SLIM 2014
Consortium

2D/3D time stepping for imaging and inversion

Xiang Li

SLIM

University of British Columbia

Motivation

- speed and accuracy
- preprocessing on the time axis (muting, windowing, AGC, etc)
- mathematic principles (adjoint test, gradient test, etc)
- easy to read and update
- "memory free" (using backward modeling for background wavefield)
- Matlab (easy for sponsors to use, test and modify)

Wave equation in time domain

continuous form

$$\frac{\partial^2 f(x,y,z,t)}{\partial^2 x^2} + \frac{\partial^2 f(x,y,z,t)}{\partial y^2} + \frac{\partial^2 f(x,y,z,t)}{\partial z^2} - \frac{1}{v^2}\frac{\partial^2 f(x,y,z,t)}{\partial t^2} = q(s_x, s_y, s_z, t)$$

linear algebra form

$$\mathbf{A}\mathbf{u} = \mathbf{q}$$

$\mathbf{A}$: time domain forward modeling matrix

$\mathbf{u}$: vectorized wave field of all time steps and modeling grids

$\mathbf{q}$: source

One time step of continuous form

SLIM

$$\frac{f(x-1,y,z,t_k)-2f(x,y,z,t_k)+f(x+1,y,z,t_k)}{\triangle x^2}+\frac{f(x,y-1,z,t_k)-2f(x,y,z,t_k)+f(x,y+1,z,t_k)}{\triangle y^2}$$
$$+\frac{f(x,y,z-1,t_k)-2f(x,y,z,t_k)+f(x,y,z+1,t_k)}{\triangle z^2}$$
$$-\frac{1}{v(x,y,z)^2}\frac{f(x,y,z,t_{k-1})-2f(x,y,z,t_k)+{\color{red}f(x,y,z,t_{k+1})}}{\triangle t^2}=q(s_x,s_y,s_z,t_k)$$

One time step of continuous form

$f(x-1, y, z,$ … $t_k) + f(x, y+1, z, t_k)$

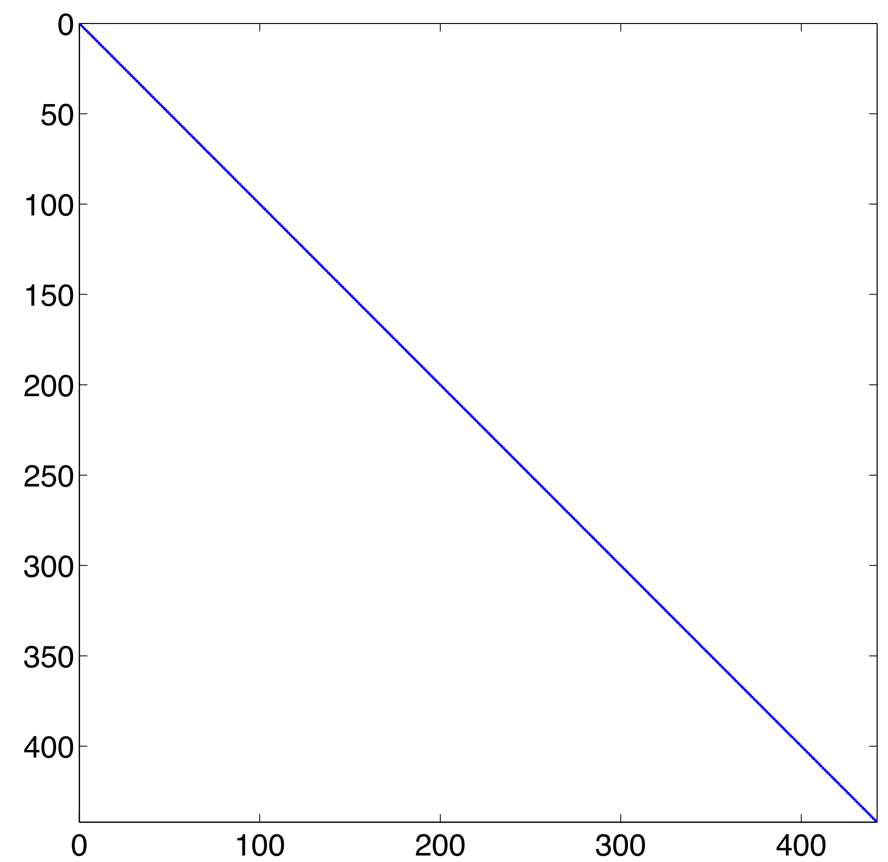
$t_k) + f(x, y, z+1, t_k)$

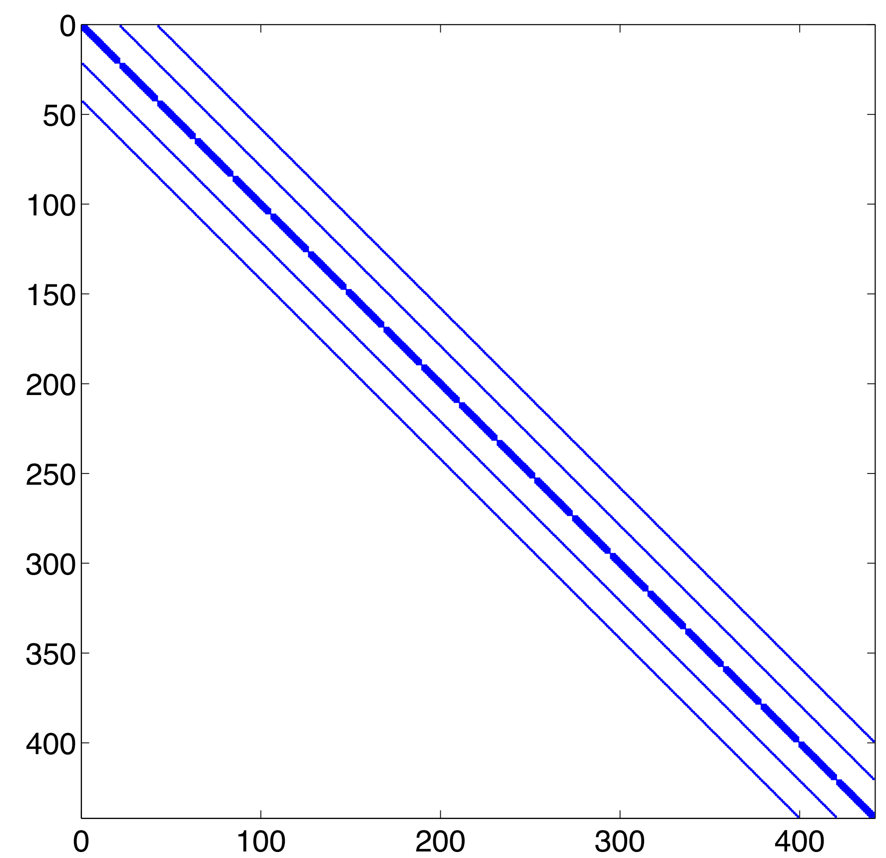
$_{k+1})$ $= q(s_x, s_y, s_z, t_k)$

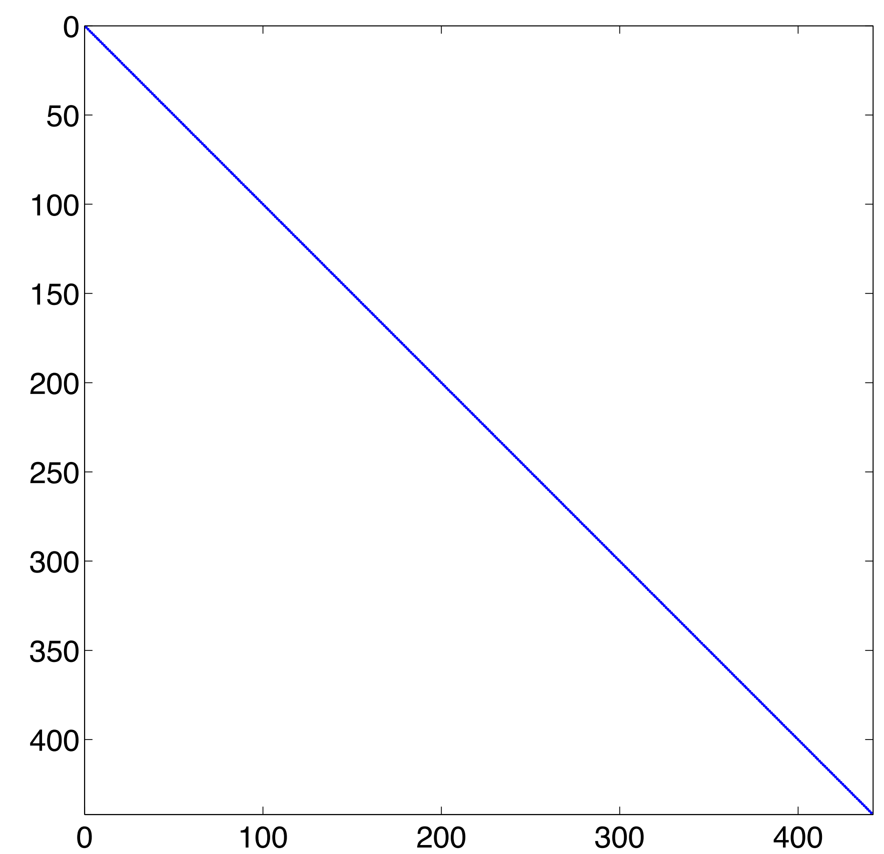
```
for l=1:lk-1%l: time loop; i: x dimension loop; j: z dimension loop
  clf;
    u(x0,z0,l)=wavelet(l);
    for i=3:lx-2%calculate inner wavefield
        for j=3:lz-2
            if (i==x0 && j==z0) %not calculate wavefield at the source position
                continue;
            end
             e=(v(i,j)*tx)^2;
              f=(v(i,j)*tz)^2;

             if(l==1)%Finite difference of wave equation
                 u(i,j,2)=(1/1)*(2*u(i,j,1)+e*(-1/12*(u(i+2,j,1)+u(i-2,j,1))+4/3*(u(i+1,j,1)+...
                     u(i-1,j,1))-2.5*u(i,j,1))+f*(-1/12*(u(i,j+2,1)+u(i,j-2,1))+4/3*(u(i,j+1,1)+...
                     u(i,j-1,1))-2.5*u(i,j,1)));
             else
                u(i,j,l+1)=(1/1)*(2*u(i,j,l)-u(i,j,l-1)+e*(-1/12*(u(i+2,j,l)+u(i-2,j,l))+...
                     4/3*(u(i+1,j,l)+u(i-1,j,l))-2.5*u(i,j,l))+f*(-1/12*(u(i,j+2,l)+...
                    u(i,j-2,l))+4/3*(u(i,j+1,l)+u(i,j-1,l))-2.5*u(i,j,l)));%Wu lv's
             end
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%    Calculate boundary when l=1 j=2b                                %%%%%%%%%
%%%%    Attention: Because the code is scend-order in time;             %%%%%%%%%
%%%%    So we have to know the wavefield value at the 'l-1' where       %%%%%%%%%
%%%%    was set zeros in advance, so we have to calculate the          %%%%%%%%%
%%%%    boundarys of j = 1 and j = 2 separately when l=1                %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(l==1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% boundary when l=1,j=2 (already pass the test)
    for i=2:lx-1
        if(x0==i && z0==2)   continue;
        end
        u(i,2,2)=2*u(i,2,1)+(v(i,2)*tx)^2*(u(i+1,2,1)-2*u(i,2,1)+u(i-1,2,1))+...
           (v(i,2)*tz)^2*(u(i,2+1,1)-2*u(i,2,1)+u(i,2-1,1));%Wu linqin's
    end
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

One time step of linear algebra form

$$\mathbf{A}_1\mathbf{u}^{t-1} + \mathbf{A}_2\mathbf{u}^{t} + \mathbf{A}_3\mathbf{u}^{t+1} = \mathbf{q}^{t}$$







$$\mathbf{A_1} = \mathrm{diag}(\frac{1}{\mathbf{v}^2\triangle t^2}) \qquad \mathbf{A_2} = \nabla^2 - 2\mathrm{diag}(\frac{1}{\mathbf{v}^2\triangle t^2}) \qquad \mathbf{A_3} = \mathrm{diag}(\frac{1}{\mathbf{v}^2\triangle t^2})$$

One time step of linear algebra form

$$\mathbf{A}_1\mathbf{u}^{t-1} + \mathbf{A}_2\mathbf{u}^{t} + \mathbf{A}_3\mathbf{u}^{t+1} = \mathbf{q}^{t}$$

```
% ---------------------------- loop over time step ----------------------------
for ti = 1:Num_of_time_step

    %  forward modeling solve for A U  = Q
    %
    % [ A1                           ]  U1    Q1
    % | A2 A1                        |  U2    Q2
    % | A3 A2 A1                     |  U3  = Q3
    % |              . . .           |  U4    Q4
    % [                    A3 A2 A1]  U5    Q5

    U3  = Ps * Q(:,ti) - A2 * U2(:) - A1 * U1(:);

    U3  = A3_inv * U3(:);
```

Time-stepping method for wave equation

one time step

$$\mathbf{A}_1\mathbf{u}^{t-1} + \mathbf{A}_2\mathbf{u}^{t} + \mathbf{A}_3\mathbf{u}^{t+1} = \mathbf{q}^{t}$$

all time steps

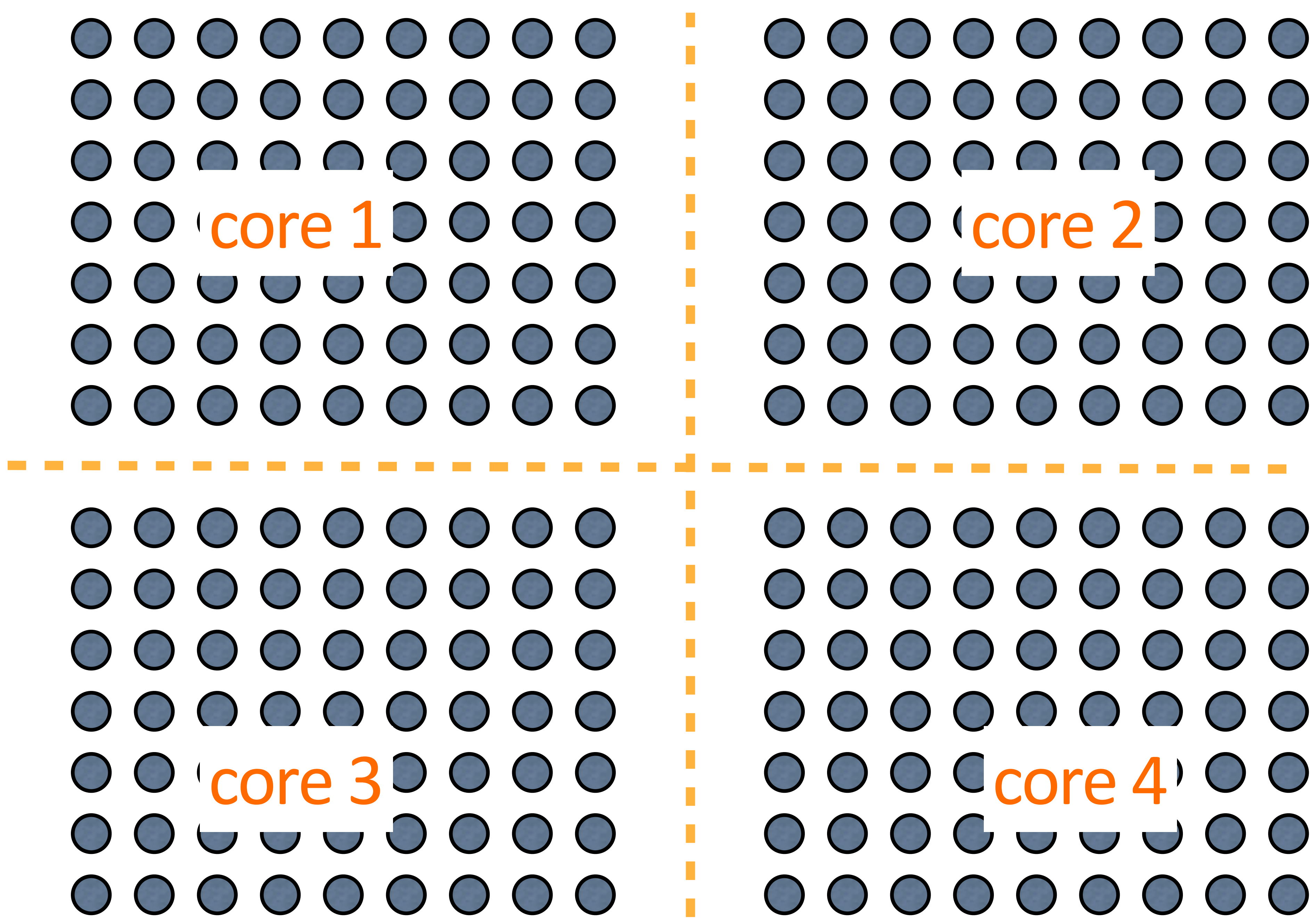
like Gaussian elimination

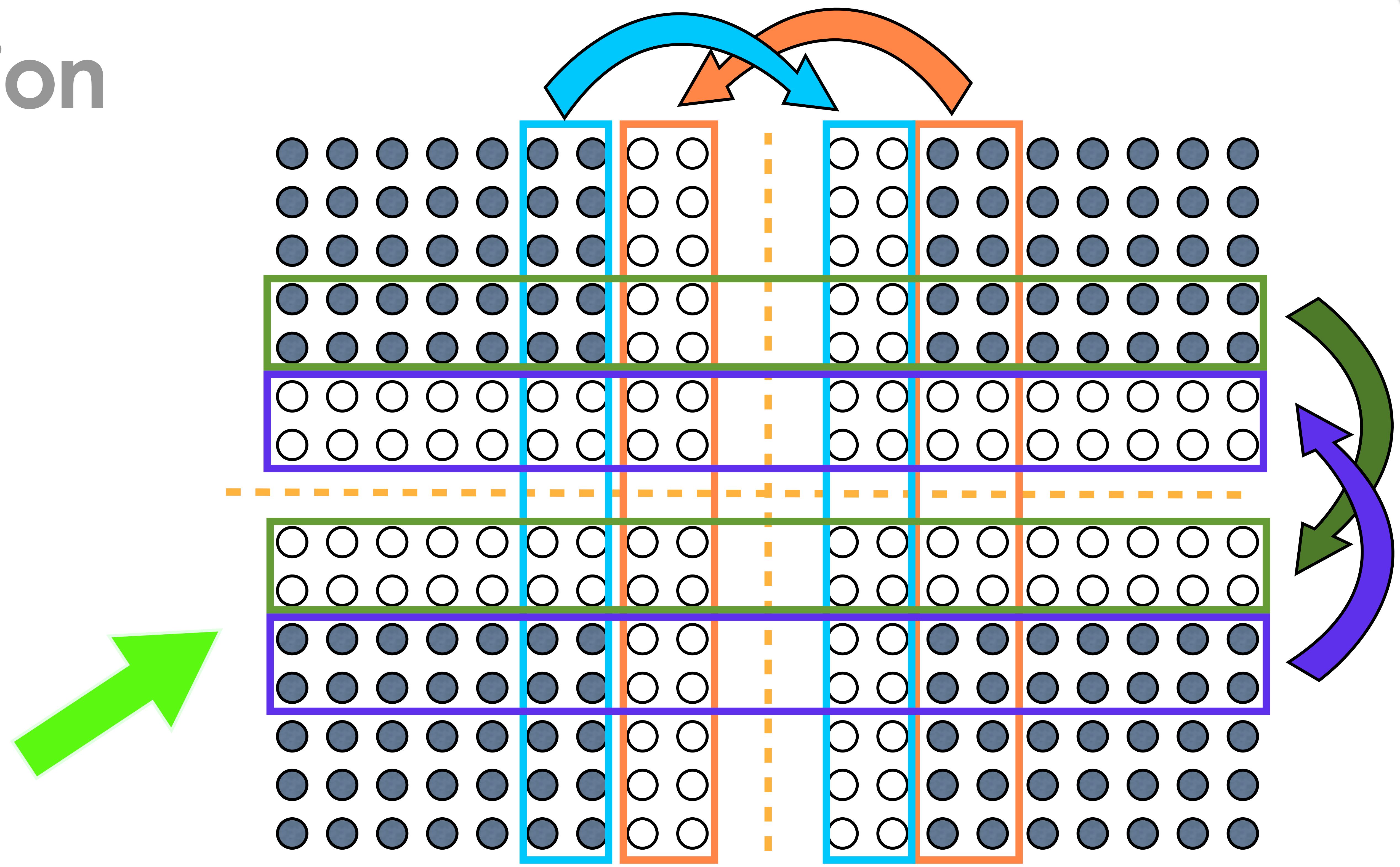
$$\begin{array}{cc}
 & \begin{array}{c} \mathbf{u}^{-1} \\ \mathbf{u}^{0} \end{array} \\
\begin{array}{r} \mathbf{A}_1 \; \mathbf{A}_2 \\ \mathbf{A}_1 \\ \\ \\ \\ \\ \\ \end{array}
\begin{pmatrix}
\mathbf{A}_3 & & & & & & \\
\mathbf{A}_2 & \mathbf{A}_3 & & & & & \\
\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & & & \\
 & \ddots & \ddots & \ddots & & & \\
 & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3
\end{pmatrix}
&
\begin{pmatrix}
\mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \\ \vdots \\ \mathbf{u}^t \\ \vdots \\ \mathbf{u}^{Nt}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{q}^1 \\ \mathbf{q}^2 \\ \mathbf{q}^3 \\ \vdots \\ \mathbf{q}^t \\ \vdots \\ \mathbf{q}^{Nt}
\end{pmatrix}
\end{array}$$

$$\mathbf{A}\mathbf{u} = \mathbf{q}$$

Domain decomposition

wavefield @ $t-1$ & t

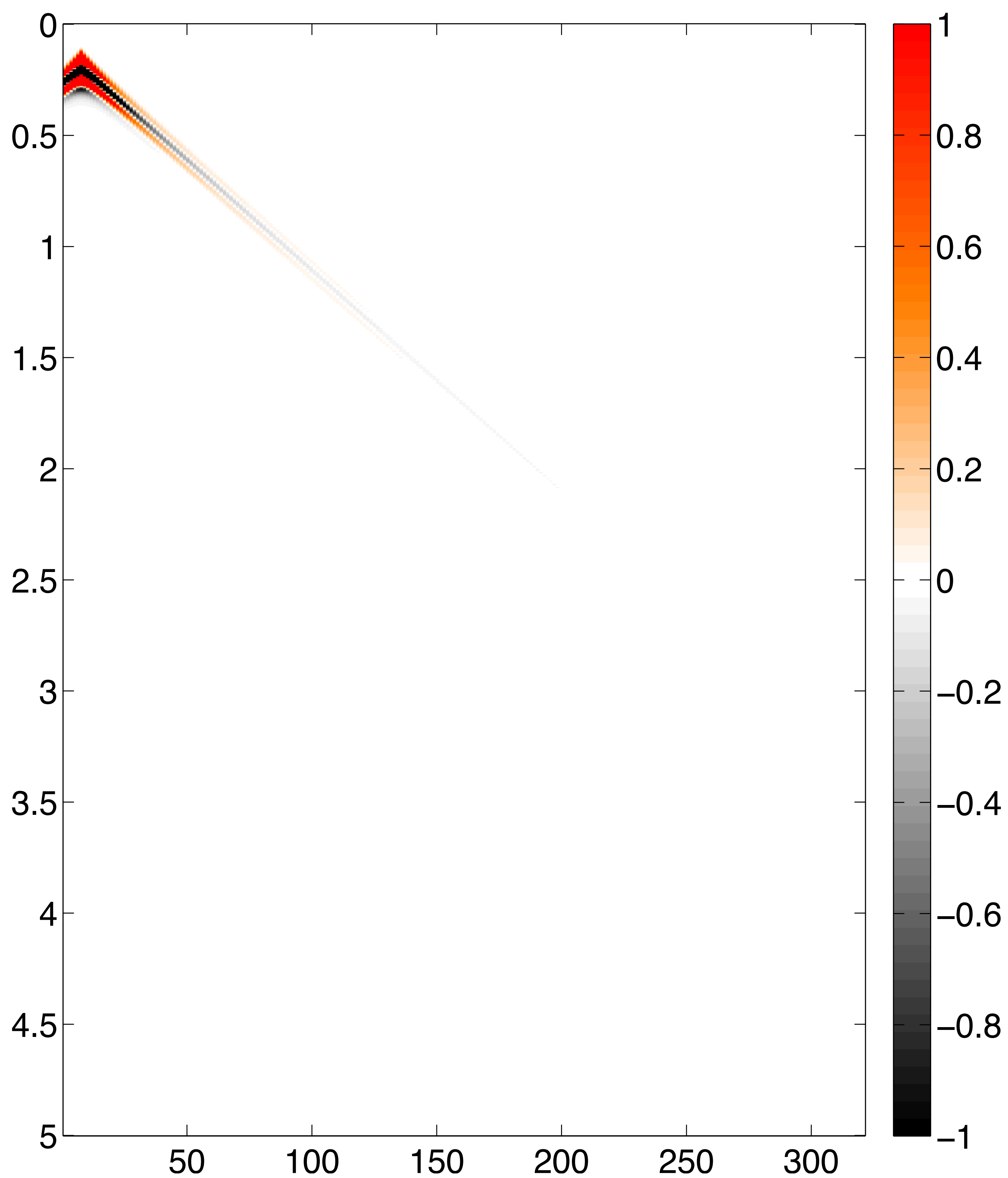




wavefield @ $t+1$

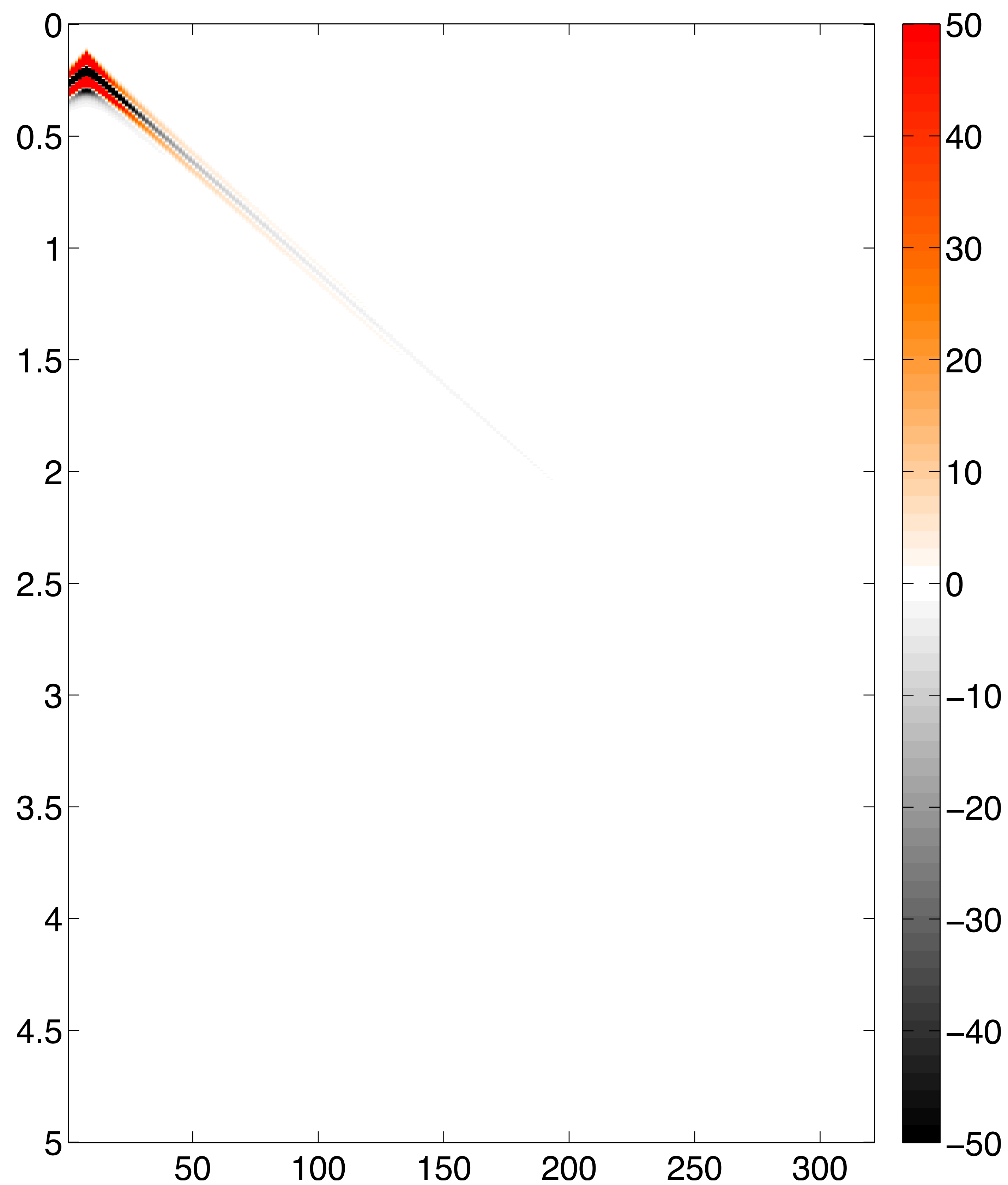
Comparison with iWAVE

SLIM



Elapsed time is 38 seconds.

IWAVE



Elapsed time is 42 seconds.

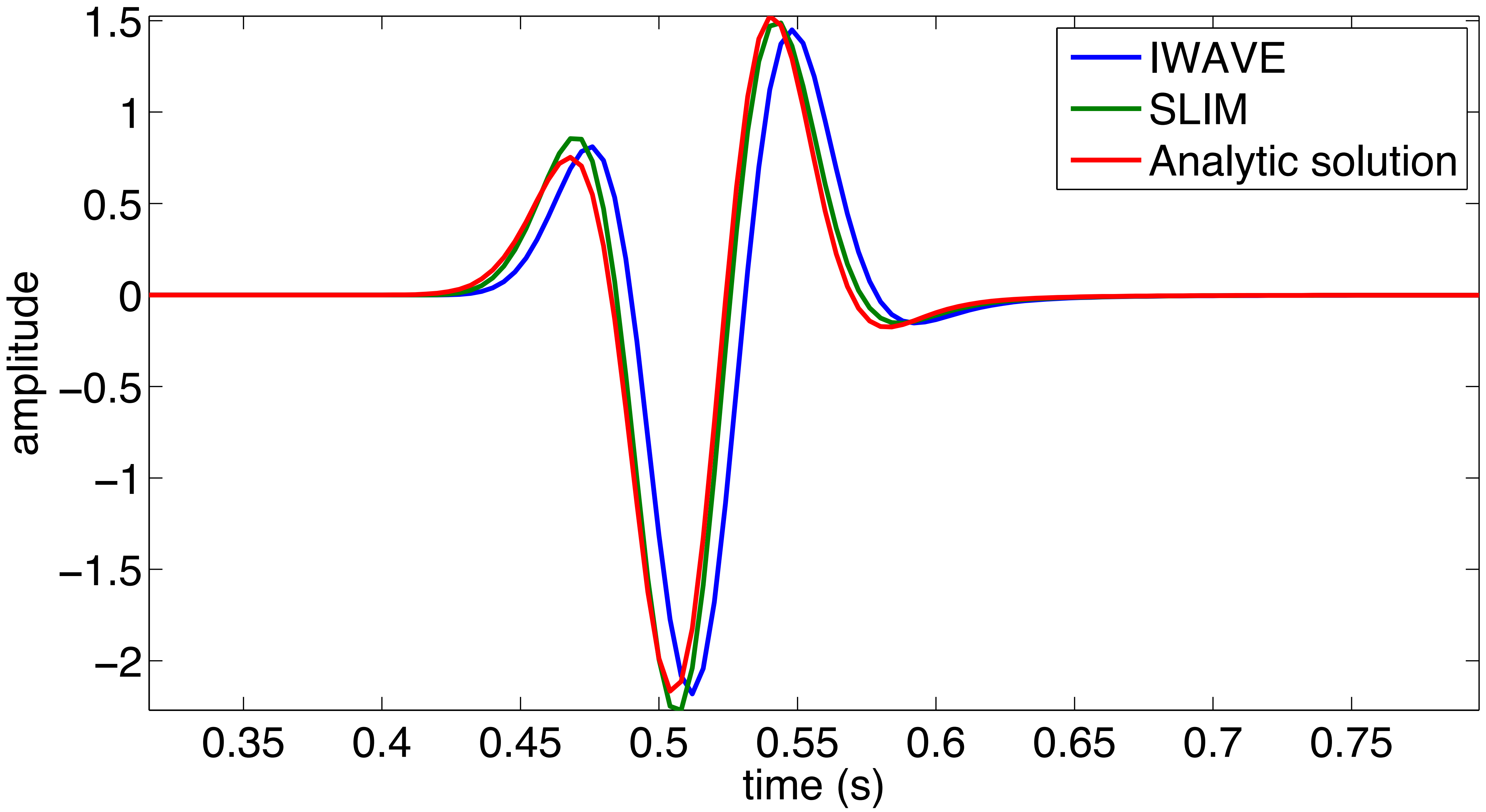
constant velocity model (2000m/s), size of which is 501x501

1000 time samples with 2s length.

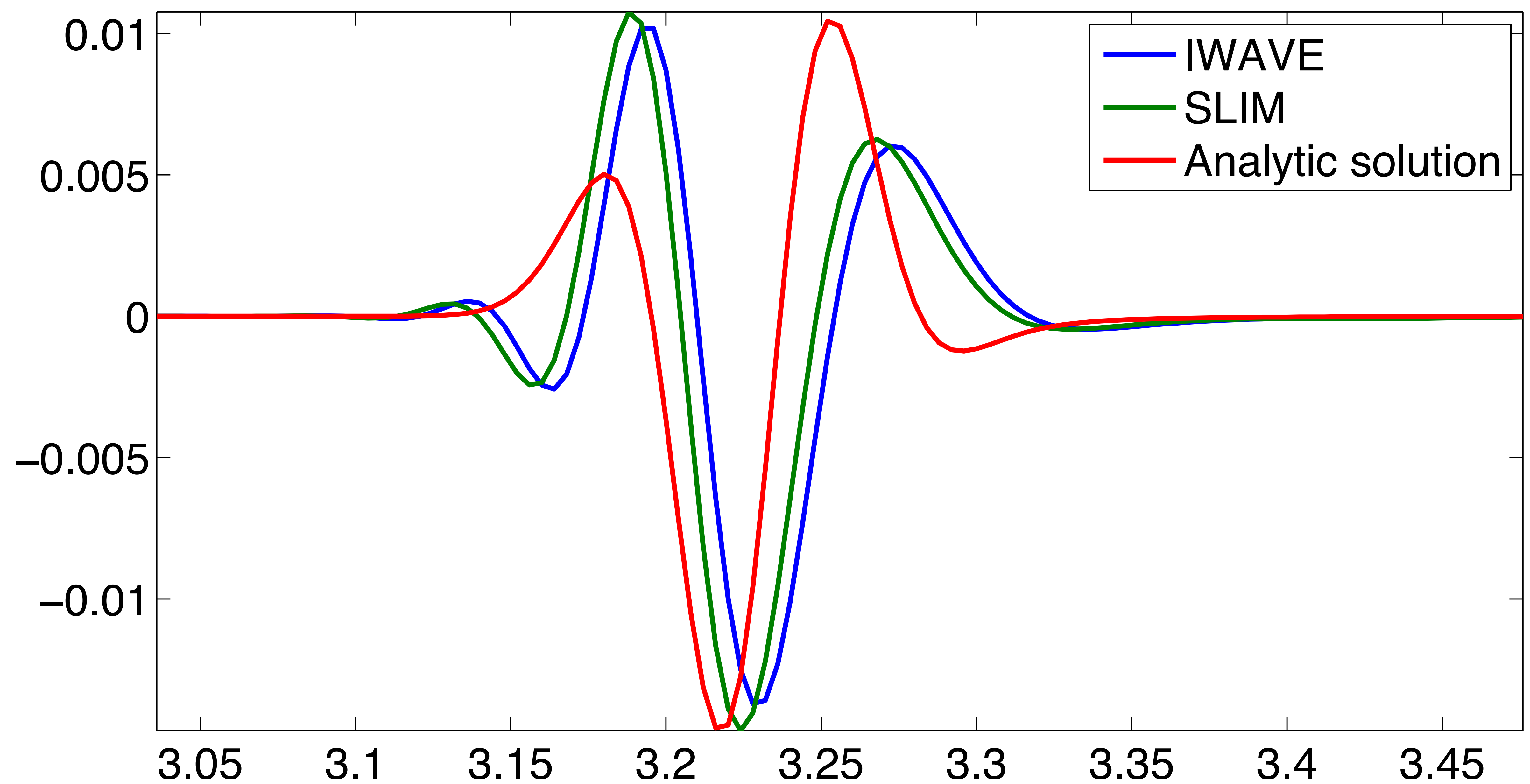
with free surface.

iWave is a C++ framework for wave simulation from Rice University [William Symes 2011]

Trace to trace



trace 30



trace 235

Jacobian of FWI objective and its adjoint

FWI objective function

$$\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} \| \underbrace{\mathbf{d}_i - \mathbf{P}_r \mathbf{A}(\mathbf{m})^{-1} \mathbf{q}_i}_{\delta \mathbf{d}_i} \|_2^2$$

$\mathbf{d}_i$: one observed shot record

$\mathbf{P}_r$: receiver restriction operator

$\mathbf{m}$: model slowness

gradient of FWI objective (action of Jacobian adjoint or RTM operator)

$$\nabla \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[- \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1} \mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$

Born modeling (action of Jacobian or Born modeling operator)

$$\widehat{\delta \mathbf{d}} = \mathbf{vec} \left[-\mathbf{P}_r \mathbf{A}^{-1} \left[\mathrm{diag} [\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1} \mathbf{Q})] (\mathbf{I}_{nt} \otimes \delta \mathbf{m}) \right] \right] = \mathbf{J} \delta \mathbf{m}$$

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{ns}]$$

SLIM

Jacobian of FWI objective and its a

$\mathbf{J}$ and $\mathbf{J}^{\mathrm{T}}$

need to pass adjoint test !!!

$$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} > = < \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >$$

FWI objective function

$$\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} \| \underbrace{\mathbf{d}_i - \mathbf{P}_r \mathbf{A}(\mathbf{m})^{-1} \mathbf{q}_i}_{\delta\mathbf{d}_i} \|_2^2$$

$\mathbf{d}_i$:

$\mathbf{P}_r$:

$\mathbf{m}$:

gradient of FWI objective (action of Jacobian adjoint or RTM operator)

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[-\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta\mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta\mathbf{d}$$

Born modeling (action of Jacobian or Born modeling operator)

$$\widehat{\delta\mathbf{d}} = \mathbf{vec} \left[-\mathbf{P}_r \mathbf{A}^{-1} \left[\mathrm{diag}[\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{Q})] (\mathbf{I}_{nt} \otimes \delta\mathbf{m}) \right] \right] = \mathbf{J}\delta\mathbf{m}$$

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{ns}]$$

Importance of adjoint test

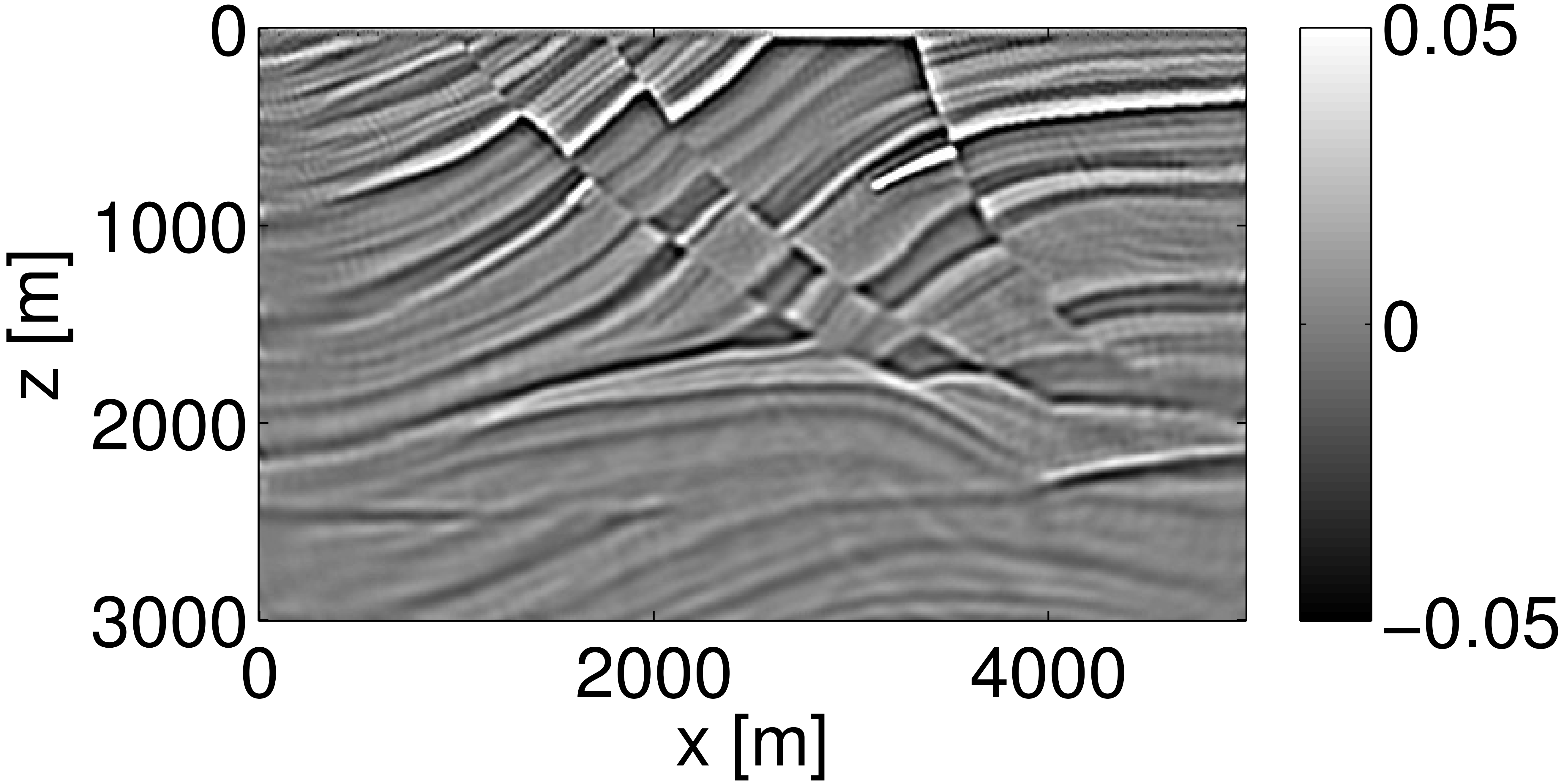
least-squares migration

$$\delta\mathbf{m} = \arg\min_{\delta\mathbf{m}} \|\delta\mathbf{d} - \mathbf{J}\delta\mathbf{m}\|_2^2$$

analytic solution

$$\delta\mathbf{m} = (\underbrace{\mathbf{J}^{\mathrm{T}}\mathbf{J}}_{\text{Gauss-Newton Hessian of FWI objective}})^{-1}\underbrace{\mathbf{J}^{\mathrm{T}}\delta\mathbf{d}}_{\text{RTM}}$$

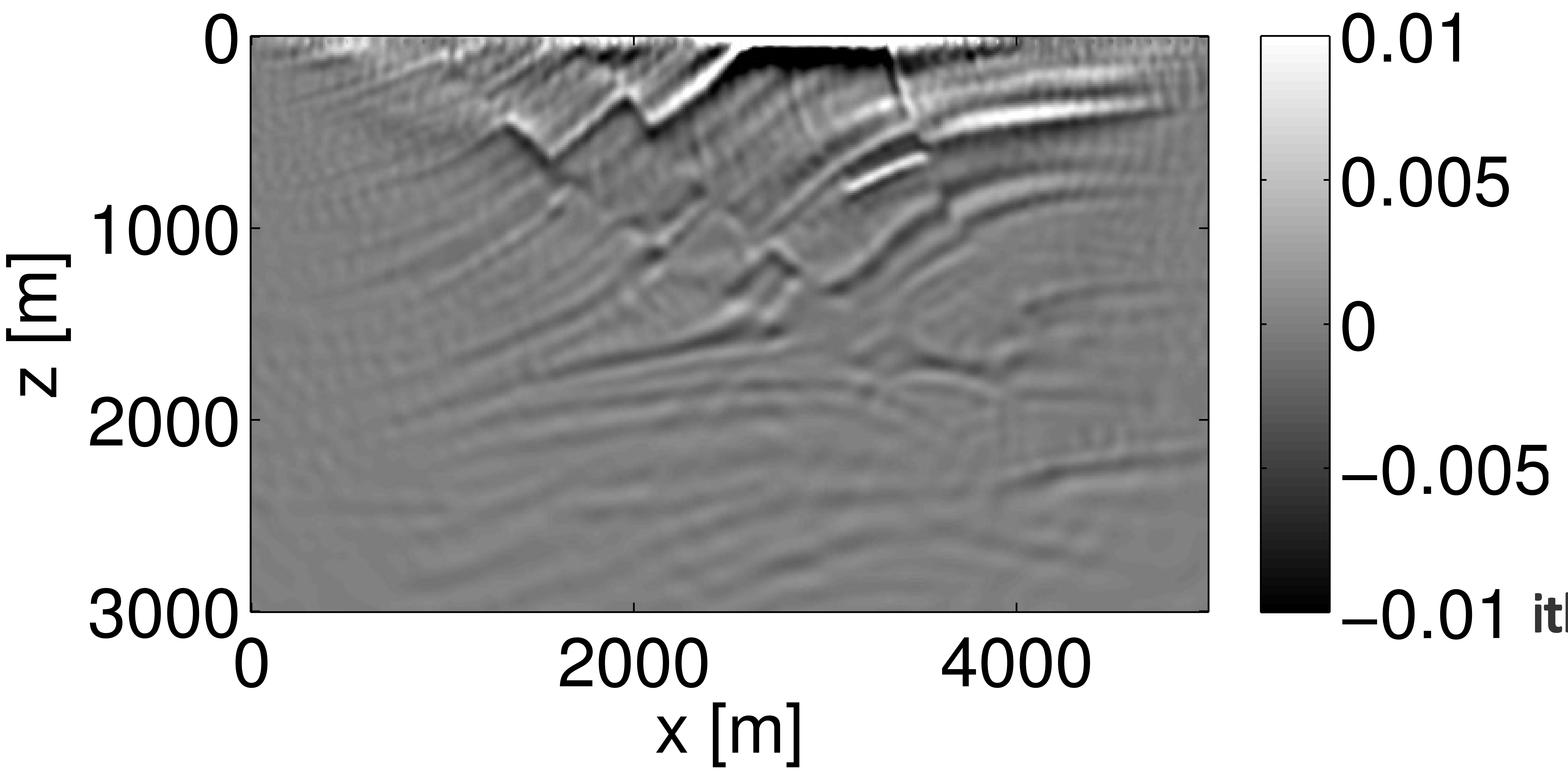
SLIM



4-25 Hz with 0.5 Hz spacing
50 LSQR iterations

perfectly pass adjoint test

$$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} > / < \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >= 1$$



itle

almost pass adjoint test

$$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} > / < \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >= 1.01$$

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[-\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta\mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}}\delta\mathbf{d}$$

Tristan van Leeuwen 2012

Gradient of FWI objective (RTM)

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[-\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$

forward modeling

$$\begin{pmatrix} \mathbf{A}_3 & & & & & & \\ \mathbf{A}_2 & \mathbf{A}_3 & & & & & \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \\ \vdots \\ \mathbf{u}^t \\ \vdots \\ \mathbf{u}^{Nt} \end{pmatrix} = \begin{pmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \mathbf{q}^3 \\ \vdots \\ \mathbf{q}^t \\ \vdots \\ \mathbf{q}^{Nt} \end{pmatrix}$$

SLIM

Gradient of FWI objective (RTM)

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[- \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$

forward modeling

adjoint modeling

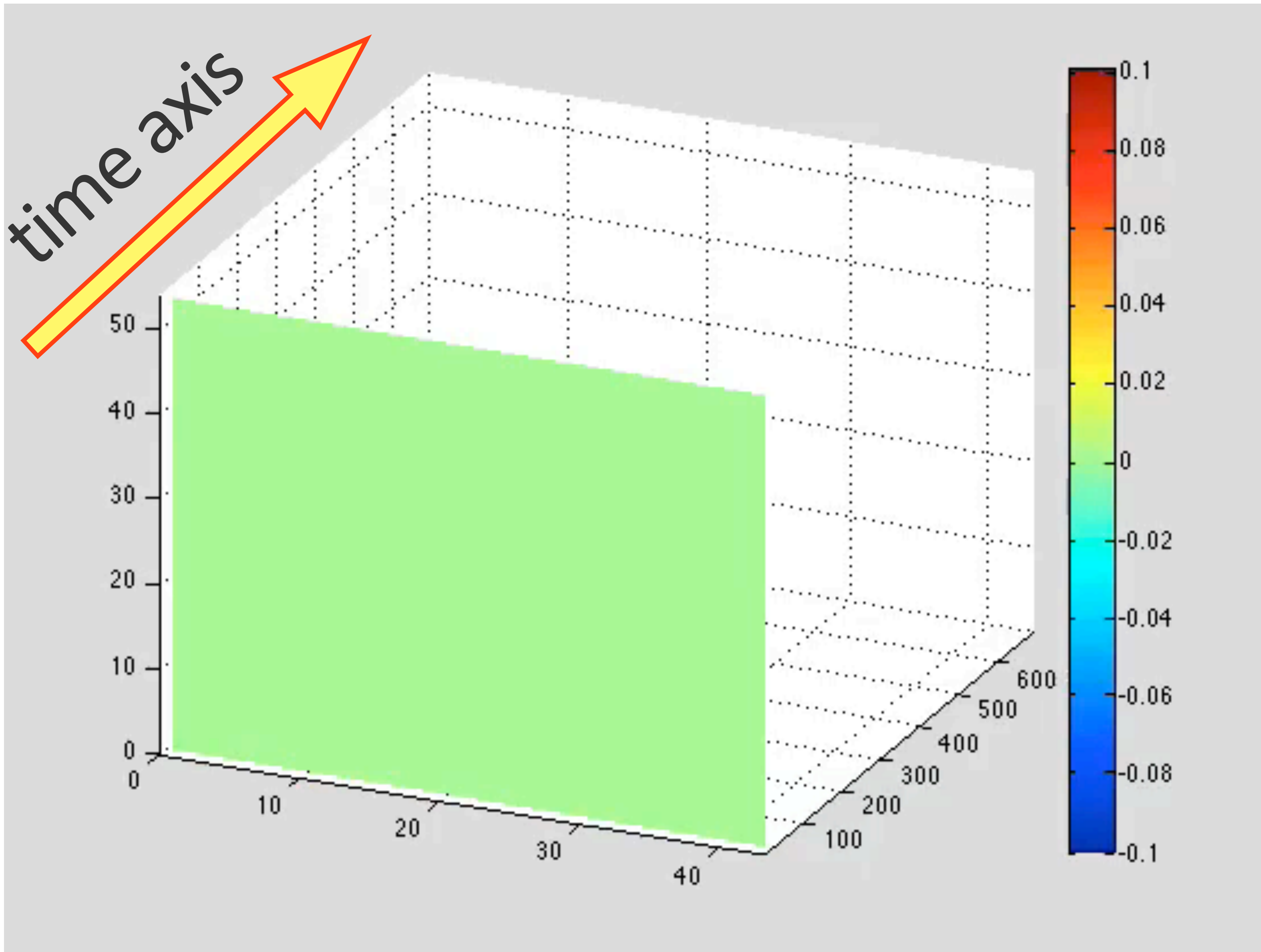
$$\begin{pmatrix} \mathbf{A}_3 & & & & & & \\ \mathbf{A}_2 & \mathbf{A}_3 & & & & & \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \\ \vdots \\ \mathbf{u}^t \\ \vdots \\ \mathbf{u}^{Nt} \end{pmatrix} = \begin{pmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \mathbf{q}^3 \\ \vdots \\ \mathbf{q}^t \\ \vdots \\ \mathbf{q}^{Nt} \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{A}_3^{\mathrm{T}} & \mathbf{A}_2^{\mathrm{T}} & \mathbf{A}_1^{\mathrm{T}} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{A}_3^{\mathrm{T}} & \mathbf{A}_2^{\mathrm{T}} & \mathbf{A}_1^{\mathrm{T}} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_3^{\mathrm{T}} & \mathbf{A}_2^{\mathrm{T}} & \mathbf{A}_1^{\mathrm{T}} \\ & & & & & \mathbf{A}_3^{\mathrm{T}} & \mathbf{A}_2^{\mathrm{T}} \\ & & & & & & \mathbf{A}_3^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{v}^1 \\ \mathbf{v}^2 \\ \mathbf{v}^3 \\ \vdots \\ \mathbf{v}^t \\ \vdots \\ \mathbf{v}^{Nt} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_r^{\mathrm{T}}\delta\mathbf{d}^1 \\ \mathbf{P}_r^{\mathrm{T}}\delta\mathbf{d}^2 \\ \mathbf{P}_r^{\mathrm{T}}\delta\mathbf{d}^3 \\ \vdots \\ \mathbf{P}_r^{\mathrm{T}}\delta\mathbf{d}^t \\ \vdots \\ \mathbf{P}_r^{\mathrm{T}}\delta\mathbf{d}^{Nt} \end{pmatrix}$$

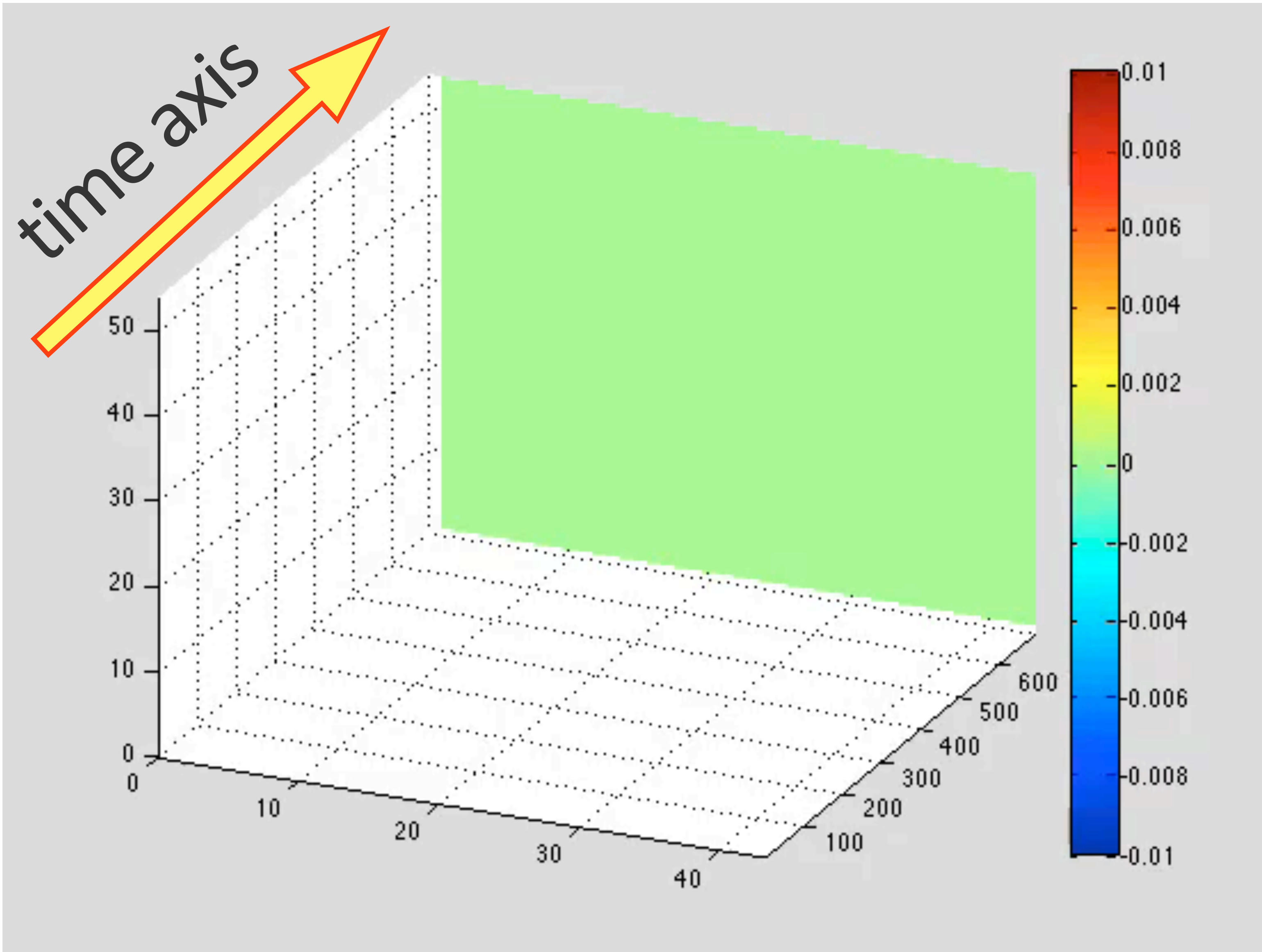
Gradient of FWI objective (RTM)

$$\nabla \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[- \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1} \mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$

forward modeling



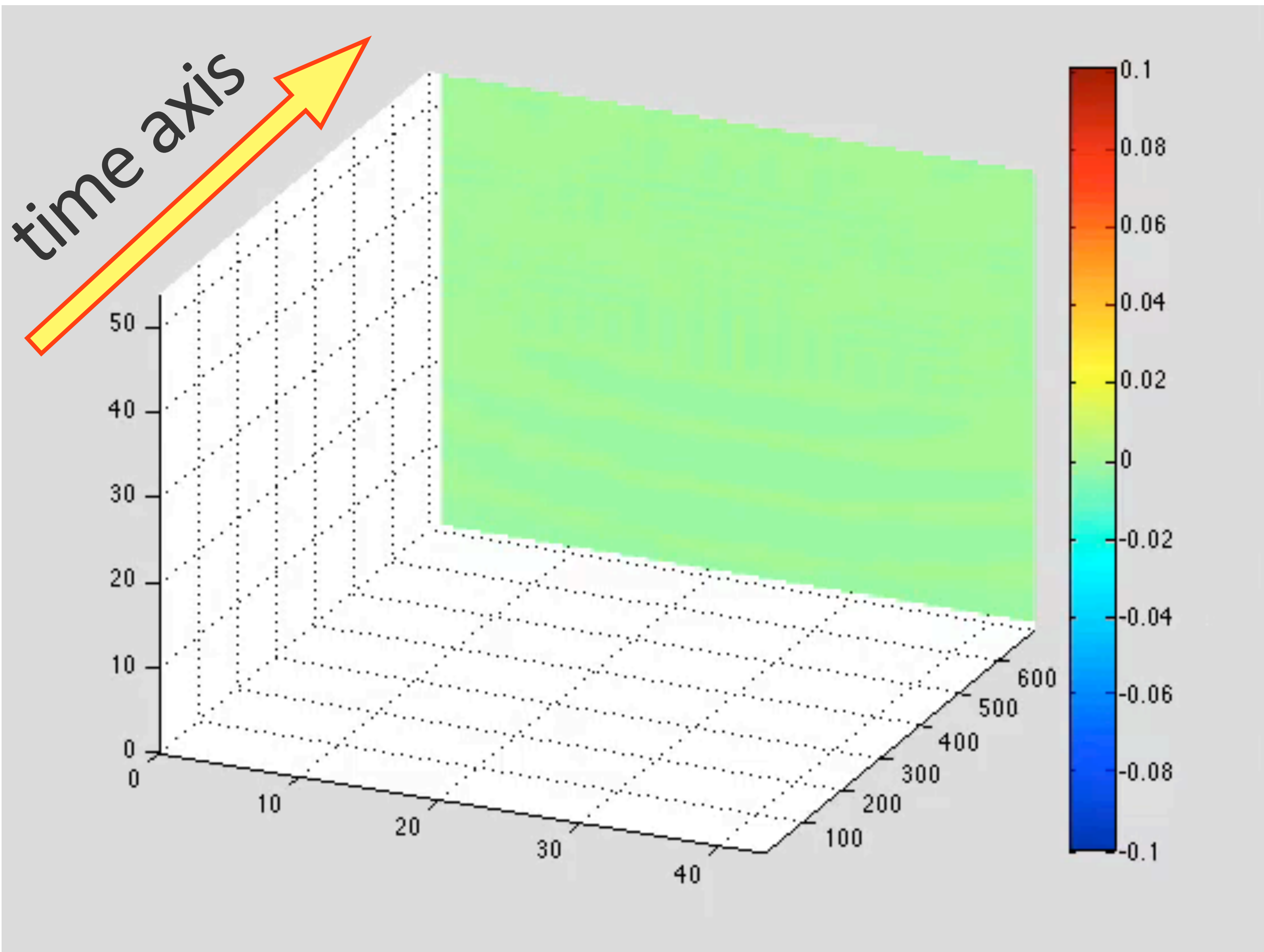
adjoint modeling



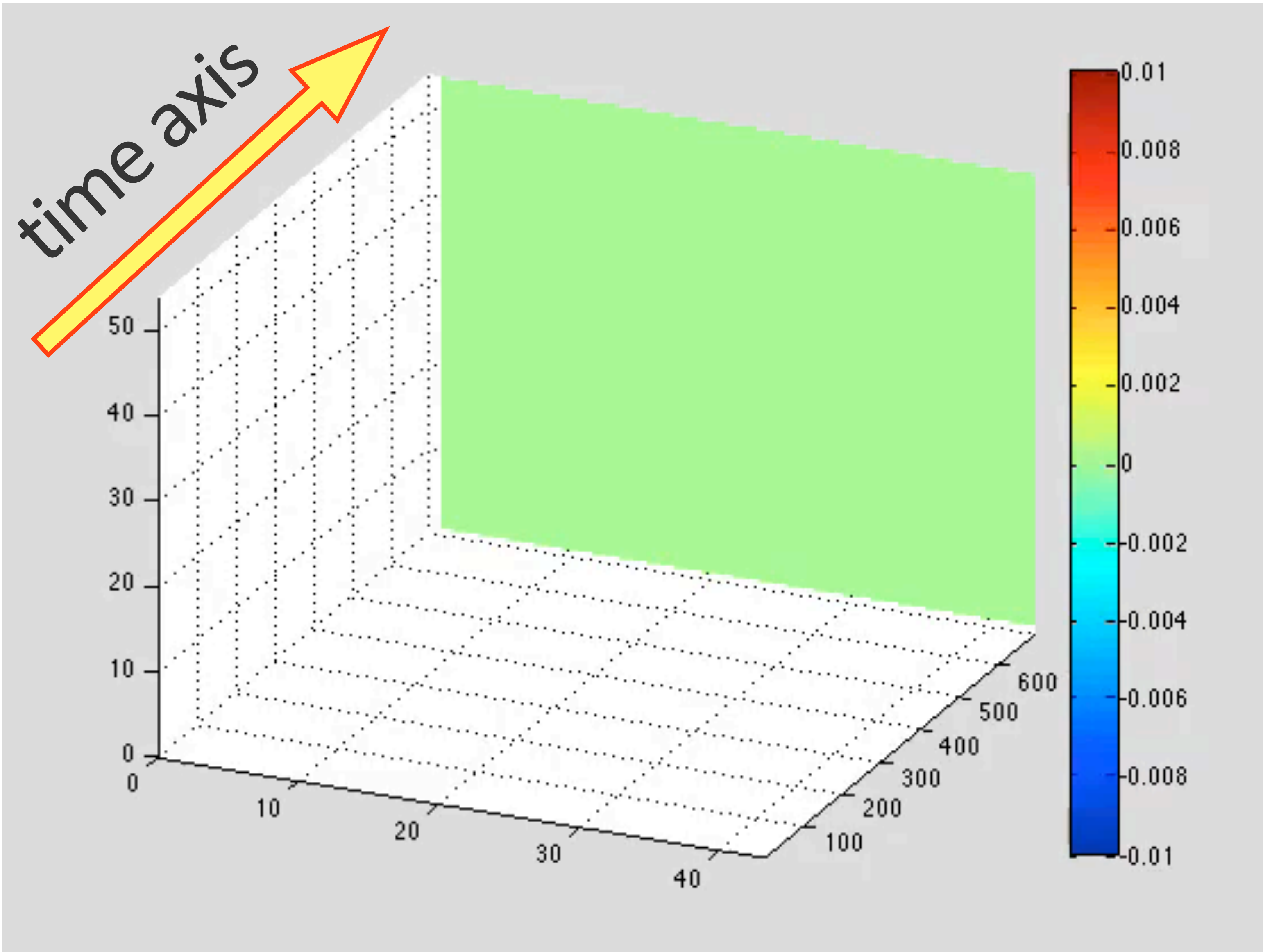
Gradient of FWI objective (RTM)

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[-\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$

backward modeling



adjoint modeling



Backward modeling with checkpoints

one time step of *forward modeling*

$$\mathbf{A}_1\mathbf{u}^{t-1} + \mathbf{A}_2\mathbf{u}^{t} + \mathbf{A}_3\mathbf{u}^{t+1} = \mathbf{q}^{t}$$

one time step of *backward modeling*

$$\mathbf{A}_1\mathbf{u}^{t-1} + \mathbf{A}_2\mathbf{u}^{t} + \mathbf{A}_3\mathbf{u}^{t+1} = \mathbf{q}^{t}$$

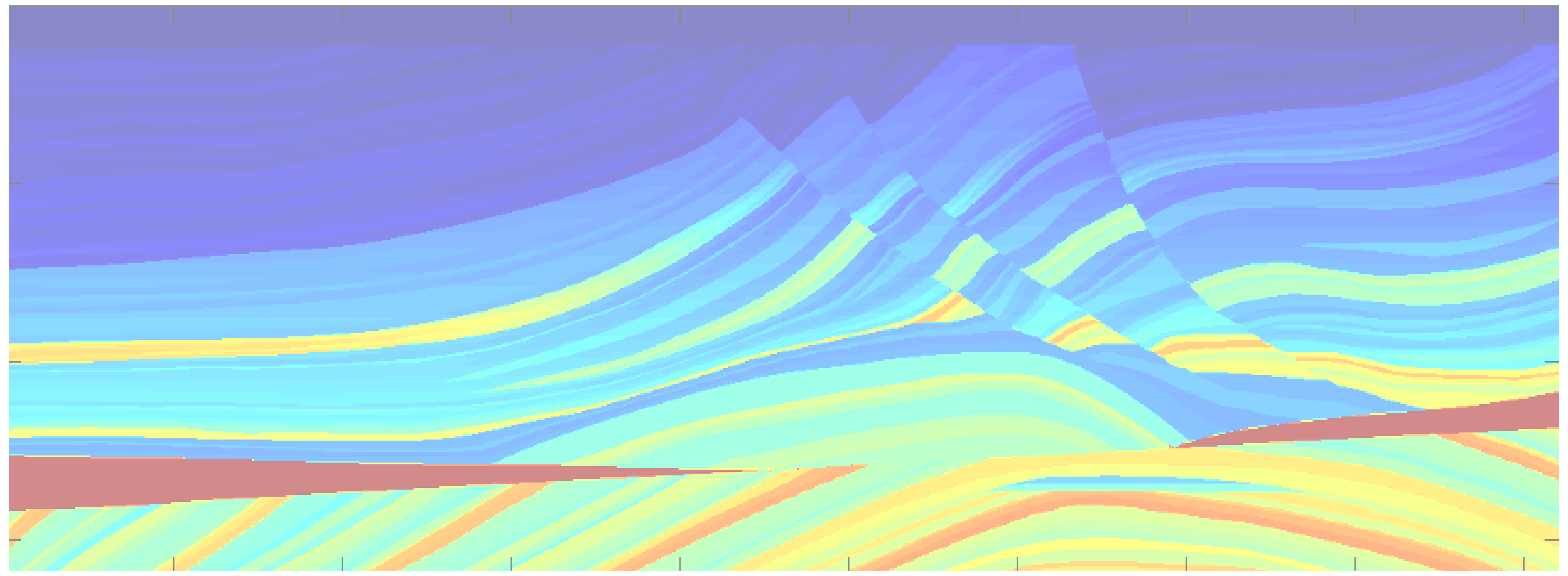
solve the ***EXACT*** forward wave equation

$$\begin{pmatrix} \mathbf{A}_3 & & & & & & \\ \mathbf{A}_2 & \mathbf{A}_3 & & & & & \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \\ \vdots \\ \mathbf{u}^t \\ \vdots \\ \mathbf{u}^{Nt-1} \\ \mathbf{u}^{Nt} \end{pmatrix} = \begin{pmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \mathbf{q}^3 \\ \vdots \\ \mathbf{q}^t \\ \vdots \\ \mathbf{q}^{Nt} \end{pmatrix}$$

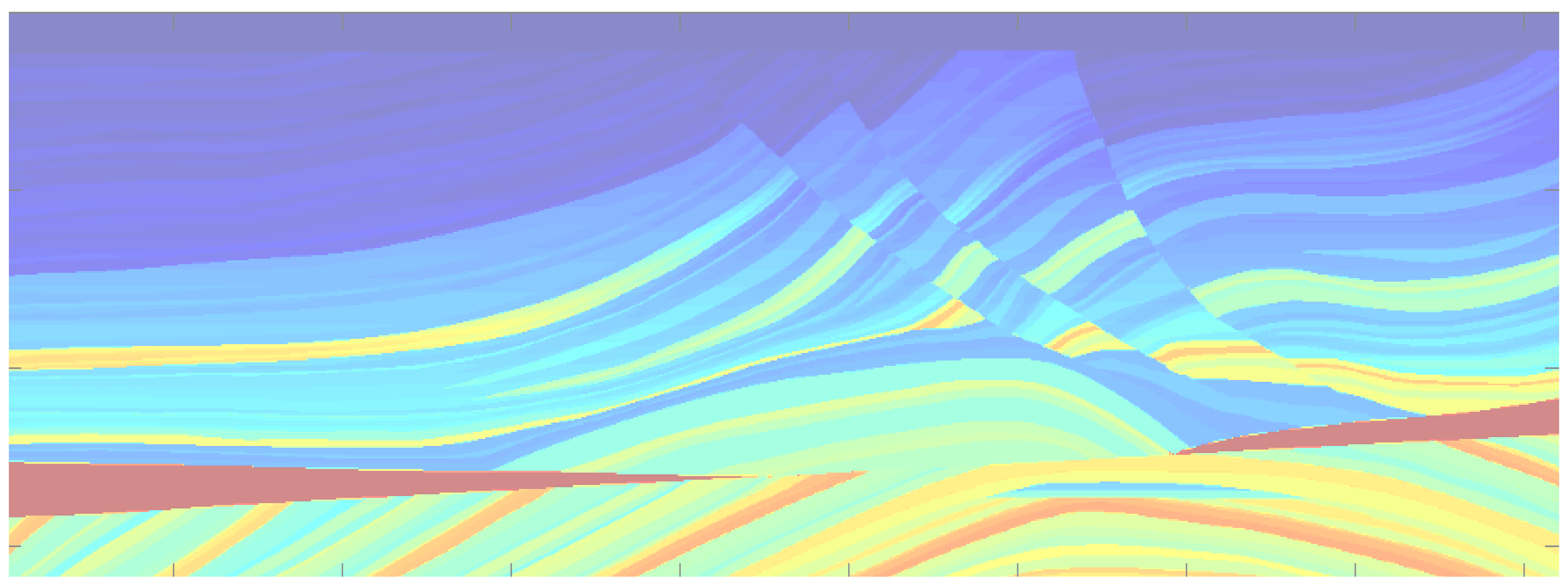
save the wavefield at the last two time steps

Yder Massonand, etc; 2013
Eric Dussaud, etc; 2008

Numerical experiment



forward modeling



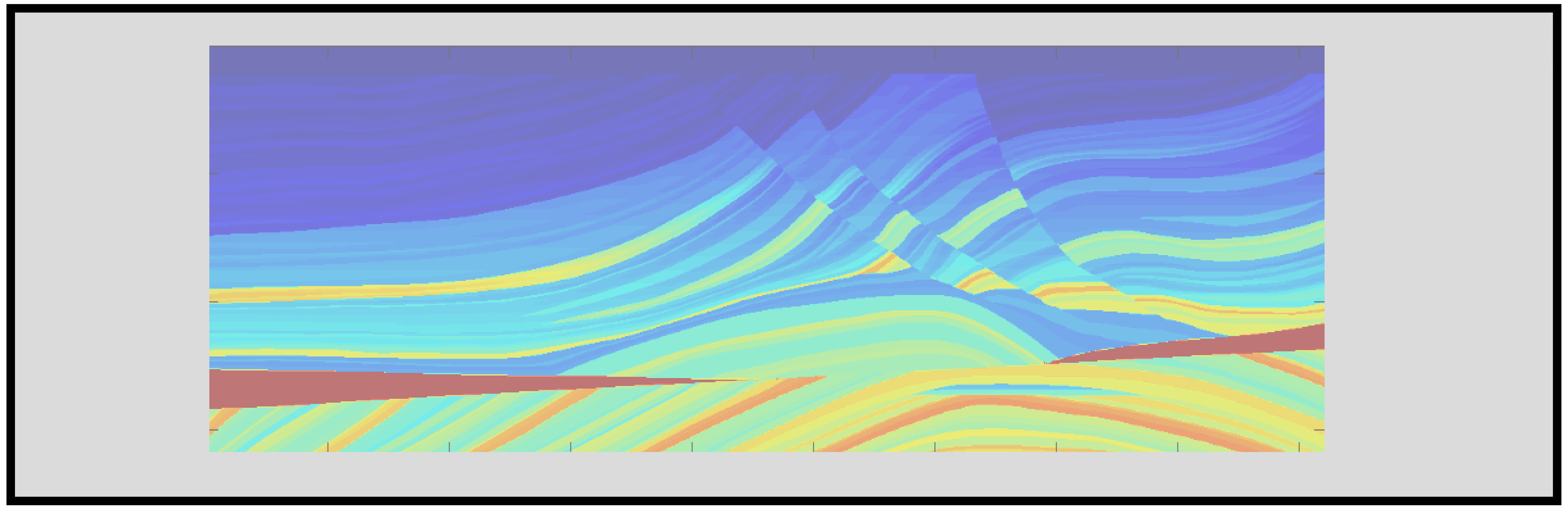
backward modeling

This example uses Marmousi velocity model, the size of which is 319x921, 3000 time samples with 3s length.

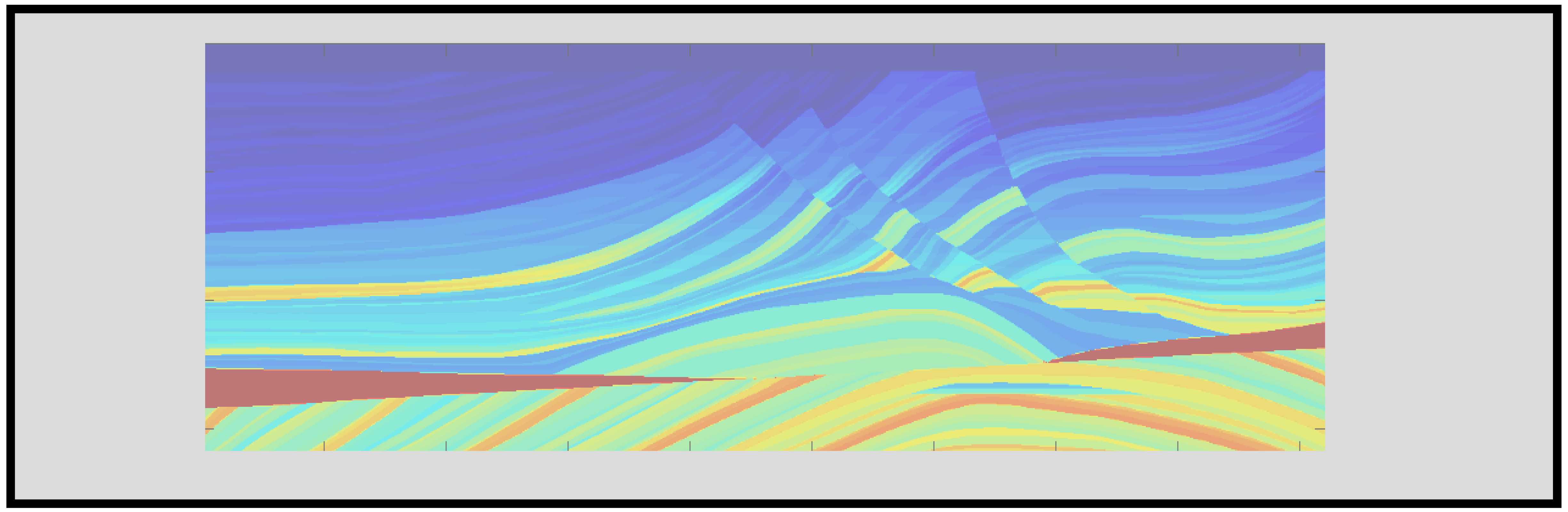
with free surface.

Video time is the actuarial *modeling time + plotting time*

Numerical experiment



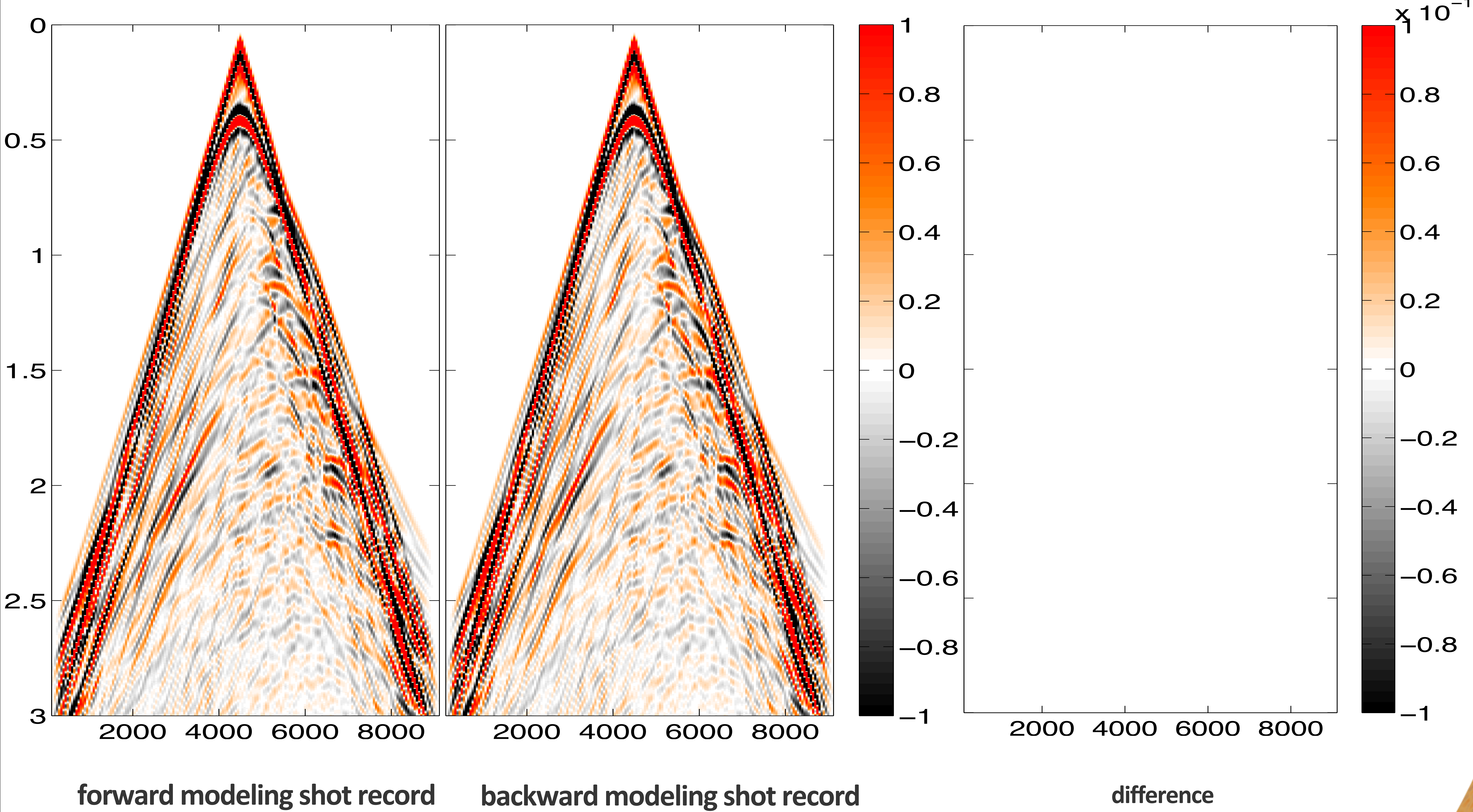
forward modeling

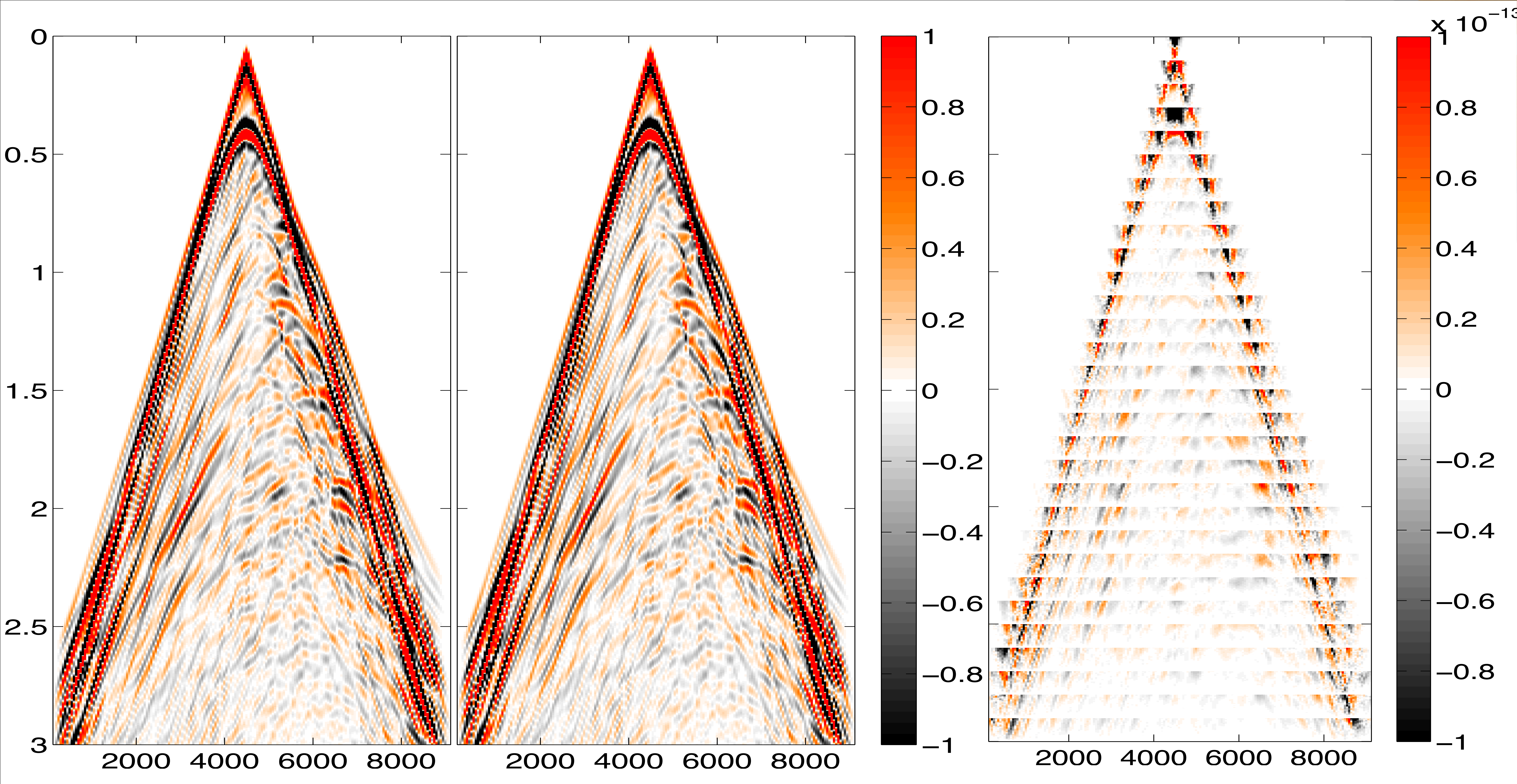


backward modeling

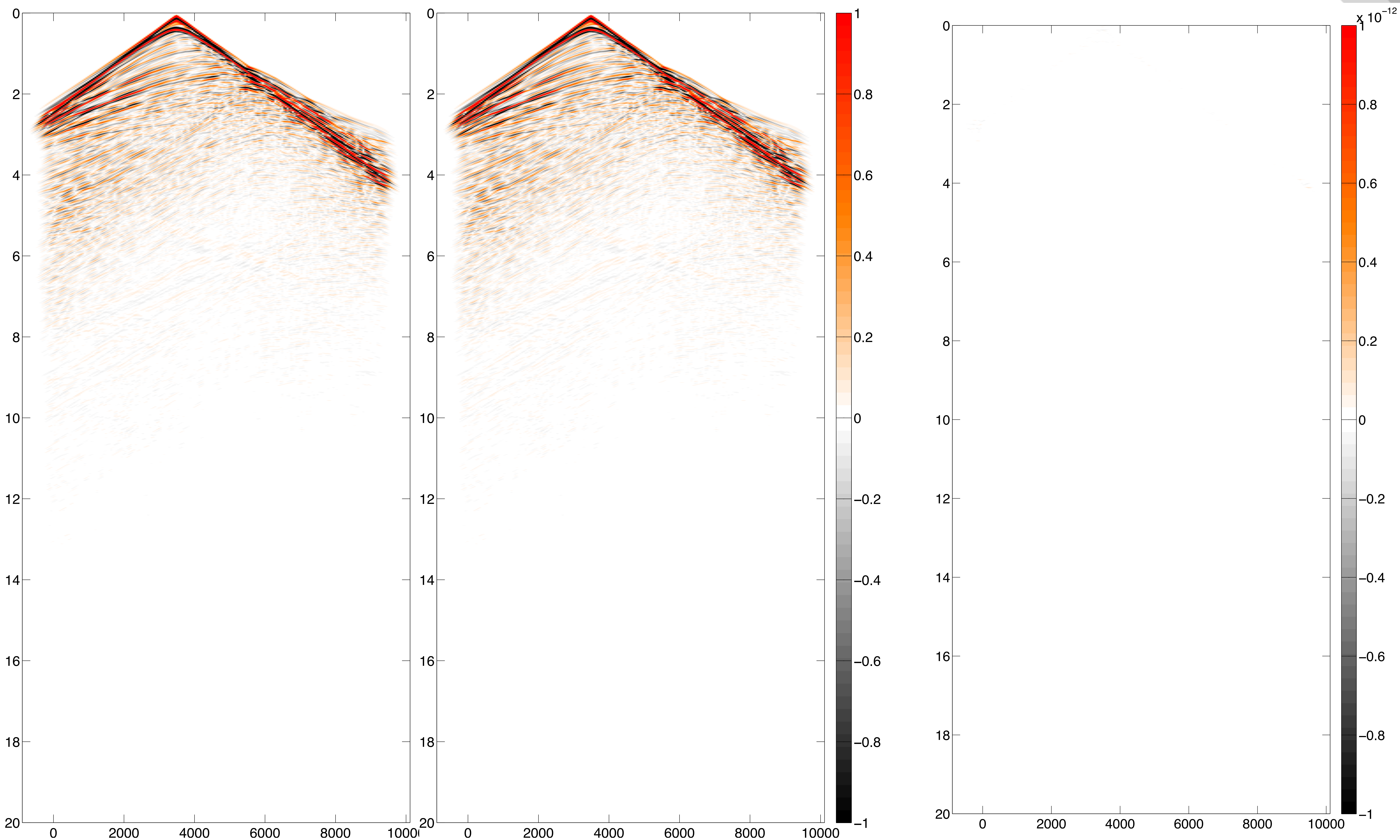
This example uses Marmousi velocity model, the size of which is 319x921, 3000 time samples with 3s length.
with free surface.

Video time is the actuarial *modeling time + plotting time*


0
0.5
1
1.5
2
2.5
3
2000
4000
6000
8000
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
x 10^−11
forward modeling shot record
backward modeling shot record
difference



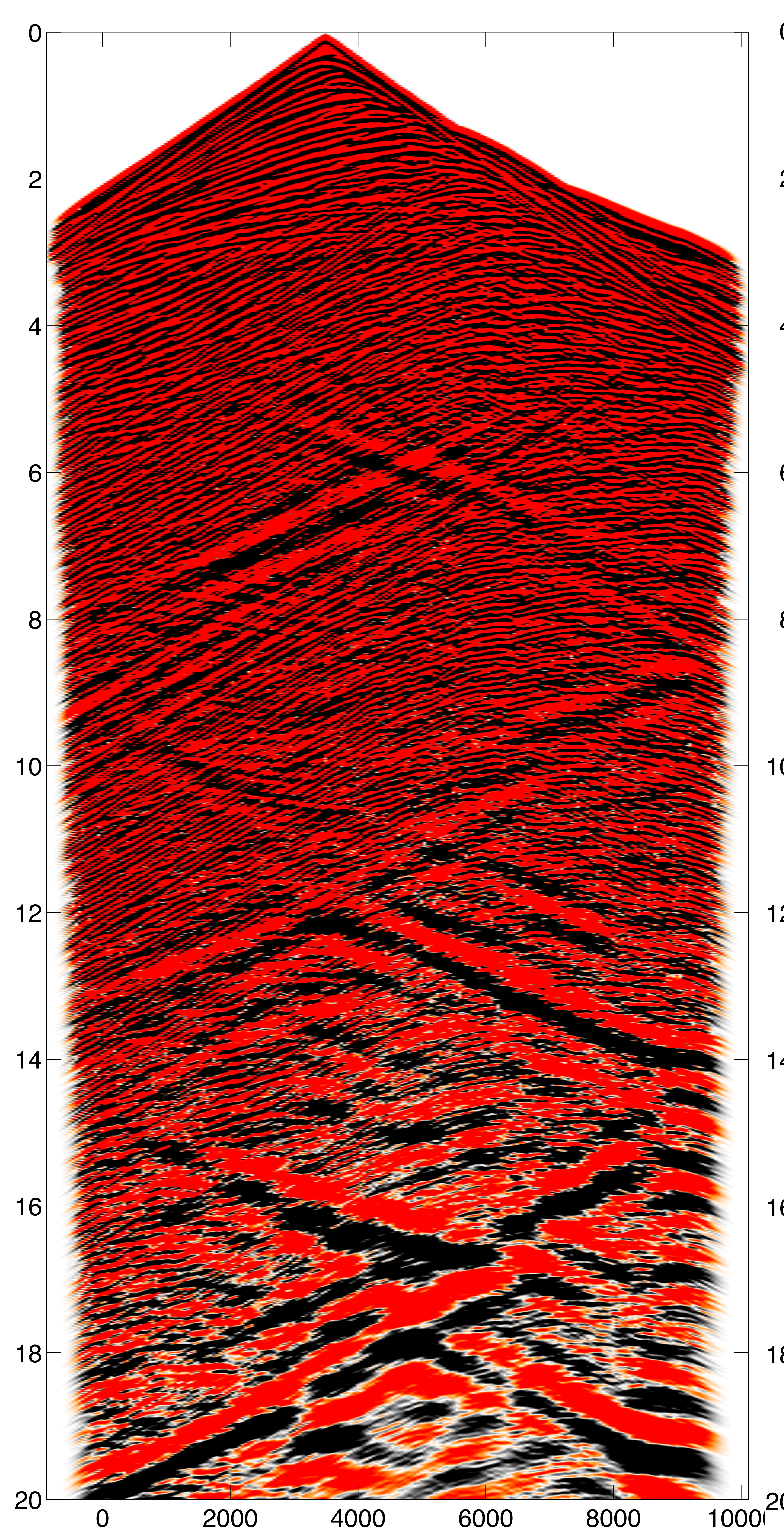
forward modeling shot record **backward modeling shot record** **difference**



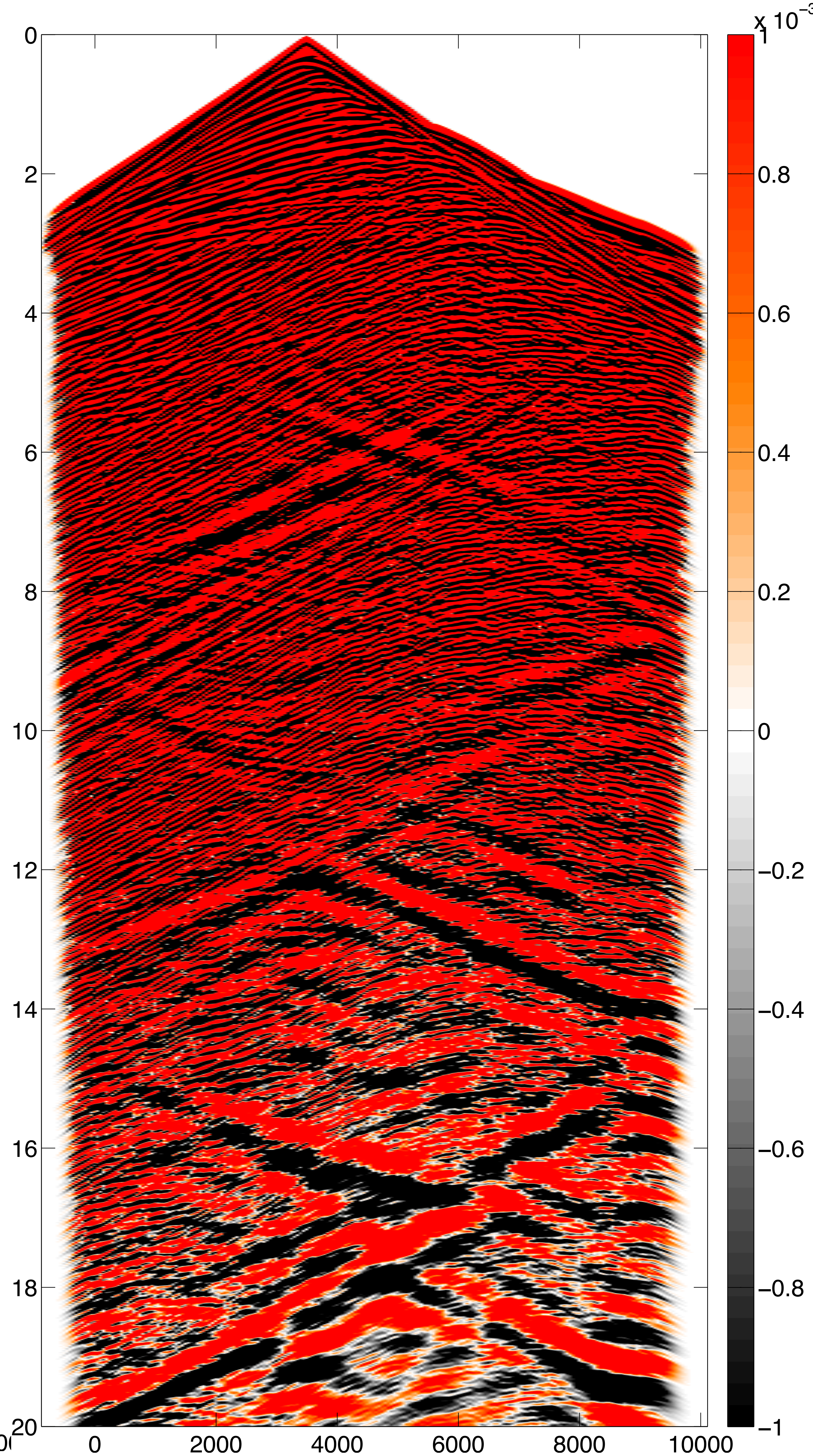
 forward modeling shot record

backward modeling shot record

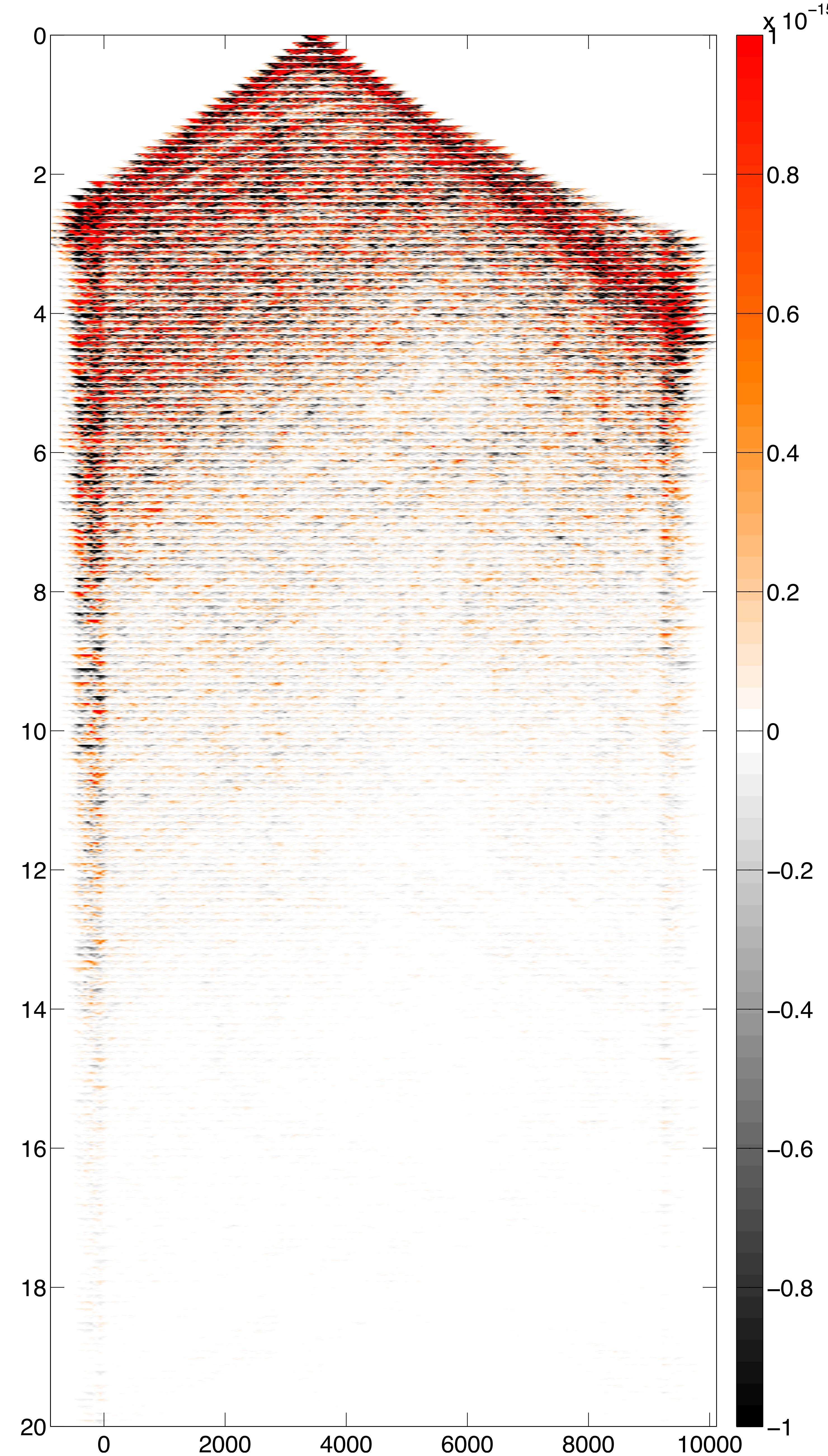
difference



forward modeling shot record



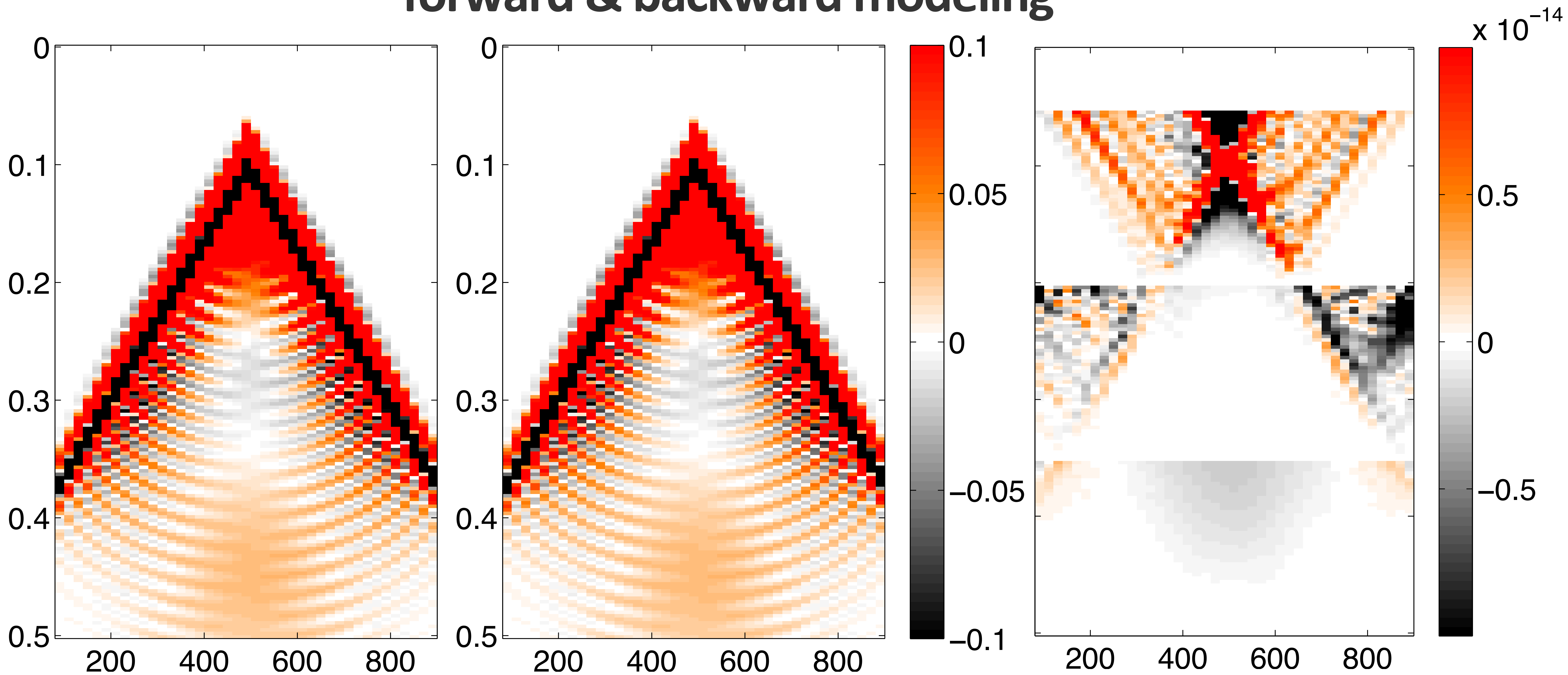
backward modeling shot record



difference

Numerical experiment

forward & backward modeling



forward modeling

backward modeling

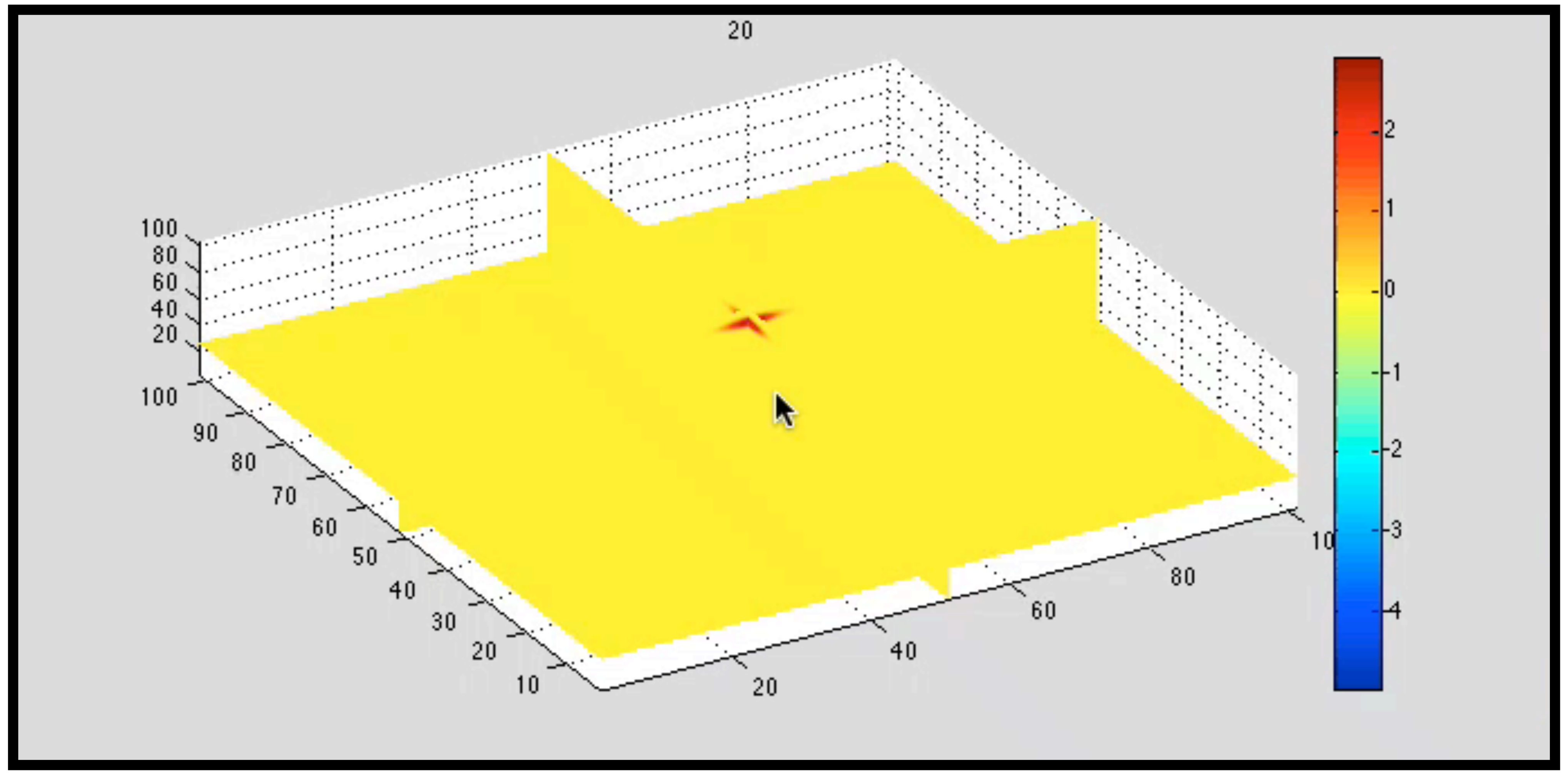
difference

This example uses constant velocity model (2000m/s), the size of which is 101x101.

We use a very bad stability condition

Video time is the actuarial *modeling time + plotting time*

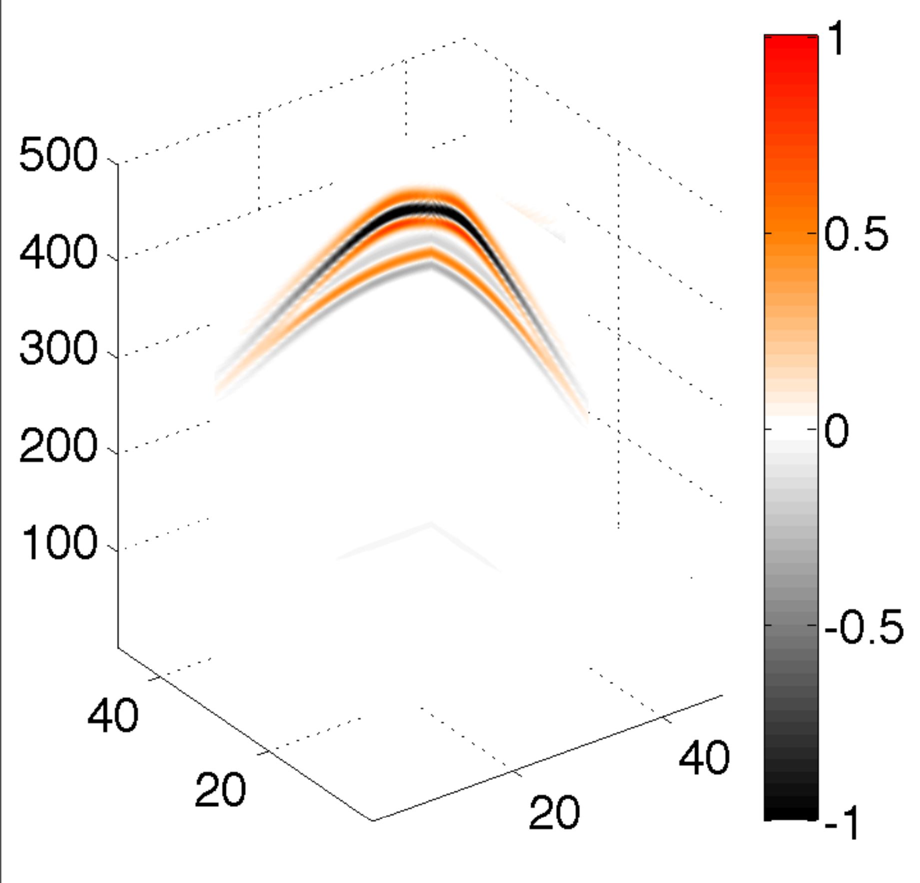
Numerical experiment



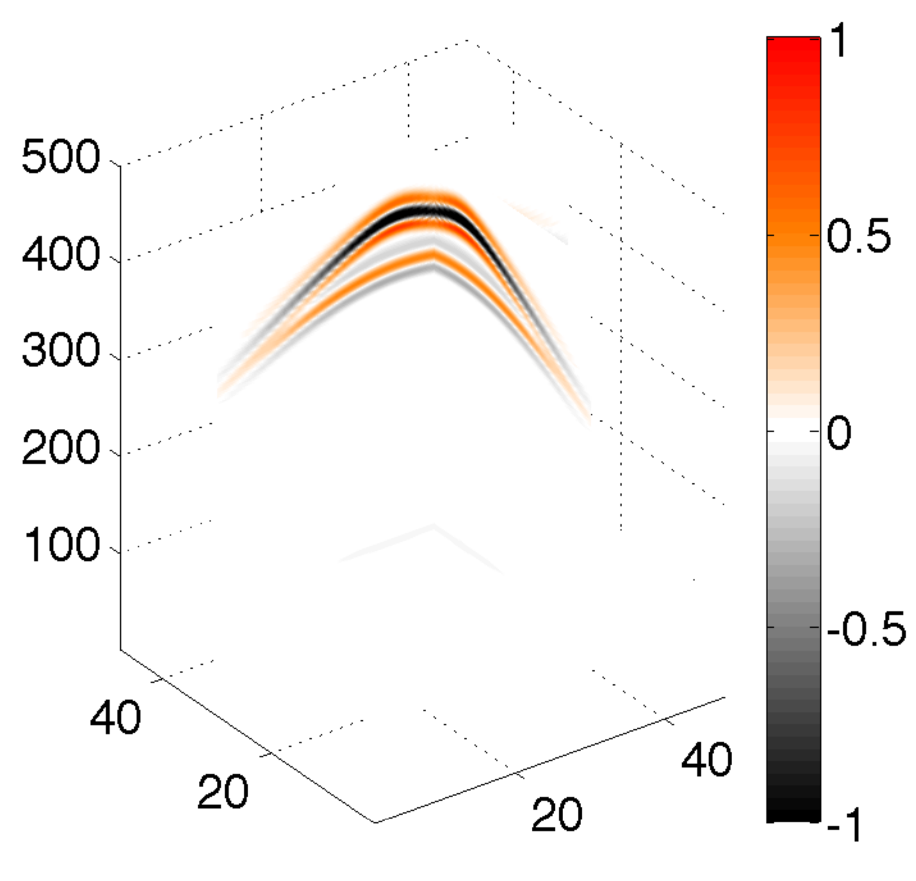
forward & backward modeling

This example uses 2-layer velocity model, the size of which is 101x101x101.

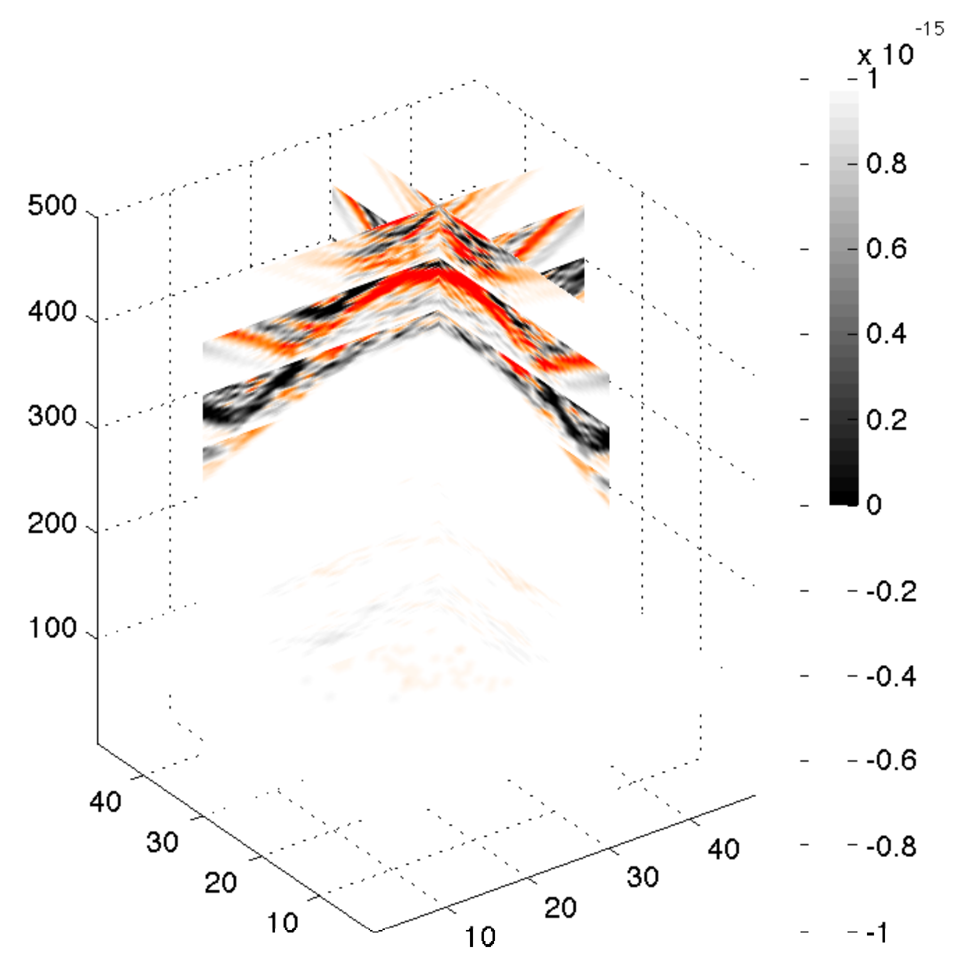
Video time is the actuarial *modeling time + plotting time*



forward modeling



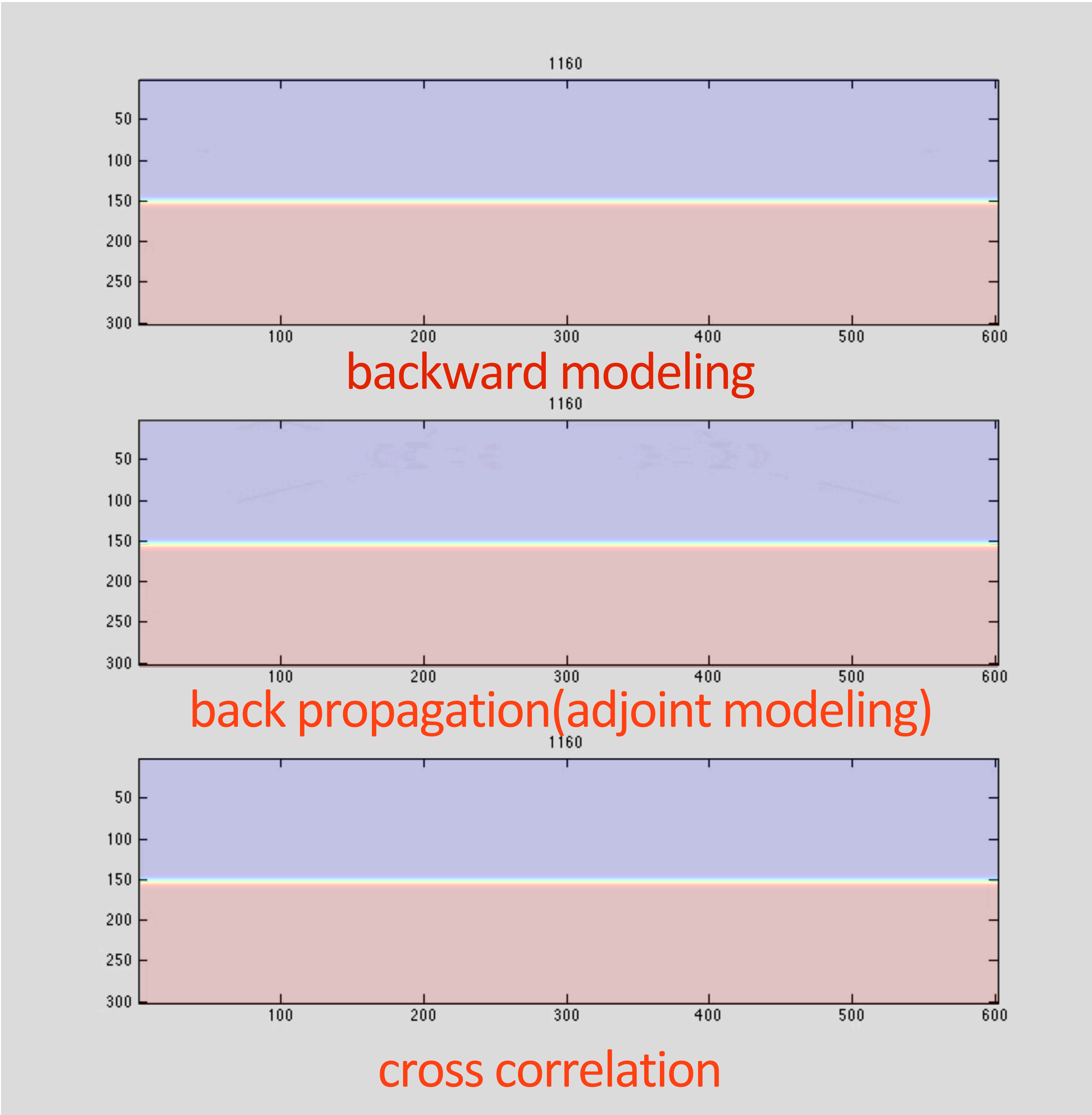
backward modeling



difference

Gradient of FWI objective function

$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[- \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$



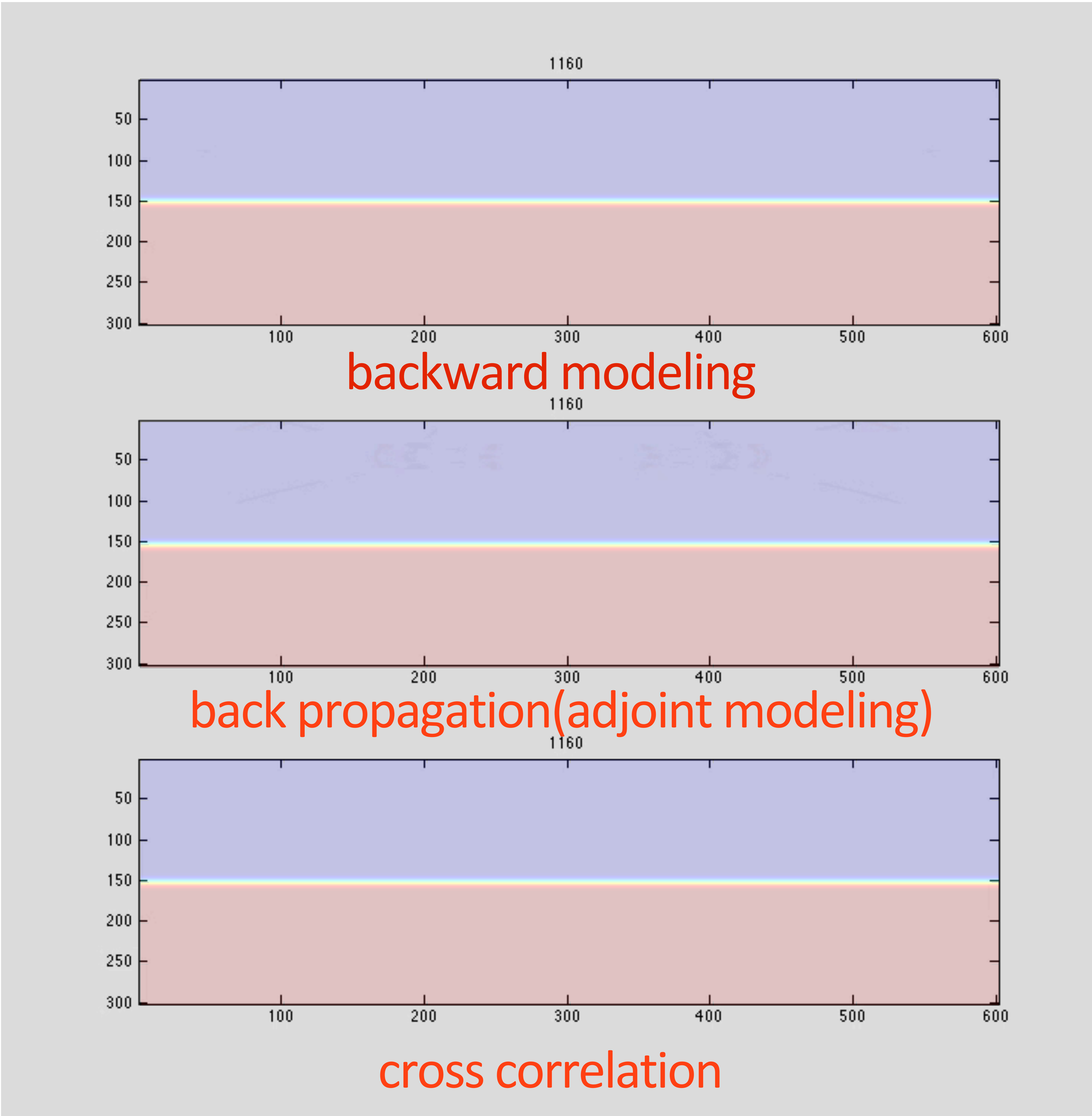
This example uses 2 layer velocity model, the size of which is 300x600, 1334 time samples with 4s length.

with freesurface.

Video time is the actuarial *gradient computing time + plotting time*

Gradient of FWI objective function

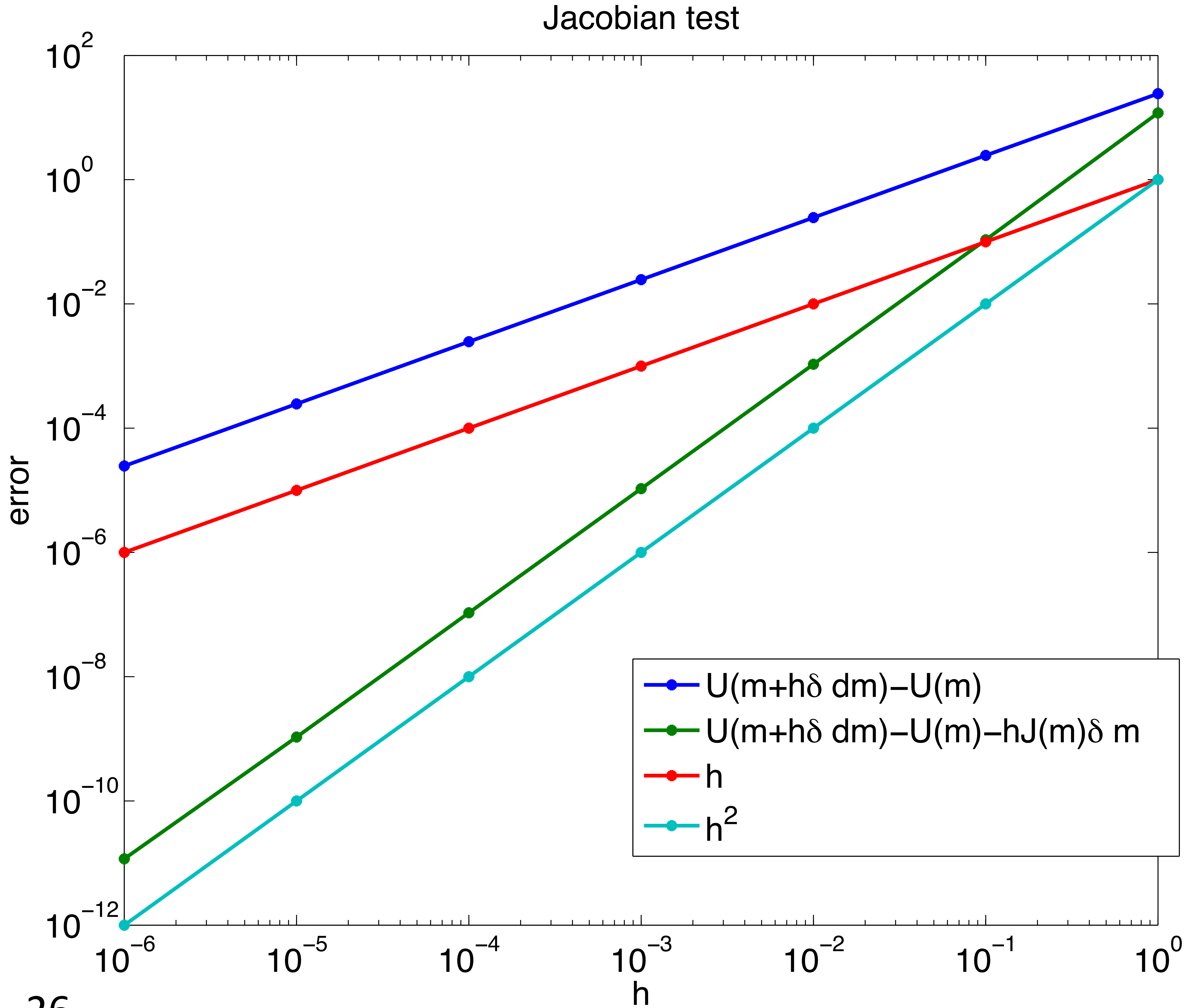
$$\nabla\Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \mathbf{P}_t \left[- \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} (\mathbf{A}^{-1}\mathbf{q}_i) \right) \odot \left[(\mathbf{A}^{\mathrm{T}})^{-1} \mathbf{P}_r^{\mathrm{T}} \delta \mathbf{d}_i \right] \right] = \mathbf{J}^{\mathrm{T}} \delta \mathbf{d}$$




```
for ti = Num_of_time_step:-1:1

    %  forward modeling solve for A U  = Q
    %
    % [ A1                           ]  U1    Q1
    % | A2 A1                        |  U2    Q2
    % | A3 A2 A1                     |  U3  = Q3
    % |             . . .            |  U4    Q4
    % [                    A3 A2 A1]  U5    Q5

    % wavefield completed while ti = 3
    if ti > 2
    % -------------- load check point per time step ---------------------
        if chk_point_idx(ti-2) == 1...end
    end % end of if
    % do  U_temp =  diag(dA U)
    if ti == 1
        U_temp = -(dA1 * U1(:));
    elseif ti == 2
        U_temp = -(dA2 * U2(:) + dA1 * U1(:));
    else
        U_temp = -(dA3 * U3(:) + dA2 * U2(:) + dA1 * U1(:));
    end

    % generate source wavelet
    du                     = prepare_adjoint_source(input_data,fwdpara,fwdpara_dis,si);

    % adjoint modeling  solve for AT V = dU
    %
    % [A1T A2T A3T                 ]  V1    dU1
    % |             . . .          |  V2    dU2
    % |                 A1T A2T A3T|  V3  = dU3
    % |                     A1T A2T|  V4    dU4
    % [                         A1T]  V5    dU5
    %

    if ti == Num_of_time_step
        V3(1:end)   = full( (Pr' * du(:,ti)));
    elseif ti == Num_of_time_step-1
        V3(1:end)   = Pr' * du(:,ti) - A2' * V2(:);
    else
        V3(1:end)   = Pr' * du(:,ti) - A2' * V2(:) - A3' * V1(:);
    end
    V3(1:end) = A1_inv' *  V3(:);
    % ------------ domain decomposition send data around ---------
    if fwdpara_dis.num_model ~=1...end
    % -------------- end of domain decomposition -----------------------

    g = g + reshape(U_temp,size(V3)).*V3;
```

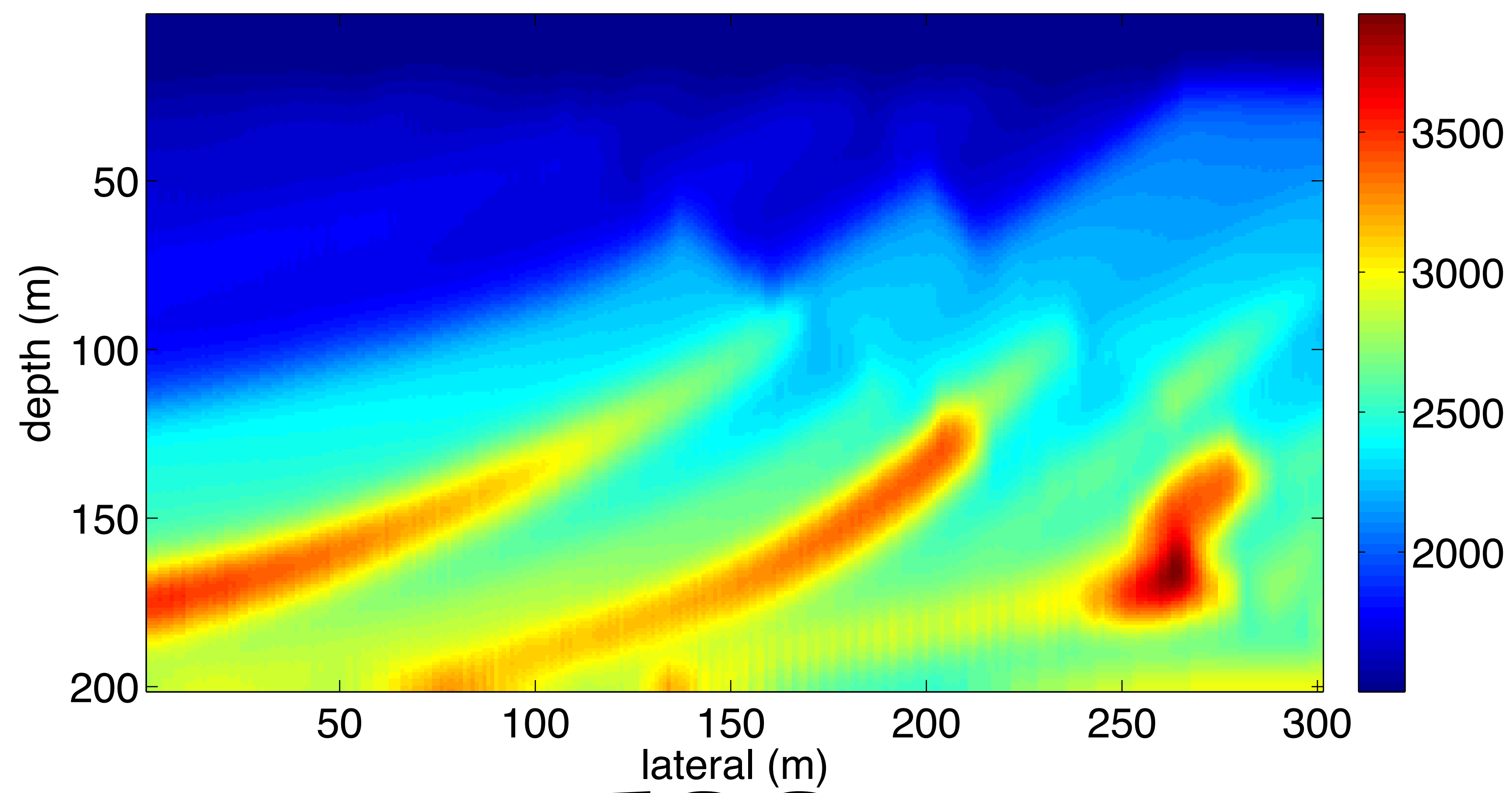

Wednesday, December 10, 14

Jacobian test



considering the Taylor approximation

$$\mathbf{U}(\mathbf{m} + h\delta\mathbf{m}) = \mathbf{U}(\mathbf{m}) + h\mathbf{J}(\mathbf{m})\delta\mathbf{m} + \mathcal{O}(h^2)$$



$\mathbf{m}$

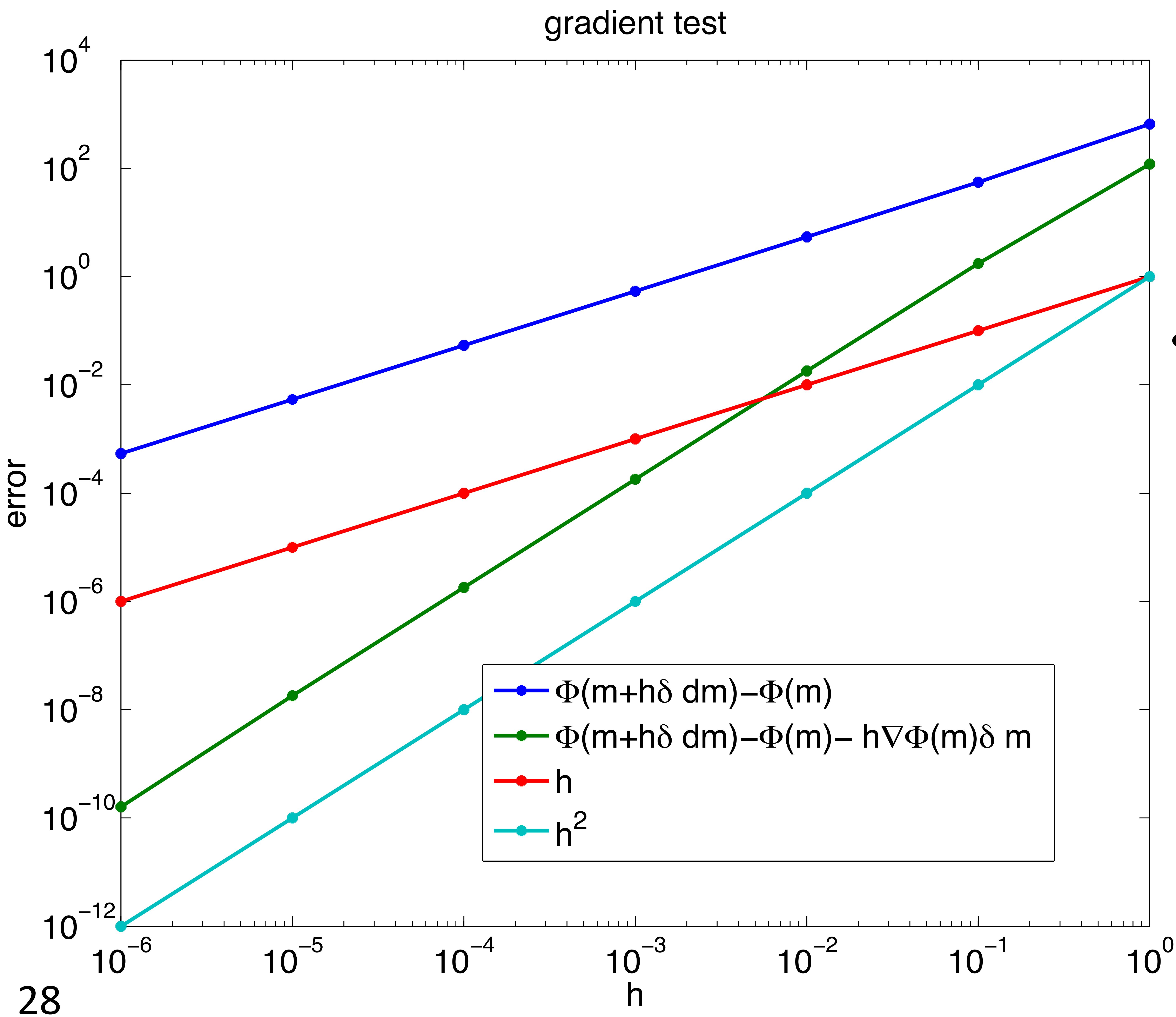
Adjoint test

	$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} >$	$< \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >$	$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} > = < \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >$
2D one layer	198.6919	198.6919	3.69E-13
2D Marmousii	2.48E+04	2.48E+04	2.01E-09
3D one layer	9.0273	9.0273	4.09E-14

The adjoint test will ensure that our implementation of the adjoint satisfies the formal definition of the adjoint

$$< \mathbf{J}\delta\mathbf{m}, \delta\mathbf{d} > = < \delta\mathbf{m}, \mathbf{J}^{\mathrm{T}}\delta\mathbf{d} >$$

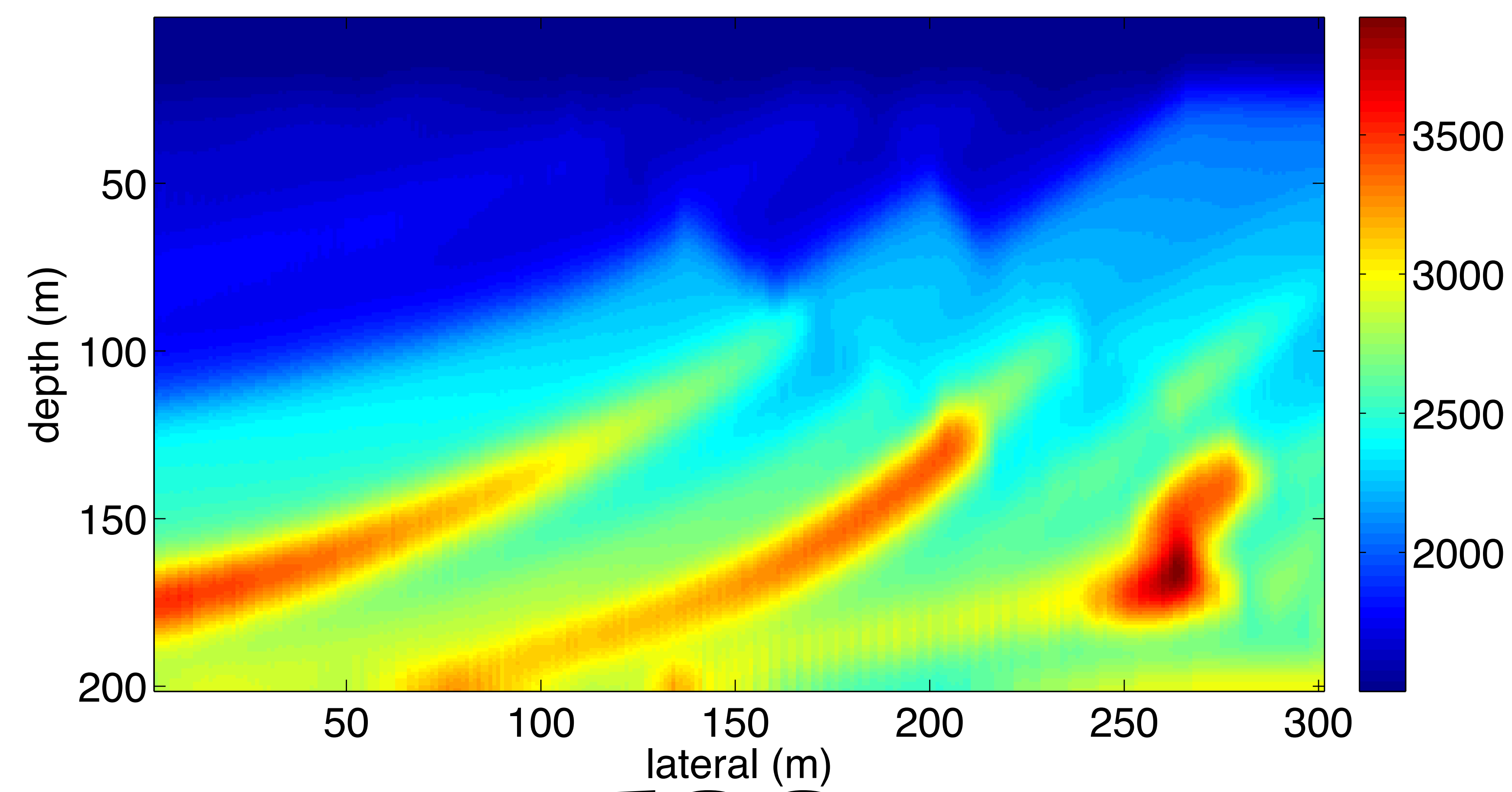
Gradient test



considering the Taylor approximation of FWI objective

$$\Phi(\mathbf{m}) = \frac{1}{2}\|\mathbf{d} - \mathbf{P}_r \mathbf{A}(\mathbf{m})^{-1}\mathbf{q}\|_2^2$$

$$\Phi(\mathbf{m} + h\delta\mathbf{m}) = \Phi(\mathbf{m}) + h\nabla\Phi(\mathbf{m})\delta\mathbf{m} + \mathcal{O}(h^2)$$



$$\mathbf{m}$$

Applications (Chevron 2014 benchmark)

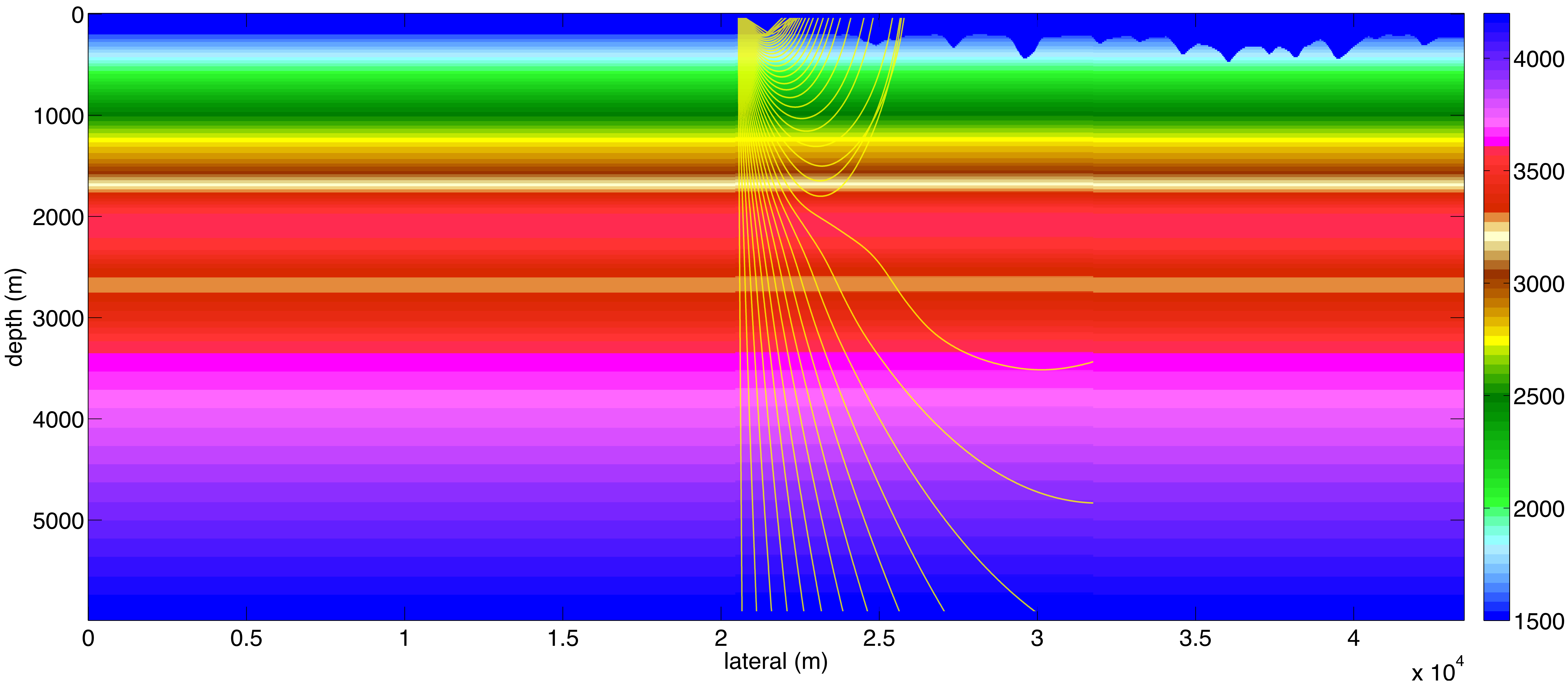
- 1600 shots: dS = 25 m, Source depth = 15 m
- 321 hydrophone recs/shot: dR = 25 m, Recevr depth = 15 m
- Max offset = 8000 m
- Record time = 8.0 s, sample rate 4 ms
- Vp water = constant = 1510 m/s
- With free surface multiples present in the data
- Isotropic Elastic

Inversion settings

- Time domain approach with frequency domain FWI precondition
- 3-10 Hz inversion
- 4 frequency bands
- 4 GN updates for each frequency band
- Each GN update uses 5 LSQR iterations

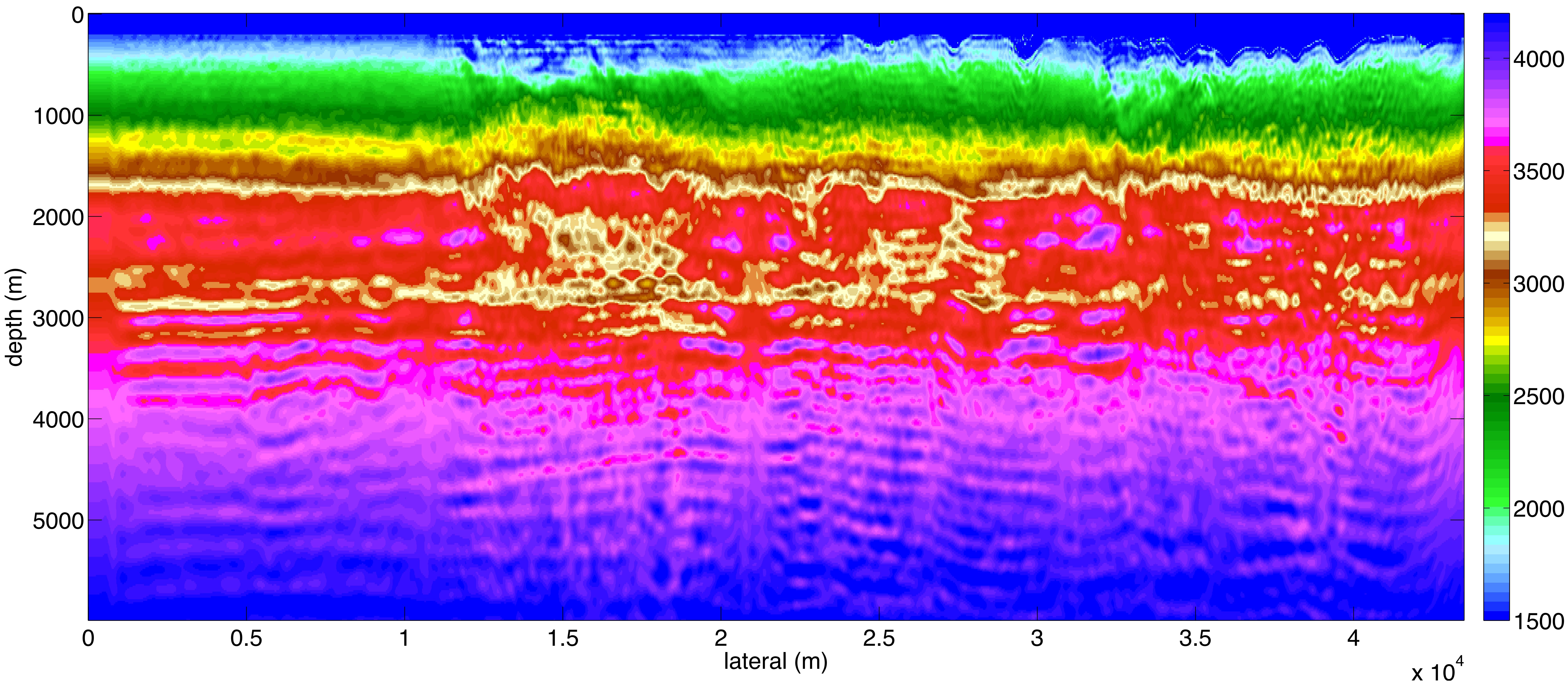
Initial model

10m grid



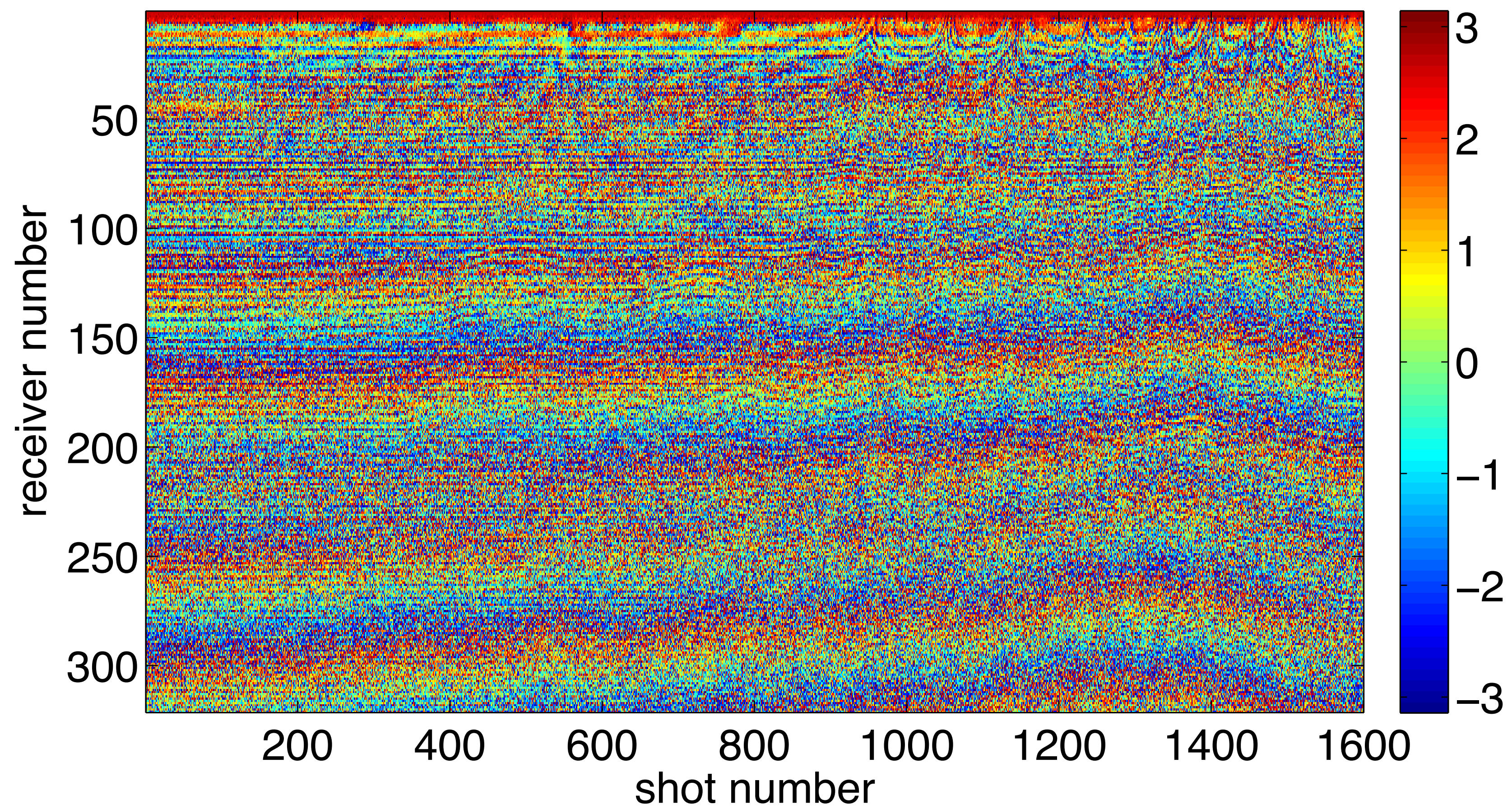
FWI result

time domain result

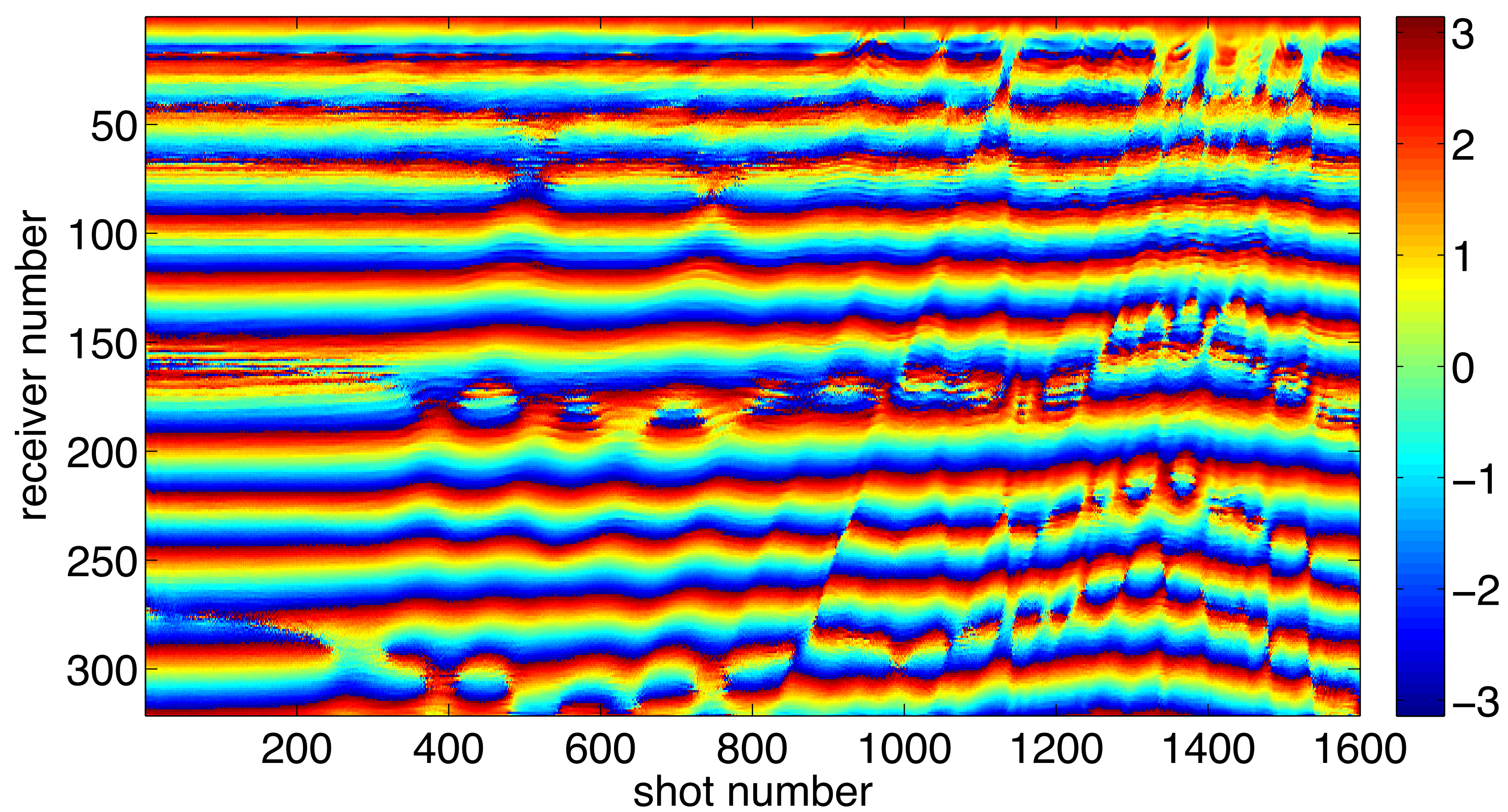


Data in frequency VS data in time

2Hz

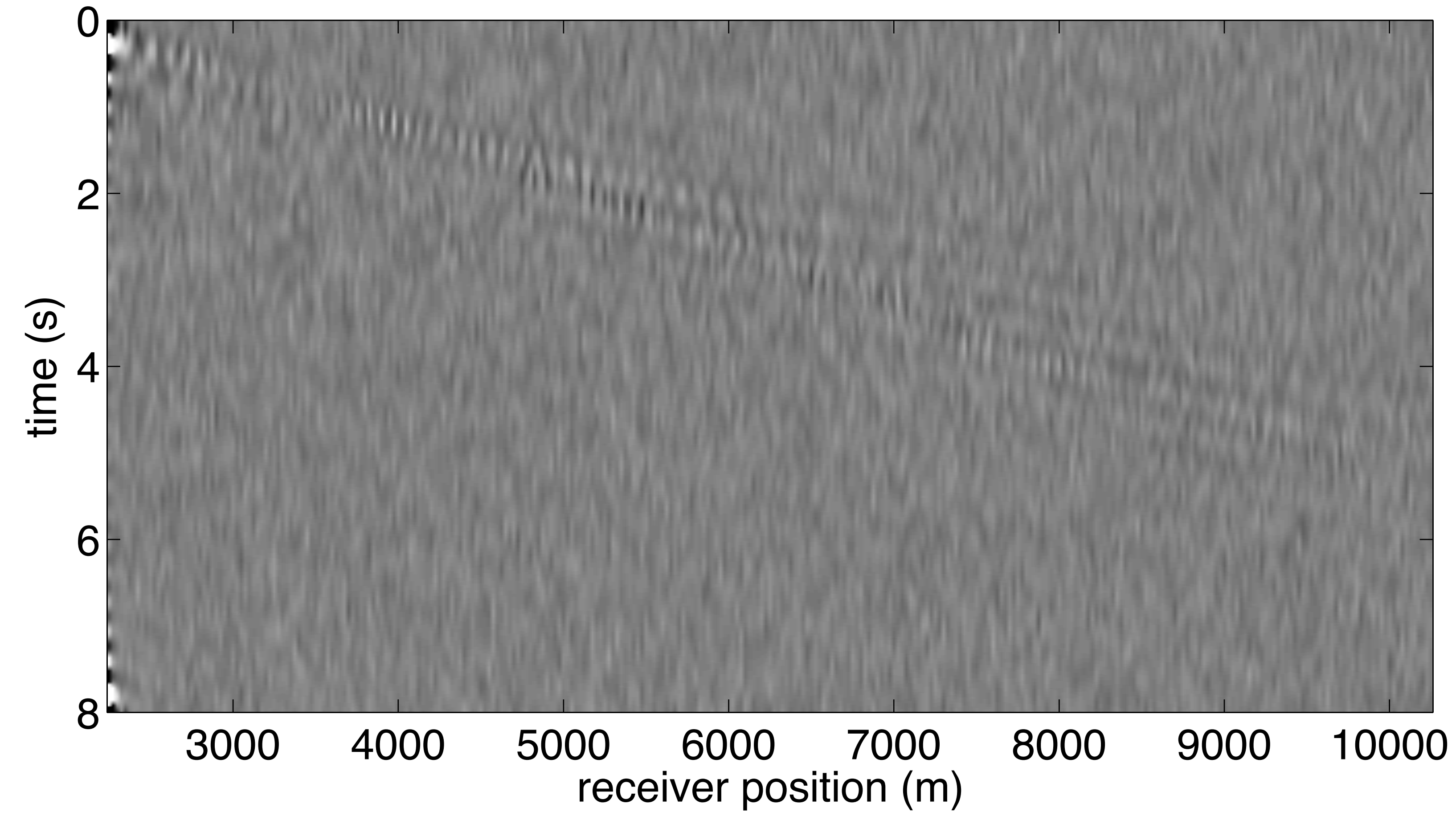


3Hz

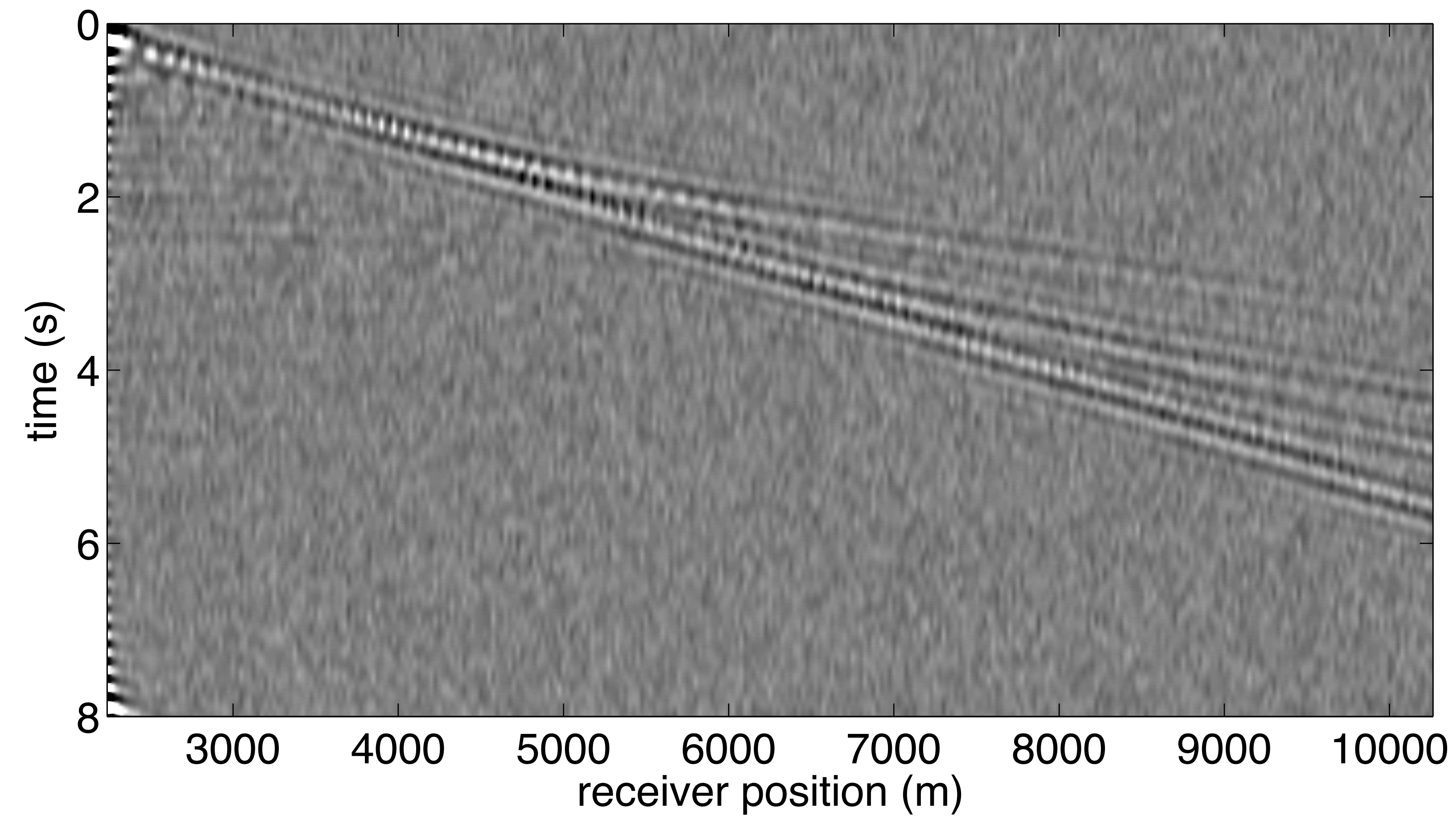


frequency domain data for inversion

0-3Hz



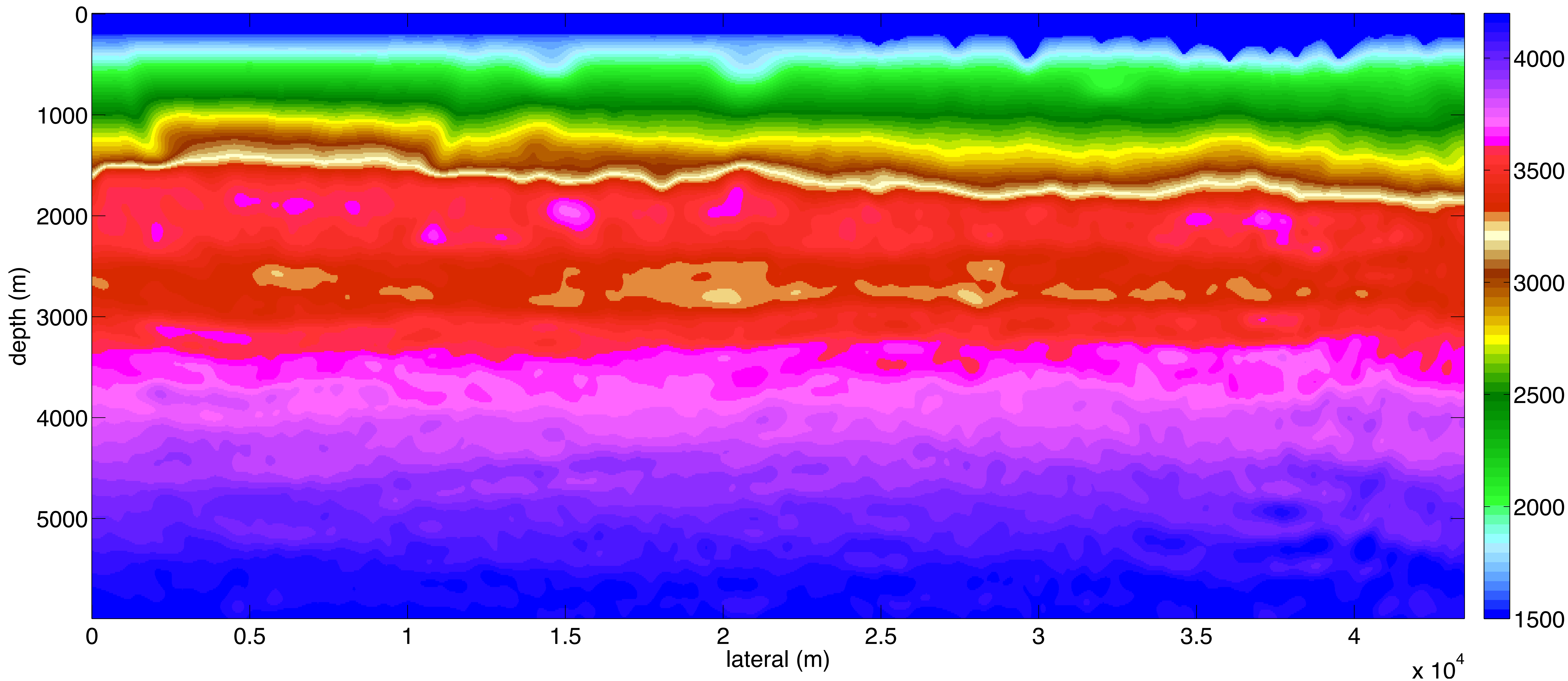
0-5Hz



time domain data for inversion

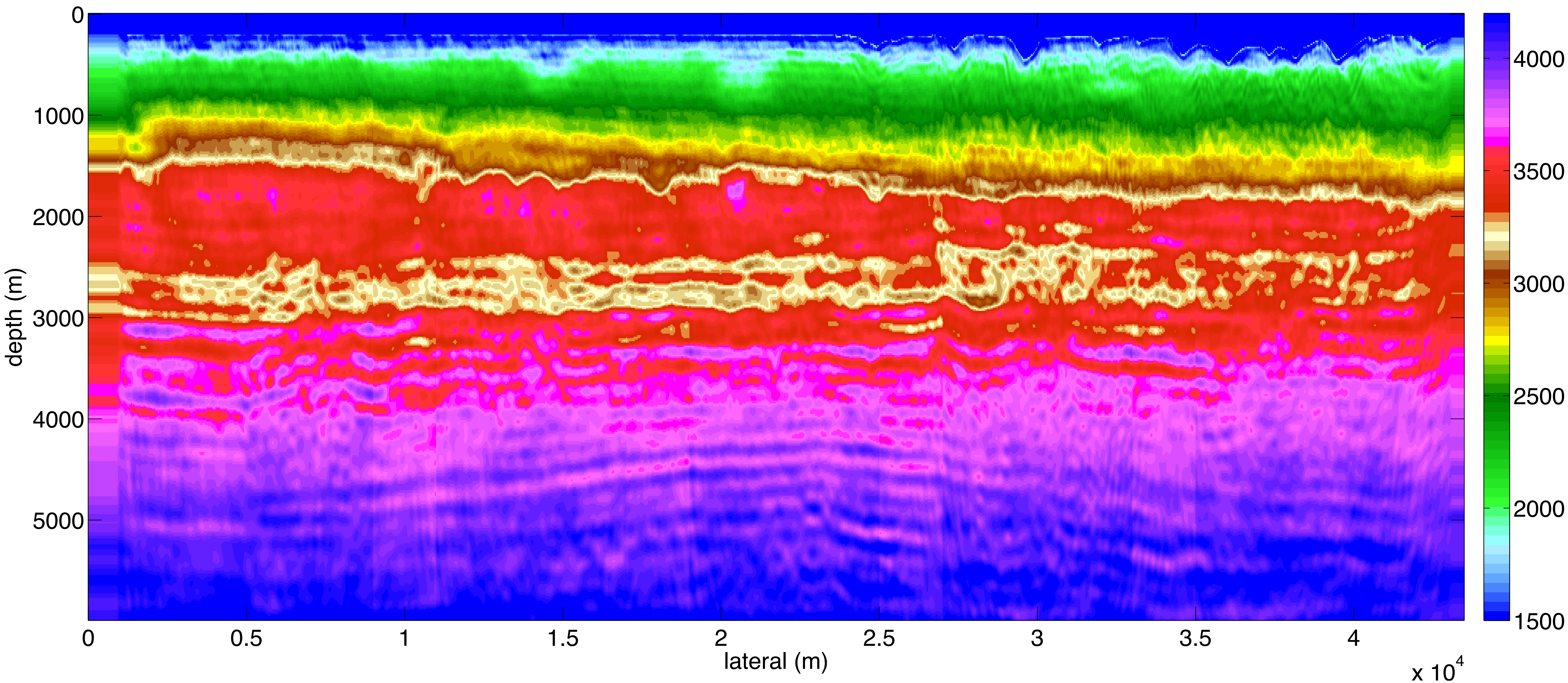
Smoothed frequency domain result

use this as a initial model for time domain FWI

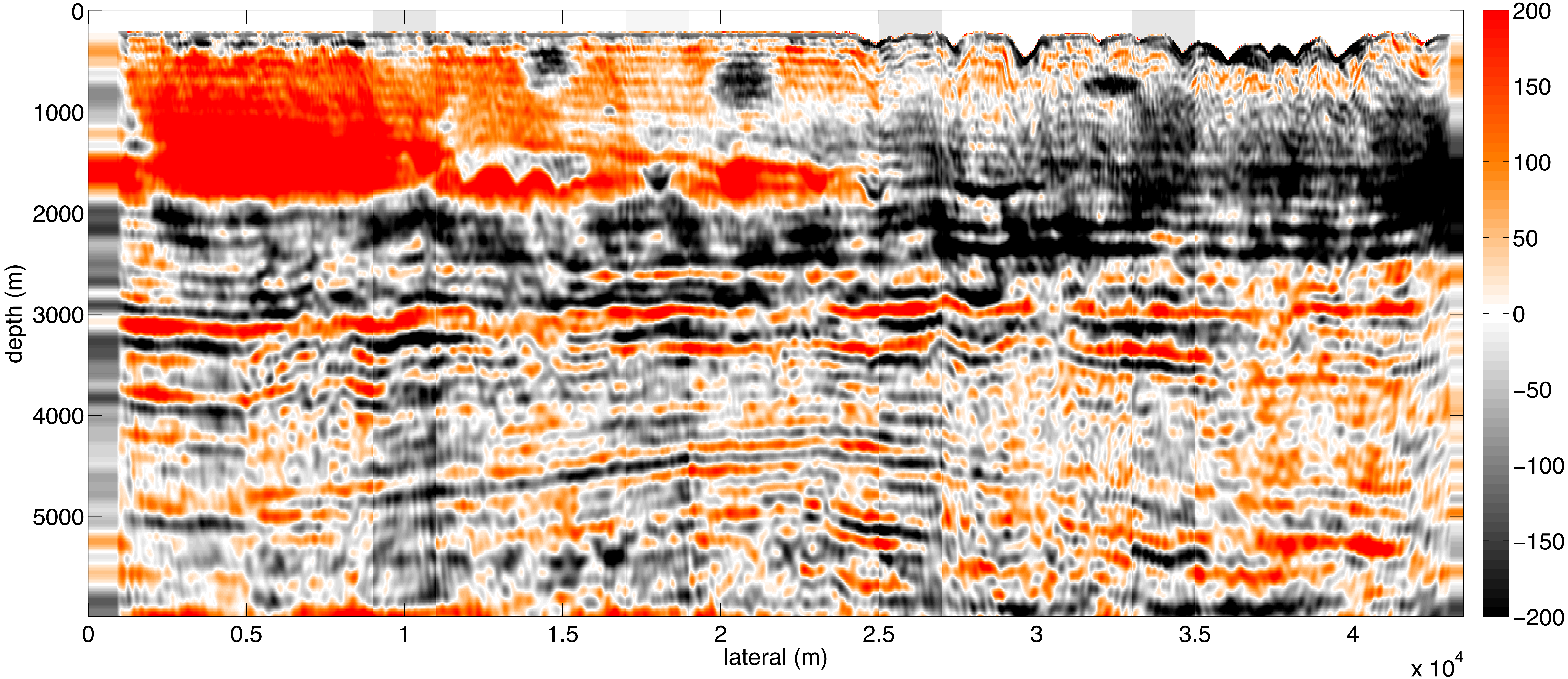


FWI in time domain

3-10Hz Gauss-Newton with density and free surface

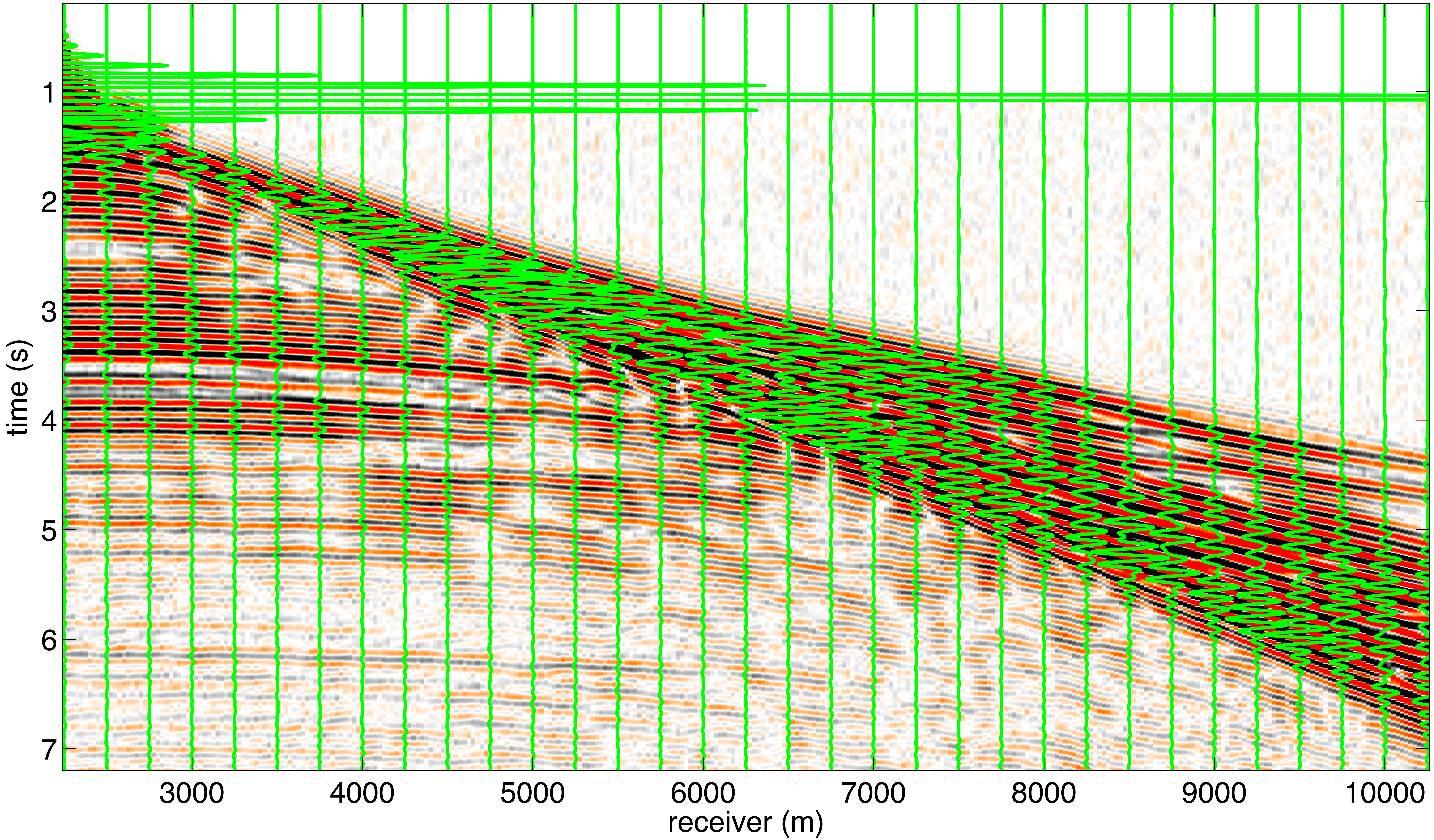


Difference



depth (m)
0
1000
2000
3000
4000
5000
lateral (m)
0
0.5
1
1.5
2
2.5
3
3.5
4
x 10^4
200
150
100
50
0
−50
−100
−150
−200

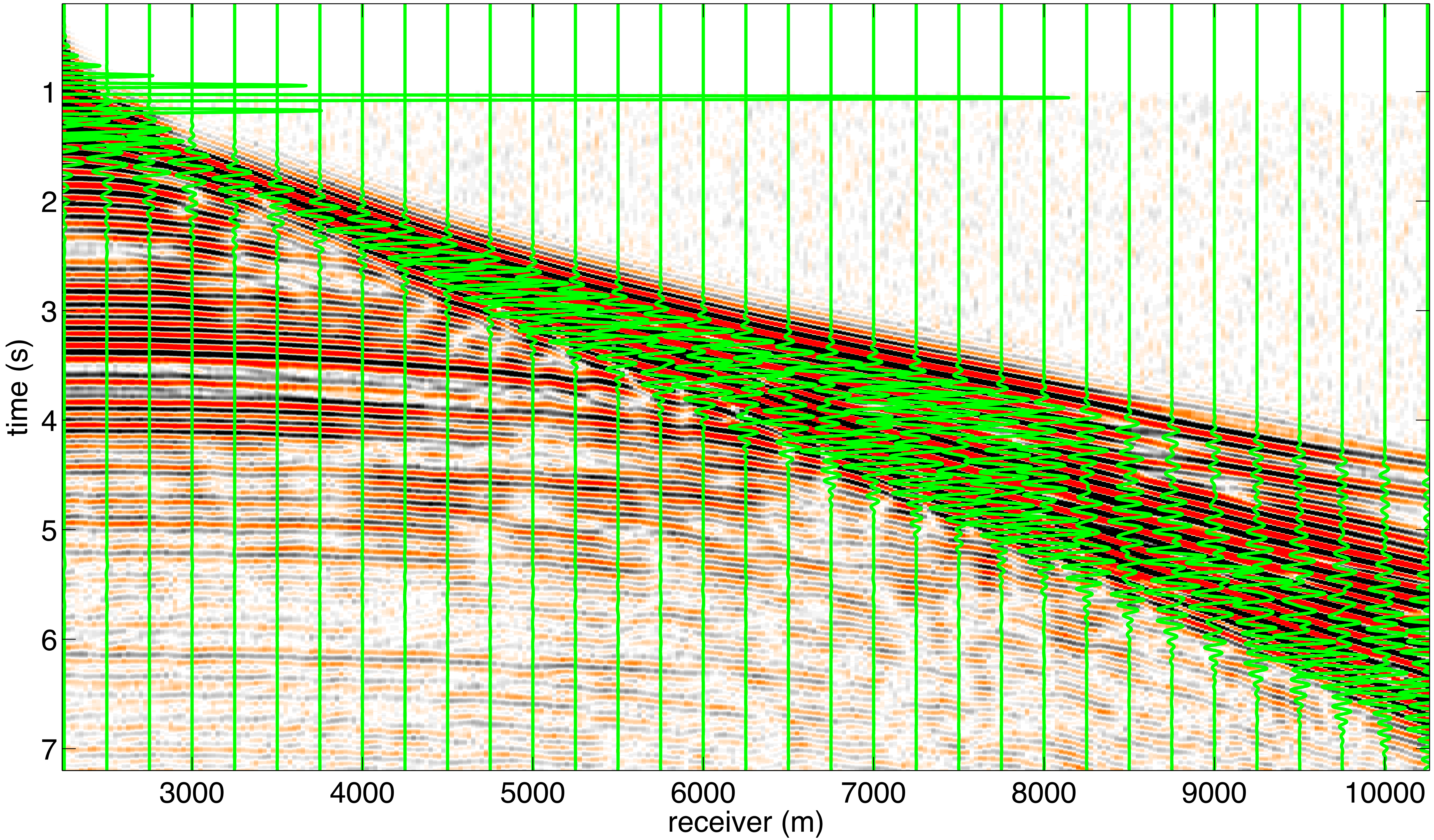
Data comparison



observed shot record

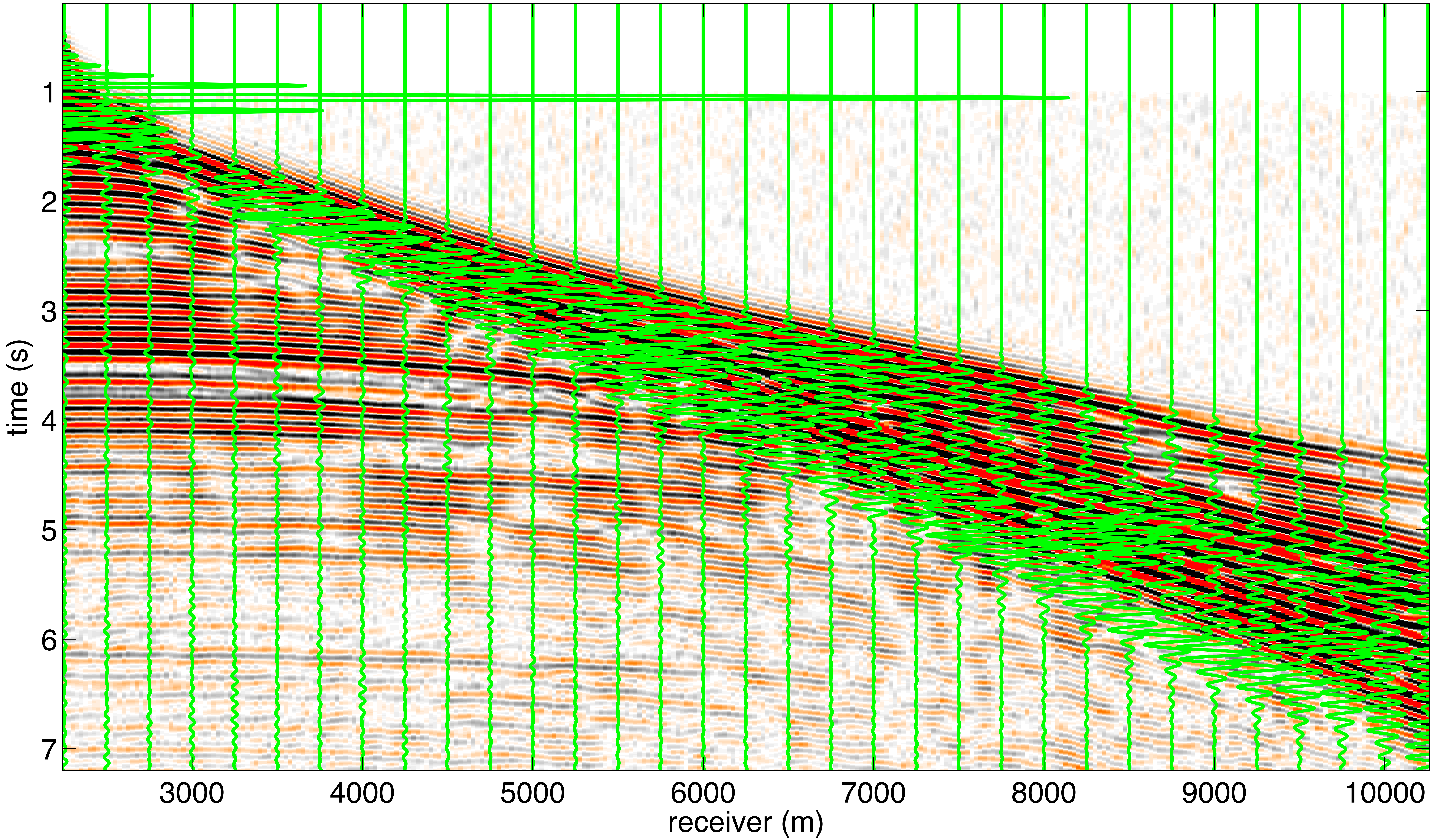
background is the observed data. Same for the rest

Data comparison



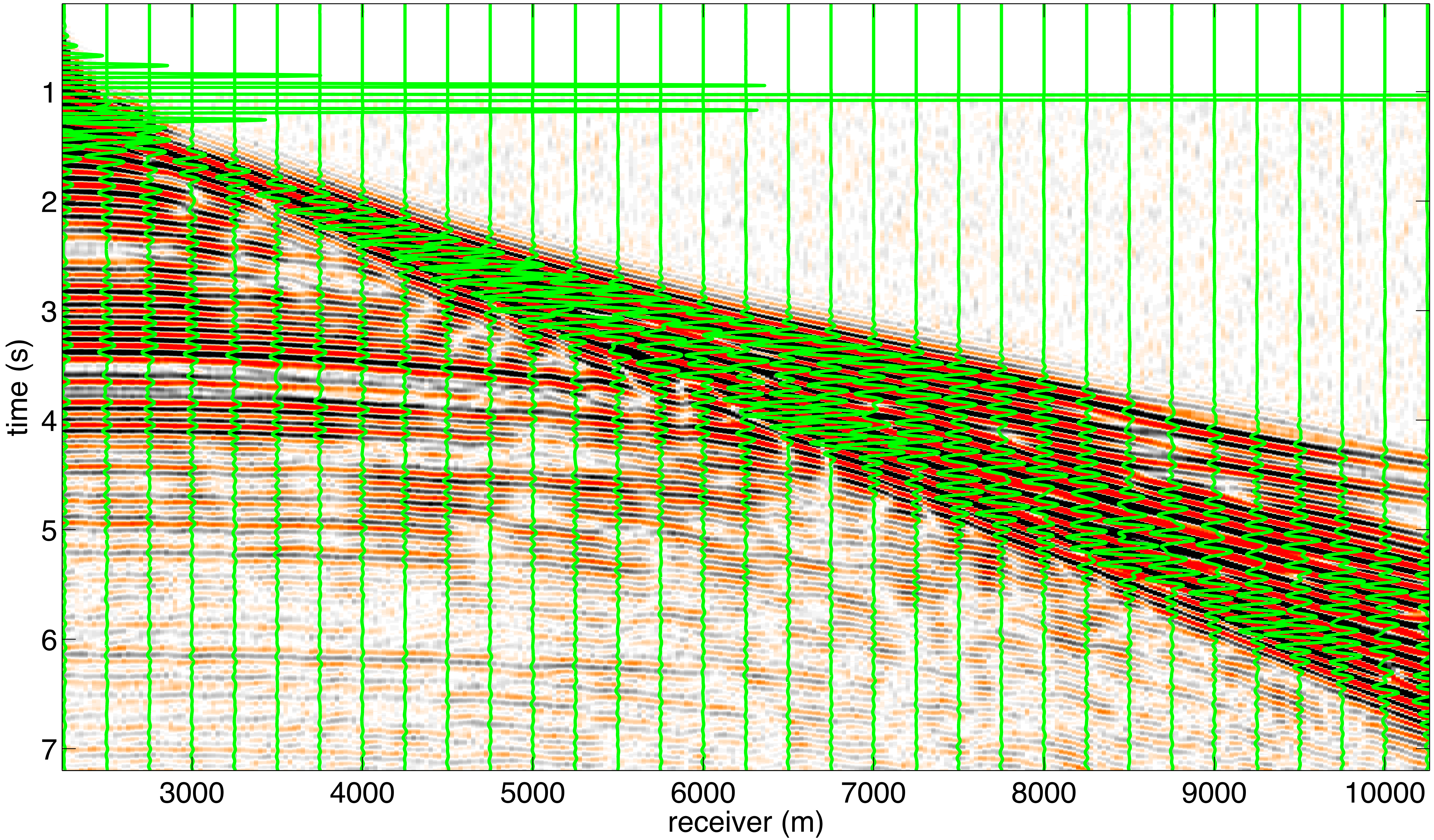
shot record from initial model

Data comparison



shot record from FWI result

Data comparison



observed shot record

Future plan

modeling

- TTI
- elastic

imaging

- imaging with sparse regularization

FWI

- MGN FWI

TTI forward modeling and backward modeling

$$\epsilon = 0.2$$

$$\delta = 0.02$$

$$\theta = 0.3\pi$$

SLIM

Acknowledgements

Thank you for your attention !

https://www.slim.eos.ubc.ca/

SINBAD

NSERC CRSNG

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, CGG, Chevron, ConocoPhillips, ION, Petrobras, PGS, Total SA, WesternGeco, and Woodside.