Full Waveform Inversion of Crosswell Seismic Data
Using Automatic Differentiation

Wenyuan Liao
Department of Math. & Stat., University of Calgary

Danping Cao
School of Geosciences, China University of Petroleum

CREWES Tech Talk
February 1, 2013
<table>
<thead>
<tr>
<th>1. Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Adjoint State Method</td>
</tr>
<tr>
<td>3. Automatic Differentiation(AD)</td>
</tr>
<tr>
<td>4. FWI using AD</td>
</tr>
<tr>
<td>5. Model Test</td>
</tr>
<tr>
<td>6. Conclusions</td>
</tr>
</tbody>
</table>
Introduction

Fixed Receivers – varying sources

Workflow of FWI
Introduction

- Mathematical Formulation: PDE-constrained Optimization

\[ \mathcal{J}(m) = \frac{1}{2} \int_0^{t_f} \sum_{i=1}^{N_r} (d_{obs}^i - d_{cai}(m))^2 \, dt + \kappa \|m\| \]

where

- \( m \): Model parameter (wave velocity)
- \( d_{obs}^i \): Observational data
- \( d_{cai}(m) \): Synthetic seismogram based on \( m \) through the wave eq.
- \( \kappa \|m\| \): Regularity term (Optional, depending on prior knowledge)

The inverse problem is solved through

\[ \min_{m \in \mathbb{H}(\Omega)} \mathcal{J}(m) \]
Introduction

- PDE-Constrained Optimization: Gradient Calculation

$$\frac{\partial J}{\partial m} = - \int_0^{t_f} \sum_{i=1}^{N_r} \left( (d_{obs}^i - d_{cal}^i) \cdot \frac{\partial d_{cal}^i}{\partial u} \cdot \frac{\partial u}{\partial m} \right) dt + \kappa \frac{\partial \|m\|}{\partial m}$$

Direct computation of $\frac{\partial u}{\partial m}$ is difficult and expensive!

Adjoint-state method is an effective way to resolve this issue
<table>
<thead>
<tr>
<th>1. Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Adjoint State Method</td>
</tr>
<tr>
<td>3. Automatic Differentiation(AD)</td>
</tr>
<tr>
<td>4. FWI using AD</td>
</tr>
<tr>
<td>5. Model Test</td>
</tr>
<tr>
<td>6. Conclusions</td>
</tr>
</tbody>
</table>
Adjoint State Method

- Define the cost functional as

\[ J(m) = J(u(m), m) \]

which may depend on the model parameter implicitly if no regularity term.

The governing PDE (acoustic wave equation in this case) is stated as

\[ L(u(m), m) = 0 \]

Here \( L \) is an operator defining the initial-boundary value problem of the wave equation.
Introduce a perturbation to the parameter:
\[ \delta m \Rightarrow \delta u \Rightarrow \delta J \]

\[ \mathbb{L}(u, m) = 0 \quad + \quad \mathbb{L}(u + \delta u, m + \delta m) = 0 \]

\[ \delta J = \left( \frac{\partial J(u, m)}{\partial m} - \left\langle \xi, \frac{\partial \mathbb{L}(u, m)}{\partial m} \right\rangle \right) \delta m \]

where the adjoint-state variable is defined as

\[ \left[ \left( \frac{\partial \mathbb{L}(u, m)}{\partial u} \right)^* \right] \xi = \left[ \frac{\partial J(u, m)}{\partial u} \right] \Rightarrow \xi = \left[ \left( \frac{\partial \mathbb{L}(u, m)}{\partial u} \right)^* \right]^{-1} \left[ \frac{\partial J(u, m)}{\partial u} \right] \]
Adjoint State Method: Lagrange Multipliers

Redefine a new cost functional as

$$\tilde{J}(u, m, \xi) = J(u, m) - \langle \xi, \mathbb{L}(u, m) \rangle$$

Solving the unconstrained optimization problem we obtain the gradient as

$$\frac{\partial \tilde{J}(u, m, \xi)}{\partial m} = \frac{\partial J(u, m)}{\partial m} - \left\langle \xi, \frac{\partial \mathbb{L}(u, m)}{\partial m} \right\rangle$$

where

$$\frac{\partial \tilde{J}(u, m, \xi)}{\partial u} = 0 \Rightarrow \frac{\partial J(u, m)}{\partial u} - \left( \frac{\partial \mathbb{L}(u, m)}{\partial u} \right)^* \xi = 0 \Rightarrow \xi = \left[ \left( \frac{\partial \mathbb{L}(u, m)}{\partial u} \right)^* \right]^{-1} \frac{\partial J(u, m)}{\partial u}$$

Adjoint -> Discretization or Discretization -> Adjoint
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Introduction</strong></td>
</tr>
<tr>
<td><strong>2. Adjoint State Method</strong></td>
</tr>
<tr>
<td><strong>3. Automatic Differentiation (AD)</strong></td>
</tr>
<tr>
<td><strong>4. FWI using AD</strong></td>
</tr>
<tr>
<td><strong>5. Model Test</strong></td>
</tr>
<tr>
<td><strong>6. Conclusions</strong></td>
</tr>
</tbody>
</table>
Automatic differentiation

- Automatic Differentiation (AD), sometimes alternatively called algorithmic differentiation, is a set of techniques to numerically evaluate the derivative of a function specified by a computer program.
Forward mode AD

\[ f(x_1, x_2) \]

\[ \dot{w}_5 = \dot{w}_3 + \dot{w}_4 \]

\[ \dot{w}_4 = \cos(w_1) \dot{w}_1 \]

\[ \dot{w}_3 = \dot{w}_1 w_2 + w_1 \dot{w}_2 \]

\( w_1, w_2 \in \{0, 1\} \)
Reverse mode AD

\[
\begin{align*}
\tilde{f} &= \tilde{w}_5 = 1 \text{ (seed)} \\
\tilde{w}_4 &= \tilde{w}_5 \frac{\partial w_5}{\partial w_4} = \tilde{w}_5 \cdot 1 \\
\tilde{w}_3 &= \tilde{w}_5 \frac{\partial w_5}{\partial w_3} = \tilde{w}_5 \cdot 1 \\
\tilde{w}_2 &= \tilde{w}_3 \frac{\partial w_3}{\partial w_2} = \tilde{w}_3 w_1 \\
\tilde{w}_1^a &= \tilde{w}_4 \cos(w_1) \\
\tilde{w}_1^b &= \tilde{w}_3 w_2 \\
\tilde{x}_1 &= \tilde{w}_1^a + \tilde{w}_1^b = \cos(x_1) + x_2 \\
\tilde{x}_2 &= \tilde{w}_2 = x_1
\end{align*}
\]
AD tools

- **AD Model Builder** (C/C++)
- **ADC** (C/C++)
- **ADF** (Fortran77, Fortran95)
- **ADIC** (C/C++)
- **ADIFOR** (Fortran77)
- **ADiMat** (MATLAB)
- **ADMAT / ADMIT** (MATLAB)
- **ADOL-C** (C/C++)
- **ADOL-F** (Fortran95)
- **APMonitor** (Interpreted)
- **AUTODIF** (C/C++)
- **AutoDiff.NET** (.NET)
- **AUTO_DERIV** (Fortran77/95)
- **ColPack** (C/C++)
- **COSY INFINITY** (Fortran77/95, C/C++)
- **CppAD** (C/C++)
- **CTaylor** (C/C++)
- **FAD** (C/C++)
- **FADBAD/TADIFF** (C/C++)
- **FFADLib** (C/C++)
- **GRESS** (Fortran77)
- **HSL_AD02** (Fortran95)
- **INTLAB** (MATLAB)
- **NAGWare Fortran 95** (Fortran77, Fortran95)
- **OpenAD** (C/C++, Fortran77/95)
- **PCOMP** (Fortran77)
- **pyadolc** (python)
- **pyccppad** (Interpreted, python)
- **Rapsodia** (C/C++, Fortran95)
- **Sacado** (C/C++)
- **TAF** (Fortran77, Fortran95)
- **TAMC** (Fortran77)
- **TAPENADE** (C/C++, Fortran77/95)
- **TaylUR** (Fortran95)
- **The Taylor Center** (independent)
- **TOMLAB / MAD** (MATLAB)
- **TOMLAB / TomSym** (MATLAB)
- **Treeverse / Revolve** (C/C++, Fortran77/95)
- **YAO** (C/C++
1. Introduction
2. Adjoint State Method
3. Automatic Differentiation (AD)
4. FWI using AD
5. Model Test
6. Conclusions
Workflow of FWI

\[ \Delta m = - \left[ \frac{\partial^2 J(m_0)}{\partial m^2} \right]^{-1} \frac{\partial J(m_0)}{\partial m} \]

\[ J(m) = \frac{1}{2} \left( d_{\text{obs}} - d_{\text{calc}}(m) \right)^\top \left( d_{\text{obs}} - d_{\text{calc}}(m) \right) \]
FWI solution one by one

Forward Modeling:
- 2\textsuperscript{nd}-order in time and 4\textsuperscript{th}-order in space
- Stagger-grid finite difference
- PML absorbing boundary

Model Updating:
- L-BFGS selected
- Limited memory required
- Quasi-Newton method

Gradient Calculation:
- Adjoint state method
- AD tools used
- TAPENADE test
FWI workflow with AD

1. Initial model
2. Forward modeling
3. Misfit calculation
4. Model updating
5. Automatic Differentiation

Converged?

Y

N

AD tool solve the gradient according to the forward modeling program
Benefit of FWI with AD

- Simplify the gradient calculation
- Focus on forward modeling and optimization method
- High efficiency forward modeling program will lead to high efficiency gradient calculation code
- FWI workflow is simplified
Accuracy of Gradient calculation

Gradient calculation:
- True model
- Synthetic record
- Initial model

Gradient by TAPENADE

Gradient by central difference quotient
Outline

1. Introduction
2. Adjoint State Method
3. Automatic Differentiation (AD)
4. FWI using AD
5. Model Test
6. Conclusions
Model test 1

Model: 101 X 101
Spatial sample: 1m
Time sample: 0.1ms
Source: ricker wavelet
Main frequency: 180 Hz
Boundary: PML
Inversion result - 1 shot

true model

initial model

10 iteration

30 iteration

50 iteration

100 iteration

300 iteration

1000 iteration
Inversion result - different shot

Initial model

1 shot

3 shot

5 shot

7 shot

11 shot
Model test 2

Model: 101 X 101
Spatial sample: 1m
Time sample: 0.1ms
Source: ricker wavelet
Main frequency: 180 Hz
Boundary: PML
Inversion result - different shot

- True model
- Initial model
- 1 shot
- 3 shots
- 5 shots
- 7 shots
- 11 shots
- 21 shots
Inversion result - 1 shot
Conclusion

• Automatic differentiation (AD) is a promising yet not popular approach in Geoscience.

• The gradient calculated through AD is accurate.

• The full waveform inversion workflow is simplified with the usage of the AD tool.

• Model tests show that the full waveform inversion method with AD is effective and efficient in the inversion of the crosswell seismic data.
Future work

- Improve the forward modeling: finite difference 4th-order in time
- Test the large-scale data inversion using checkpoint technology
- Test with other AD tools, and Optimization algorithms
- Test the surface seismic inversion
- Address inverse modeling issues under the current framework
- Test on other types of wave equations
Acknowledgement

• Gary Margrave
• CREWES
• POTSI
• NSERC
• Department of Math. & Stat.
• Chinese Scholarship Council, National Nature Science Foundation of China