Seismic full waveform inversion algorithms and their numerical behaviour

MATLAB Energy Conference
Kris Innanen
Nov 2020
• Introduction to the CREWES Project
• The seismic exploration / monitoring problem & the CREWES Matlab toolbox
• Seismic full waveform inversion (FWI) science and technology
  o Academic applications to field-scale problems
  o New formulations, new parameters, new data modes, new goals
• FWI in Matlab: tour of a simple implementation
• Recent research results
  o New parameters and uncertainty
  o New data modes (fiberoptic or “DAS” seismology)
  o New goals (direct determination of rock physics properties)
• Next steps
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
CREWES Matlab Toolbox

CREWES Toolbox Version: 2078

CREWES Matlab Toolbox (ZIP) 155.19 MB
MD5 checksum: b6918a4dca78b19c0e0e159d7 *crewes.zip

Sample data to accompany Methods of Seismic Data Processing (ZIP) 13.65 MB
MD5 checksum: 402597c854a3c08e81e654b91 *NMESdata.zip

Guide to the CREWES Matlab toolbox (PDF) 6.98 MB
MD5 checksum: 38648c7a6ac29914e13ce2a9027d *NumMeth.pdf

Introductory seismic data processing course (PDF) 88.09 MB
MD5 checksum: 01e78195b1e8ddc3c69ba2c78de *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 notation. The subset of CMSL covered by NMES has been checked for consistency and
The CREWES Matlab toolbox (www.crewes.org)
CREWES Matlab Toolbox

CREWES Toolbox Version: 2078

CREWES Matlab Toolbox (ZIP) 155.19 MB
MD5 checksum: 8914badca7809a00e1c059d7 *crewes.zip

Sample data to accompany Methods of Seismic Data Processing (ZIP) 13.65 MB
MD5 checksum: e823f97c834e33ed8ee81e63b591 *NMESdata.zip

Guide to the CREWES Matlab toolbox (PDF) 6.98 MB
MD5 checksum: c906848c7a6ac09164c32a0027d *NumMeth.pdf

Introductory seismic data processing course (PDF) 88.09 MB
MD5 checksum: 01e78195bd1e8d0c3c069e2c780c *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 (CMLSL) 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 CMLSL. Both library and text are intended for teaching and research in exploration seismology. The complete CMLSL is a large collection of geophysical codes that has grown by accretion over time with limited planning or rationalization. The subset of CMLSL covered by NMES has been checked for consistency and
The CREWES Matlab toolbox (www.crewes.org)
Full waveform inversion – concept and technology
BP plans for significant growth in deepwater Gulf of Mexico
8 January 2019

- Approval of major expansion at Atlantis field supports strategy of growing advanced 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 barrels 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 advanced 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 online in 2020, the project is expected to boost production at the platform by an estimated 36,000 barrels of oil equivalent a day (boe/d) at its peak. It will also access the eastern area of the field where 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 Starke 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 Wolfspur seismic acquisition source is being planned for Thunder Horse and Atlantis to better understand the reservoirs. Wolfspur uses ultraslow frequencies during seismic surveys, allowing geophysicists to see deeper below salt layers and enable 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

\[ \phi \]

\[ m_{\text{true}} \]

\[ m_{\text{start}} \]

"cross-talk"

\[ \approx 10^3 \]

\[ V_P, V_S, \rho \]
Full waveform inversion – tenets and concepts

**INPUT**

- starting model: $V_P\{V_S, \rho\}$

**INPUT**

- observed data

---

**simulate data in current medium**

iterate

---

**update:**

- $V_P \leftarrow V_P + \Delta V_P$
- $V_S \leftarrow V_S + \Delta V_S$
- $\rho \leftarrow \rho + \Delta \rho$

---

**form residuals**

- $\Sigma$ sensitivities $\times$ residuals

---

**uncalibrated update**

- $\Sigma$

---

**calibrate update, compute**

- $\Delta V_P\{\Delta V_S, \Delta \rho\}$
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
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

% Module 6. "mod6_1_FWIdomain_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_FWIdomain_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
endif

% 2. Set up numerical parameters
FWI in Matlab – code setup for a simple implementation

```matlab
mod6_1_FW interleaved DRIVER.m

1 % Module 6. "mod6_1_FW interleaved DRIVER.m"
2 % 2D 1-parameter acoustic frequency domain FWI with synthetic
3 % data and choosable model; full tour. Author: S. Keating 2018, small
4 % additions by K. Innanen, 2019.
5
6 clear; close all;
7 plot_model_only = 'y';

8
9 % 1. Select velocity model
10 crewes_FW interleaved BALLMODEL;
11
12 profileindex = 25;
13 veltruemat = reshape(vel_true,nz,nx);
14 veltrueprofile = veltruemat(:,profileindex);
15
16 if ( plot_model_only == 'y' )
17
18 figure,
19 subplot(2,2,[1 2]),
20 plot(1:nz,veltrueprofile,'k-');
21 axis([1,50,2000,2900]);
22 grid; grid minor;
23 xlabel('Depth z (m)'); ylabel('Velocity');
24 subplot(2,2,4),
25 imagesc(veltruemat); colormap('gray'); colorbar;
26 xlabel('Lateral position x (m)'); ylabel('Depth z (m)');
27 return
18 end

19 % 2. Set up numerical parameters
```
FWI in Matlab – code setup for a simple implementation

%% 2. Set up numerical parameters

% Perfectly matched layer parameters
PMI\_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 = (nx+2*PMI\_thick)*(nx+2*PMI\_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(sz)), (nz-2)*ones(size(sz))];
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));

%% 4. Set up frequencies

[S,R] = crewes\_FWI\_fdomain\_DEFINEACQUISITION(sz, sx, rz, rx, nz, nx, PMI\_thick);
21

FWI in Matlab – code setup for a simple implementation

%% 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
NML = (nx+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 intercept
s_initial=2; % initial source
s_end=nx-2; % end source
sx=s_initial:s_inter:s_end; % x-coordinate
sz = 3*ones(size(sx));

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

% Source coordinates (units of grid number)
r_inter=1; % receiver intercept
r_initial=2; % initial receiver
r_end=nx-1; % end receiver
rx=r_initial:r_inter:r_end; % x-coordinate 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

```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 tanning 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
```
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

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, ootype, 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 in Matlab – code setup for a simple implementation

```matlab
%% 5.
FDFDf

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =

% figur
image
caxis
title
drawn

dz,dx);

rangevel);
```
FWI – code setup for a simple implementation

%% 5.
FDFDf

%% 6.
D = c

%% 7.
% Set
numit
optyp
range

% Mai
vel =

% figur
image
caxis	title
drawn
FWI in Matlab – code setup for a simple implementation

%% 5.
FDFDf

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Main
vel =

% Show the image
figure
image
caxis
title
drawn

rangevel); dz, dx);
FWI in Matlab – code setup for a simple implementation

```matlab
%% 5.
FDFDf
%% 6.
D = c
%% 7.

% Set
numit
optyp
range

% Main
vel =

figure
image
caxis
title
drawn
```
FWI in Matlab

- Code setup for a simple implementation

```matlab
mod6_1_FWI

%% 5.
FDFDf

%% 6.
D = c

%% 7.

% Set
numit
optyp
range

% Mai
vel =

figur
image
caxis
title
drawn
```

dz, dx);
FWI in Matlab – code setup for a simple implementation

```matlab
mod6_1_FWl
98
99
100 -
101 % 5.
102 FDFDf
103 -
104 % 6.
105 D = c
106 -
107 % 7.
108 -
109 % Set
110
111
112 % Mai
113 -
114 -
115 -
116 -
117 -
118 -
119 -
20
21 % figur
22 % image
23 % caxis
24 % title
25 % drawn
```
FWI in Matlab – code setup for a simple implementation

```matlab
98 mod6_1_FWI
99 function
100 
101 % Set
102 numit
103 optyp
104 range
105 % Mai
106 vel =
107 figur
108 image
109 caxis
110 title
111 drawn
112
113 dz,dx);
114 rangevel);
```
% 5.
FDFDf
% 6.
D = c
% 7.
% Set
numit
optyp
range
% Mai
vel =
figur
image
caxis
title
drawn

dz, dx);
rangevel);
FWI in Matlab – code setup for a simple implementation

```matlab
% 5.
FD FDFd
% 6.
D = c
% 7.

% Set
numit
optyp
range

% Mai
vel =

figur
image
caxis
title
drawn

dz, dx);
rangevel);
```
FWI in Matlab – code setup for a simple implementation

```matlab
%% 5.
FDFDf

%% 6.
D = c

%% 7.
Set
numit
optyp
range

%% Mai
vel =

figur
image
caxis
title
drawn
```

Iteration 19
FWI in Matlab – code setup for a simple implementation

```
mod6_1_FWIm
98 %      5.
00 - FDFDf
01 %  6.
02 D = c
03 -
04 %  7.
05 % Set
06 numit
07 - optyp
08 range
10 -
11 % Mai
12 vel =
13 -
14 - figur
15 - image
16 - caxis
17 - title
18 - drawn
```

dz,dx);

rangevel);
FWI in Matlab

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

iteration 23
FWI in Matlab – code setup for a simple implementation

```
mod6_1_FW1
98 5.
99 FDFDf
00 D = c
01 7.
02 Set
03 numit
04 optyp
05 range
11 % Mai
12 vel =
13 figur
14 image
15 caxis
17 title
18 drawn
```

Iteration 25
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.
New science and new methods from FWI in Matlab

\( m_{\text{true}} \)  

\( m_{\text{start}} \)  

"cross-talk"  

\( \Phi \)

\[ V_P, V_S, \rho \]

\( \approx 10^3 \)

\( \approx 10^3 \)
New science and new methods from FWI in Matlab

New science and new methods from FWI in Matlab

New science and new methods from FWI in Matlab

New science and new methods from FWI in Matlab

New science and new methods from FWI in Matlab

• 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