir de quantidades previamente calculadas em passos (de tempo) anteriores.

Esquema de Diferenças Finitas Implícito

$$y_{n+1} = x_n + f(y_{n+1})$$

- O esquema de diferenças finitas é dito implícito se as saídas da atualização y_{n+1} dependem de si mesmas.
- São geralmente formulados como um sistema de equações $Ax = b$.

Diferenças Finitas Implícito e Explícito

» A discretização por diferenças finitas da equação da onda sísmica no domínio do tempo conduz à um esquema explícito.

Equação da Onda no Domínio do Tempo

$$\nabla^2 p(\mathbf{r}, t) - \frac{1}{v^2(\mathbf{r})} \frac{\partial^2 p(\mathbf{r}, t)}{\partial t^2} = 0,$$

» Por outro lado, a discretização por diferenças finitas da equação da onda sísmica no domínio da frequência conduz à um esquema implícito.

Equação da Onda no Domínio da Frequência

$$\nabla^2 P(\mathbf{r}, \omega) + \frac{\omega^2}{v^2(\mathbf{r})} P(\mathbf{r}, \omega) = 0$$

DISCRETIZAÇÃO DA EQUAÇÃO DA ONDA SÍSMICA

Formulação da Equação da Onda Acústica

Formulação: Equação da Onda Acústica

$$\nabla^2 p(\mathbf{r},t) - \frac{1}{v^2(\mathbf{r})}\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2} = \frac{\partial^2 i_v(\mathbf{r},t)}{\partial t^2}\delta(\mathbf{r}-\mathbf{r}_s),$$

- $\mathbf{r} = (r_x, r_y, r_z)$ é o vetor posição definido no domínio $\Omega \subset \mathbb{R}^{n_{sd}}$, $n_{sd} = 2, 3$,
- $t \in [0, T]$ é o tempo,
- $p(\mathbf{r}, t)$ é o campo de pressão,
- $v(\mathbf{r})$ é o campo de velocidades,
- i_v a fonte sísmica na posição $\mathbf{r} = \mathbf{r}_s$,
- δ é a delta de Dirac,
- $p(\mathbf{r}, 0) = \partial p(\mathbf{r}, 0)/\partial t = 0$ para $\mathbf{r} \in \Omega$ são as condições iniciais e $p(\mathbf{r}, t) = 0$ em $\partial\Omega$ a condição de contorno.

Discretização das Derivadas Temporal e Espaciais

» DISCRETIZAÇÃO POR DIFERENÇAS FINITAS DA DERIVADA DE SEGUNDA ORDEM

$$\frac{\partial^2 f}{\partial i^2} \approx \frac{1}{\Delta i^2}\left[c_0 f_0 + \sum_{m=1}^{N/2} c_m(f_m + f_{-m})\right],$$

- c_m são os coeficientes da discretização da derivada de segunda ordem,
- i diz respeito a variável de interesse,
- N é a ordem de discretização da derivada.
  - $N = 2$ » diferença finita de segunda ordem,
  - $N = 4$ » diferença finita de quarta ordem,
  - ....

Discretização das Derivadas Temporal e Espaciais

Escolhendo $N = 2$ para a discretização da derivada de segunda ordem, tem-se:

$$\frac{\partial^2 f}{\partial i^2} \approx \frac{1}{\Delta i^2} \left[c_0 f_0 + \sum_{m=1}^{2/2} c_m (f_m + f_{-m}) \right],$$

Expandindo:

$$\frac{\partial^2 f}{\partial i^2} \approx \frac{1}{\Delta i^2} \left[c_0 f_0 + c_1 (f_1 + f_{-1}) \right],$$

Que é discretização centrada de segunda ordem, que vimos nas aulas anteriores.

Discretização das Derivadas Temporal e Espaciais

A função que iremos derivar é o campo de pressão $p(x, y, z, t)$. Além disso, a derivada será aplicada para todas as variáveis, ou seja,

$$\frac{\partial^2 p(x,y,z,t)}{\partial t^2}; \ \frac{\partial^2 p(x,y,z,t)}{\partial x^2}; \ \frac{\partial^2 p(x,y,z,t)}{\partial y^2}; \ \frac{\partial^2 p(x,y,z,t)}{\partial z^2}$$

Utilizando a derivada centrada de segunda ordem, tem-se a aproximação para a derivada temporal:

$$\frac{\partial^2 p(x,y,z,t)}{\partial t^2} \approx \frac{1}{\Delta t^2}[c_0 p(x,y,z,t) + c_1(p(x,y,z,t+\Delta t) + p(x,y,z,t-\Delta t))],$$

onde Δt é denominado taxa de variação temporal.

Discretização das Derivadas Temporal e Espaciais

Já as aproximações para as derivadas espacias, assumindo uma malha espacial regular $h = \Delta x = \Delta y = \Delta z$, são:

$$\frac{\partial^2 p(x,y,z,t)}{\partial x^2} \approx \frac{1}{h^2}[c_0 p(x,y,z,t) + c_1(p(x+h,y,z,t) + p(x-h,y,z,t))],$$

$$\frac{\partial^2 p(x,y,z,t)}{\partial y^2} \approx \frac{1}{h^2}[c_0 p(x,y,z,t) + c_1(p(x,y+h,z,t) + p(x,y-h,z,t))],$$

$$\frac{\partial^2 p(x,y,z,t)}{\partial z^2} \approx \frac{1}{h^2}[c_0 p(x,y,z,t) + c_1(p(x,y,z+h,t) + p(x,y,z-h,t))],$$

Discretização das Derivadas Temporal e Espaciais

Seja $p^n_{i,j,k}$ a representação mais compacta da função que representa ocampo de pressão $p(x, y, z, t)$.

$$\frac{\partial^2 p(x,y,z,t)}{\partial t^2} \approx \frac{1}{\Delta t^2}[c_0 p^n_{i,j,k} + c_1(p^{n+\Delta t}_{i,j,k} + p^{n-\Delta t}_{i,j,k})],$$

$$\frac{\partial^2 p(x,y,z,t)}{\partial x^2} \approx \frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i+h,j,k} + p^n_{i-h,j,k})],$$

$$\frac{\partial^2 p(x,y,z,t)}{\partial y^2} \approx \frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i,j+h,k} + p^n_{i,j-h,k})],$$

$$\frac{\partial^2 p(x,y,z,t)}{\partial z^2} \approx \frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i,j,k+h} + p^n_{i,j,k-h})],$$

Discretização das Derivadas Temporal e Espaciais

Voltando a equação da onda sísmica:

$$\nabla^2 p(\mathbf{r},t) - \frac{1}{v^2(\mathbf{r})}\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2} = \frac{\partial^2 i_v(\mathbf{r},t)}{\partial t^2}\delta(\mathbf{r}-\mathbf{r}_s),$$

Substituindo as diferenças finitas para cada derivada e assumindo que $v_{i,j,k}$ é a representaçao discreta para o campo de velocidades $v(\mathbf{r})$ e que $f^n_{i,j,k}$ a representação discreta para a derivada segunda da fonte sísmica $\partial^2 i_v(\mathbf{r},t)/\partial t^2$, temos:

$$\begin{aligned}
&\frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i+h,j,k} + p^n_{i-h,j,k})] \\
&+\frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i,j+h,k} + p^n_{i,j-h,k})] \\
&+\frac{1}{h^2}[c_0 p^n_{i,j,k} + c_1(p^n_{i,j,k+h} + p^n_{i,j,k-h})] \\
-\frac{1}{v_{i,j,k}}&\frac{1}{\Delta t^2}[c_0 p^n_{i,j,k} + c_1(p^{n+\Delta t}_{i,j,k} + p^{n-\Delta t}_{i,j,k})] = f^n_{i,j,k}
\end{aligned}$$

Discretização das Derivadas Temporal e Espaciais

Rearranjando os termos chega-se à expressão discretizada da equação da onda sísmica para o caso tridimensionais (3D):

$$
\begin{aligned}
p_{i,j,k}^{n+\Delta t} = C_{i,j,k}[3.0 * c_0 * p_{i,j,k}^{n} + \\
& c_1(p_{i+h,j,k}^{n} + p_{i-h,j,k}^{n} + p_{i,j+h,k}^{n} \\
& + p_{i,j-h,k}^{n} + p_{i,j,k+h}^{n} + p_{i,j,k-h}^{n})] \\
& + 2.0 * p_{i,j,k}^{n} - p_{i,j,k}^{n-\Delta t} + v_{i,j,k}^{2} * f_{i,j,k}^{n},
\end{aligned}
$$

onde $C_{i,j,k} = (v_{i,j,k}\Delta t/h)^2$ e os coeficientes das diferenças finitas $c_0 = -2.0$ e $c_1 = 1.0$.

» Para o caso bidimensional (2D):

$$
\begin{aligned}
p_{i,k}^{n+\Delta t} = C_{i,k}[2.0 * c_0 * p_{i,k}^{n} \\
& + c_1(p_{i+h,k}^{n} + p_{i-h,k}^{n} + p_{i,k+h}^{n} + p_{i,k-h}^{n})] \\
& + 2.0 * p_{i,k}^{n} - p_{i,k}^{n-\Delta t} + v_{i,k}^{2} * f_{i,k}^{n},
\end{aligned}
$$

onde $C_{i,k} = (v_{i,k}\Delta t/h)^2$

Sistema de Coordenadas da Malha de Diferenças Finitas

» Domínio $\Omega = [0, dimZ] \times [0, dimX] \subset \mathbb{R}^2$
» Eixo X: Extensão do Domínio (dimX: quantidade de pontos no eixo X)
» Eixo Z: Profundidade do Domínio (dimZ: quantidade de pontos no eixo Z)

IMPLEMENTAÇÃO COMPUTACIONAL

Descrição das Variáveis do Problema

```
h                                        # espaçamento da malha (m)
dt                                       # taxa de variação temporal (s)
dimX                                     # extensão do domínio
dimZ                                     # profundidade do domínio
Ntotal                                   # tempo total discreto
cut_frequency                            # frequência de corte (Hz)

seismic_souce = np.zeros(Ntotal)         # fonte sísmica
velocity     = np.zeros((dimZ, dimX))    # campo de velocidades
C            = np.zeros((dimZ, dimX))    # matriz auxilar

wavefield01 = np.zeros((dimZ, dimX))     # campo de pressão em n - dt
wavefield02 = np.zeros((dimZ, dimX))     # campo de pressão em n
wavefield03 = np.zeros((dimZ, dimX))     # campo de pressão em n + dt

seismic_signal = np.zeros(Ntotal)        # sinal sísmico em 1 receptor

shot_x, shot_z                           # posicionamento fonte sísmica
rec_x, rec_z                             # posicionamento receptores
```

Kernel Propagação do Campo de Ondas

```
for n in range(0, Ntotal):

    #kernel da equação da onda sísmica
    for i in range(1, dimZ - 1):
        for k in range(1, dimX - 1):
            wavefield03[i,k] = C[i,k] * (2.0*c[0]*wavefield02[i, k]
                                + c[1]*(wavefield02[i,k-1] + wavefield02[i,k+1]
                                      + wavefield02[i-1,k] + wavefield02[i+1,k]))
                                      + 2.0*wavefield02[i,k] - wavefield01[i,k]

    wavefield02[shot_z,shot_x] = wavefield02[shot_z,shot_x]
                                 + fonte[n]*(velocity[shot_z,shot_x]**2)

    seismic_signal[n] = wavefield02[rec_z,rec_x]

    # Atualização do campo de onda
    wavefield01 = np.copy(wavefield02)
    wavefield02 = np.copy(wavefield03)
```

Inicialização Variáveis Auxiliares

» Campo de velocidades, coeficiente de diferenças finitas e matriz auxiliar $C_{i,k}$.

```
# campo de velocidades
for i in range(0, dimZ):
    for k in range(0, dimX):
        velocity[i][k] = 2000.0
```

```
# coeficiente de diferenças finitas
c = (-2.0, 1.0)                # para a discretização de segunda ordem

# matriz auxiliar
for i in range(0, dimZ):
    for k in range(0, dimX):
        C[i,k] = (velocity[i,k]*dt/h)**2
```

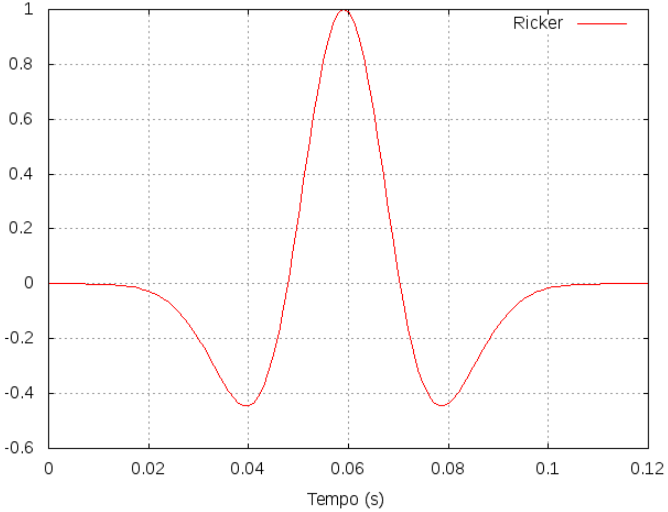
Inicialização Variáveis Auxiliares

Fonte sísmica: Derivada segunda da função Gaussiana.
Necessário definir a frequência de corte.

```
# fonte sísmica
t = np.zeros(Ntotal)                        # matriz de tempos
TF = 2 * sqrt(pi) / cut_frequency           # Período da função Gaussiana
fc = cut_frequency / (3. * sqrt(pi))        # Frequência central

for n in range(Ntotal):
    t[n] = ((n) * dt - TF)
    fonte[n] =  -(-exp(-pi * (pi * fc * t[n]) ** 2)
              * (1 - 2 * pi * (pi * fc * t[n])
              * (pi * fc * t[n]) ) )
```

Fonte Sísmica: Ricker

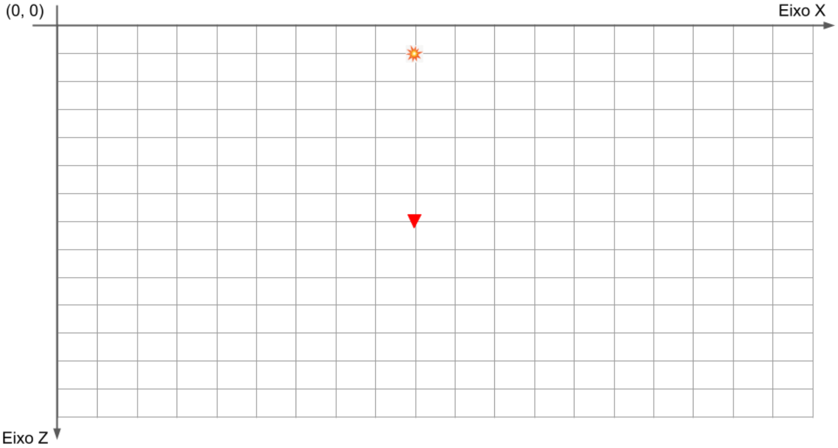
» Derivada segunda da função Gaussiana



ATIVIDADES PRÁTICAS

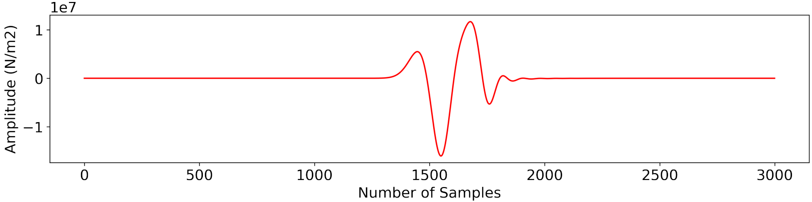
Atividade 1: O Sinal Sísmico

» Campo de velocidades constante e igual a 1500 m/s
» Propagação do campo de ondas considerando uma fonte sísmica
» Registro do campo de ondas em um único receptor



Atividade 1: O Sinal Sísmico

- Dimensão do domínio: $\Omega = [0, 300] \times [0, 200]$
- Localização da fonte sísmica: [shot_x, shot_z] = [150, 5]
- Localização do receptor: [rec_x, rec_z] = [150, 100]



Atividade 2: Discretização Espacial de Alta Ordem

» Discretização da equação da onda sísmica para o caso bidimensional (2D) utilizando as diferenças fintias de segunda ordem no tempo e quarta ordem no espaço:

$$
\begin{aligned}
p_{i,k}^{n+\Delta t} = C_{i,k}[2.0 * c_0 * p_{i,k}^n & \\
& + c_1(p_{i+h,k}^n + p_{i-h,k}^n + p_{i,k+h}^n + p_{i,k-h}^n) \\
& + c_2(p_{i+2h,k}^n + p_{i-2h,k}^n + p_{i,k+2h}^n + p_{i,k-2h}^n)] \\
& + 2.0 * p_{i,k}^n - p_{i,k}^{n-\Delta t} + v_{i,k}^2 * f_{i,k}^n,
\end{aligned}
$$

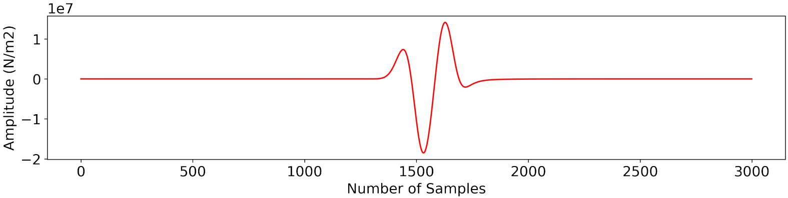
onde $C_{i,k} = (v_{i,k}\Delta t/h)^2$ e os coeficientes das diferenças finitas $c_0 = -5/3$, $c_1 = 4/3$ e $c_2 = -1/12$.

Atividade 2: Discretização Espacial de Alta Ordem

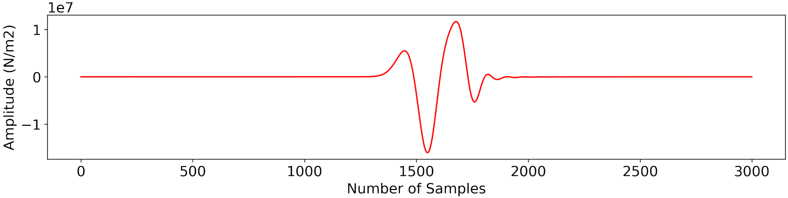
```
c = (-5.0/2.0, 4.0/3.0, -1.0/12.0)      # coeficiente de diferenças finitas

for n in range(0, Ntotal):
    #kernel da equação da onda sísmica
    for i in range(2, dimZ - 2):
        for k in range(2, dimX - 2):
            wavefield03[i,k] = C[i,k] * (2.0*c[0]*wavefield02[i, k]
                                + c[1]*(wavefield02[i,k-1] + wavefield02[i,k+1]
                                      + wavefield02[i-1,k] + wavefield02[i+1,k])
                                + c[2]*(wavefield02[i,k-2] + wavefield02[i,k+2]
                                      + wavefield02[i-2,k] + wavefield02[i+2,k]))
                                      + 2.0*wavefield02[i,k] - wavefield01[i,k]

    wavefield02[shot_z,shot_x] = wavefield02[shot_z,shot_x]
                                   + fonte[n]*(velocity[shot_z,shot_x]**2)

    seismic_signal[n] = wavefield02[rec_z,rec_x]

    # Atualização do campo de onda
    wavefield01 = np.copy(wavefield02)
    wavefield02 = np.copy(wavefield03)
```

Atividade 2: O Sinal Sísmico

» Sinal sísmico para a discretização de quarta ordem no espaço

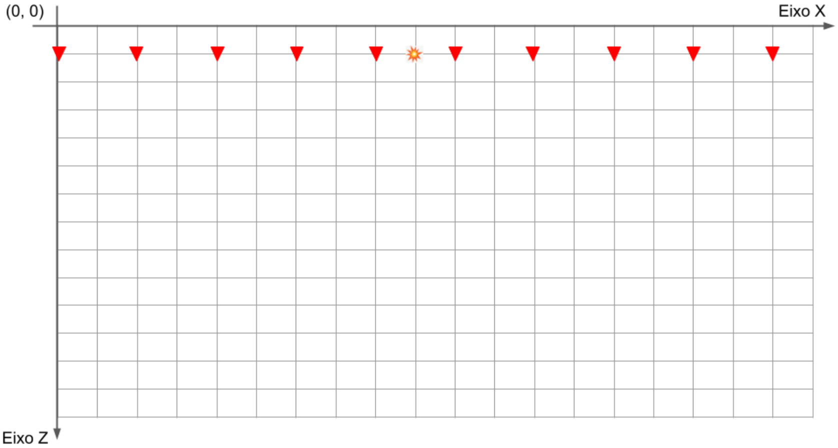


» Sinal sísmico para a discretização de segunda ordem no espaço



Atividade 3: O Sismograma - Aquisição de Spread Fixo

» Campo de velocidades constante e igual a 1500 m/s
» Propagação do campo de ondas considerando uma fonte sísmica
» Registro do campo de ondas em receptores expalhados ao redor da localização da fonte sísmica

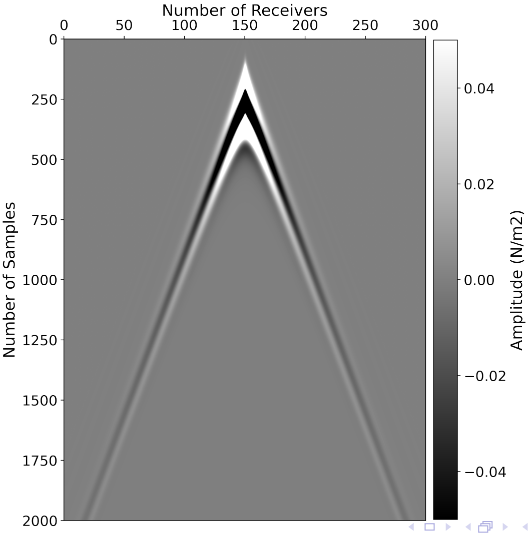


Atividade 3: O Sismograma - Aquisição de Spread Fixo

```
c = (-5.0/2.0, 4.0/3.0, -1.0/12.0)      # coeficiente de diferenças finitas

for n in range(0, Ntotal):
    #kernel da equação da onda sísmica
    for i in range(2, dimZ - 2):
        for k in range(2, dimX - 2):
            wavefield03[i,k] = C[i,k] * (2.0*c[0]*wavefield02[i, k]
                                + c[1]*(wavefield02[i,k-1] + wavefield02[i,k+1]
                                      + wavefield02[i-1,k] + wavefield02[i+1,k])
                                + c[2]*(wavefield02[i,k-2] + wavefield02[i,k+2]
                                      + wavefield02[i-2,k] + wavefield02[i+2,k]))
                                      + 2.0*wavefield02[i,k] - wavefield01[i,k]

    wavefield02[shot_z,shot_x] = wavefield02[shot_z,shot_x]
                                 + fonte[n]*(velocity[shot_z,shot_x]**2)

    for irec in range (dimX):
            seismogram[n, irec] = wavefield03[rec_z,irec]

    # Atualização do campo de onda
    wavefield01 = np.copy(wavefield02)
    wavefield02 = np.copy(wavefield03)
```

Atividade 3: O Sismograma - Aquisição de Spread Fixo

- Localização da fonte sísmica: [shot_x, shot_z] = [150, 5]
- Localização do receptor: rec_x = [0, dimX], rec_z = 5



Exercícios

1. Utilizar o código de propagação da onda sísmica para simular a propagação do campo de ondas em um modelo de velocidades de camadas paralelas.

2. Utilizar o código de propagação da onda sísmica para simular uma aquisição do tipo split-spread. Utilize o modelo de velocidades de camadas paralelas.

3. Utilizar o código de propagação da onda sísmica para simular uma aquisição do tipo end-on. Utilize o modelo de velocidades de camadas paralelas.

4. Utilizar o código de propagação da onda sísmica para simular uma aquisição do tipo Ocean Bottom Cable (OBC). Utilize o modelo de velocidades de camadas paralelas.