

# Report on research project

Ariane Ducellier

May 6, 2016

## 1 How did I do it?

### 1.1 Seismic Unix

#### 1.1.1 Command lines

I transform the SEGY files into Seismic Unix files with:

```
segypread tape=filename.SEGY verbose=1 endian=0 > filename.su
```

The option endian=0 is for a little-endian computer. Use endian=1 for a big-endian computer. See getEndian code in C to know if your computer is little-endian or big-endian.

I visualize the Seismic Unix files with:

```
suximage < filename.su perc=X
```

Data values outside  $[-X * V_{max} / 100, X * V_{max} / 100]$  are clipped.

To save the image in a file, I use:

```
supsimage < filename.su > filename.eps
```

I get the header values with:

```
surange < filename.su
```

I transform the Seismic Unix files into binary files with:

```
sustrip < filename.su > filename.bin
```

I get a subset of the data with:

```
suwind < filename.su > filename_sub.su key=X min=minX max=maxX
```

If I have an SU file in the wrong format (little-endian instead of big-endian), I can use:

```
suoldtonew < file_wrong_format.su > file_right_format.su
```

### 1.1.2 Description of the header for Seismic Unix

Keyword	Signification
trac1	Trace sequence number within line
tracr	Trace sequence number within reel
fldr	Original field record number
tracf	Trace sequence number within original field record
ep	Energy source point number
trid = 1	Seismic data
offset	Distance from source point to receiver group
gelev	Receiver group elevation
depth	Source depth below surface
sca1	Scalar for elevations and depths (+ multiplier, - divisor)
sca10	Scalar for coordinates (+ multiplier, - divisor)
sx	X source coordinate
gx	X receiver group coordinate
counit=1	Length in meters or feet
ns	Number of sample in this trace
dt	Sample interval of this trace in microseconds
d1	X cdp coordinate
d2	In-line number
f2	Cross-line number

## 1.2 Dataset

### 1.2.1 P-wave velocity

The values in the header are:

Parameter	Value
trac1	1 - 3820
tracr	1 - 3820
fldr	1
tracf	1 - 3820
ep	1
sca1	-10000
sca10	-10000
sx	0 - 477375000
gx	0 - 477375000
ns	1200
dt	5000

There are 3820 points along the X-axis for the P-wave velocity, with 12.5 m spacing, that is the coordinate X goes from 0 to 47737.5 m. There are 1200 points along the Z-axis, with 5 m spacing, that is the coordinate Z goes from 0 to - 5995 m.

In Matlab, it gives a 1200 \* 3820 matrix. In Python, it gives a 3820 \* 1200 array (using C-like index order to reshape the array). The size of the binary file is 1200 \* 3820 \* 4 = 18336000 bytes.

### 1.2.2 Seismic data

The values in the header are:

Parameter	Value
tracr	1 - 513600
fldr	1 - 1600
tracf	1 - 321
ep	1 - 1600
offset	0 - 8000
gelev	-150000
sdepth	150000
scalel	-10000
scalco	-10000
sx	10000000 - 409750000
gx	10000000 - 489750000
ns	2001
dt	4000

There are 1600 sources, with 25 m spacing, that is the coordinate  $X_S$  goes from 1000 to 40975 m. For each source, the first receiver is located at  $X_R = X_S$ . There are 321 receivers, with 25 m spacing, that is the coordinate  $X_R$  goes from  $X_S$  to  $X_S + 8000$ . Sources and receivers are located 15 m below the surface. There are 2001 time steps with time step = 0.004 s. The total duration is thus equal to 8 s.

In Matlab, it gives a  $1600 * 2001 * 321$  array (**I am not sure about the order**). The size of the binary file is  $1600 * 2001 * 321 * 4 = 4110854400$  bytes. The binary file for one shot gives a  $2001 * 321$  matrix in Matlab. In Python, it gives a  $321 * 2001$  array (using C-like index order to reshape the array). The size of the binary file is  $2001 * 321 * 4 = 2569284$  bytes.

### 1.3 SPECFEM2D

#### 1.3.1 Code version and modifications

SPECFEM2D was downloaded on Tiger on November 25<sup>th</sup> 2015. Git version is 9b4ce1d44156ea86a7560574a3476d6f3db2defb.

I modified files:

**meshfem2D/read\_parameter\_file.F90** Modification of lines 693 and 701: "gll" becomes "legacy"

**specfem2d/finalize\_simulation.F90** Modification of lines 91 and 106: "15.5" becomes "20.10"

#### 1.3.2 How to generate the input files

- Use the binary file containing the initial P-wave velocity (file *Vp.bin*) to create an input model with density, P-wave velocity, and S-wave velocity at each coordinate x, z (file *input\_model.dat*). We also obtain a file with the bathymetry (file *bathymetry.dat*). This is done with Matlab code *create\_model.m*. We take  $V_S = \frac{V_P}{\sqrt{3}}$  and  $\rho = 310V_P^{0.25}$  (Gardner's relationship).
- Run SPECFEM2D with the appropriate number of spectral elements in each direction. Use the options MODEL = default and SAVE\_MODEL = legacy in the parameter file *Par\_file*.
- Get the temporary model file of the material properties as an output of this first run (file *proc000000\_model\_velocity.dat\_input*). In particular, we will need the coordinates of all the GLL points.
- Use the input model (file *input\_model.dat*) and the temporary model file (file *proc000000\_model\_velocity.dat\_input*) to get the actual material properties file that will be used in the simulations (file *proc000000\_model\_velocity.dat\_input*). This is done with Matlab code *mesh\_model.m*.
- Use the wavelet file (file *Wavelet.txt*) to interpolate the source function with the chosen time step and number of steps. This is done with Matlab code *create\_source.m*.
- Run SPECFEM2D. For the meshing part, use the options MODEL = legacy and SAVE\_MODEL = default in the parameter file *Par\_file*.

### 1.3.3 Parameters of the simulation

The maximum P-wave velocity is 4219 m/s. The minimum P-wave velocity (outside of the sea water) is 1589 m/s. With the S-wave velocity equal to:  $V_S = \frac{V_P}{\sqrt{3}}$ , the minimum S-wave velocity is 917 m/s.

- **First simulation:** We take 955 elements in the x direction and 120 elements in the z direction, that is the size of the elements is around 50 m. The corresponding maximum frequency is around 4 Hz for the elastic part (taking 4.5 points per S-wave length) and 5.5 Hz for the acoustic part (taking 5.5 points per P-wavelength). We take  $dt = 0.001$  s, which corresponds to a CFL = 0.084.
- **Second simulation:** We take 1910 elements in the x direction and 240 elements in the z direction, that is the size of the elements is around 25 m. The corresponding maximum frequency is around 8 Hz for the elastic part (taking 4.5 points per S-wave length) and 11 Hz for the acoustic part (taking 5.5 points per P-wavelength). We take  $dt = 0.0005$  s, which corresponds to a CFL = 0.084.
- **Third simulation:** We take 4774 elements in the x direction and 600 elements in the z direction, that is the size of the elements is around 10 m. The corresponding maximum frequency is around 20 Hz for the elastic part (taking 4.5 points per S-wave length) and 27.5 Hz for the acoustic part (taking 5.5 points per P-wavelength). We take  $dt = 0.0002$  s, which corresponds to a CFL = 0.084.

### 1.3.4 Source function

If the source is located in the acoustic part, the acoustic wave equation is:

$$\frac{\partial^2 \phi}{\partial t^2} - \alpha^2 \nabla^2 \phi = -\frac{f(t)}{\kappa} \quad (1)$$

In the case of a forward simulation,  $f(t)$  is read in the file *source.txt*. In the case of an adjoint simulation:

- If seismotype = 1,  $f(t)$  is read in the files *Ux\_file\_single.su.adj* or *AA.Sxxxx.BXX.adj*
- If seismotype = 4 or 6,  $f(t)$  is read in the files *Up\_file\_single.su.adj* or *AA.Sxxxx.PRE.adj*

If the source is located in the elastic part, the elastic wave equation is:

$$\frac{\partial^2 \vec{u}}{\partial t^2} - \vec{\nabla} \cdot ((\alpha^2 - 2\beta^2)\vec{\nabla} \cdot \vec{u}I + \beta^2(\vec{\nabla} \otimes \vec{u} + (\vec{\nabla} \otimes \vec{u})^T)) = \vec{f}(t) \quad (2)$$

In the case of a forward simulation,  $\vec{f}(t)$  is read in the file *source.txt* and multiplied by  $-\sin \theta$  for the X-component and  $\cos \theta$  for the Z-component. In the case of an adjoint simulation:

- If seismotype = 1,  $f(t)$  is read in the files *Ux\_file\_single.su.adj* and *Uz\_file\_single.su.adj* or *AA.Sxxxx.BXX.adj* and *AA.Sxxxx.BXZ.adj*

## 1.4 Useful tools

To make a movie:

```
ffmpeg -i image%2d.jpg -vcodec mpeg4 movie.mp4
```

## 2 Conjugate gradient method

## 3 Adjoint method and objective functional

To carry out the inversion, we want to minimize an objective functional  $\chi(m)$ :

$$\chi(m) = \chi(u(m)) \text{ with } m = (m_i) \text{ with } i = 1, \dots, p \text{ and } u = (u_i) \text{ with } i = 1, \dots, n \quad (3)$$

where  $m$  is the model and  $u$  is the acoustic potential or the displacement field. The relationship between the displacement field and the model is described by the acoustic or the seismic wave equations, that can be written under the form:

$$l_i(u, m) = f_i \text{ for } i = 1, \dots, l \quad (4)$$

The acoustic or seismic wave equations can be also be written in a matrix form as:

$$l_i(u, m) = \sum_{j=1}^n L_{ij}(m) u_j \text{ for } i = 1, \dots, l \quad (5)$$

Thus we have:

$$\frac{\partial l_i}{\partial u_j} = L_{ij}(m) \text{ for } i = 1, \dots, l \text{ and } j = 1, \dots, n \quad (6)$$

### 3.1 Adjoint method

To carry out the minimization with the conjugate-gradient method, we need to compute the values of the  $p$  components of the gradient of the objective functional  $\frac{\partial \chi}{\partial m_i}$ . In order to do this, we use the adjoint-state method as explained in Plessix (2006). We can write the gradient of the objective functional  $\chi$  as:

$$\frac{\partial \chi}{\partial m_i} \delta m_i = \sum_{j=1}^n \frac{\partial \chi}{\partial u_j} \frac{\partial u_j}{\partial m_i} \delta m_i \text{ for } i = 1, \dots, p \quad (7)$$

By summing all the components of the gradient, we get:

$$\left( \frac{\partial \chi}{\partial m_i} \right)^T (\delta m_i) = \left( \left( \frac{\partial u_j}{\partial m_i} \right) \left( \frac{\partial \chi}{\partial u_j} \right) \right)^T (\delta m_i) \quad (8)$$

$$\left( \frac{\partial \chi}{\partial m_i} \right)^T (\delta m_i) = \left( \frac{\partial \chi}{\partial u_i} \right)^T \left( \frac{\partial u_j}{\partial m_i} \right)^T (\delta m_i) \quad (9)$$

with  $(\delta u_i) = \left( \frac{\partial u_j}{\partial m_i} \right)^T (\delta m_i)$ .

From Equation (4), we can write:

$$\left( \frac{\partial l_j}{\partial m_i} \right)^T (\delta m_i) + \left( \frac{\partial l_j}{\partial u_i} \right)^T (\delta u_i) = 0 \text{ for } j = 1, \dots, l \quad (10)$$

By using all the components of the acoustic or the seismic wave equations, we get:

$$\left( \frac{\partial l_i}{\partial m_j} \right) (\delta m_i) + \left( \frac{\partial l_i}{\partial u_j} \right) (\delta u_i) = 0 \quad (11)$$

By multiplying with an adjoint vector  $u^\dagger = (u_i^\dagger)$  with  $i = 1, \dots, l$ , we get:

$$(u_i^\dagger)^T \left( \frac{\partial l_i}{\partial m_j} \right) (\delta m_i) + (u_i^\dagger)^T \left( \frac{\partial l_i}{\partial u_j} \right) (\delta u_i) = 0 \quad (12)$$

By summing Equations (9) and (12), we get:

$$\left( \frac{\partial \chi}{\partial m_i} \right)^T (\delta m_i) = \left( \frac{\partial \chi}{\partial u_i} \right)^T (\delta u_i) + (u_i^\dagger)^T \left( \frac{\partial l_i}{\partial m_j} \right) (\delta m_i) + (u_i^\dagger)^T \left( \frac{\partial l_i}{\partial u_j} \right) (\delta u_i) \quad (13)$$

that is:

$$\left( \frac{\partial \chi}{\partial m_i} \right)^T (\delta m_i) = (\delta u_i)^T \left( \frac{\partial \chi}{\partial u_i} \right) + (\delta u_i)^T \left( \frac{\partial l_i}{\partial u_j} \right)^T (u_i^\dagger) + (u_i^\dagger)^T \left( \frac{\partial l_i}{\partial m_j} \right) (\delta m_i) \quad (14)$$

We look for the value of the adjoint state  $(u_i^\dagger)$  such that:

$$\left( \frac{\partial \chi}{\partial u_i} \right) = - \left( \frac{\partial l_i}{\partial u_j} \right)^T (u_i^\dagger) \quad (15)$$

In that case, we have:

$$\left( \frac{\partial \chi}{\partial m_i} \right)^T (\delta m_i) = (u_i^\dagger)^T \left( \frac{\partial l_i}{\partial m_j} \right) (\delta m_i) \quad (16)$$

that is:

$$\frac{\partial \chi}{\partial m_j} = \left(u_i^\dagger\right)^T \left(\frac{\partial l_i}{\partial m_j}\right) \text{ for } j = 1, \dots, p \quad (17)$$

where  $\left(\delta u_i^\dagger\right)^T \left(\frac{\partial l_i}{\partial m_j}\right)$  is the sensitivity kernel for the  $j$ th model parameter.

### 3.2 Augmented functional

We define the augmented functional  $\mathcal{L}$  by:

$$\mathcal{L}(u, u^\dagger, m) = \chi(u(m)) + \left(u_i^\dagger\right)^T ((l_i(u, m)) - (f_i)) \quad (18)$$

If  $u$  is a solution of the acoustic or the seismic wave equations, then we have:

$$\mathcal{L}(u, u^\dagger, m) = \chi(u(m)) \quad (19)$$

The adjoint state  $u^\dagger$  is independent of the model  $m$ , thus we can write:

$$\left(\frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial u_i}\right)^T \left(\frac{\partial u_i}{\partial m_j}\right) + \frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial m_j} = \frac{\partial \chi}{\partial m_j} \text{ for } j = 1, \dots, p \quad (20)$$

By summing all the components of the gradient, we get:

$$\left(\frac{\partial u_j}{\partial m_i}\right) \left(\frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial u_i}\right) + \left(\frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial m_i}\right) = \left(\frac{\partial \chi}{\partial m_i}\right) \quad (21)$$

We choose  $u^\dagger$  such that  $\left(\frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial u_i}\right) = (0)$ , that is:

$$\left(\frac{\partial \chi}{\partial u_i}\right) + \left(\frac{\partial l_i}{\partial u_j}\right)^T \left(u_i^\dagger\right) = (0) \quad (22)$$

Equation (22) defining the adjoint state is the same as Equation (15) in the previous subsection. From Equation (19), we also have:

$$\left(\frac{\partial \chi}{\partial m_i}\right) = \left(\frac{\partial \mathcal{L}(u, u^\dagger, m)}{\partial m_i}\right) \quad (23)$$

that is:

$$\left(\frac{\partial \chi}{\partial m_i}\right) = \left(\frac{\partial l_i}{\partial m_j}\right)^T \left(u_i^\dagger\right) \quad (24)$$

Equation (24) defining the sensitivity kernel is the same as Equation (16) in the previous subsection.

### 3.3 Objective functional

The acoustic or seismic wave equations can be written using initial boundary conditions for each of the  $S$  sources:

$$\begin{cases} (u_i^s(0)) = (0) \text{ for } s = 1, \dots, S \\ \left(\frac{\partial u_i^s(0)}{\partial t}\right) = (0) \text{ for } s = 1, \dots, S \\ (l_i(u^s(t), m)) = (f_i^s(t)) \text{ for } s = 1, \dots, S \end{cases} \quad (25)$$

The operator  $l$  can be written as:

$$(l_i(u(t), m)) = \left(\rho_i \frac{\partial^2 u_i(t)}{\partial t^2}\right) + (L'_{ij})(u_i(t)) \quad (26)$$

The objective functional is defined by:

$$\chi(m) = \frac{1}{2} \sum_{s,r} \int_0^T \left((S_{s,ri})^T (u_i^s(t)) - d_{s,r}(t)\right)^2 dt \quad (27)$$

where  $S_{s,r}$  is a restriction operator onto the receiver position  $r$  for source  $s$ .

The augmented functional  $\mathcal{L}$  becomes then:

$$\begin{aligned} \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1}) &= \frac{1}{2} \sum_{s,r} \int_0^T \left( (S_{s,ri})^T (u_i^s(t)) - d_{s,r}(t) \right)^2 dt \\ &+ \sum_s \int_0^T \left( u_i^{\dagger s}(t) \right)^T \left( \left( \rho_i \frac{\partial^2 u_i^s(t)}{\partial t^2} \right) + (L'_{ij}) (u_i^s(t)) - (f_i^s(t)) \right) dt \\ &+ \sum_s \left( u_i^{\dagger s0} \right)^T (u_i^s(0)) + \sum_s \left( u_i^{\dagger s1} \right)^T \left( \frac{\partial u_i^s(0)}{\partial t} \right) \end{aligned} \quad (28)$$

where  $u^{\dagger s0}$  and  $u^{\dagger s1}$  are the adjoint states associated to the initial boundary conditions for source  $s$ . Using two integrations by part, we get:

$$\begin{aligned} &\int_0^T \left( u_i^{\dagger s}(t) \right)^T \left( \rho_i \frac{\partial^2 u_i^s(t)}{\partial t^2} \right) dt = \\ &\int_0^T \left( \frac{\partial^2 u_i^{\dagger s}(t)}{\partial t^2} \right)^T (\rho_i u_i^s(t)) dt \\ &+ \left( u_i^{\dagger s}(T) \right)^T \left( \rho_i \frac{\partial u_i^s(T)}{\partial t} \right) - \left( u_i^{\dagger s}(0) \right)^T \left( \rho_i \frac{\partial u_i^s(0)}{\partial t} \right) \\ &- \left( \frac{\partial u_i^{\dagger s}(T)}{\partial t} \right)^T (\rho_i u_i^s(T)) + \left( \frac{\partial u_i^{\dagger s}(0)}{\partial t} \right)^T (\rho_i u_i^s(0)) \end{aligned} \quad (29)$$

The augmented functional  $\mathcal{L}$  is thus equal to:

$$\begin{aligned} \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1}) &= \frac{1}{2} \sum_{s,r} \int_0^T \left( (S_{s,ri})^T (u_i^s(t)) - d_{s,r}(t) \right)^2 dt \\ &+ \sum_s \int_0^T \left( \frac{\partial^2 u_i^{\dagger s}(t)}{\partial t^2} \right)^T (\rho_i u_i^s(t)) dt \\ &+ \sum_s \left( u_i^{\dagger s}(T) \right)^T \left( \rho_i \frac{\partial u_i^s(T)}{\partial t} \right) - \sum_s \left( u_i^{\dagger s}(0) \right)^T \left( \rho_i \frac{\partial u_i^s(0)}{\partial t} \right) \\ &- \sum_s \left( \frac{\partial u_i^{\dagger s}(T)}{\partial t} \right)^T (\rho_i u_i^s(T)) + \sum_s \left( \frac{\partial u_i^{\dagger s}(0)}{\partial t} \right)^T (\rho_i u_i^s(0)) \\ &+ \sum_s \int_0^T \left( u_i^{\dagger s}(t) \right)^T ((L'_{ij}) (u_i^s(t)) - (f_i^s(t))) dt \\ &+ \sum_s \left( u_i^{\dagger s0} \right)^T (u_i^s(0)) + \sum_s \left( u_i^{\dagger s1} \right)^T \left( \frac{\partial u_i^s(0)}{\partial t} \right) \end{aligned} \quad (30)$$

To compute the adjoint state  $(u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1})$  for each source  $s$ , we need to compute all the partial derivatives  $\frac{\partial \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1})}{\partial u_i^s(t)}$ ,  $\frac{\partial \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1})}{\partial u_i^s(0)}$ ,  $\frac{\partial \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1})}{\partial \frac{\partial u_i^s(0)}{\partial t}}$  and  $\frac{\partial \mathcal{L}(u^s, u^{\dagger s}, u^{\dagger s0}, u^{\dagger s1})}{\partial \frac{\partial u_i^s(T)}{\partial t}}$ , and set them to 0. For each source  $s$ , we thus have the adjoint state equations:

$$\begin{cases} \sum_r \left( (S_{s,ri})^T (u_i^s(t)) - d_{s,r}(t) \right) (S_{s,ri}) + \left( \rho_i \frac{\partial^2 u_i^{\dagger s}(t)}{\partial t^2} \right) + (L'_{ij})^T (u_i^{\dagger s}(t)) = (0) \\ \left( \rho_i \frac{\partial u_i^{\dagger s}(0)}{\partial t} \right) + \left( u_i^{\dagger s0} \right) = (0) \\ - \left( \rho_i \frac{\partial u_i^{\dagger s}(T)}{\partial t} \right) = (0) \\ - \left( \rho_i u_i^{\dagger s}(0) \right) + \left( u_i^{\dagger s1} \right) = (0) \\ \left( \rho_i u_i^{\dagger s}(T) \right) = (0) \end{cases} \quad (31)$$

To get the adjoint state for each source  $s$ , we need to solve the system:

$$\begin{cases} (u_i^{\dagger s}(T)) = (0) \\ \left(\frac{\partial u_i^{\dagger s}(T)}{\partial t}\right) = (0) \\ (l'_i(u_i^{\dagger s}(t), m)) = (f_i^{\prime s}(t)) \end{cases} \quad (32)$$

where the operator  $l'$  is equal to:

$$(l'_i(u(t), m)) = \left(\rho_i \frac{\partial^2 u_i(t)}{\partial t^2}\right) + (L'_{ij})^T (u_i(t)) \quad (33)$$

and the source function  $f^{\prime s}(t)$  is equal to:

$$(f_i^{\prime s}(t)) = - \sum_r \left( (S_{s,ri})^T (u_i^s(t)) - d_{s,r}(t) \right) (S_{s,ri}) \quad (34)$$

This is equivalent to solving the system:

$$\begin{cases} (u_i^{\dagger s}(0)) = (0) \\ \left(\frac{\partial u_i^{\dagger s}(0)}{\partial t}\right) = (0) \\ (l'_i(u_i^{\dagger s}(t), m)) = (f_i^{\prime s}(T - t)) \end{cases} \quad (35)$$

The matrix  $(L'_{ij})$  is symmetric, thus we can solve the adjoint problem using the same method as the one we used to solve the forward problem, but using a different source.

### 3.4 Summary of the procedure

For each source  $s$ , we solve the forward problem:

$$\begin{cases} (u_i^s(0)) = (0) \\ \left(\frac{\partial u_i^s(0)}{\partial t}\right) = (0) \\ (l_i(u_i^s(t), m)) = (f_i(t)) \end{cases} \quad (36)$$

where  $f(t)$  is the wavelet given in the Chevron dataset. Then, for each source  $s$  we compute the adjoint source  $f^{\prime s}(t)$ :

$$(f_i^{\prime s}(t)) = - \sum_r \left( (S_{s,ri})^T (u_i^s(T - t)) - d_{s,r}(T - t) \right) (S_{s,ri}) \quad (37)$$

using the residual between the acoustic potential and the waveform given in the Chevron dataset for each receiver associated to the current source. Then, for each source, we solve the adjoint problem using the adjoint source. We can then compute the gradient for each source  $s$ :

$$\frac{\partial \chi}{\partial m_j} = \int_0^T \left( u_i^{\dagger s}(T - t) \right)^T \left( \frac{\partial l_i}{\partial m_j} \right) dt \text{ for } j = 1, \dots, p \quad (38)$$

where  $\int_0^T \left( \delta u_i^{\dagger s}(T - t) \right)^T \left( \frac{\partial l_i}{\partial m_j} \right) dt$  is the sensitivity kernel for the  $j$ th model parameter.

We then sum the gradient of all the sources to get the gradient that will be used in the conjugate-gradient method.

## 4 Articles and books

Introduction to seismology. Peter M. Shearer. Cambridge: Cambridge University Press, 2009. QE534.3 .S455 2009

An introduction to seismology, earthquakes, and earth structure. Seth Stein, Michael Wysession. Malden, MA: Blackwell Pub., 2003. QE534.3 .S74 2003

Modern global seismology. Edited by Thorne Lay, Terry C. Wallace. San Diego: Academic Press, c1995. QE534.2 .M62 1995

Principles of seismology. Agustín Udias. Cambridge ; New York: Cambridge University Press, 1999. QE534.2 .U35 1999



The seismic wavefield. B.L.N. Kennett. Cambridge ; New York: Cambridge University Press, 2001-2002. QE541 .K46 2001

Seismic wave propagation in stratified media. Brian Kennett. Acton, A.C.T.: ANU E Press, 2009. QE538.5 .K46 1983

Quantitative seismology. Keiiti Aki, Paul G. Richards. Sausalito, CA: University Science Books, 2009. QE539 .A64

Theoretical global seismology. F.A. Dahlen and Jeroen Tromp. Princeton, N.J.: Princeton University Press, 1998. QE534.2 .D34 1998