Efficient gradient computation of a misfit function for FWI using the adjoint method

B. Galuzzi, department of Earth Sciences “A. Desio”, University of Milan
E. Stucchi, department of Earth Sciences “A. Desio”, University of Milan

24 Novembre 2016

GNGTS 2016, Lecce
Sessione 3.3: Modelling, aspetti teorici e tecnologie innovative
1. Seismic inversion using FWI
2. Seismic modelling
3. The importance of misfit gradient for FWI
4. Computation of misfit gradient using the adjoint method
5. An example of application
Full Waveform Inversion (Tarantola, 1986)

Estimation of a geological macro-model of subsurface from active seismic data, by means of:

1) Seismic modelling algorithm → Numerical solution of wave equation (predicted data)

2) Misfit function → Difference between predicted and observed data

3) Seismic inversion algorithm → Optimization of the misfit function → estimation of the predicted model

Stopping criteria: maximum number of iterations, minimum threshold,...
The seismic modelling

The 2D acoustic wave equation...

\[ \begin{aligned}
\dot{q}(\vec{x}, t) &= v(\vec{x})^2 \Delta p(\vec{x}, t) + \delta(\vec{x} - \vec{x}_0)s(t) \\
\dot{p}(\vec{x}, t) &= q(\vec{x}, t)
\end{aligned} \]

Initial conditions: \( p(\vec{x}, 0) = q(\vec{x}, 0) = 0, \ \forall \vec{x} \in D \)

...and its approximation by an explicit finite difference method

\[ \begin{aligned}
q_{i,j}^{t+1/2} &= q_{i,j}^{t-1/2} + dt v_{i,j}^2 \hat{\Delta} p_{i,j} + dt \delta(\vec{x}_{i,j} - \vec{x}_0) s^t \\
p_{i,j}^{t+1} &= p_{i,j}^t + dt q_{i,j}^{t+1/2}
\end{aligned} \]

- \( dt \): time sampling
- \( dx \): space sampling
- \( \hat{\Delta} \): approximation of Laplacian
- 2nd order in time and 4th in space

- \( t \in [0, T] \): recording time
- \( \vec{x} \in D(x, z) \subset \mathbb{R}^2 \): space domain
- \( \vec{x}_0 \): the location of the source
- \( s(t) \): the seismic wavelet
- \( v(\vec{x}) \): the acoustic wave velocity
Numerical considerations

1. Numerical stability:

\[ dt < \frac{dx}{v_{\lambda}}, \]

1. \( \lambda \in (0.5, 1) \): Courant number

2. Numerical dispersion

\[ dx < \frac{v_{\min}}{f_{\max}n}, \]

1. \( v_{\min} \): minimum velocity of the model
2. \( f_{\max} \): maximum frequency of \( s(t) \)
3. \( n \approx 5 \): points per wavelength

3. Absorbing boundary conditions (Cerjan, 1985):

\[
\begin{align*}
q_{i,j}^{t+1/2} &= G_{i,j} \left( q_{i,j}^{t-1/2} + \frac{dtv_{i,j}^2}{dx^2} \Delta p_{i,j}^t + dt \delta(\tilde{x}_{i,j} - x_0)s^t \right) \\
p_{i,j}^{t+1} &= G_{i,j} \left( p_{i,j}^t + dtq_{i,j}^{t+1/2} \right)
\end{align*}
\]

The Gaussian taper factor

\[ G_{i,j} = \begin{cases} 
1 & \text{inside the computational domain} \\
\in [0.92,1] & \text{inside the absorbing boundary layers}
\end{cases} \]
Misfit function for seismic inversion

The aim of FWI is to find an optimal Earth model $\overline{m}$, that minimizes a misfit function:

$$\overline{m} = \min_{m \in M} X(m) = \min_{m \in M} \| p(m) - p_0 \|$$

- $p_0$ : observed data
- $p(m)$: predicted data (using seismic modelling)
- $M$ : the set of geological models

A classical functional for FWI is the $L_2$ distance between the observed and the synthetic seismograms:

$$X(m) = \frac{1}{2} \sum_{s=1}^{n_s} \left( \sum_{r=1}^{n_r} \left( \int_0^T [p(m, t, \tilde{x}^r_s, m) - p_0(t, \tilde{x}^r_s, \tilde{x}^s)]^2 dt \right) \right),$$

- $n_r$ : number of receivers
- $n_s$ : number of the sources
- $T$ : registration time
- $\tilde{x}^r$ : position of r-th receiver
- $\tilde{x}^s$ : position of the s-th source

Due to space discretization, the geological models $m \in M$ are discretized with components $m_{i,j,k}$:

$$m_{i,j,k} = [v_p(x_{i,j,k}), v_s(x_{i,j,k}), \rho (x_{i,j,k}), ...]$$

1) Number of unknowns = (number of grid nodes) * (number of geological parameters)

2) $X(m)$ is a generally a complicated non-linear functional of $m$
Optimization algorithms for seismic inversion

Local iterative minimization algorithms

\[ m_{k+1} = m_k + \gamma_k h_k \]

- \( h_k \) is the descend direction
- \( \gamma_k > 0 \) is the step length
- \( X(m_{k+1}) < X(m_k) \)

A descend direction is given by:

\[ h_k = -\nabla_m X(m_k) \]  \( \text{the gradient} \) at the k-th iteration

The Misfit gradient is useful to guide the minimization algorithm to a local minimum of \( X(m) \)

How compute the gradient \( \nabla_m X(m_k) \)?
The adjoint method

Powerful tool to shorten the time required for computing the gradient (Fichner, 2011)

Acoustic equation + $L^2$ misfit

\[
\frac{\partial X(\hat{m}_{i,j})}{\partial m_{i,j}} = -\frac{2}{3} \int_0^T p^*(\hat{m}_{i,j}, \hat{x}_{i,j}, t) \ast \hat{p}(\hat{m}_{i,j}, \hat{x}_{i,j}, t) dt
\]

- $p$ is the regular solution of the wave equation
- $p^T$ is the solution of the adjoint equation

To compute the time integral \(\rightarrow\) simultaneously $p$ and $p^*$ for each time step, in all the domain $D$

**Regular solution $p$**

1) Wave equation scheme, forward in time (from 0 to T),
to compute the adjoint source ($g^*$)
to store $p$ only inside the absorbing boundaries

2) Wave equation scheme, reverse in time (from T to 0)
to re-compute $p$ inside the domain
using the storing $p$ inside the absorbing boundaries

**Adjoint solution $p^*$**

Wave equation scheme, backward in time (from T to 0)
to satisfy the final conditions (t=T)
after the adjoint source computation

N.B The storing of $p$ inside the absorbing boundaries to prevent numerical instability ($G_{i,j}^{-1}>1$) during backward propagation
Scheme for the adjoint gradient computation

1) Computation of the $p$, forward in time (Storing $p$ inside the absorbing layers for each time step)

2) Computation of $g^*$ using $p$ and $p_0$

3) Computation of $p$ and $p^*$, backward in time (using the stored $p$, inside the absorbing layers)

only 3 computation of wave equation is required: 1 forward modelling+2 backward modelling
Example of application

1. Observed data

- Wavelet (Ricker with $f_r=18\text{ hz}$)

2. Local optimization method

\[
\begin{align*}
m_{k+1} &= m_k + \gamma_k h_k \\
\beta_k &= \frac{||\nabla mX(m_{k+1})||^2}{||\nabla mX(m_k)||^2} \\
h_{k+1} &= -\nabla mX(m_k) + \beta_k h_k
\end{align*}
\]

- $\nabla mX(m_k)$ is the gradient at the k-th iteration
- $\gamma_k$ obtained by a line search procedure on $\phi_k = X(m_k + \gamma_k h_k)$ (linear or quadratic interpolation)
- $T_{\text{comp}} \approx n_{\text{iter}} \times (T_{\text{comp}}(h_k) + T_{\text{comp}}(\gamma_k)) \approx n_{\text{iter}} \times (\text{about 4-5 modellings})$

The time grid
- $dt = 0.002\text{s}$
- $T=6\text{s}$

The modelling grid
- $nx = 192$
- $nz = 48$
- $dx = dz = 24\text{m}$
3. The starting model (smooth version of the true model)

- The gradient emphasizes the main differences between the starting and the true model.
- Note the high values of the gradient near the sources.
- The gradient of the misfit function can guide the inversion procedure to the true model.
Results

- The final model is close to the global minimum solution.
- The inversion procedure has highlighted the main geological structures that were not clear in the smooth model.
- The main differences between the final and true model are at the border where the illumination is poor.

Note the misfit reduction from 1 to about 0.05
Conclusion

1. We have implemented the adjoint method to compute the gradient of a misfit function in an efficient way.

2. The low computational time required by the adjoint method and the low storage memory employed make this implementation of particular interest especially in applications such as the Full Waveform Inversion.

3. As an example of application, we carried out a local FWI on a portion of the Marmousi model. The final model, after the local inversion fairly match the true model.

4. As a future work we plan to apply this method for real data.

...Thank you for your attention!
How compute the misfit gradient?

The finite difference approximation

Computation of the partial derivatives by using a finite difference approximation:

\[
\left[ \frac{\partial X(\hat{m}_{i,j})}{\partial m_{1,1}} \right] \approx \frac{X(\hat{m}_{1,1} + \epsilon \delta \hat{m}_{1,1}, \hat{m}_{1,2}, \ldots) - X(\hat{m}_{1,1} - \epsilon \delta \hat{m}_{1,1}, \hat{m}_{1,2}, \ldots)}{2\epsilon}
\]

with a small \( \epsilon > 0 \)

Problems:
1. The choice of \( \epsilon \) is not obvious
2. To obtain the gradient is necessary to compute two forward modelling for each spatial derivative
3. This method is impracticable when the number of unknowns is large and the forward modelling is computationally expensive
Example of seismic acquisition

1. The observed data

- Wavelet (Ricker with $f_0=18$ Hz)
- Seismogram of the source on the left
- Seismogram of the source on the right

Spread layout
- 190 receivers
- 2 sources

Marmousi model

Depth (km)

<table>
<thead>
<tr>
<th>km/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3.5</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4.5</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Length (km)

0 1 2 3 4 5 6 7 8 9
2. The forward modelling
- \( nx = 384 \)
- \( nz = 122 \)
- \( dx = dz = 24\text{m} \)
- \( dt = 0.002\text{s} \)

3. The inversion procedure
- Misfit function: \( L^2 \) difference
- Starting model: 1D
- True model: Marmousi
- Optimization algorithm: steepest descend

\[ m_{k+1} = m_k - \gamma_k \nabla m X(m_k) \]

4. The predicted model

We need to compute the gradient of the misfit function for the predicted model
Finite difference vs adjoint method

- The two methods produce very similar solutions.
- The computational time of the adjoint method is considerably reduced:
  - **Adjoint method**: 1 forward modelling + 2 backward modelling
  - **Classical gradient method**: $2 \times 384 \times 122$ forward modelling
Formula for the adjoint state method

• Misfit function:

\[ X(v) = \frac{1}{2} \left( \int_{G} \int_{0}^{T} [ p(t, \vec{x}, v) - p_0(t, \vec{x}) ]^2 \, dt \, dx \right) = \langle X_1(v) \rangle, \]

with \( X_1(v) = \frac{1}{2} \left( [ p(t, \vec{x}, v) - p_0(t, \vec{x}) ] \right)^2 \)

• Wave equation:

\[ \ddot{p}(\vec{x}, t) - v^2 \Delta p(\vec{x}, t) = f(x, t) \]

With \( L \) linear operator
Formula for the adjoint state method

• Misfit function gradient:

\[
\nabla_v X(v) = \langle \nabla_p X_l(p(v, \vec{x}, t)) \ast \nabla_v p(v, \vec{x}, t) \rangle.
\]

• Wave equation gradient:

1) \( \nabla_v L(p(v, \vec{x}, t), v) = \nabla_v f(x, t) \)

2) \( \nabla_v L(p(v, \vec{x}, t), v) + \nabla_p L(p(v, \vec{x}, t), v) \ast \nabla_v p(v, \vec{x}, t) = 0 \)

3) \( \nabla_v L(p(v, \vec{x}, t), v) \ast g^* + \nabla_p L(p(v, \vec{x}, t), v) \ast \nabla_v p(v, \vec{x}, t) \ast g^* = 0 \)

4) \( \langle \nabla_v L(p(v, \vec{x}, t), v) \ast g^* + \nabla_p L(p(v, \vec{x}, t), v) \ast \nabla_v p(v, \vec{x}, t) \ast g^* \rangle = 0 \)
Formula for the adjoint state method

- Combining the two relations:

\[
\nabla_m X(m) = (\nabla_p X_l(p(m, \tilde{x}, t)) \ast \nabla_p p(m, \tilde{x}, t) + \nabla_m L(p(m, \tilde{x}, t), m) \ast g^* + \nabla_p L(p(m, \tilde{x}, t), m) \ast \nabla_m p(m, \tilde{x}, t) \ast g^*)
\]

\[
\nabla_m X(m) = \left( (\nabla_p X_l(p(m, \tilde{x}, t)) + \nabla_p L(p(m, \tilde{x}, t), m) \ast g^* \right) \ast ( \nabla_m p(m, \tilde{x}, t) + \nabla_m L(p(m, \tilde{x}, t), m) \ast g^*)
\]

To remove the \( \nabla m p(m, \tilde{x}, t) \), we use the adjoint term \( g^* \), imposing:

\[
\nabla_p L(p(m, \tilde{x}, t), m) \ast g^* = - \nabla_p X_l(p(m, \tilde{x}, t))
\]

If \( L \) is a linear operator, then

\[
L(g^*(m, \tilde{x}, t), m) = -\nabla_p X_l(p(m, \tilde{x}, t)), \text{ that is is the adjoint equation}
\]