Seismic full waveform inversion algorithms and their numerical behaviour

MATLAB Energy Conference

Kris Innanen

Nov 2020

NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience

Outline

- Introduction to the CREWES Project
- The seismic exploration / monitoring problem & the CREWES Matlab toolbox
- Seismic full waveform inversion (FWI) science and technology
    - *Academic applications to field-scale problems*
    - *New formulations, new parameters, new data modes, new goals*
- FWI in Matlab: tour of a simple implementation
- Recent research results
    - *New parameters and uncertainty*
    - *New data modes (fiberoptic or "DAS" seismology)*
    - *New goals (direct determination of rock physics properties)*
- Next steps

Research group introduction – who we are, what we do

CREWES

UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience

NSERC
CRSNG

CANADA FIRST
RESEARCH EXCELLENCE FUND

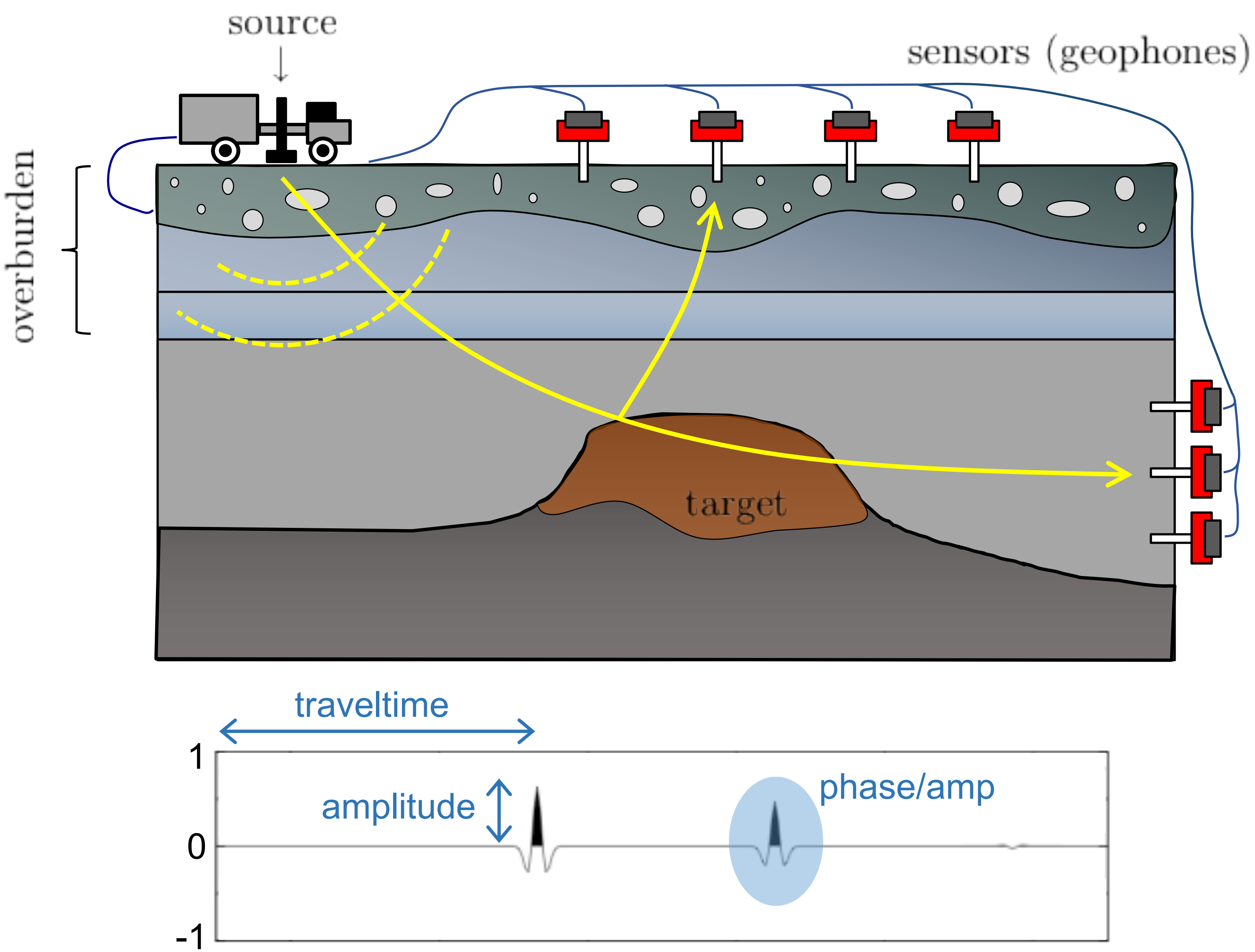
CMC RESEARCH INSTITUTES

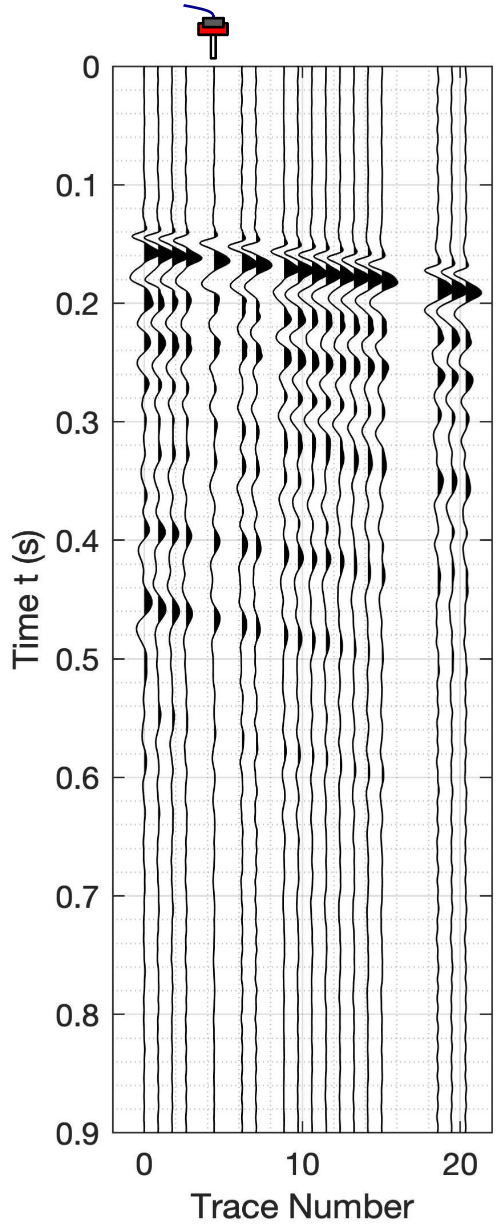
Acceleware
CGG
Chevron
CNOOC Intl
Devon
Halliburton
INOVA Geo.
Petrobras
Petronas
RIPED, CNPC
Aramco SC
SINOPEC
TGS

Do the science needed to create the next generation of seismic exploration and monitoring technology.

- Seismic acquisition: DAS and broadband
- Modelling, processing and imaging
- Large geo-computational problems
- Seismic inversion
    - *Elastic FWI for reservoir properties*
    - *Fractures and fluids*

The seismic exploration / monitoring problem





(Some of the) seismic data processing tasks

- Processing
  - *Denoising – deconvolution – sorting – interpolation*
  - *Demultiple – inverse Q filtering*
  - *Velocity analysis – traveltime corrections – stacking*
- Modelling
  - *Ray-tracing – eikonal solvers – reflectivity – AVO modeling*
  - *Finite difference – Q reflectivity*
- Imaging
  - *time migration – depth migration – converted wave migration*
  - *least-squares migration – reverse time migration*
- Inversion
  - *AVO inversion – Q estimation – tomography*

The CREWES Matlab toolbox *(www.crewes.org)*

Data Science

Full Waveform Inversion

Converted Waves

Joint Inversion

Graduate Theses

Explorer Programs

Free Seismology Textbook & Software

Online Papers

Publications

Errata

Field Work

CREWES Matlab Toolbox

CREWES Toolbox Version: 2078

CREWES Matlab Toolbox (ZIP) 155.19 MB
MD5 checksum:b6918abcada78fb19c0eec1e59d7 *crewes.zip

Sample data to accompany Methods_of_Seismic_Data_Processing (ZIP) 13.65 MB
MD5 checksum:c402597c854a23cd8ee81c654b91 *NMESdata.zip

Guide to the CREWES Matlab toolbox (PDF) 6.98 MB
MD5 checksum:e38648c7a6ae99164c13c2a9027d *NumMeth.pdf

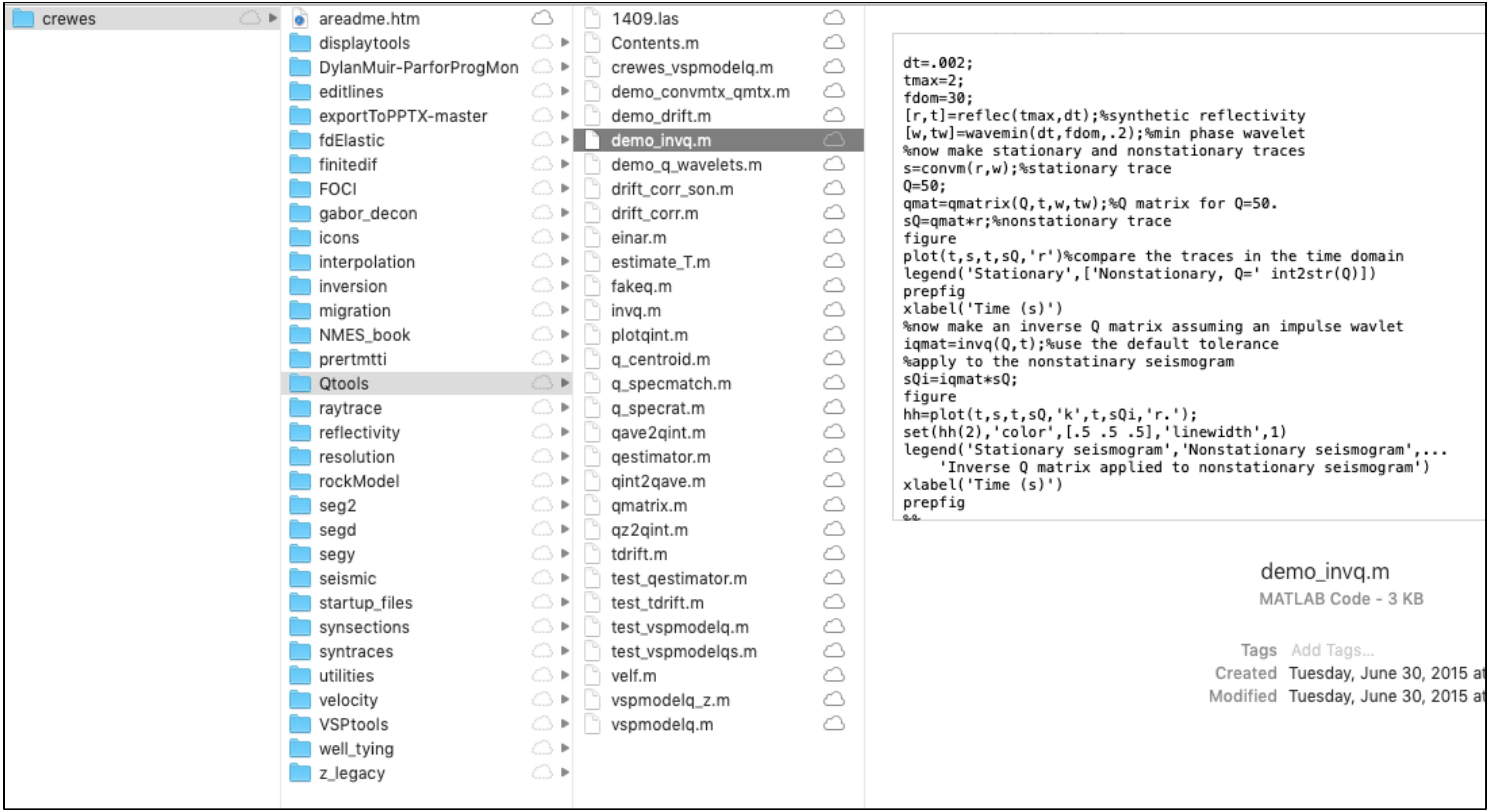
Introductory seismic data processing course (PDF) 88.09 MB
MD5 checksum:01e78195b1e8dd0c3c69ba2c78dc *Methods_of_Seismic_Data_Processing.pdf

Primary author: Gary F. Margrave
Primary maintainer: support@crewes.org
Current development environment: Matlab 2020a
Snapshot (crewes.zip) updated: Daily

Introduction

This software is called the CREWES MATLAB Software Library (CMSL) and accompanies the textbook Numerical Methods of Exploration Seismology: With Algorithms in MATLAB (NMES) by Gary F. Margrave and Michael P. Lamoureux (Cambridge University Press, 2019). An older, less complete, free version is provided here. The textbook discusses a subset of the CMSL. Both library and text are intended for teaching and research in exploration seismology. The complete CMSL is a large collection of geophysical codes that has grown by accretion over time with limited planning or regulation. The subset of CMSL covered by NMES has been checked for consistency and

The CREWES Matlab toolbox *(www.crewes.org)*



The CREWES Matlab toolbox *(www.crewes.org)*

- Data Science
- Full Waveform Inversion
- Converted Waves
- Joint Inversion
- Graduate Theses
- Explorer Programs
- **Free Seismology Textbook & Software**
- Online Papers
- Publications
- Errata
- Field Work

CREWES Matlab Toolbox

CREWES Toolbox Version: 2078

CREWES Matlab Toolbox (ZIP) 155.19 MB
MD5 checksum:b6918abcada78fb19c0eec1e59d7 *crewes.zip

Sample data to accompany Methods_of_Seismic_Data_Processing (ZIP) 13.65 MB
MD5 checksum:c402597c854a23cd8ee81c654b91 *NMESdata.zip

Guide to the CREWES Matlab toolbox (PDF) 6.98 MB
MD5 checksum:e38648c7a6ae99164c13c2a9027d *NumMeth.pdf

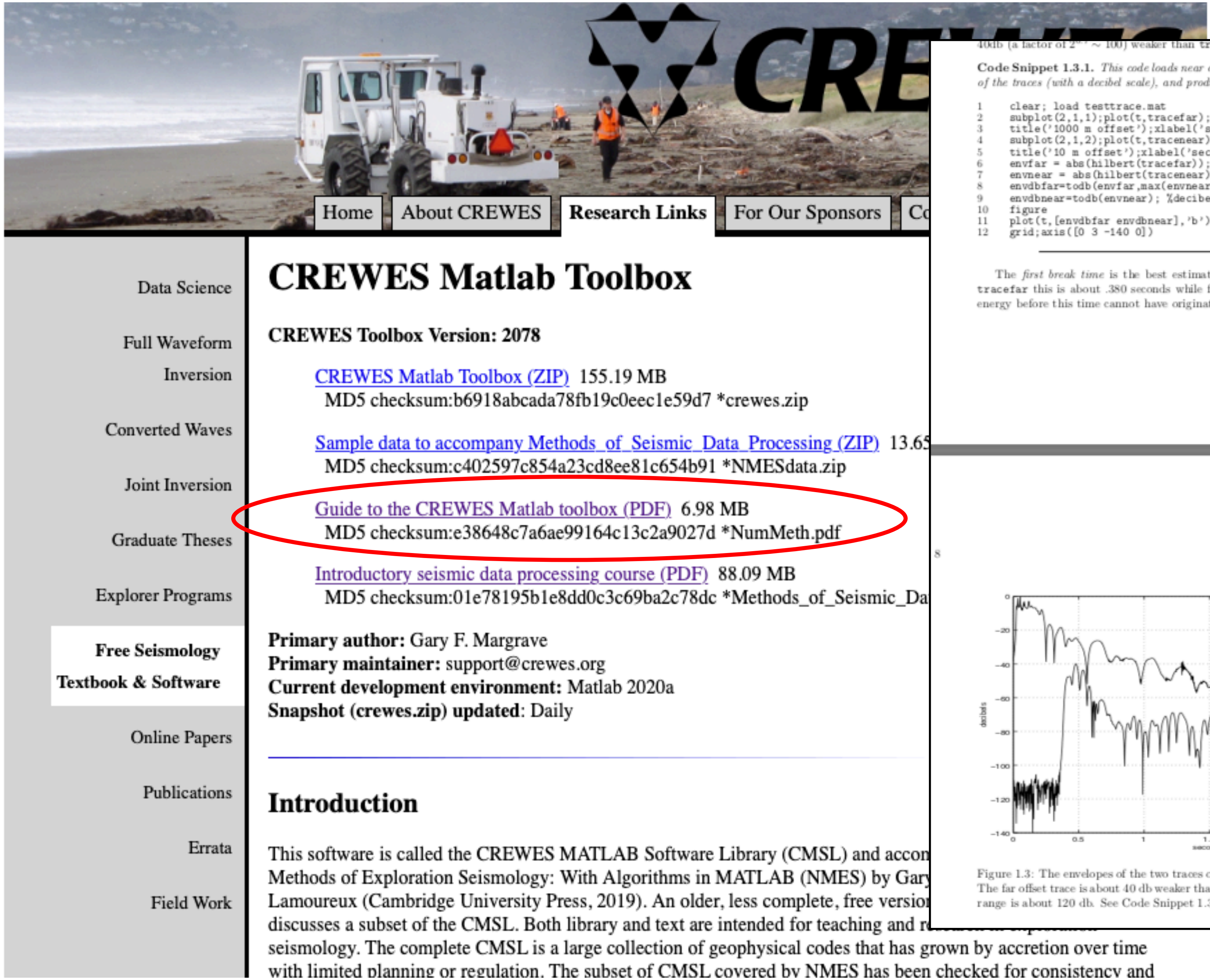
Introductory seismic data processing course (PDF) 88.09 MB
MD5 checksum:01e78195b1e8dd0c3c69ba2c78dc *Methods_of_Seismic_Data_Processing.pdf

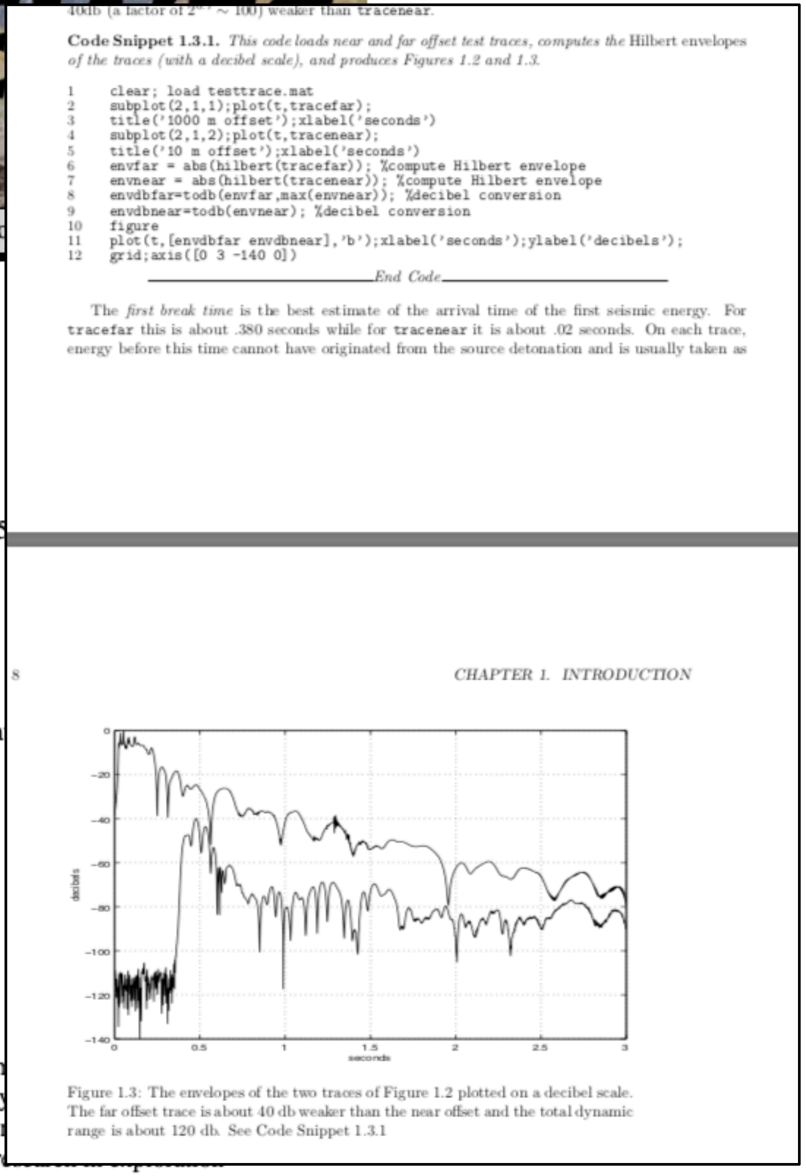
Primary author: Gary F. Margrave
Primary maintainer: support@crewes.org
Current development environment: Matlab 2020a
Snapshot (crewes.zip) updated: Daily

Introduction

This software is called the CREWES MATLAB Software Library (CMSL) and accompanies the textbook Numerical Methods of Exploration Seismology: With Algorithms in MATLAB (NMES) by Gary F. Margrave and Michael P. Lamoureux (Cambridge University Press, 2019). An older, less complete, free version is provided here. The textbook discusses a subset of the CMSL. Both library and text are intended for teaching and research in exploration seismology. The complete CMSL is a large collection of geophysical codes that has grown by accretion over time with limited planning or regulation. The subset of CMSL covered by NMES has been checked for consistency and

The CREWES Matlab toolbox *(www.crewes.org)*





40db (a factor of $2^{6.7} \sim 100$) weaker than tracenear.

Code Snippet 1.3.1. *This code loads near and far offset test traces, computes the* Hilbert envelopes *of the traces (with a decibel scale), and produces Figures 1.2 and 1.3.*

```
clear; load testtrace.mat
subplot(2,1,1);plot(t,tracefar);
title('1000 m offset');xlabel('seconds')
subplot(2,1,2);plot(t,tracenear);
title('10 m offset');xlabel('seconds')
envfar = abs(hilbert(tracefar)); %compute Hilbert envelope
envnear = abs(hilbert(tracenear)); %compute Hilbert envelope
envdbfar=todb(envfar,max(envnear)); %decibel conversion
envdbnear=todb(envnear); %decibel conversion
figure
plot(t,[envdbfar envdbnear],'b');xlabel('seconds');ylabel('decibels');
grid;axis([0 3 -140 0])
```

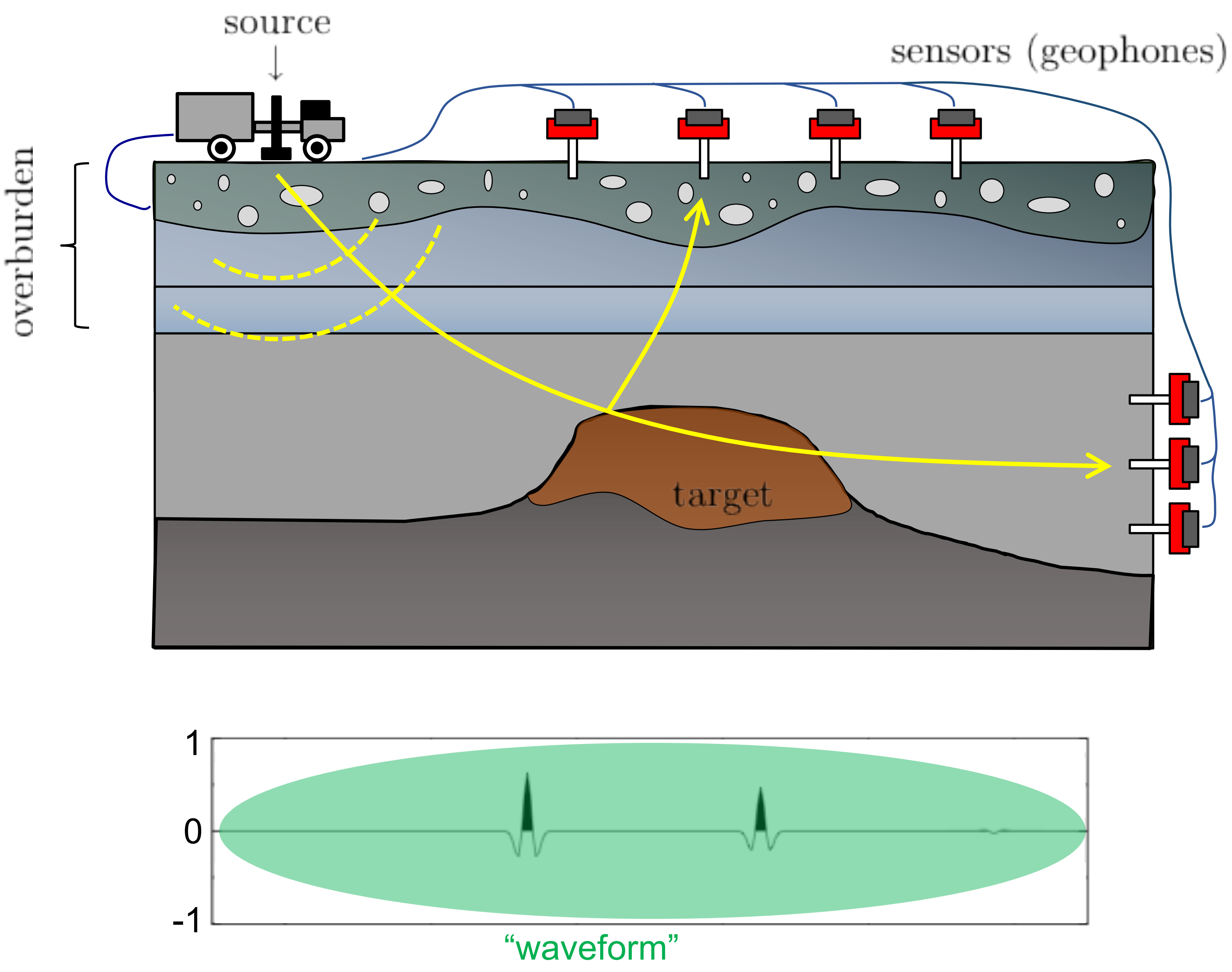
End Code

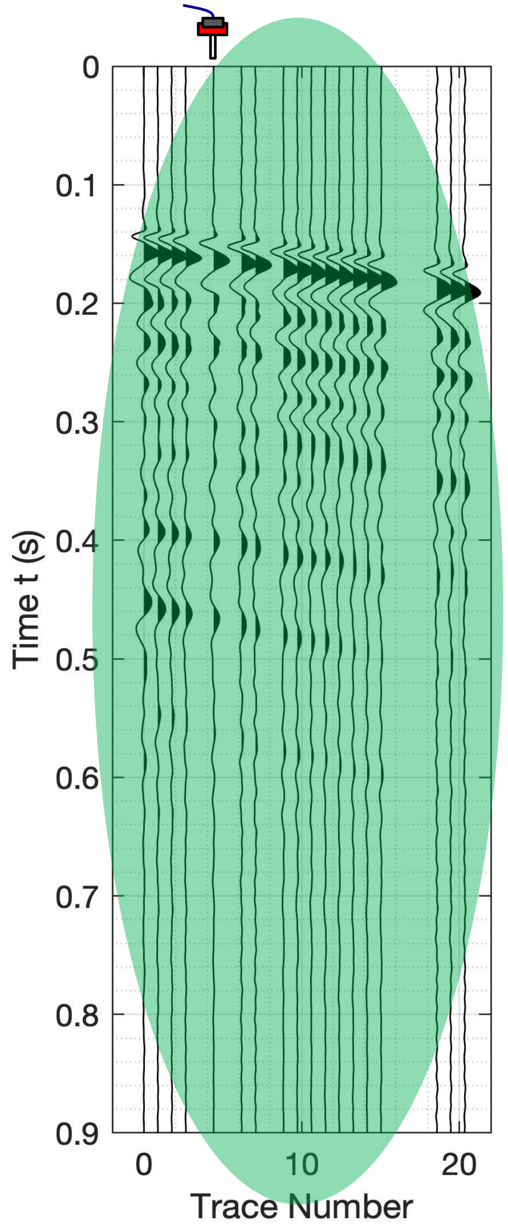
The *first break time* is the best estimate of the arrival time of the first seismic energy. For tracefar this is about .380 seconds while for tracenear it is about .02 seconds. On each trace, energy before this time cannot have originated from the source detonation and is usually taken as

8 CHAPTER 1. INTRODUCTION

Figure 1.3: The envelopes of the two traces of Figure 1.2 plotted on a decibel scale. The far offset trace is about 40 db weaker than the near offset and the total dynamic range is about 120 db. See Code Snippet 1.3.1

Full waveform inversion – concept and technology





Full waveform inversion – technology impact

BP plans for significant growth in deepwater Gulf of Mexico

8 January 2019

- Approval of major expansion at Atlantis field supports strategy of growing advantaged oil production around existing production hubs.
- Recent BP breakthrough in seismic imaging identifies 1 billion barrels of additional oil in place at Thunder Horse field.
- New discoveries near Na Kika platform provide additional development opportunities.
- BP expects to grow net Gulf production to around 400,000 boe/d in the next decade.

HOUSTON – BP announced today that it has approved a major expansion at the Atlantis field in the U.S. Gulf of Mexico and has also identified significant additional oil resources that could create further development opportunities around the production hubs it operates in the region.

The $1.3 billion Atlantis Phase 3 development is the latest example of BP's strategy of growing advantaged oil production through its existing production facilities in the Gulf. The approval for this latest development comes after recent BP breakthroughs in advanced seismic imaging and reservoir characterization revealed an additional 400 million barrels of oil in place at the Atlantis field.

BP approves Atlantis expansion

Atlantis Phase 3 will include the construction of a new subsea production system from eight new wells that will be tied into the current platform, 150 miles south of New Orleans. Scheduled to come onstream in 2020, the project is expected to boost production at the platform by an estimated 38,000 barrels of oil equivalent a day (boe/d) gross at its peak. It will also access the eastern area of the field where the advanced imaging and reservoir characterization identified additional oil in place.

"Atlantis Phase 3 shows how our latest technologies and digital techniques create real value – identifying opportunities, driving efficiencies and enabling the delivery of major projects. Developments like this are building an exciting future for our business in the Gulf," said Starlee Sykes, BP's regional president for the Gulf of Mexico and Canada.

Advanced seismic imaging boosts Thunder Horse resources

The proprietary algorithms developed by BP enhance a seismic imaging technique known as Full Waveform Inversion (FWI), allowing seismic data that would have previously taken a year to analyze to be processed in only a few weeks. Application of this technology and reservoir characterization has now identified a further 1 billion barrels of oil in place at the Thunder Horse field.

BP's leadership in seismic acquisition and imaging is a result of sustained investment in technology and high-performance computing. Following a successful field trial at the Mad Dog field, further advanced seismic imaging with ocean bottom nodes and BP's proprietary Wolfspar seismic acquisition source is being planned for Thunder Horse and Atlantis to better understand the reservoirs. Wolfspar uses ultra-low frequencies during seismic surveys, allowing geophysicists to see deeper below salt layers and enabling better planning of where to drill wells.

Source: BP.com 2019

Full waveform inversion – tenets and concepts

Tenets

- Explain each datum in terms of a field which satisfies a partial differential equation
- Reduce extraction of secondary data (e.g., traveltimes) to a minimum
- Engage tools of local, iterative numerical optimization

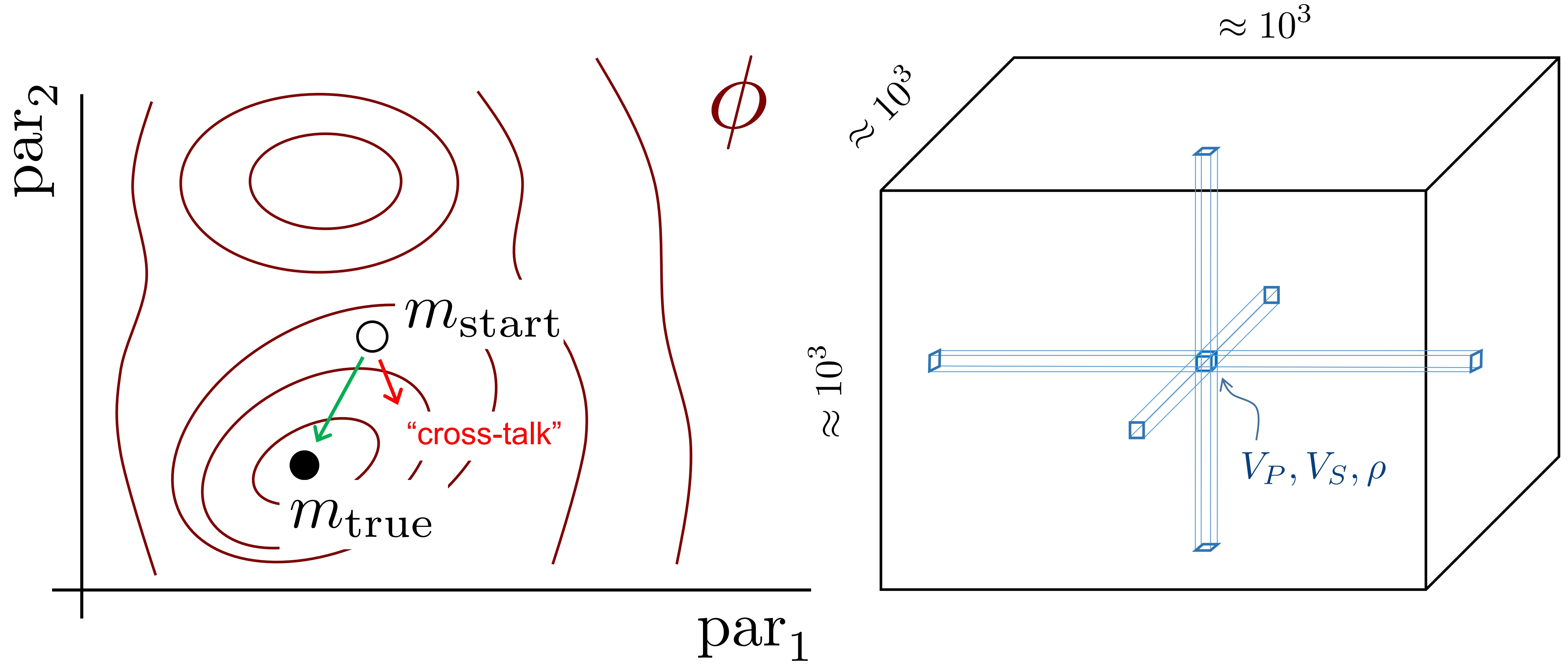
Tasks

- Accurate/efficient wavefield simulation – 3D elastodynamic FD or FE
- Algorithms with minimal simulations
- Design updates: physics, intuition, parameterization, data usage, optimization

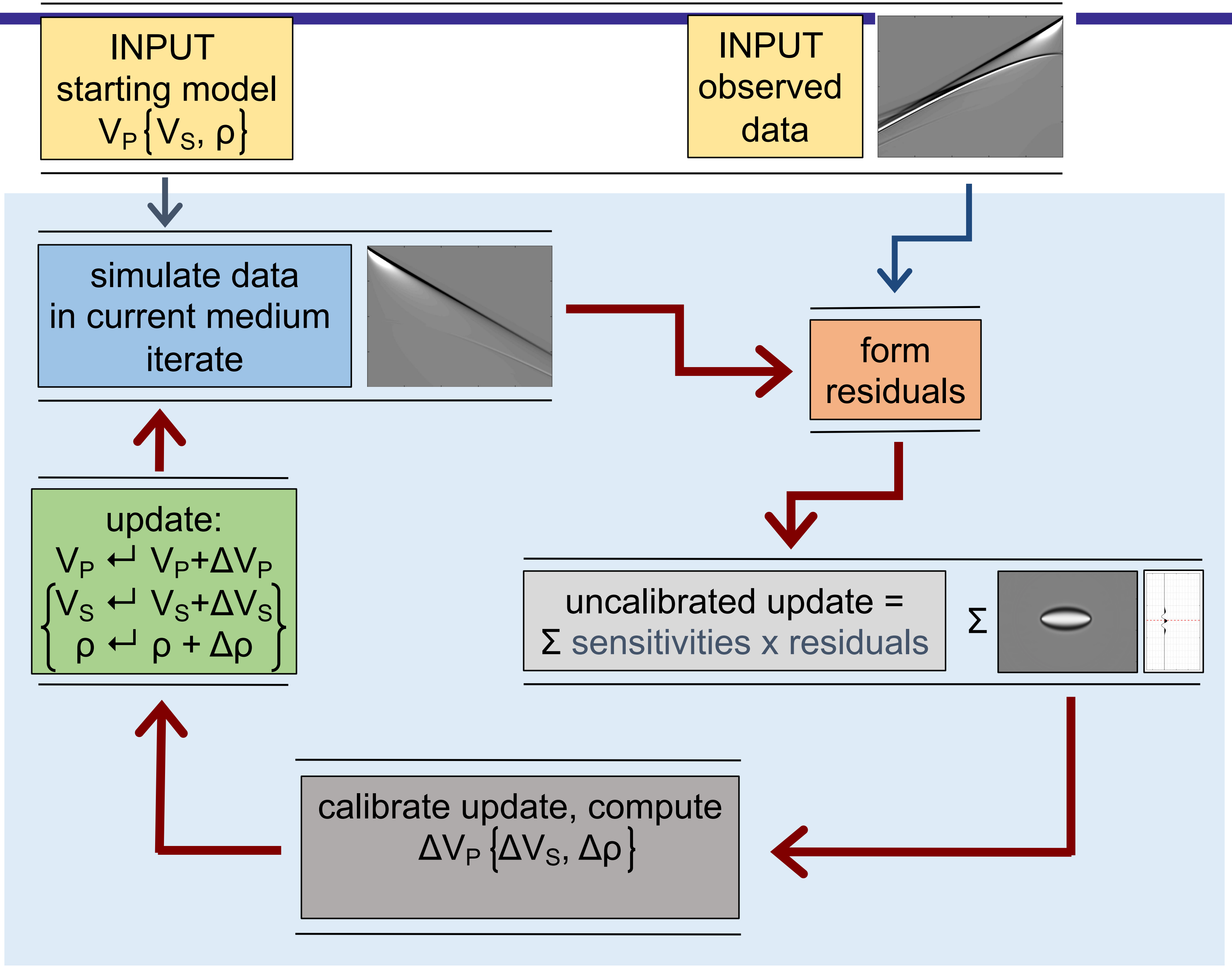
Challenges

- Computational expense
- Data incompleteness
- Parameter trade-offs / confusion

Full waveform inversion – tenets and concepts



Full waveform inversion – tenets and concepts



Full waveform inversion – tenets and concepts

Tenets

- Explain each datum in terms of a field which satisfies a partial differential equation
- Reduce extraction of secondary data (e.g., traveltimes) to a minimum
- Engage tools of local, iterative numerical optimization

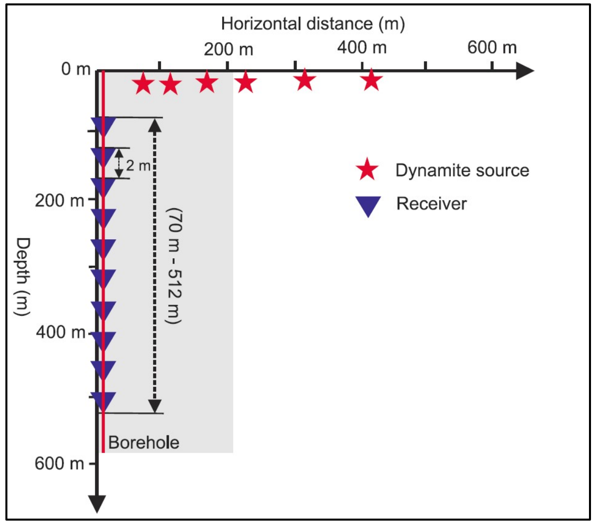
Tasks

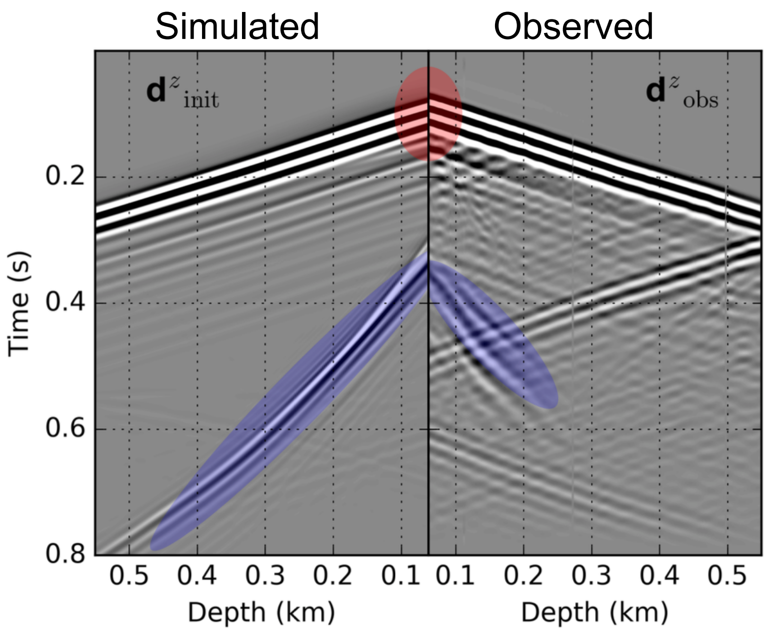
- Accurate/efficient wavefield simulation – 3D elastodynamic FD or FE
- Algorithms with minimal simulations
- Design updates: physics, intuition, parameterization, data usage, optimization

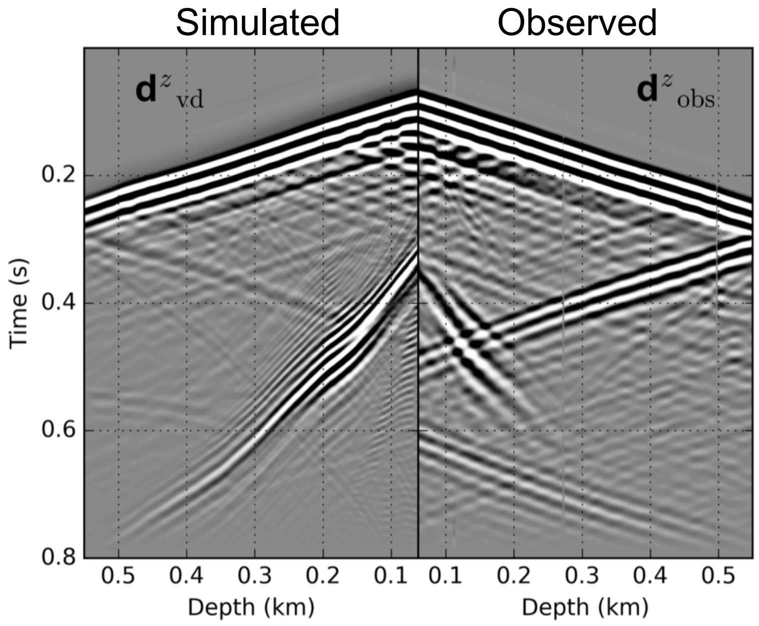
Challenges

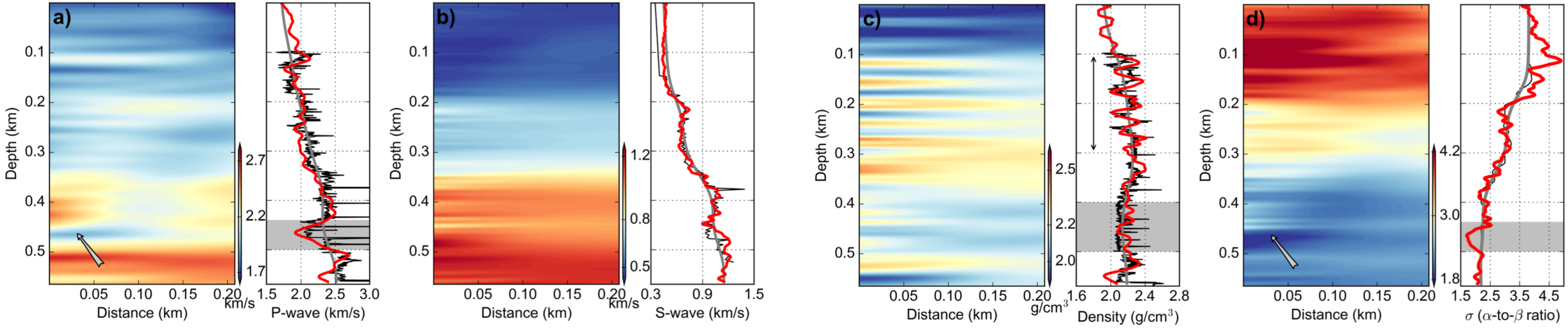
- Computational expense
- Data incompleteness
- Parameter trade-offs / confusion

CREWES FWI research projects – field cases









W. Pan, K. A. Innanen and Y. Geng, *Elastic FWI and parameterization analysis applied to walk-away vertical seismic profile data for unconventional (heavy oil) reservoir characterization*, 2018: **GJI**, 213, 3, 1934-1968.

Full waveform inversion – tenets and concepts

Tenets

- Explain each datum in terms of a field which satisfies a partial differential equation
- Reduce extraction of secondary data (e.g., traveltimes) to a minimum
- Engage tools of local, iterative numerical optimization

Tasks

- Accurate/efficient wavefield simulation – 3D elastodynamic FD or FE
- Algorithms with minimal simulations
- Design updates: physics, intuition, parameterization, data usage, optimization

Challenges

- Computational expense
- Data incompleteness
- Parameter trade-offs / confusion

Can be addressed with Matlab implementations of FWI

FWI in Matlab – code setup for a simple implementation

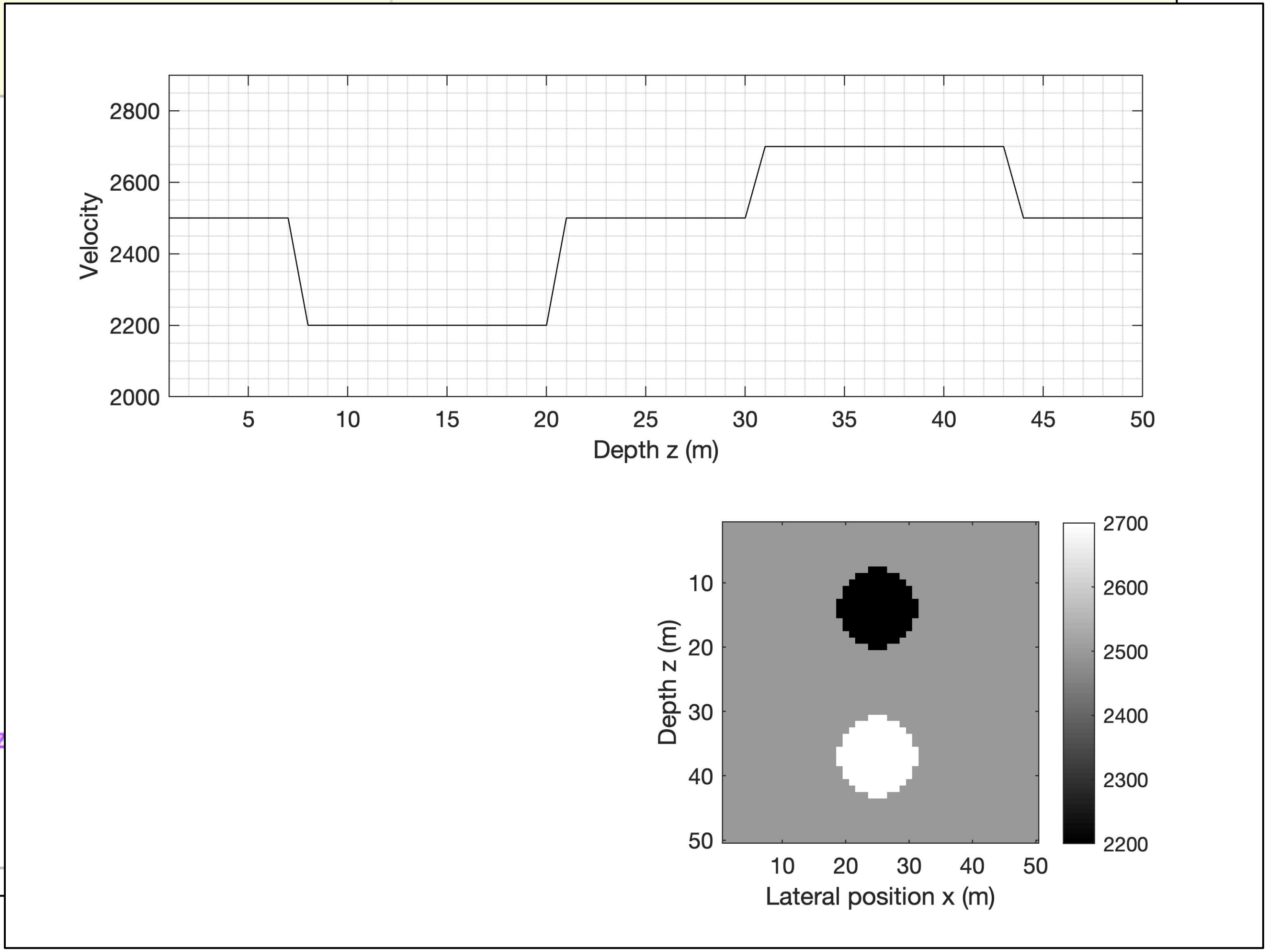
mod6_1_FWI_fdomain_DRIVER.m

```
% Module 6.  "mod6_1_FWI_fdomain_DRIVER.m"
% 2D 1-parameter acoustic frequency domain FWI with synthetic
% data and choosable model; full tour.  Author: S. Keating 2018, small
% additions by K. Innanen, 2019.

clear; close all;
plot_model_only = 'y';

%% 1. Select velocity model
crewes_FWI_fdomain_BALLMODEL;

profileindex = 25;
veltruemat = reshape(vel_true,nz,nx);
veltrueprofile = veltruemat(:,profileindex);

if ( plot_model_only == 'y' )
    %
    figure,
    subplot(2,2,[1 2]),
    plot(1:nz,veltrueprofile,'k-');
    axis([1,50,2000,2900]);
    grid; grid minor;
    xlabel('Depth z (m)'); ylabel('Velocity');
    subplot(2,2,4),
    imagesc(veltruemat); colormap('gray'); colorbar;
    xlabel('Lateral position x (m)'); ylabel('Depth z (m)');
    return
end

%% 2. Set up numerical parameters
```

FWI in Matlab – code setup for a simple implementation

mod6_1_FWI_fdomain_DRIVER.m

```
% Module 6.  "mod6_1_FWI_fdomain_DRIVER.m"
% 2D 1-parameter acoustic frequency domain FWI with synthetic
% data and choosable model; full tour.  Author: S. Keating 2018, small
% additions by K. Innanen, 2019.

clear; close all;
plot_model_only = 'y';

%% 1. Select velocity model
crewes_FWI_fdomain_BALLMODEL;

profileindex = 25;
veltruemat = reshape(vel_true,nz,nx);
veltrueprofile = veltruemat(:,profileindex);

if ( plot_model_only == 'y' )
    %
    figure,
    subplot(2,2,[1 2]),
    plot(1:nz,veltrueprofile,'k-');
    axis([1,50,2000,2900]);
    grid; grid minor;
    xlabel('Depth z (m)'); ylabel('Velocity');
    subplot(2,2,4),
    imagesc(veltruemat); colormap('gray'); colorbar;
    xlabel('Lateral position x (m)'); ylabel('Depth z
    return
end

%% 2. Set up numerical parameters
```



FWI in Matlab – code setup for a simple implementation

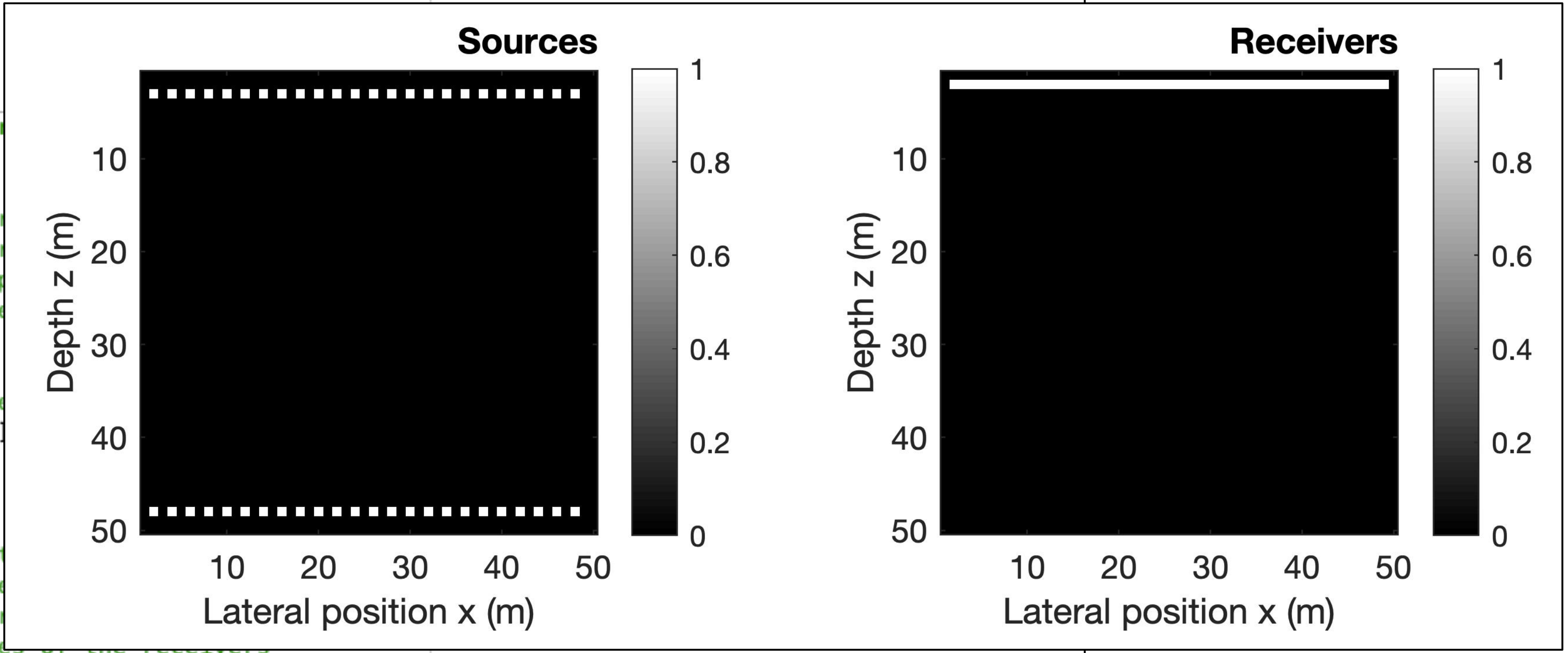
mod6_1_FWI_fdomain_DRIVER.m

```
end

%% 2. Set up numerical parameters

% Perfectly matched layer parameters
PML_thick = 9;                          % thickness of boundary layers (9-21 good values)
R_theor = 1e-3;                         % theoretical reflection coefficient (1e-3 or 1e-4)

% Additional quantities
NPML = (nz+2*PML_thick)*(nx+2*PML_thick);
vel0 = vel_initial(:);
vel_true = vel_true(:);

%% 3. Set up acquisition (sources and receivers)

% Source coordinates (units of grid number)
s_inter=2;                          % source interval
s_initial=2;                        % initial source position
s_end=nx-2;                         % end source position
sx=s_initial:s_inter:s_end;         % x-coordinates of the sources
sz = 3*ones(size(sx));

% If desired, duplicate the top sources at the bottom
sz = [3*ones(size(sx)), (nz-2)*ones(size(sx))];
sx = [sx,sx];

% Source coordinates (units of grid number)
r_inter=1;                          % receiver interval
r_initial=2;                        % initial receiver position
r_end=nx-1;                         % end receiver position
rx=r_initial:r_inter:r_end;         % x-coordinates of the receivers
rz = 2*ones(size(rx));

% Bring source and receiver simulations into modelling environment
[S,R] = crewes_FWI_fdomain_DEFINEACQUISITION(sz, sx, rz, rx, nz, nx, PML_thick);

%% 4. Set up frequencies
```

FWI in Matlab – code setup for a simple implementation

mod6_1_FWI_fdomain_DRIVER.m

```
    end

    %% 2. Set up numerical parameters

    % Perfectly matched layer parameters
    PML_thick = 9;                          % thickness of boundary layers (9-21 good values)
    R_theor = 1e-3;                         % theoretical reflection coefficient (1e-3 or 1e-4)

    % Additional quantities
    NPML = (nz+2*PML_thick)*(nx+2*PML_thick);
    vel0 = vel_initial(:);
    vel_true = vel_true(:);

    %% 3. Set up acquisition (sources and receiver

    % Source coordinates (units of grid number)
    s_inter=2;                     % source inte
    s_initial=2;                   % initial sou
    s_end=nx-2;                    % end source p
    sx=s_initial:s_inter:s_end;    % x-coordinate
    sz = 3*ones(size(sx));

    % If desired, duplicate the top sources at the
    sz = [3*ones(size(sx)), (nz-2)*ones(size(sx))]
    sx = [sx,sx];

    % Source coordinates (units of grid number)
    r_inter=1;                     % receiver int
    r_initial=2;                   % initial rece
    r_end=nx-1;                    % end receiver
    rx=r_initial:r_inter:r_end;    % x-coordinate
    rz = 2*ones(size(rx));

    % Bring source and receiver simulations into modelling environment
    [S,R] = crewes_FWI_fdomain_DEFINEACQUISITION(sz, sx, rz, rx, nz, nx, PML_thick);

    %% 4. Set up frequencies
```



FWI in Matlab – code setup for a simple implementation

mod6_1_FWI_fdomain_DRIVER.m

```matlab

%% 4. Set up frequencies

% Inversion (& modelling) occurs in bands.
numbands = 13;                          % number of bands
step=5;                                 % number of discrete freqs per band
minf = 1.0;                             % minimum frequency used
maxf = 35.0;                            % maximum frequency used

%numbands = 4;                          % number of bands
%step=3;                                 % number of discrete freqs per band
%minf = 1.0;                             % minimum frequency used
%maxf = 20.0;                            % maximum frequency used

% Frequencies gradually 'fan out' from lowest as the iterations proceed.
% Starting set (i.e., 1 of numbands, e.g., 13) 1 of "step" frequencies (step e.g., 5)
% is [minf, minf, minf, minf, ..., minf], e.g., [1,1,1,1,1].  The next
% iteration the frequencies begin fanning out, starting at minf,
% incrementing by a gradually increasing amount, such that by the 13th
% (numband'th) iteration, the frequencies are [1, ..., maxf].  These
% frequency bands are stored in a single vector freq of length
% step*numbands.

freq=zeros(1,numbands*step);                 % initialize
startfreq = minf*ones(1,numbands);           % starting frequencies
endfreq = linspace(minf,maxf,numbands);      % ending frequencies

% Fill in frequency vector.
for n=1:numbands
    freq(1 + (n-1)*step : n*step) = linspace(startfreq(n), endfreq(n), step);
end

% Include a wavelet amplitude spectral weight if desired
fwave=ones(1,length(freq));             % No weight
%fwave = exp(-((freq - 20)/10).^2);     % Exponential weight

%% 5. Set up the finite difference function to be called to create data and residuals
```

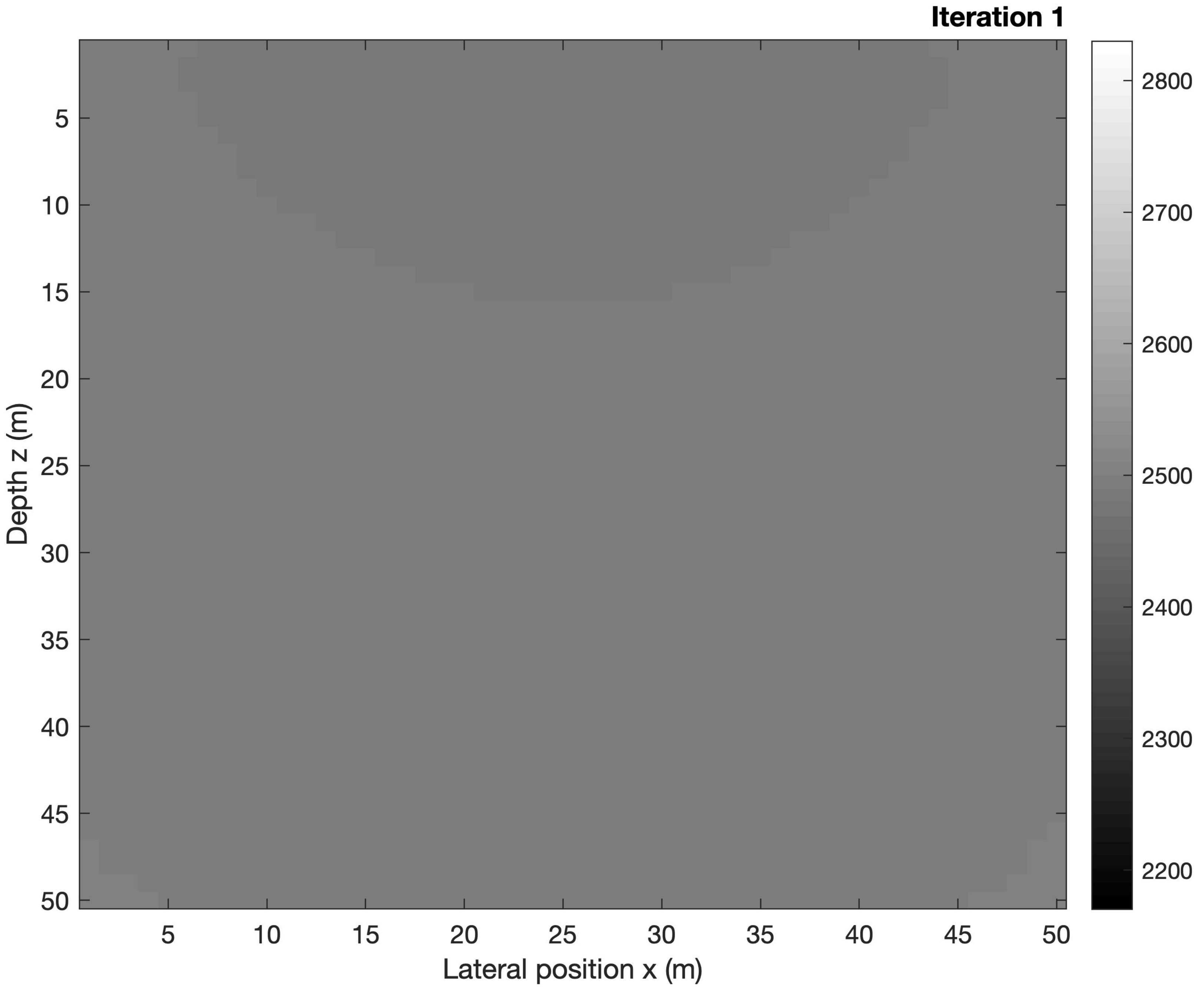
FWI in Matlab – code setup for a simple implementation

mod6_1_FWI_fdomain_DRIVER.m

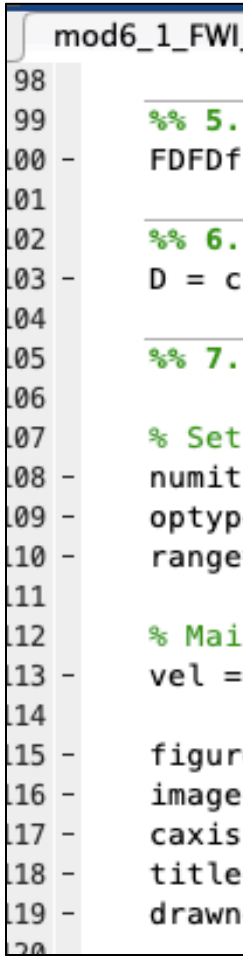
```
%% 5. Set up the finite difference function to be called to create data and residuals
FDFDfunc = @(frequency,fwave,vel)crewes_FWI_fdomain_FDFD(vel,vel0,frequency,S,fwave,PML_thick,R_theor,nz,nx,dz,dx);

%% 6. Use the finite difference function and the actual model to generate data
D = crewes_FWI_fdomain_GETDATA( FDFDfunc, freq, fwave, vel_true, R );

%% 7. Carry out FWI iterations

% Set up FWI iteration parameters
numits = 2;                             % Number of iterations per freq band
optype = 1;                             % Optimization: 1 = Steepest Descent; 2 = Gauss-Newton
rangevel = 1.1*max(abs(vel_true-vel0));  % For plotting purposes within the inversion function

% Main FWI code
vel = crewes_FWI_fdomain_FDFWI( D, freq, step, fwave, FDFDfunc, nz, nx, vel0, R, optype, numits, PML_thick, rangevel);

figure(floor(rand*10000))
imagesc(reshape(vel_true,nz,nx));
caxis([mean(vel0)-rangevel,mean(vel0)+rangevel])
title('True model')
drawnow
```

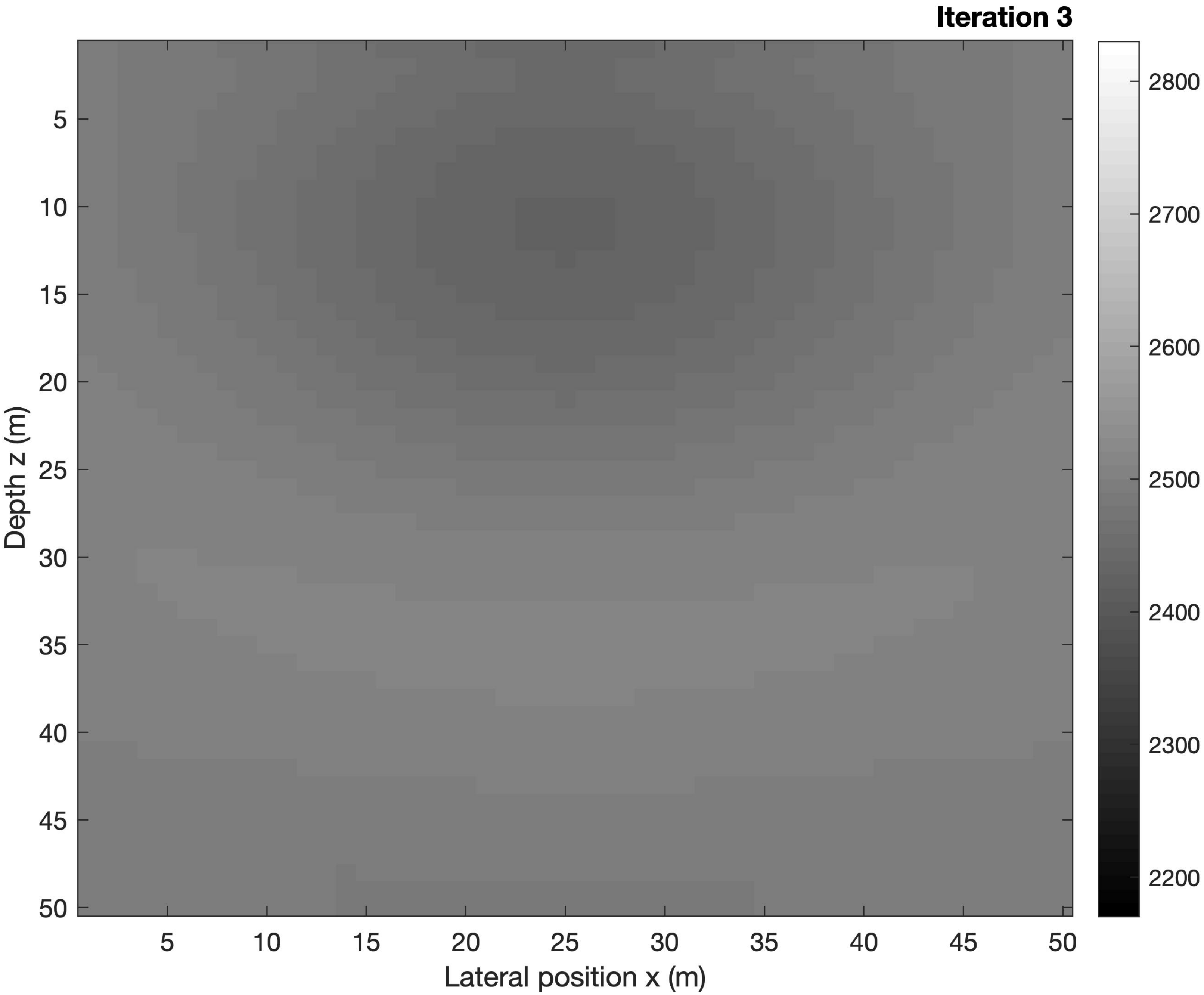
FWI

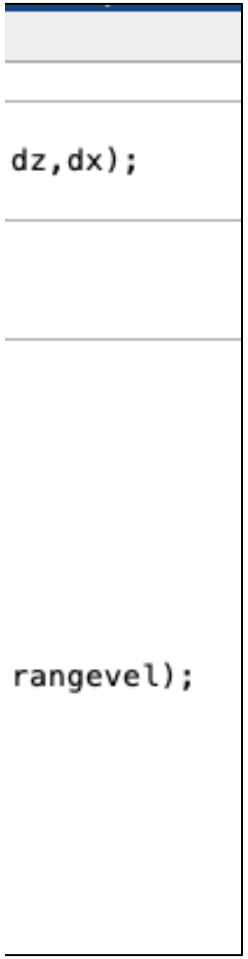
```
mod6_1_FWI_
98
99      %% 5.
100 -   FDFDf                                   dz,dx);
101
102     %% 6.
103 -   D = c
104
105     %% 7.
106
107     % Set
108 -   numit
109 -   optyp
110 -   range
111
112     % Mai
113 -   vel =                                   rangevel);
114
115 -   figur
116 -   image
117 -   caxis
118 -   title
119 -   drawn
```



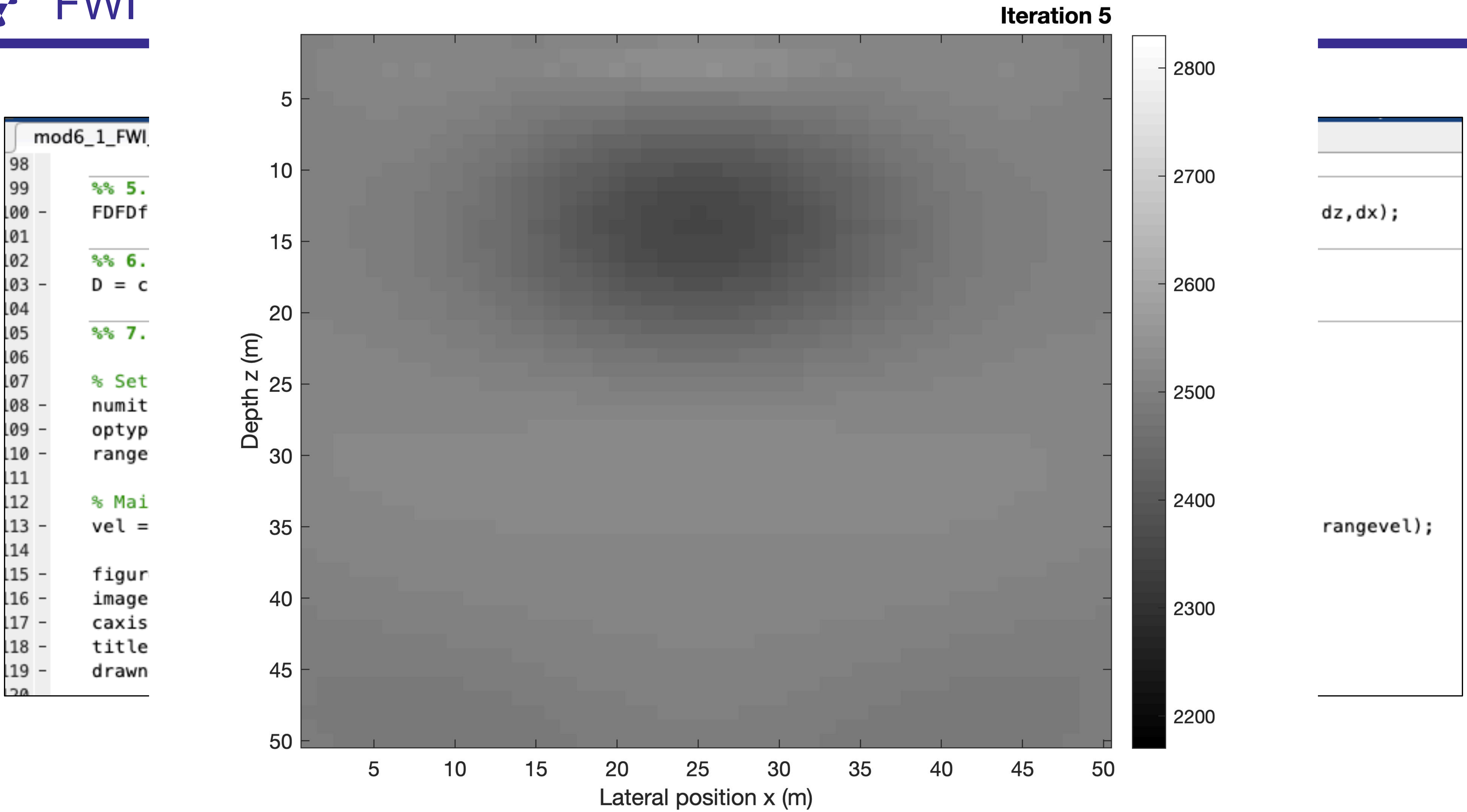
FWI


mod6_1_FWI
%% 5.
FDFDf
%% 6.
D = c
%% 7.
% Set
numit
optyp
range
% Mai
vel =
figur
image
caxis
title
drawn

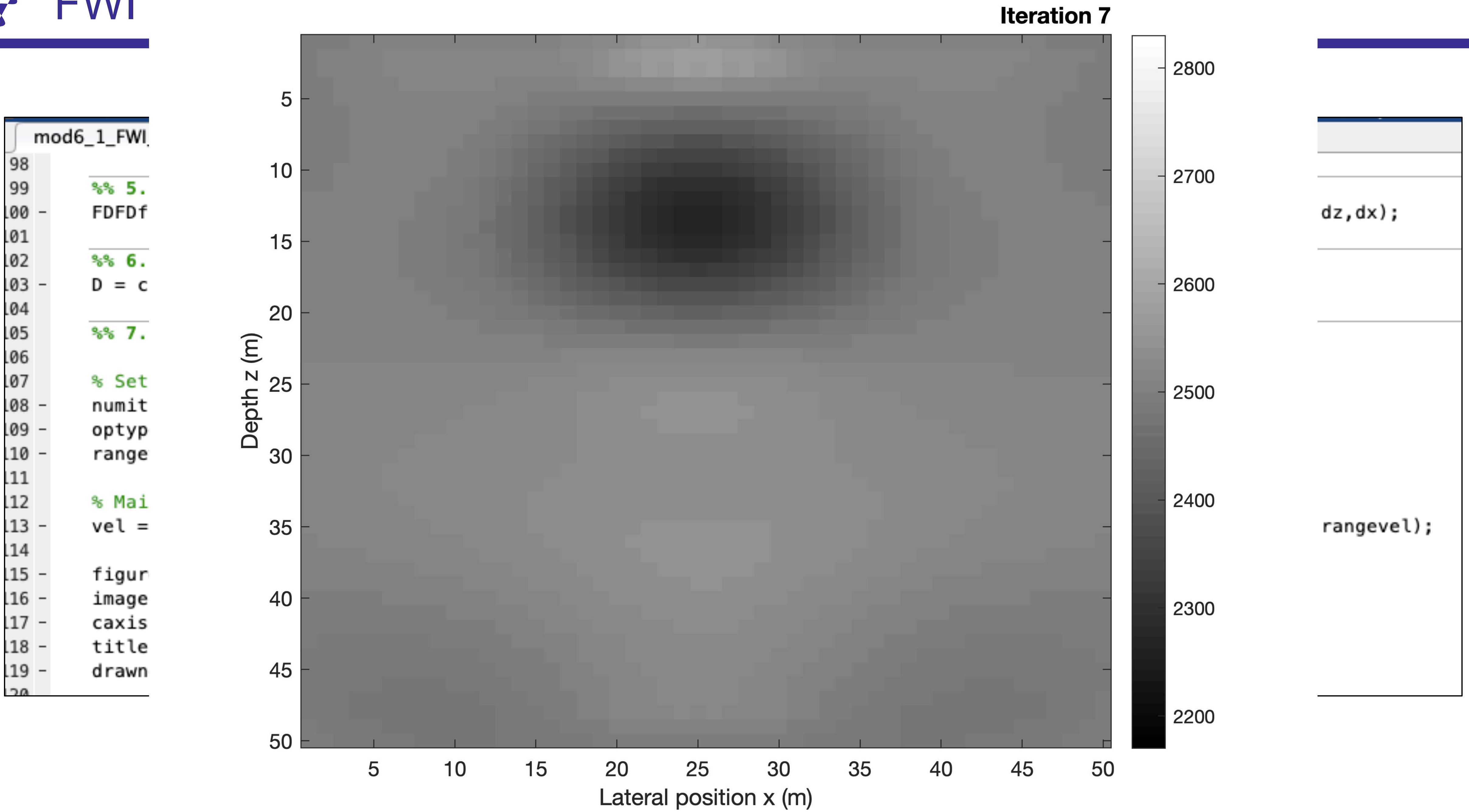

Iteration 3
Depth z (m)
Lateral position x (m)
2800
2700
2600
2500
2400
2300
2200


dz,dx);
rangevel);

FWI


Iteration 5
Depth z (m)
Lateral position x (m)
5
10
15
20
25
30
35
40
45
50
2800
2700
2600
2500
2400
2300
2200
mod6_1_FWI
%% 5.
FDFDf
%% 6.
D = c
%% 7.
% Set
numit
optyp
range
% Mai
vel =
figur
image
caxis
title
drawn
dz,dx);
rangevel);

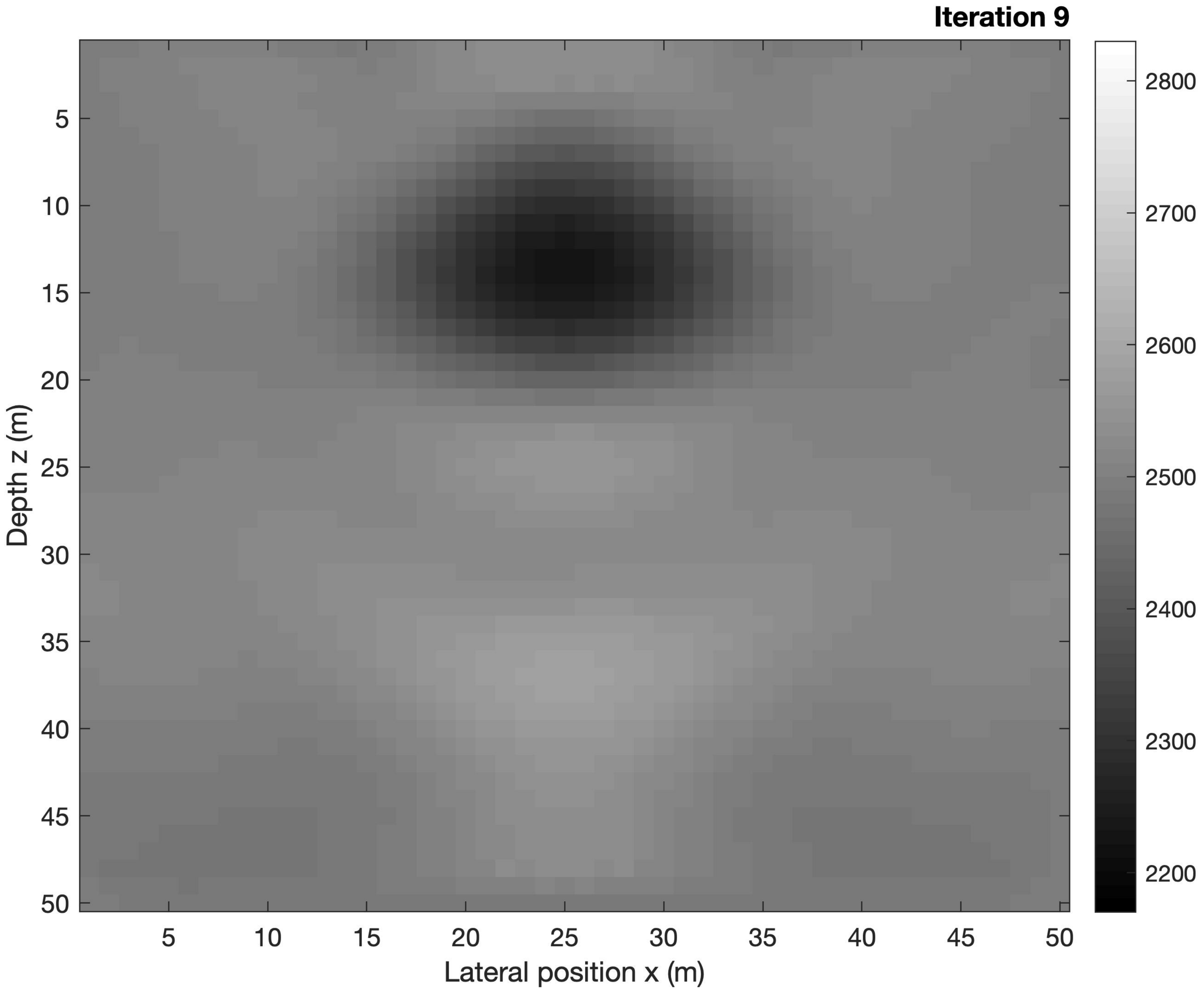
FWI


mod6_1_FWI_
%% 5.
FDFDf
%% 6.
D = c
%% 7.
% Set
numit
optyp
range
% Mai
vel =
figur
image
caxis
title
drawn
dz,dx);
rangevel);
Iteration 7
Depth z (m)
Lateral position x (m)
2800
2700
2600
2500
2400
2300
2200

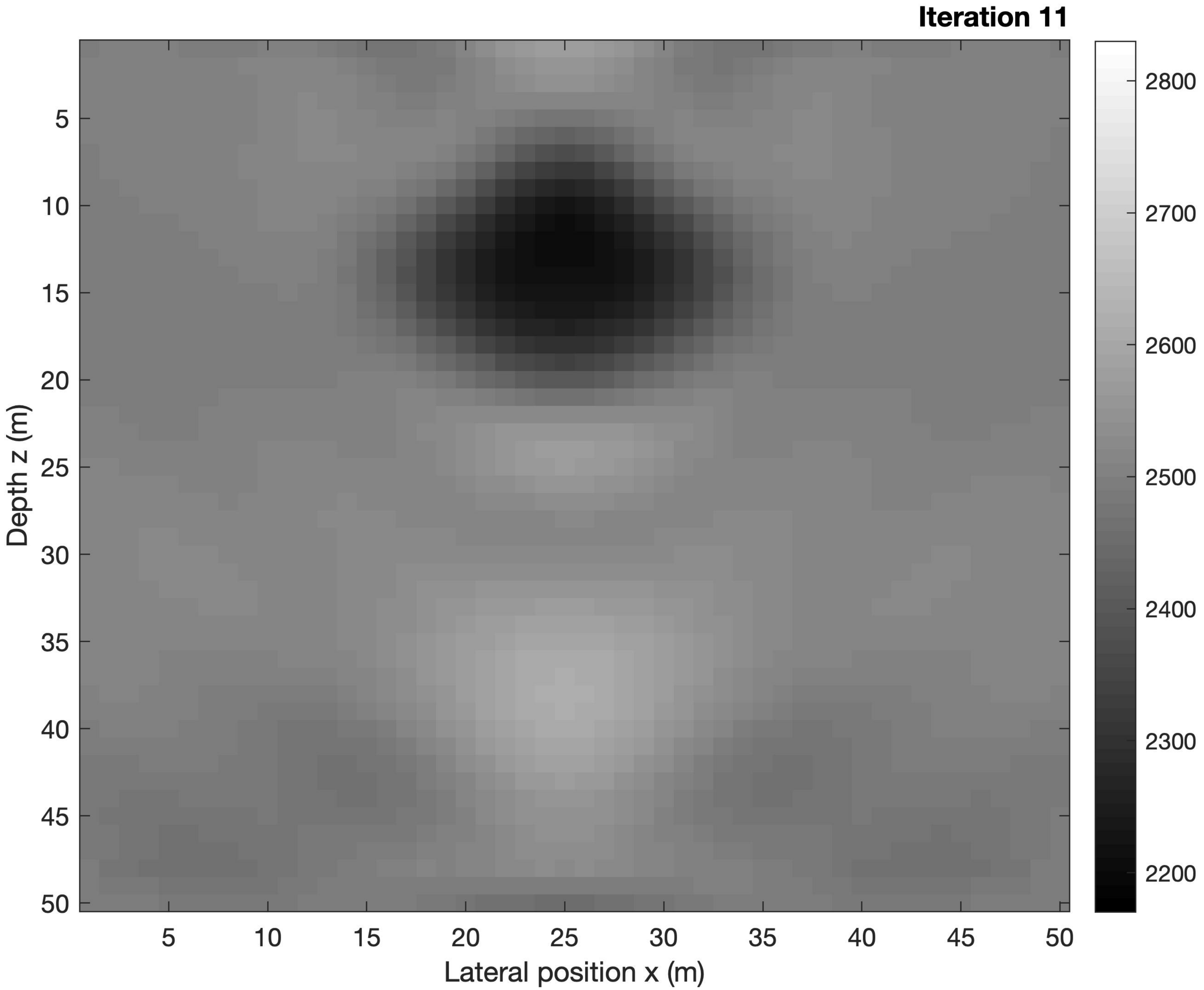
FWI

mod6_1_FWI

```
%% 5.
FDFDf                                        dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                                        rangevel);

figur
image
caxis
title
drawn
```



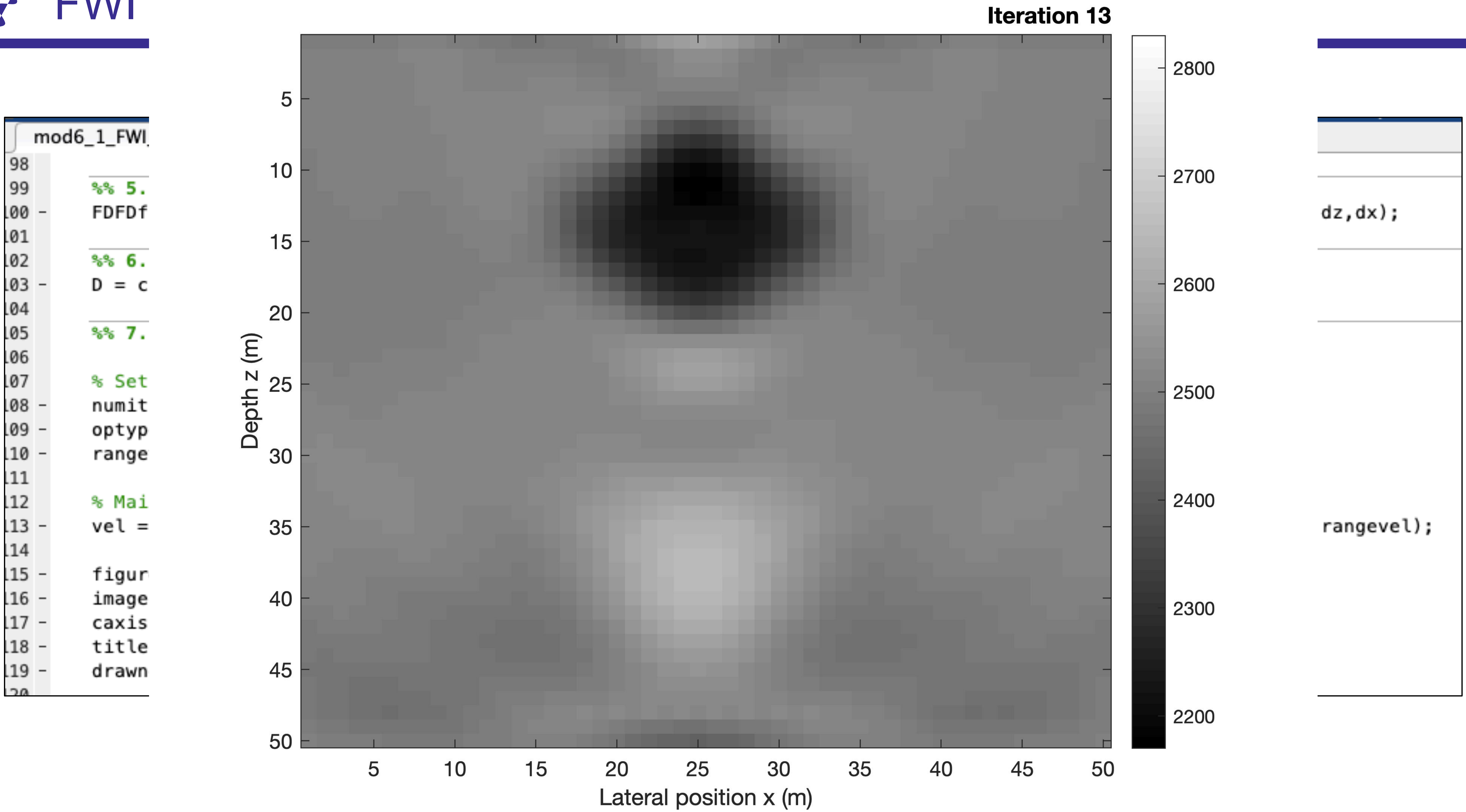
FWI



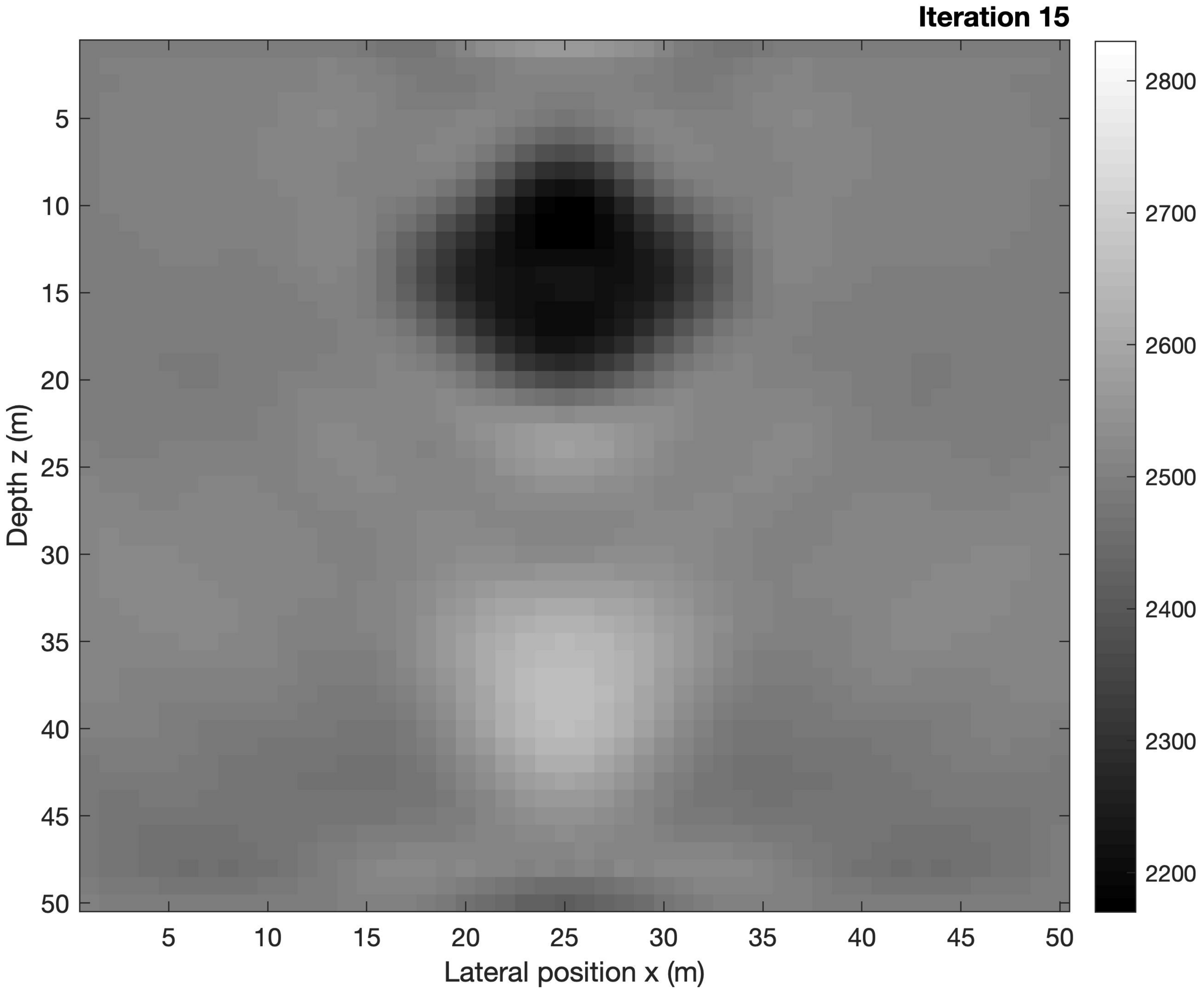
mod6_1_FWI

```
%% 5.
FDFDf                                   dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                                   rangevel);

figur
image
caxis
title
drawn
```

FWI


Iteration 13
Depth z (m)
Lateral position x (m)
2800
2700
2600
2500
2400
2300
2200
mod6_1_FWI_
%% 5.
FDFDf
%% 6.
D = c
%% 7.
% Set
numit
optyp
range
% Mai
vel =
figur
image
caxis
title
drawn
dz,dx);
rangevel);

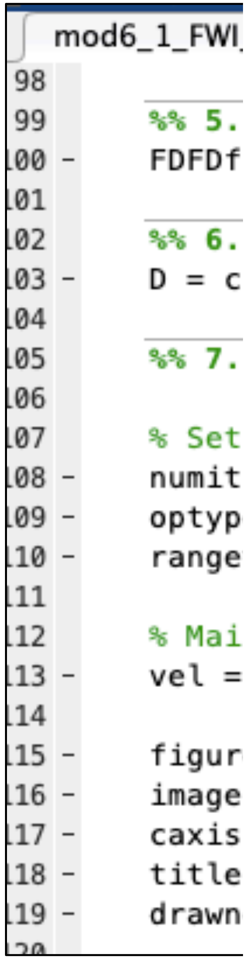
FWI



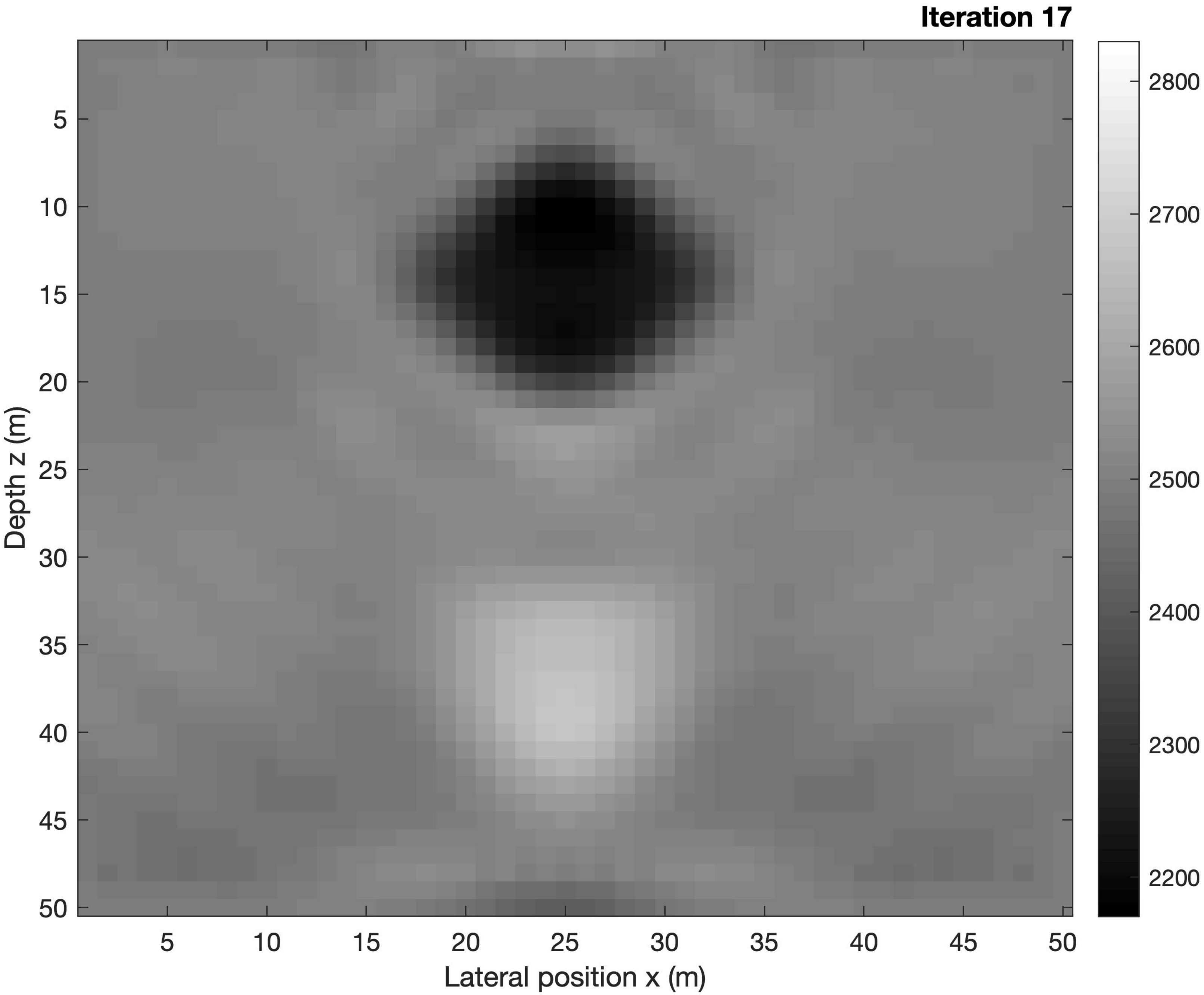
mod6_1_FWI_

```
%% 5.
FDFDf                                   dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                                   rangevel);

figur
image
caxis
title
drawn
```

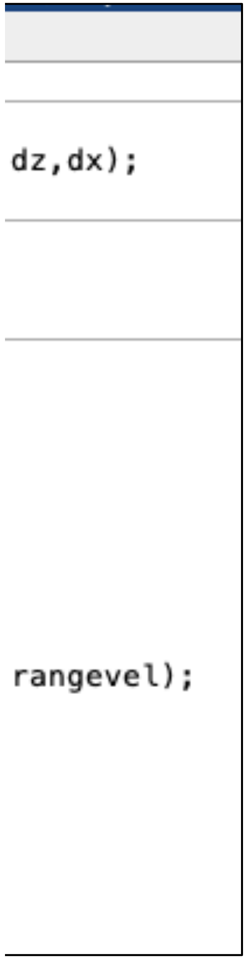
FWI



mod6_1_FWI
98
99 %% 5.
100 - FDFDf
101
102 %% 6.
103 - D = c
104
105 %% 7.
106
107 % Set
108 - numit
109 - optyp
110 - range
111
112 % Mai
113 - vel =
114
115 - figur
116 - image
117 - caxis
118 - title
119 - drawn



Iteration 17
Depth z (m)
Lateral position x (m)
5
10
15
20
25
30
35
40
45
50
2800
2700
2600
2500
2400
2300
2200

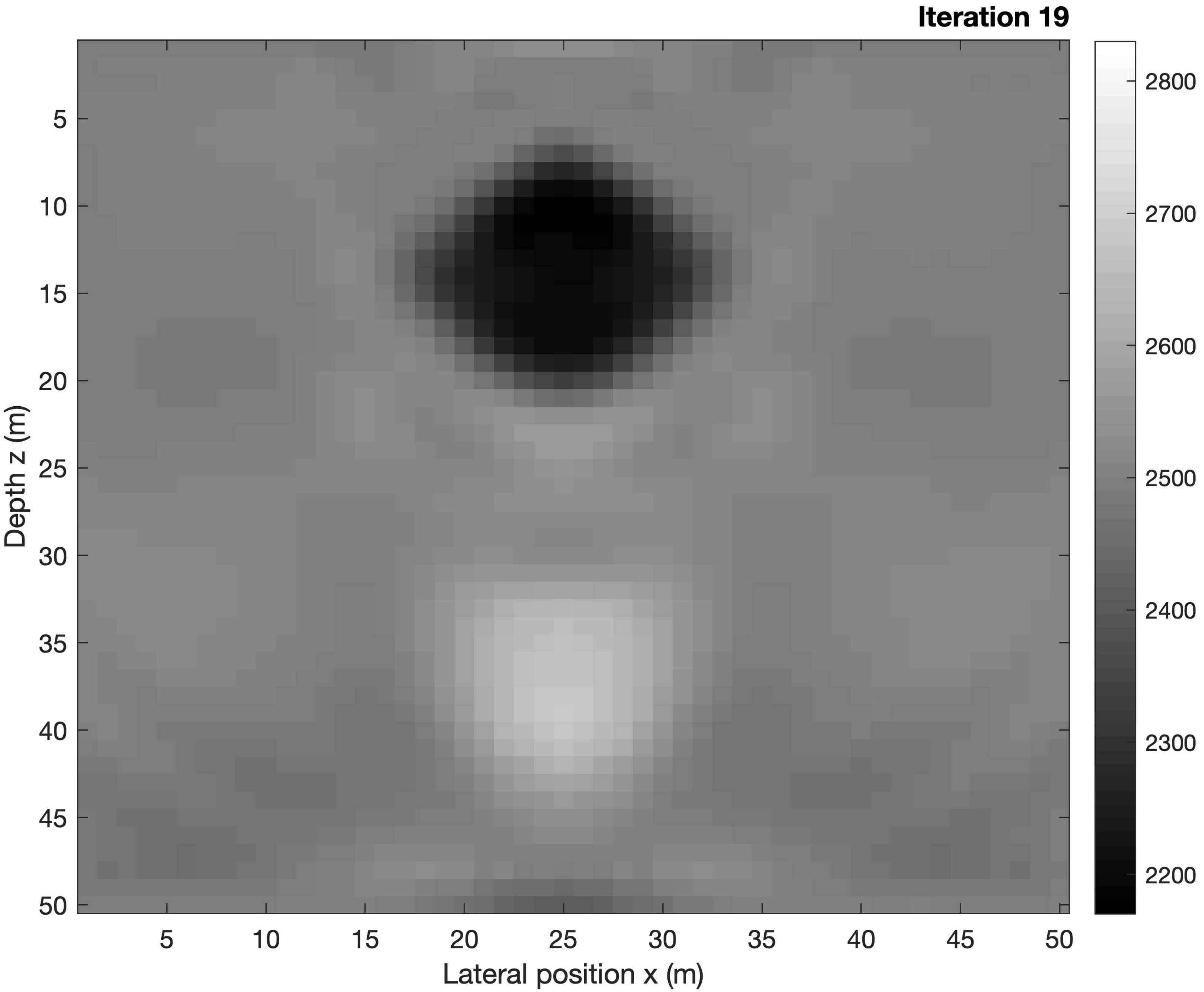


dz,dx);
rangevel);

FWI

mod6_1_FWI

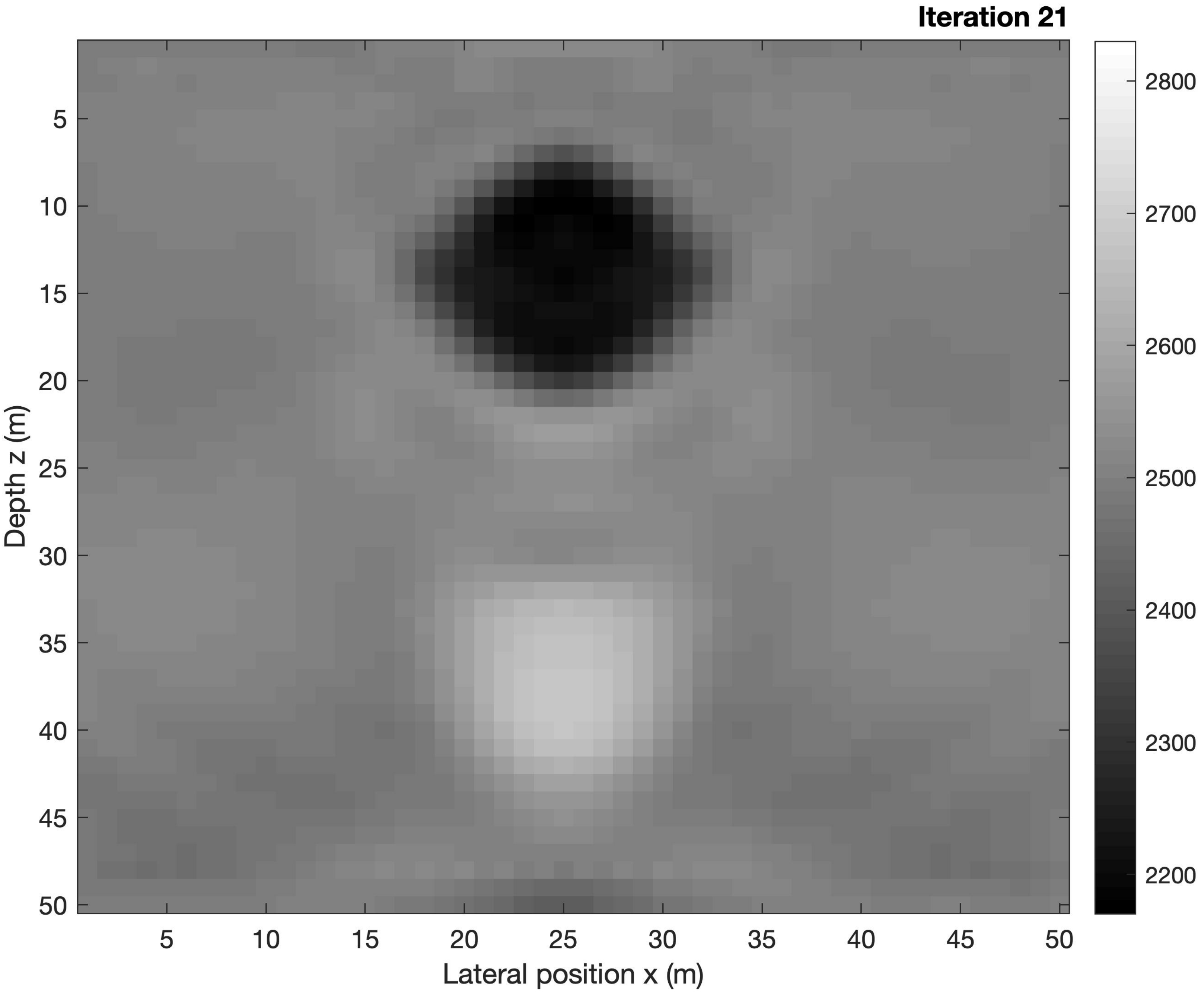
```
%% 5.
FDFDf                dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                rangevel);

figur
image
caxis
title
drawn
```



FWI

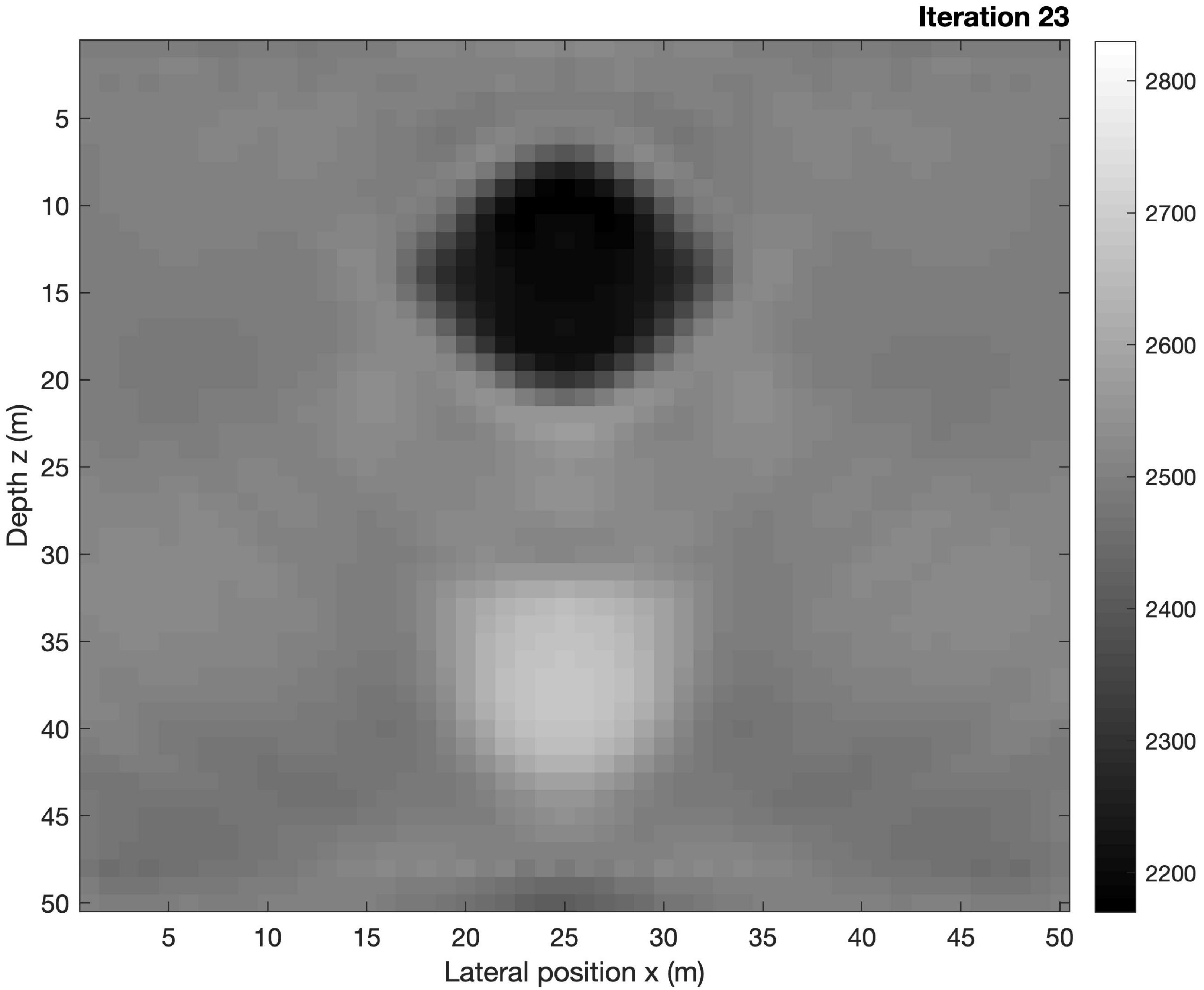
mod6_1_FWI_

```
%% 5.
FDFDf                dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                rangevel);

figur
image
caxis
title
drawn
```



FWI

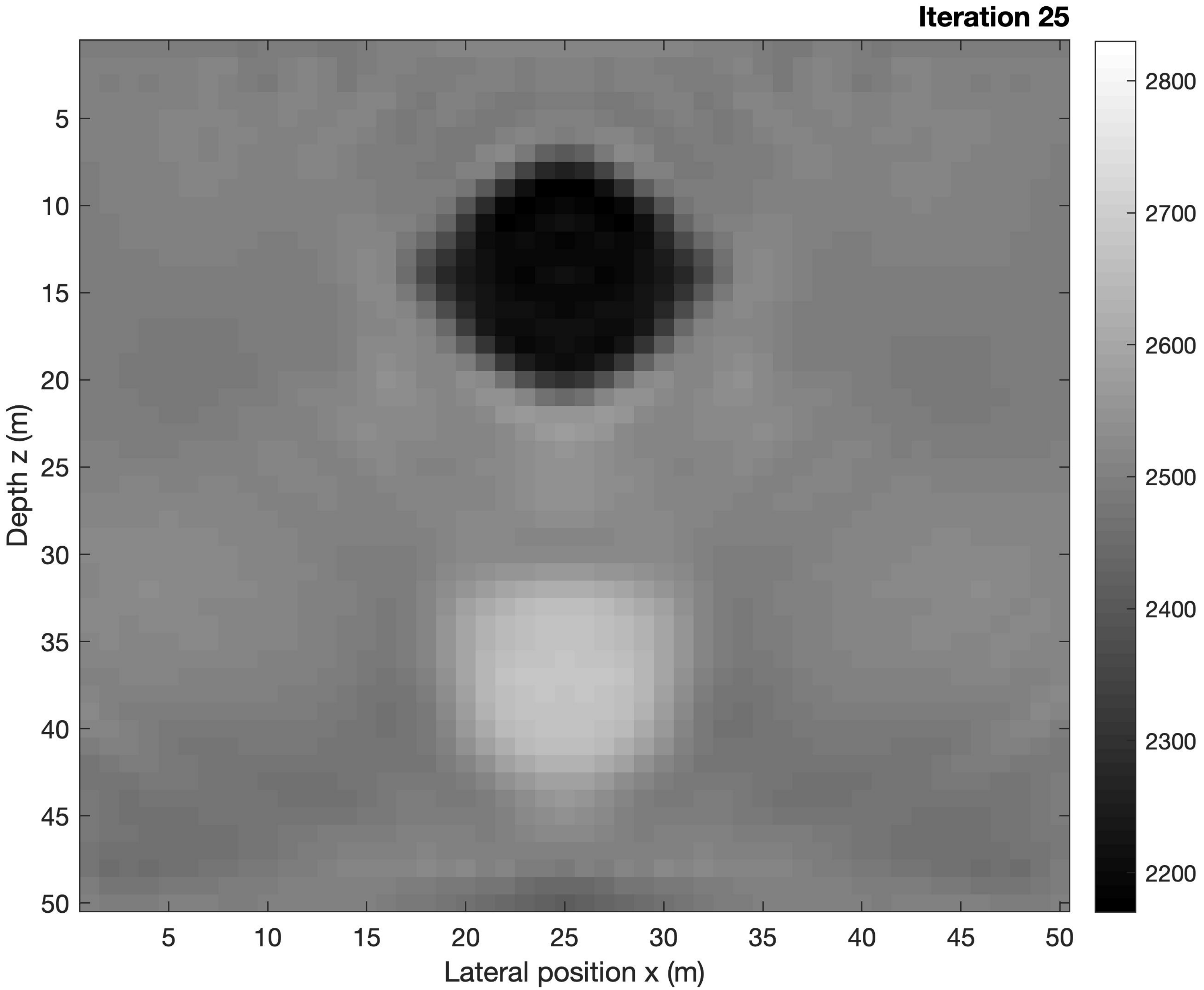
```
mod6_1_FWI_
98
99      %% 5.
100 -   FDFDf                                   dz,dx);
101
102     %% 6.
103 -   D = c
104
105     %% 7.
106
107     % Set
108 -   numit
109 -   optyp
110 -   range
111
112     % Mai
113 -   vel =                                   rangevel);
114
115 -   figur
116 -   image
117 -   caxis
118 -   title
119 -   drawn
```



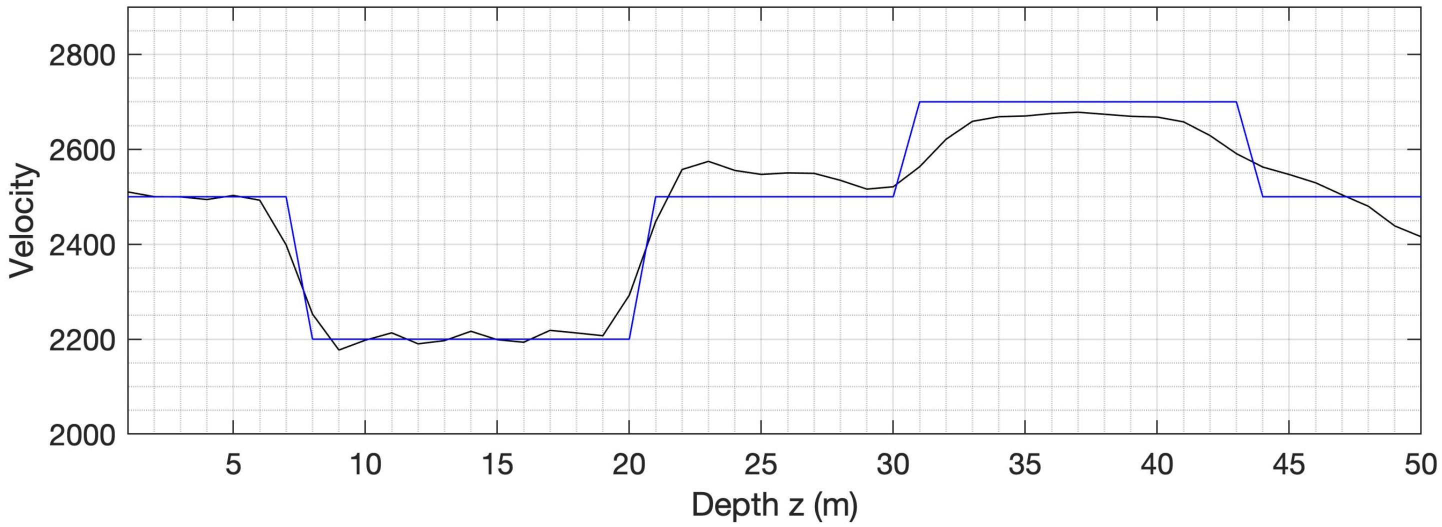
FWI

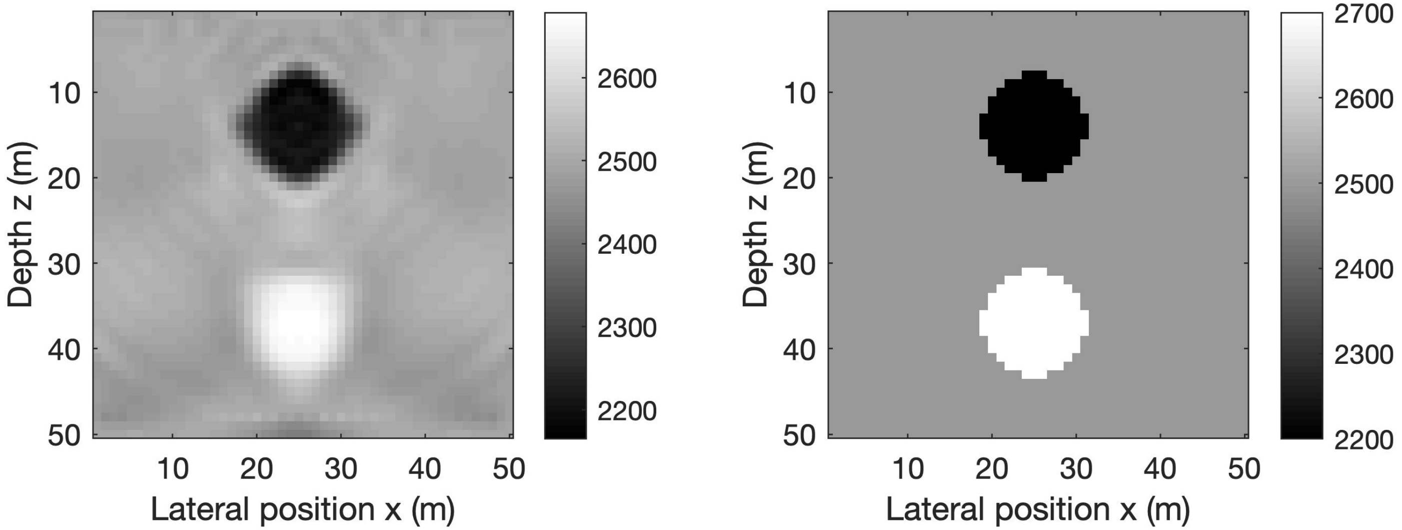
mod6_1_FWI

```
%% 5.
FDFDf                                   dz,dx);

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =                                   rangevel);

figur
image
caxis
title
drawn
```



FWI in Matlab – code setup for a simple implementation

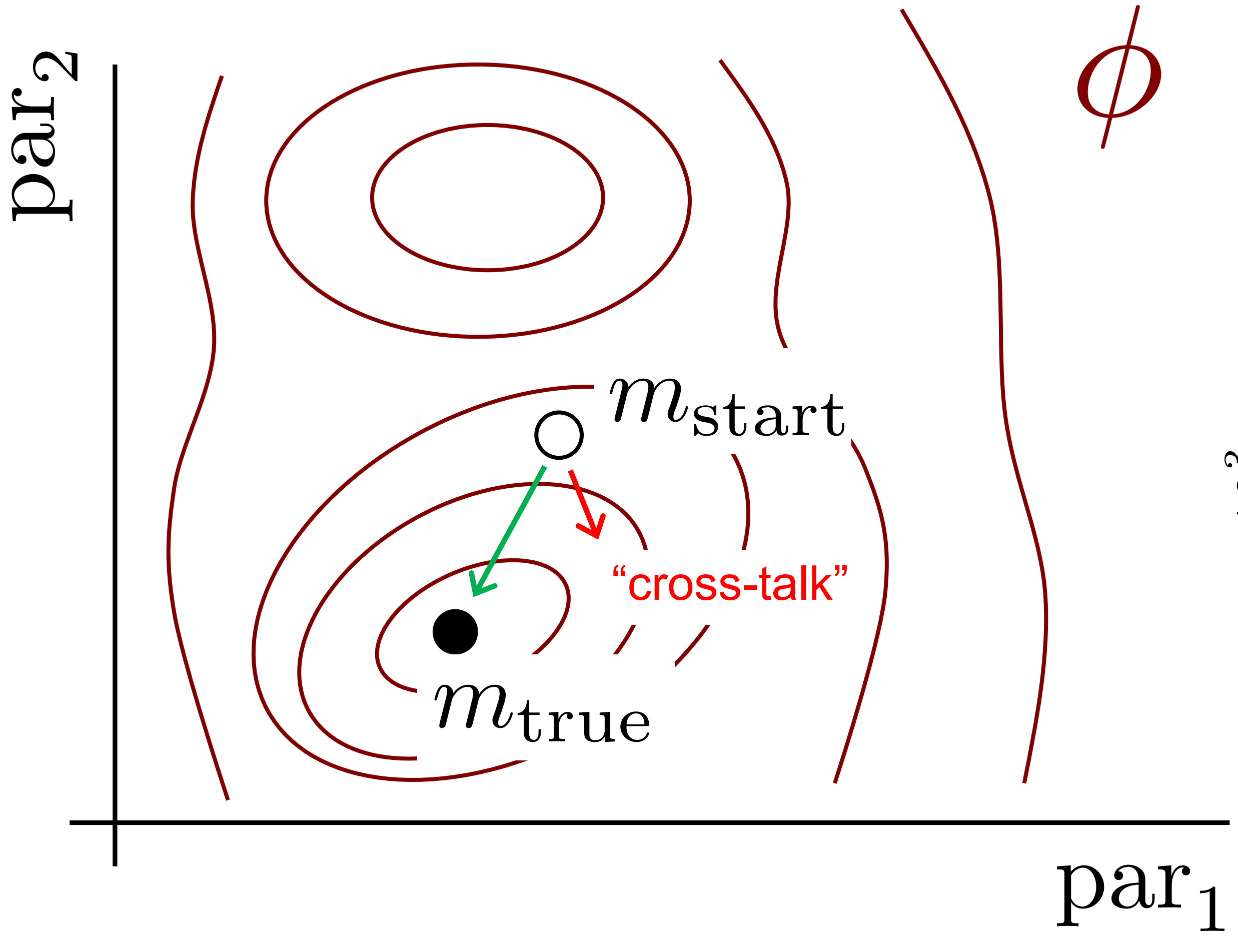




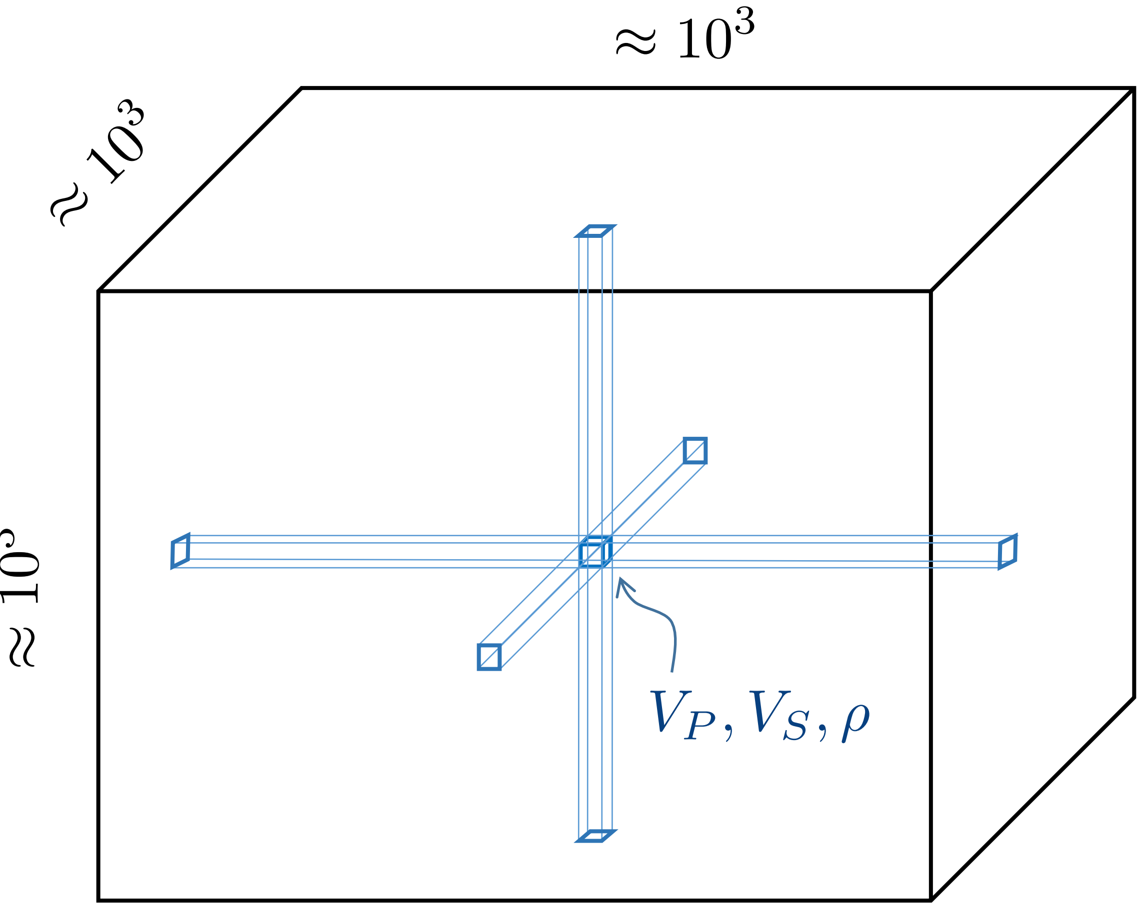
Result for specific choices of

- True model
- Frequency bands
- Source configuration
- Receiver configuration
- Optimization type
- Introduction of frequencies

The Matlab environment, through treatment of small to medium sized problems, allows the response / numerical behaviour of FWI to be tested

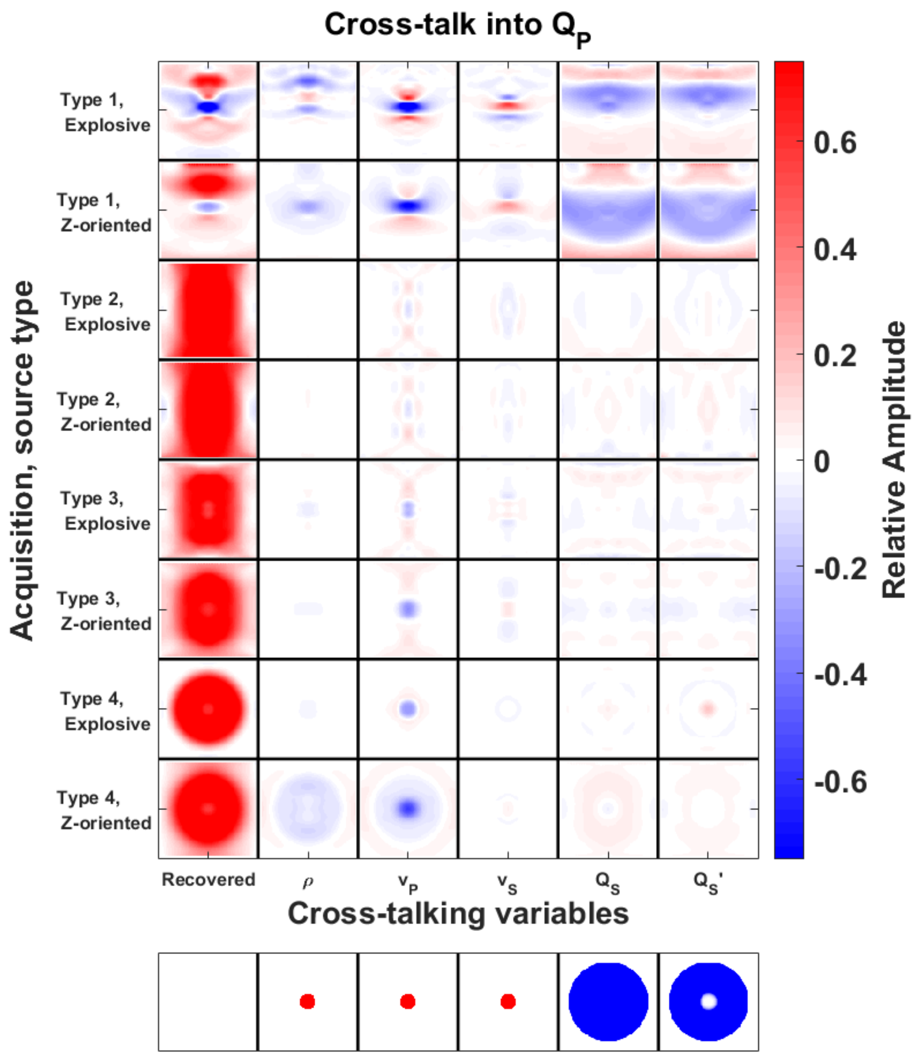


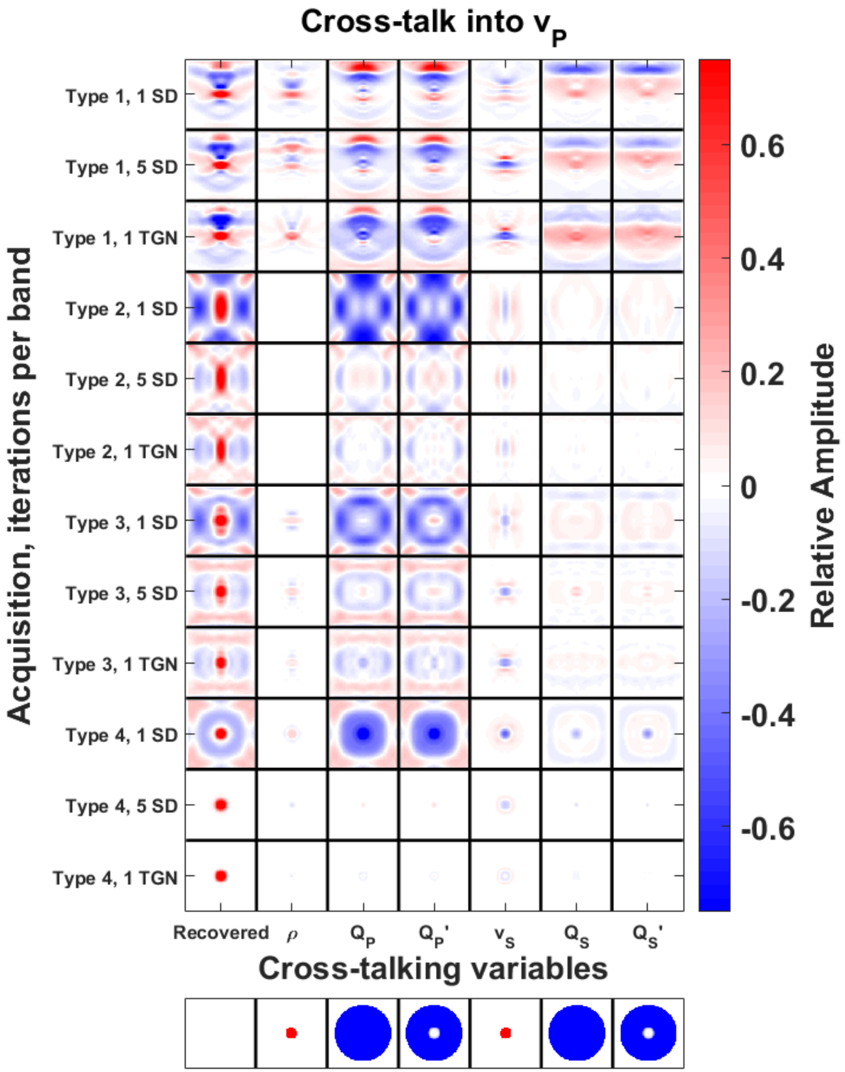
ϕ
par_2
par_1
m_{start}
m_{true}
“cross-talk”



$\approx 10^3$
$\approx 10^3$
$\approx 10^3$
V_P, V_S, ρ

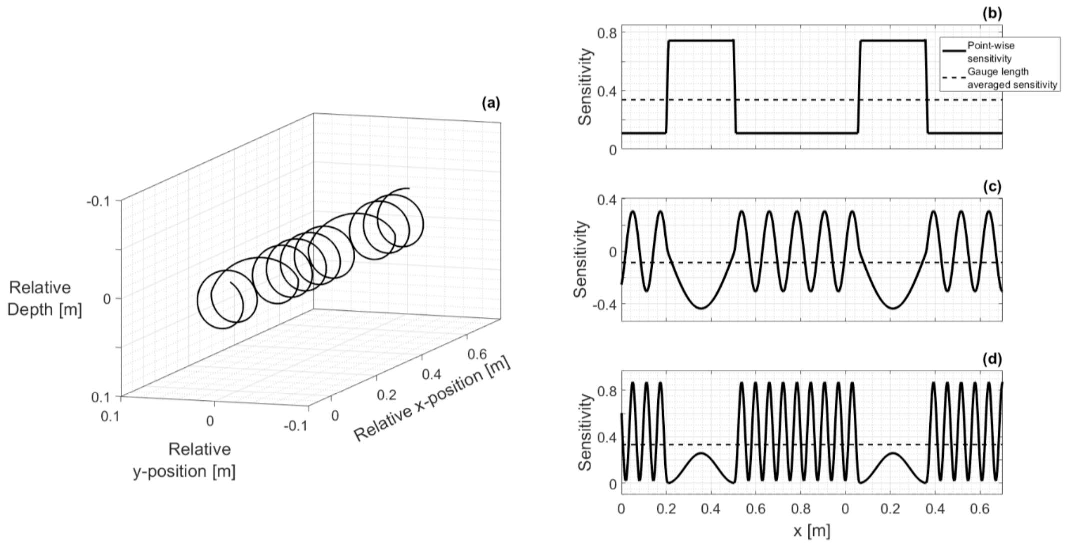
New science and new methods from FWI in Matlab

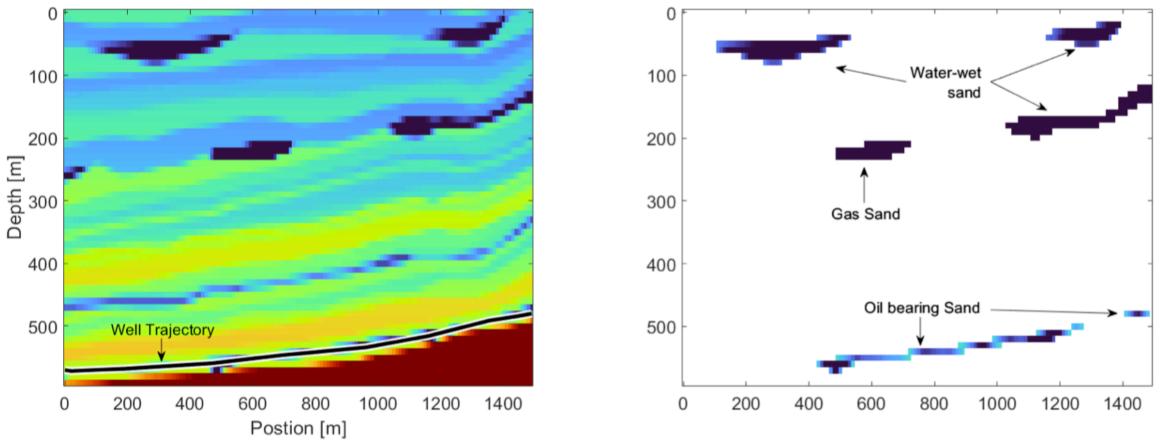




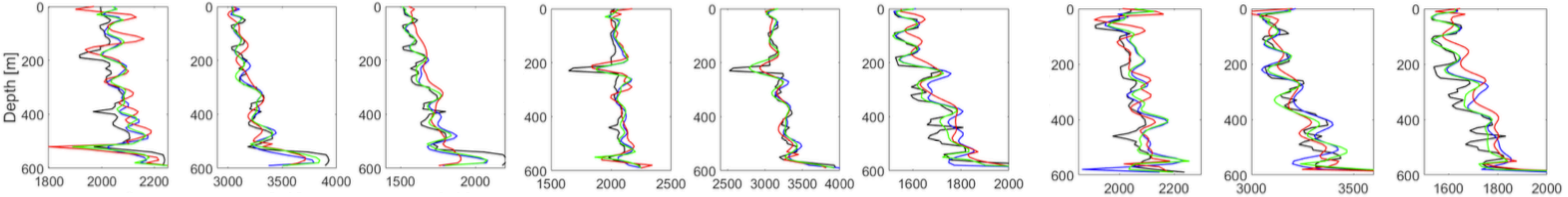
S. D. Keating and K. A. Innanen, *Parameter cross-talk and leakage between spatially-separated unknowns in viscoelastic full waveform inversion*, 2020: **Geophysics**, 85, 4.

New science and new methods from FWI in Matlab

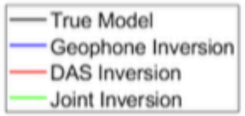




Profile 250m | Profile 550m | Profile 1100m

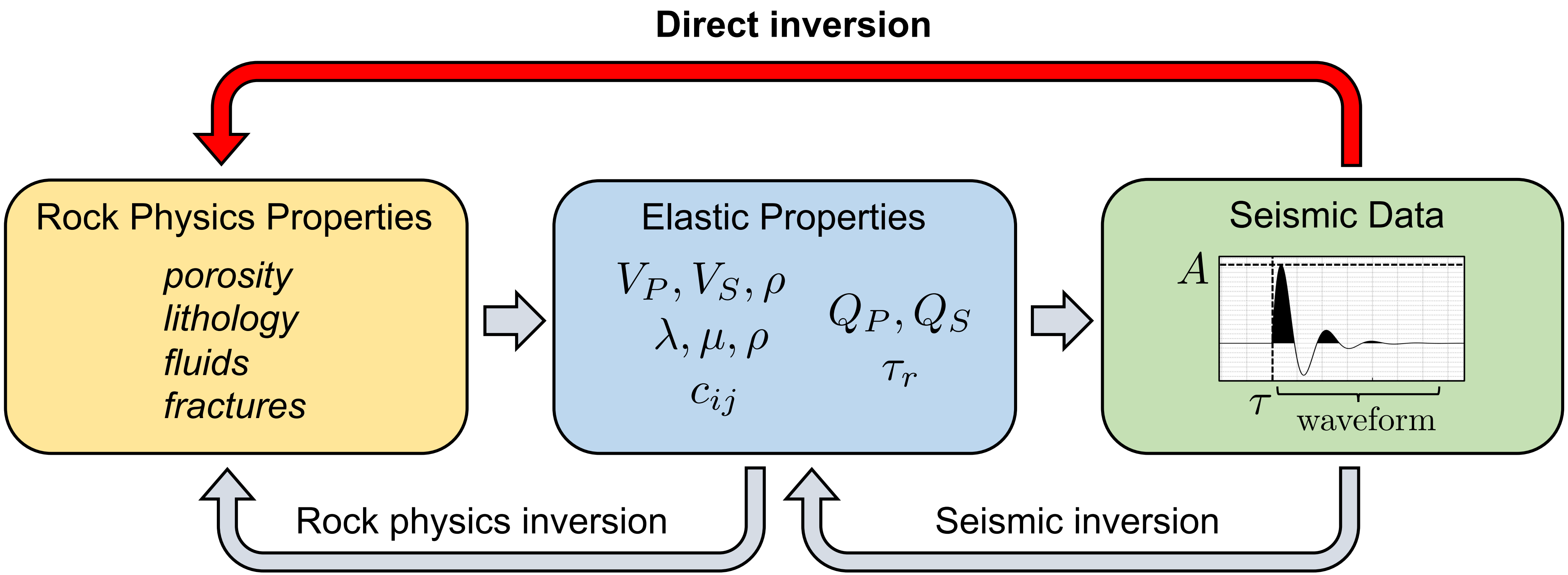


Density (kg/m^3) | P-wave velocity (m/s) | S-wave velocity (m/s) | Density (kg/m^3) | P-wave velocity (m/s) | S-wave velocity (m/s) | Density (kg/m^3) | P-wave velocity (m/s) | S-wave velocity (m/s)



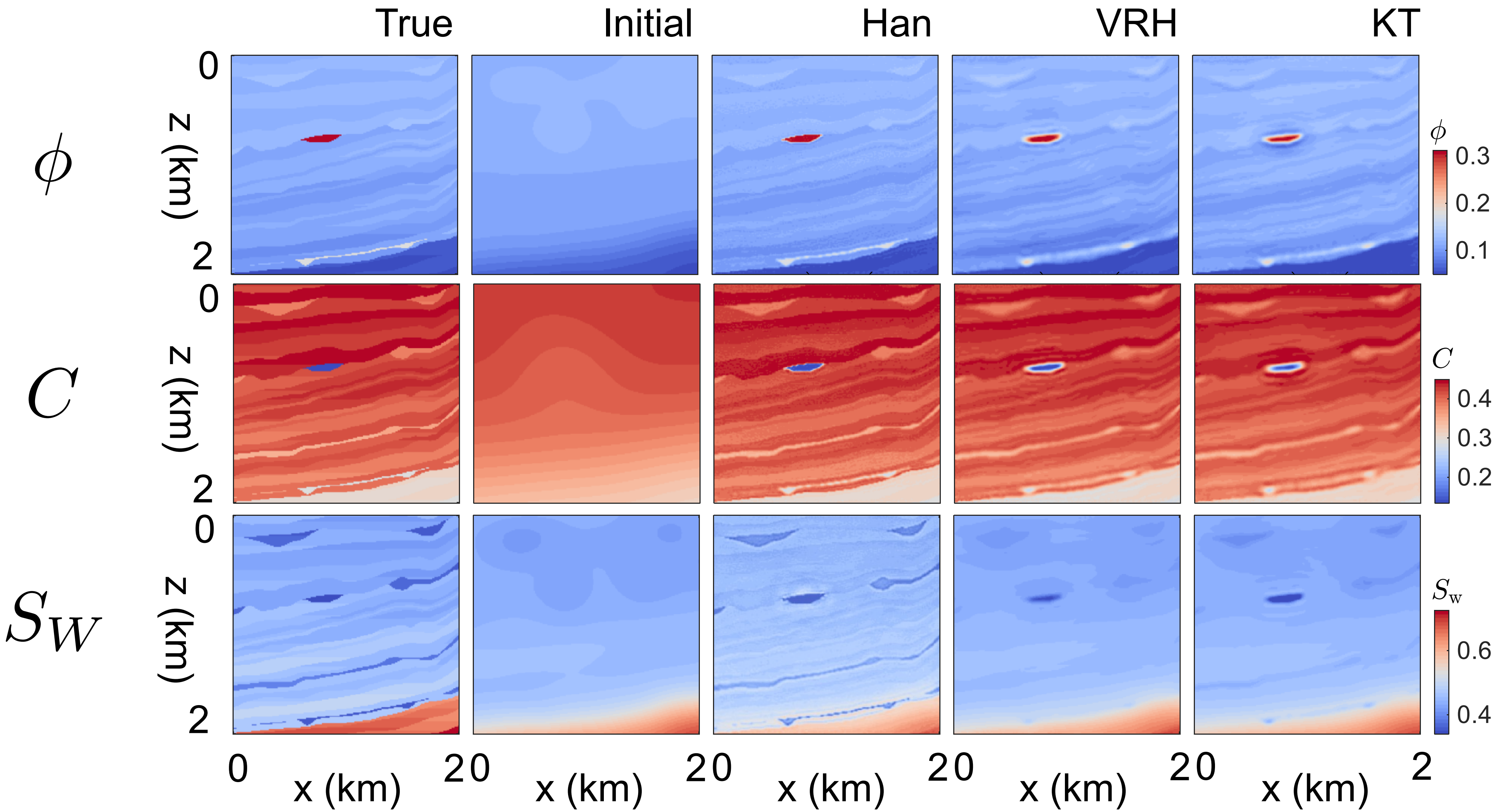
M. V. Eaid, S. D. Keating and K. A. Innanen, *Multi-parameter seismic elastic full waveform inversion with combined geophone and shaped fiberoptic cable data*, 2020: **Geophysics**, 85, 6.

New science and new methods from FWI in Matlab



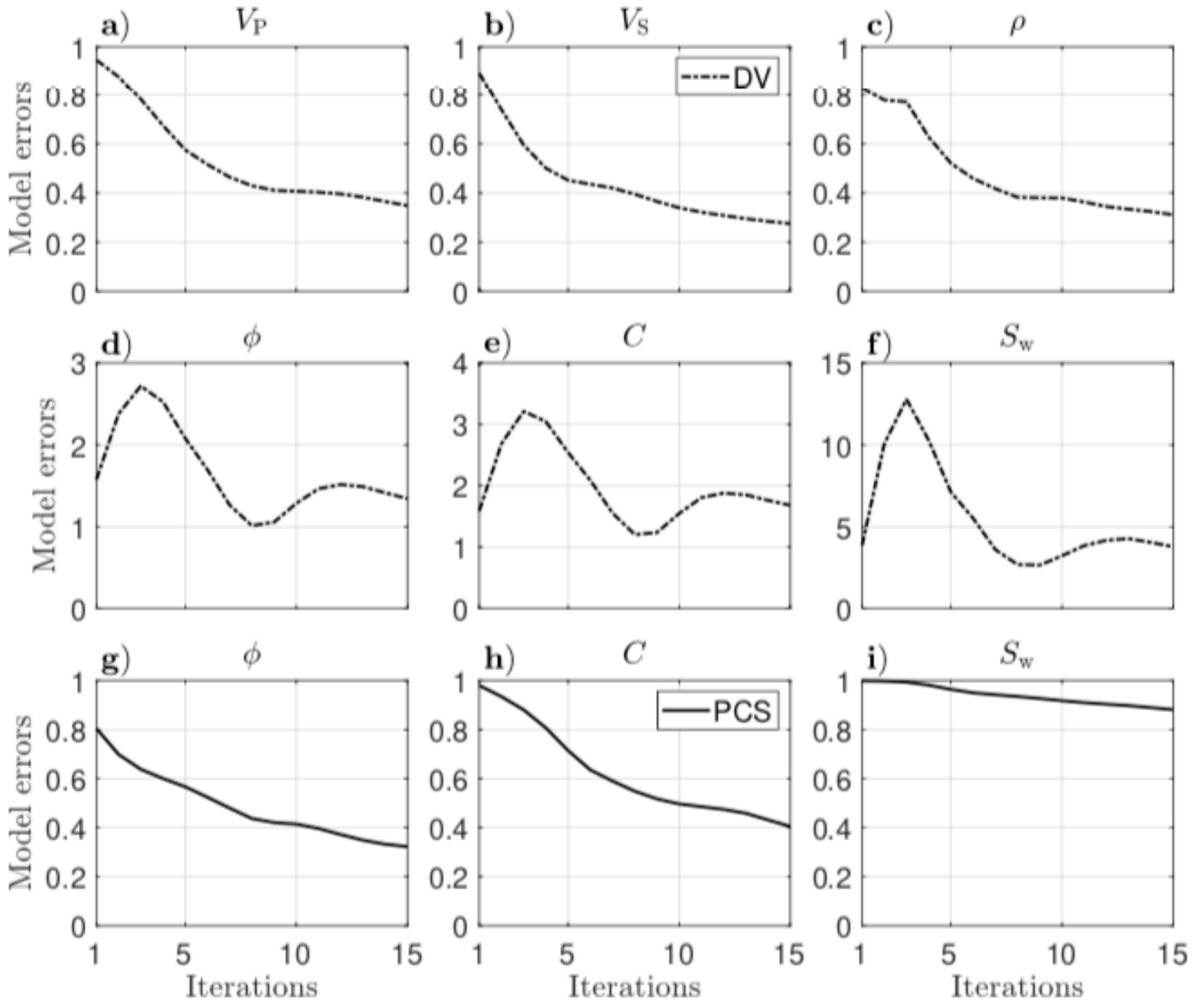
Q. Hu, S. D. Keating and K. A. Innanen, *Direct updating of rock physics properties using elastic full waveform inversion*, 2020: **in review**.

New science and new methods from FWI in Matlab



Q. Hu, S. D. Keating and K. A. Innanen, *Direct updating of rock physics properties using elastic full waveform inversion*, 2020: **in review**.

New science and new methods from FWI in Matlab



Elastic property model error with iteration

Rock physics property model error with iteration (indirect)

Rock physics property model error with iteration (direct)

Q. Hu, S. D. Keating and K. A. Innanen, *Direct updating of rock physics properties using elastic full waveform inversion*, 2020: **in review**.

Some next steps (2020 and beyond)

- Seismic inversion as a theory-guided data science / machine learning problem
  - “Wave equation machine”
  - Training a network to adapt wave propagation physics, produce better initial models
- Constraining FWI with prior (geological, rock physics) models and/or PDFs