Supporting Information for "Multi-SEM: Multi-solver Spectral-Element and Adjoint Methods for Computational Seismology"

Yujiang Xie[1], Catherine Rychert[1], Nicholas Harmon[1]

Qinya Liu[2], Dirk Gajewski[3]

[1]Ocean and Earth Science, University of Southampton, UK

[2]Department of Physics & Department of Earth Sciences, University of Toronto, Canada

[3]Institute of Geophysics, University of Hamburg, Germany

Contents of this file

1. Fréchet kernels in three model parameter sets
2. Hessian kernels in three model parameter sets
3. Wavefield storage method (WSM) for computing Hessian kernels
4. Figures S1 to S14, where Figures S6 to S14 are shown for the WSM

1. Fréchet kernels in three model parameter sets

Fréchet kernels are related to the first-order derivatives of the seismic data functional, χ. Assuming the perturbation of the functional as $\delta\chi$, we may have (also see Tromp et al., 2005)

$$\delta\chi = \int_V \overline{K}_m \frac{\delta \mathbf{m}}{\mathbf{m}}\, d^3\mathbf{x} = \int_V K_m \delta\mathbf{m}\, d^3\mathbf{x}, \tag{1}$$

where $\overline{K}_m$ or K_m denotes the *Fréchet* kernels, and V denotes the model volume. The kernels applied to the perturbation of the model ($\delta\mathbf{m}$) can be further expressed with respect to three different model parameterizations as (see Tromp et al., 2005; Fichtner &

Trampert, 2011a)

$$K_m \delta \mathbf{m} = \begin{pmatrix} K_\rho \\ K_\kappa \\ K_\mu \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \delta\rho \\ \delta\kappa \\ \delta\mu \end{pmatrix} = \begin{pmatrix} K_\rho \\ K_\lambda \\ K_\mu \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \delta\rho \\ \delta\lambda \\ \delta\mu \end{pmatrix} = \begin{pmatrix} K'_\rho \\ K_\alpha \\ K_\beta \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \delta\rho \\ \delta\alpha \\ \delta\beta \end{pmatrix}, \tag{2}$$

where the superscript T denotes the vector transpose. The model parameters ρ, κ and μ indicate the density, bulk and shear moduli. The λ and μ are the lamé parameters. The μ used in the two sets of model parameters is the same. The α and β are the compressional and shear wave speeds. The *Fréchet* kernels can be further expressed by a cross-correlation of the forward and adjoint fields as (see e.g., Tromp et al., 2005; Liu & Tromp, 2006)

$$\begin{pmatrix} K_\rho \\ K_\kappa \\ K_\mu \end{pmatrix} = \begin{pmatrix} K_\rho(\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\kappa(\mathbf{s}^\dagger, \mathbf{s}) \\ K_\mu(\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}, \begin{pmatrix} K_\rho \\ K_\lambda \\ K_\mu \end{pmatrix} = \begin{pmatrix} K_\rho(\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\lambda(\mathbf{s}^\dagger, \mathbf{s}) \\ K_\mu(\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}, \begin{pmatrix} K'_\rho \\ K_\alpha \\ K_\beta \end{pmatrix} = \begin{pmatrix} K'_\rho(\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\alpha(\mathbf{s}^\dagger, \mathbf{s}) \\ K_\beta(\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}. \tag{3}$$

Two approaches may be used in practice to compute the Fréchet kernels. One is the *field storage method* which first saves the forward field in space and time from the forward simulation, and then during the adjoint simulation, reads the corresponding time step of the forward wavefield into the temporary memory to conduct the calculation for the Fréchet kernel. During the time integration for kernels, only one step of the forward wavefield is read in at one time, therefore there is no need to carry the entire forward field in memory. The field storage method is suitable for small or local scale simulations, but becomes computationally prohibitive for large or global scale simulations due to the large amount of disk storage required and the frequent I/O calls. The second method is the *forward-field back-reconstruction method* which trades CPU hours with storage requirements as it only saves a very small subsets of time steps of the forward field from the forward simulation, and during the adjoint simulation, reconstructs the forward field back in time to combine the forward and adjoint wavefield directly in memory for the kernel calculation. For a purely elastic kernel calculation, only the last state of the forward field needs to be saved as the start point for the backward reconstruction during the adjoint simulation (see Tromp et al., 2005; Liu & Tromp, 2006; Tromp et al., 2008).

For the anelastic case, the parsimonious storage method (Komatitsch et al., 2016) can be used with one additional forward simulation to account for the attenuation for the adjoint source, and the forward fields are stored at selected checkpoints and recomputed during the adjoint simulation.

2. Hessian kernels in three model parameter sets

We use the Hessian operator as defined by Fichtner and Trampert (2011a), which may be rewritten as

$$H(\delta\mathbf{m}_1, \delta\mathbf{m}_2) = \int_V K_m^1 \delta\mathbf{m}_2 \; d^3\mathbf{x} = \int_V (\mathrm{H}_a + \mathrm{H}_b + \mathrm{H}_c)\delta\mathbf{m}_2 d^3\mathbf{x}, \tag{4}$$

where $K_m^1 = \mathrm{H}_a + \mathrm{H}_b + \mathrm{H}_c$ denotes the Hessian kernels, which can be expressed differently with respect to different model parameterizations.

1. When the model is given by ρ, κ, and μ, we may have

$$\mathrm{H}_a(\rho,\kappa,\mu) = \begin{pmatrix} K_\rho(\mathbf{s}^\dagger, \delta\ddot{\mathbf{s}}) \\ K_\kappa(\mathbf{s}^\dagger, \delta\mathbf{s}) \\ K_\mu(\mathbf{s}^\dagger, \delta\mathbf{s}) \end{pmatrix}, \mathrm{H}_b(\rho,\kappa,\mu) = \begin{pmatrix} K_\rho(\delta\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\kappa(\delta\mathbf{s}^\dagger, \mathbf{s}) \\ K_\mu(\delta\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}, \mathrm{H}_c(\rho,\kappa,\mu) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5}$$

2. When the model is given by ρ, λ, μ, we may have

$$\mathrm{H}_a(\rho,\lambda,\mu) = \begin{pmatrix} K_\rho(\mathbf{s}^\dagger, \delta\ddot{\mathbf{s}}) \\ K_\lambda(\mathbf{s}^\dagger, \delta\mathbf{s}) \\ K_\mu(\mathbf{s}^\dagger, \delta\mathbf{s}) \end{pmatrix}, \mathrm{H}_b(\rho,\lambda,\mu) = \begin{pmatrix} K_\rho(\delta\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\lambda(\delta\mathbf{s}^\dagger, \mathbf{s}) \\ K_\mu(\delta\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}, \mathrm{H}_c(\rho,\lambda,\mu) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6}$$

3. When the model given by ρ, α, β, we may have

$$\mathrm{H}_a(\rho,\alpha,\beta) = \begin{pmatrix} K'_\rho(\mathbf{s}^\dagger, \delta\ddot{\mathbf{s}}) \\ K_\alpha(\mathbf{s}^\dagger, \delta\mathbf{s}) \\ K_\beta(\mathbf{s}^\dagger, \delta\mathbf{s}) \end{pmatrix}, \mathrm{H}_b(\rho,\alpha,\beta) = \begin{pmatrix} K'_\rho(\delta\mathbf{s}^\dagger, \ddot{\mathbf{s}}) \\ K_\alpha(\delta\mathbf{s}^\dagger, \mathbf{s}) \\ K_\beta(\delta\mathbf{s}^\dagger, \mathbf{s}) \end{pmatrix}, \tag{7}$$

$$\mathrm{H}_c(\rho,\alpha,\beta) = \begin{pmatrix} \rho^{-1}K_\alpha(\mathbf{s}^\dagger, \mathbf{s})\delta\alpha + \rho^{-1}K_\beta(\mathbf{s}^\dagger, \mathbf{s})\delta\beta \\ \rho^{-1}K_\alpha(\mathbf{s}^\dagger, \mathbf{s})\delta\rho + \alpha^{-1}K_\alpha(\mathbf{s}^\dagger, \mathbf{s})\delta\alpha \\ \rho^{-1}K_\beta(\mathbf{s}^\dagger, \mathbf{s})\delta\rho + \beta^{-1}K_\beta(\mathbf{s}^\dagger, \mathbf{s})\delta\beta \end{pmatrix}. \tag{8}$$

Eq.(5)-eq.(8) show the link between *Fréchet* kernels (Tromp et al., 2005) and the Hessian kernels (Fichtner & Trampert, 2011a) for different model parameterizations. The H_b practically includes two parts: one is the $\mathrm{H}_b^{\langle m \rangle}$ which is due to the perturbation of the model, and the other is the $\mathrm{H}_b^{\langle s \rangle}$ which is due to the perturbation of the adjoint source.

The $\mathrm{H}_b^{\langle m \rangle}$ can be given in different model parameterizations as

$$\mathrm{H}_b^{\langle m \rangle}(\rho, \kappa, \mu) = \begin{pmatrix} K_\rho(\delta \mathbf{s}_m^\dagger, \ddot{\mathbf{s}}) \\ K_\kappa(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \\ K_\mu(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \end{pmatrix}, \tag{9}$$

$$\mathrm{H}_b^{\langle m \rangle}(\rho, \lambda, \mu) = \begin{pmatrix} K_\rho(\delta \mathbf{s}_m^\dagger, \ddot{\mathbf{s}}) \\ K_\lambda(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \\ K_\beta(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \end{pmatrix}, \tag{10}$$

$$\mathrm{H}_b^{\langle m \rangle}(\rho, \alpha, \beta) = \begin{pmatrix} K'_\rho(\delta \mathbf{s}_m^\dagger, \ddot{\mathbf{s}}) \\ K_\alpha(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \\ K_\beta(\delta \mathbf{s}_m^\dagger, \mathbf{s}) \end{pmatrix}, \tag{11}$$

where $\delta \mathbf{s}_m^\dagger$ indicates the approximate perturbed adjoint field due to only perturbation in the model. The $\mathrm{H}_b^{\langle s \rangle}$ referred to the approximate Hessian kernels defined by Fichtner and Trampert (2011a), which could be also rewritten in three model parameterizations as

$$\mathrm{H}_b^{\langle s \rangle}(\rho, \kappa, \mu) = \begin{pmatrix} K_\rho(\delta \mathbf{s}_s^\dagger, \ddot{\mathbf{s}}) \\ K_\kappa(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \\ K_\mu(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \end{pmatrix}, \tag{12}$$

$$\mathrm{H}_b^{\langle s \rangle}(\rho, \lambda, \mu) = \begin{pmatrix} K_\rho(\delta \mathbf{s}_s^\dagger, \ddot{\mathbf{s}}) \\ K_\lambda(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \\ K_\beta(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \end{pmatrix}, \tag{13}$$

$$\mathrm{H}_b^{\langle s \rangle}(\rho, \alpha, \beta) = \begin{pmatrix} K'_\rho(\delta \mathbf{s}_s^\dagger, \ddot{\mathbf{s}}) \\ K_\alpha(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \\ K_\beta(\delta \mathbf{s}_s^\dagger, \mathbf{s}) \end{pmatrix}, \tag{14}$$

where $\delta \mathbf{s}_s^\dagger$ indicates the approximate perturbed adjoint field due to only perturbation in the adjoint source.

2.1. Implementation

In principle, the approximate or full Hessian kernels can be computed by using existing spectral-element packages for wavefield generation with the perturbed wavefields computed in advance. The challenge is to compute and use these fields on the fly as shown in this work. Once these fields are computed for each or incremental time step, the Hessian kernels can be calculated by using, e.g., the *compute_kernels()* subroutine in the SPECFEM2D/3D packages (`https://geodynamics.org/cig/software/specfem2d/` and `https://geodynamics.org/cig/software/specfem3d/`), where one

just needs to substitute the regular fields with the perturbed field as indicated in eq.(5)-(14). Similar to Fréchet kernel calculation for each time step, the computation of Hessian kernels is performed at individual time step. Since only one single time step of all fields and the integrated kernels are kept in memory on the fly, the use of a sub-sampled calculation may be unnecessary.

3. Wavefield storage method (WSM) for computing Hessian kernels

The Hessian kernels can be computed when the required fields are determined. To compute the required fields, we design and use one forward simulation and three adjoint simulations (see Figure S7). The forward simulation is to compute and save four forward fields, that is $\mathbf{s}(\mathbf{m}_1)$, $\mathbf{s}(\mathbf{m}_2)$, $\ddot{\mathbf{s}}(\mathbf{m}_1)$, $\ddot{\mathbf{s}}(\mathbf{m}_2)$, where $\mathbf{m}_2 = \mathbf{m}_1 + v\delta\mathbf{m}$. The first and second adjoint simulations (Adjoint simulation I) are designed to compute and save the adjoint fields $\mathbf{s}_s^\dagger(\mathbf{m}_1)$ and $\mathbf{s}_m^\dagger(\mathbf{m}_2)$. The third adjoint simulation, the last one, is a simultaneous adjoint simulation and the Hessian calculation (Adjoint simulation II), where the adjoint simulation is to compute the adjoint field $\mathbf{s}^\dagger(\mathbf{m}_1)$ on the fly during the construction of Hessian kernels.

3.1. Models

We use two synthetic models and take the Specfem2D package as examples. The first model is a homogeneous model ($\mathbf{m}_1$) and the second model is a perturbation model ($\mathbf{m}_2 = \mathbf{m}_1 + v\delta\mathbf{m}$) relative to the homogeneous one (see Fig S6 for the compressional wave speed and the source and receiver geometry). We placed the scatter on the kernel path and set the scatter size close to the dominant wavelength to account for the perturbed fields. Both models are set to 800 $km \times 360$ km in the horizontal and vertical direction. For the mesher, we use the internal mesher of the Specfem2D package. We placed 400 elements in the horizontal direction and 360 elements in the vertical direction, leading to ~ 500 m and $\sim$

250 m grid-point spacing respectively for the mesher since 5×5 Gauss–Lobatto–Legendre (GLL) points for each element are used. We use a dense element mesh for the model to eliminate the effects of grip-point intervals to the kernel imaging since we focus on the computation of Hessian kernels here. A detailed resolution analysis or the use of external mesher tools, one can refer to Fichtner and Trampert (2011b) and Peter et al. (2011).

The model material properties for the homogeneous model is set to density 2900 kg/m^3, compressional wave speed $\alpha = 8000\ m/s$ and shear wave speed $\beta = 4800\ m/s$. We use +10% relative model perturbation to model $\mathbf{m}_1$ and the scatter perturbation is of 10 $km \times 10\ km$ located within the path that links the source and the receiver (see Fig S6b). For simplicity and to show how the Hessian kernels are computed, we use a point source and place it at $(x, z) = (100\ km, -260\ km)$. A standard Ricker wavelet with the dominant frequency of 0.5 Hz is applied. So the minimum wavelengths for the P and S waves are 16 km and 9.6 km respectively. The receiver is placed at the model surface at $(x, z) = (600\ km, 0\ km)$. For this example, we use 10,000 time steps with $dt = 0.01\ s$ for the simulation. The number of time steps and the dt can be estimated by the model setup and the phases to be investigated.

3.2. Forward simulation

Typically, the forward simulation includes two simulations, one for the model $\mathbf{m}_1$ and the other for model $\mathbf{m}_2$. Both can be performed individually or simultaneously. In the forward simulation, the fields computed at each time step or a incremental time step are saved for the two models. The seismograms for the two models are saved to compute the two adjoint sources $\mathbf{f}^\dagger(\mathbf{m}_1)$ and $\mathbf{f}^\dagger(\mathbf{m}_2)$. To facilitate the simulation, we run the two simulations for the two models simultaneously since there are sufficient memory left for each CPU. The use of a simultaneous simulation for the two models is convenient since there one just needs to input the two models and the forward fields and seismograms are computed once

a time. In the simultaneous simulation, there are $\sim$ 160/100 memory and $\sim$ 180/100 computational time required when compared to the use of the single simulation twice. The reduction in memory and computational time less than double is due to the same mesh database used for the simulation, excluding the two models imported externally. Figure S8 shows four time steps of the forward displacement fields and their perturbed fields computed from the two models. The perturbed forward fields are observed (see Figure S8i,f,c) when the forward fields pass through the scatter.

3.3. Adjoint simulation I

There are two adjoint simulations in the *Adjoint simulation I* stage (see Figure S7). The first adjoint simulation is to compute and save the adjoint field $\mathbf{s}_s^\dagger(\mathbf{m}_1)$, which accounts for the perturbation due to the adjoint source. The adjoint source $\mathbf{f}^\dagger(\mathbf{m}_2)$ computed from the measurements for model $\mathbf{m}_2$ is used (see Figure S9 for a quick view), where we use the traveltime adjoint source (Tromp et al., 2005). Figure S10 shows four time steps of the adjoint fields $\mathbf{s}^\dagger(\mathbf{m}_1)$ and $\mathbf{s}_s^\dagger(\mathbf{m}_1)$ and their perturbations $\delta\mathbf{s}_s^\dagger$. The time-reversed perturbed adjoint fields $\delta\mathbf{s}_s^\dagger$ (the third column in Figure S10) are weaker than the regular adjoint fields (the first and the second column). The second adjoint simulation in the *Adjoint simulation I* is to compute $\mathbf{s}_m^\dagger(\mathbf{m}_2)$, which accounts for the perturbation of the model, where the adjoint source $\mathbf{f}^\dagger(\mathbf{m}_1)$ (see Figure S9) computed from the measurements for model $\mathbf{m}_1$ is used. Figure S11 shows four time steps of the adjoint fields $\mathbf{s}^\dagger(\mathbf{m}_1)$ and $\mathbf{s}_m^\dagger(\mathbf{m}_2)$ and the perturbed fields $\delta\mathbf{s}_m^\dagger$. The time-reversed perturbed adjoint fields are generated when the regular fields pass through the scatter (see Figure S11i,l).

3.4. Adjoint adjoint II

The *Adjoint simulation II* is a simultaneous adjoint simulation and the Hessian kernel calculation, where the adjoint simulation is to compute $\mathbf{s}^\dagger(\mathbf{m}_1)$ on the fly, which is triggered

by the adjoint source $\mathbf{f}^{\dagger}(\mathbf{m}_1)$. In the adjoint simulation, each time step or a skipped time step of *the four forward fields and the two adjoint fields* (the saved fields) are read into the temporary memory for constructing the Hessian kernels for that time step. The final Hessian kernels are accumulated(integrated) by previous Hessian kernels computed at each counted step. In the implementation, only one time step of the Hessian kernels (i.e., the integrated Hessian kernels) is kept in the temporary memory until it is output finally. Figure S12 shows four components of the Hessian kernels: H_a, $\mathrm{H}_b^{\langle m \rangle}$, $\mathrm{H}_b^{\langle s \rangle}$, and H_c computed in this simulation. The four components individually with respect to the density can be computed when used $\ddot{\mathbf{a}}(\mathbf{m}_1)$ and $\ddot{\mathbf{a}}(\mathbf{m}_2)$. Only two forward and two adjoint fields need to be stored if without considering the density kernels.

Figure S13 shows the conventional Fréchet kernels, where only the K_α component is observed well since only the P phase on the seismograms is used for the adjoint source calculation. Figure S14 shows the full Hessian kernels investigated for the same P phase. The full Hessian kernels are obtained by summing the H_a, $\mathrm{H}_b^{\langle m \rangle}$, $\mathrm{H}_b^{\langle s \rangle}$, and H_c components together, which includes the approximate Hessian kernels $\mathrm{H}_b^{\langle s \rangle}$ (see second row in Figure S12). The computation of full Hessian kernels includes the computation of Fréchet kernels as required by the H_c calculation. The disk space required for the WSM approach is big even for the 2D example, it takes about 400 GB disk space to store the required fields even if without considering the density perturbation for the density kernel calculation.

References

Fichtner, A., & Trampert, J. (2011a). Hessian kernels of seismic data functionals based upon adjoint techniques. *Geophys. J. Int.*, *185*, 775–798.

Fichtner, A., & Trampert, J. (2011b). Resolution analysis in full waveform inversion. *Geophys. J. Int.*, *187*, 1604–1624.

Komatitsch, D., Xie, Z., Bozdağ, E., Andrade, E. S. d., Peter, D., Liu, Q., & Tromp, J. (2016). Anelastic sensitivity kernels with parsimonious storage for adjoint tomography and full waveform inversion. *Geophys. J. Int.*, *206*, 1467–1478.

Liu, Q., & Tromp, J. (2006). Finite-frequency kernels based on adjoint methods. *Bulletin of the Seismological Society of America*, *96*, 2383–2397.

Peter, D., Komatitsch, D., Luo, Y., Martin, R., Goff, N. L., Casarotti, E., ... Tromp, J. (2011). Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes. *Geophys. J. Int.*, *186*, 721–739.

Tromp, J., Komatitsch, D., & Liu, Q. (2008). Spectral-element and adjoint methods in seismology. *Communications in Computational Physics*, *3*, 1–32.

Tromp, J., Tape, C., & Liu, Q. (2005). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophys. J. Int.*, *160*, 195–216.

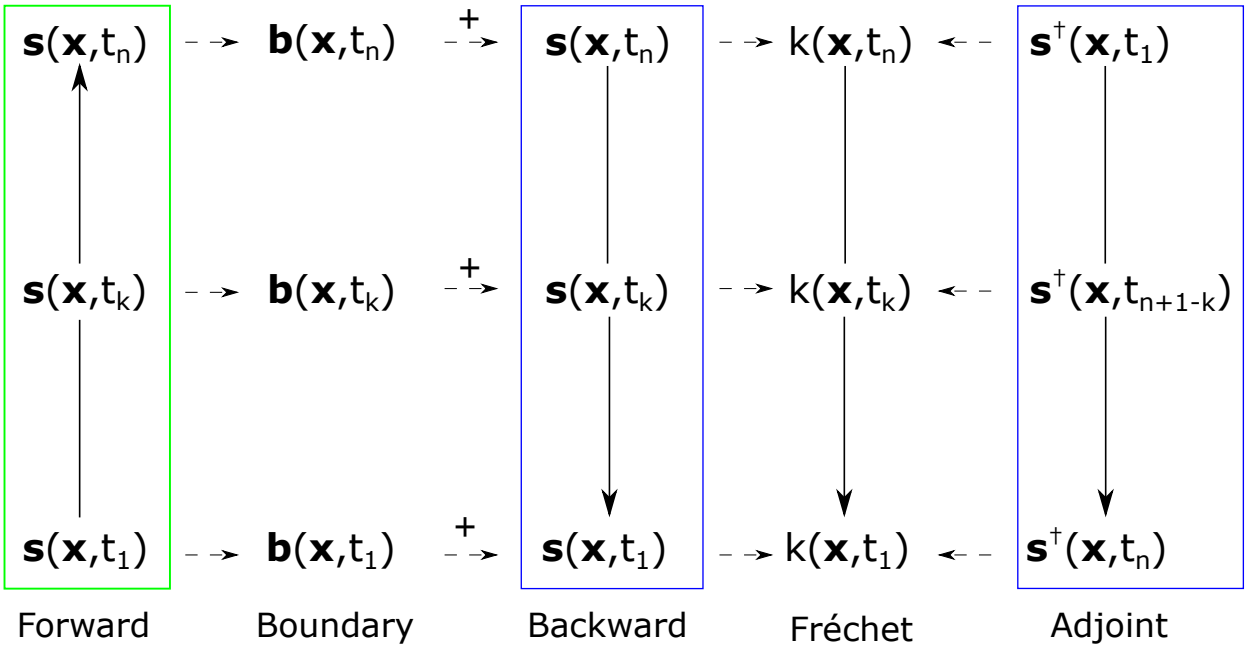

Figure S1. Forward simulation (green rectangle) and the simultaneous backward and adjoint simulation (blue rectangles) for computing the Fréchet kernels. The forward simulation is started from the first time step t_1 and ended at the last time step t_n. The absorbing boundary field $\mathbf{b}(\mathbf{x}, \mathrm{t_k})$ of each time step t_k and the last state field $\mathbf{s}(\mathbf{x}, \mathrm{t_n})$ are stored in the forward simulation. The backward simulation takes the last state field as a start point and reconstructed the forward field backward in time. In each time step, the absorbing boundary field $\mathbf{b}(\mathbf{x}, \mathrm{t_k})$ is re-injected into the backward simulation to reconstruct the forward fields (called backward fields here). The adjoint simulation is started from the time-reversed adjoint source from the receivers. The Fréchet kernels at each time step or at a sub-sampled time step are constructed on the fly based upon the backward and adjoint fields. If each time step is used, the kernels are summed at each time step until the final step as $K_m = \sum_{k=1}^{n} \mathrm{K}(\mathbf{x}, \mathrm{t_k})\delta t$, where δt is time interval in the simulation.

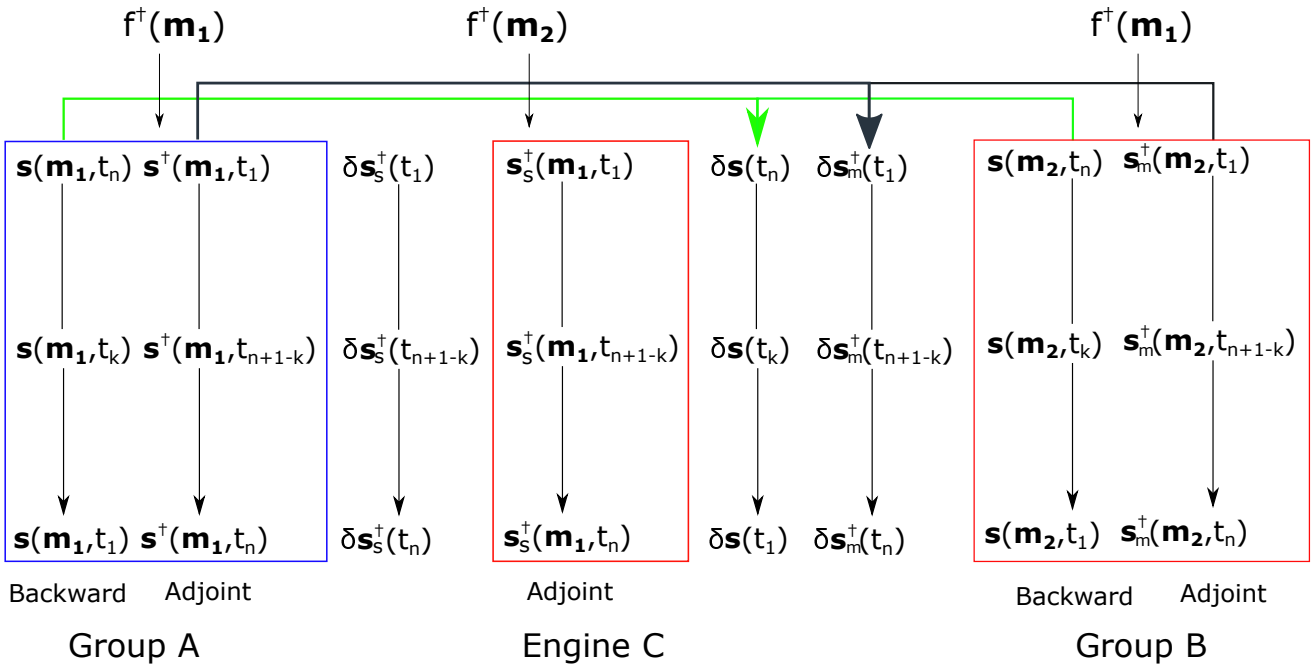

Figure S2. Simultaneous backward and adjoint simulation in the Multi-SEM, where five SEM solvers are coupled and used (see the five arrows within the three rectangles). Group A: two SEM solvers are coupled and used under the same mesh database, where one solver is used for the backward simulation and the other solver is used for the adjoint simulation. This is similar to the adjoint simulation in the computation of Fréchet kernels. Group A is designed to compute the backward and adjoint fields for model $\mathbf{m}_1$. One the right side, Group B adopts two SEM solvers to compute the backward and adjoint fields for the the perturbed model $\mathbf{m}_2$. Engine C is one solver engine designed to compute the adjoint field due to the perturbation of the adjoint source $f^\dagger(\mathbf{m}_2)$. The simulation in Engine C is the same as the adjoint simulation of Group A except the source term. Since all the fields are computed on the fly for each designed time step (each time step or a skipping time step), the perturbed fields to be used in the calculation of Hessian kernels can be determined, e.g., by the first-order finite-difference approximation.

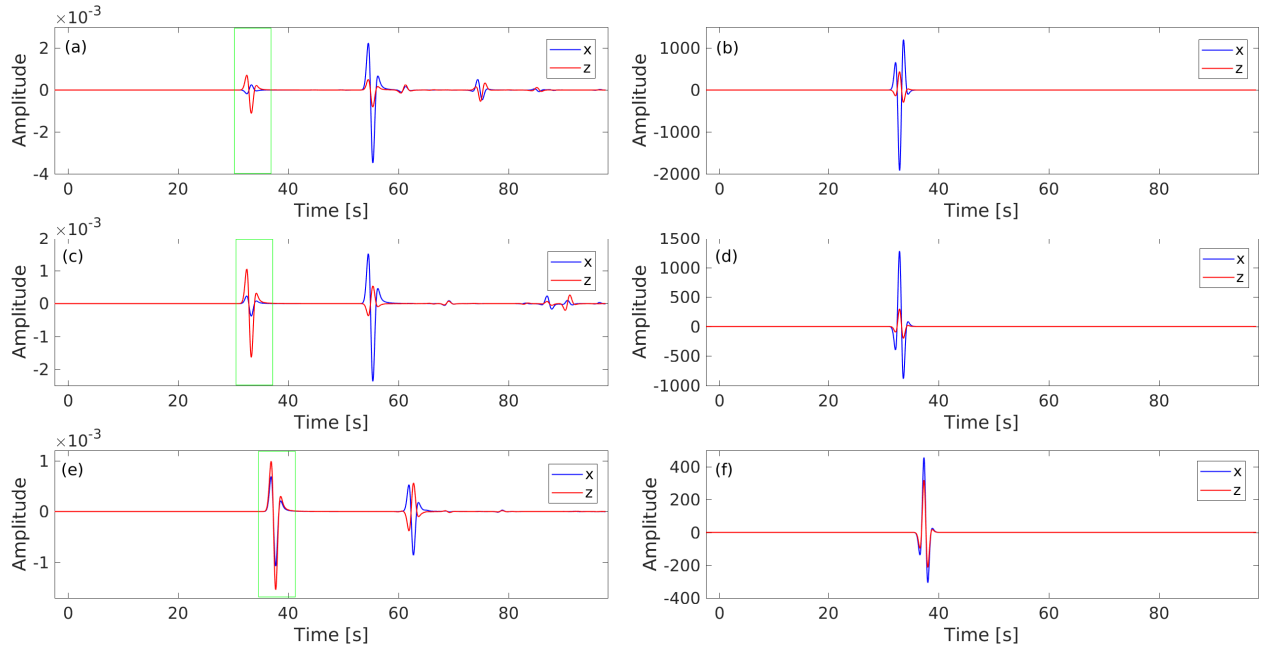

Figure S3. Two-component seismograms registered at the three stations (a,c,e) and their associated adjoint source (b,d,f) computed for the first P wave peak (green rectangles). This example uses the homogeneous model and the traveltime adjoint source.

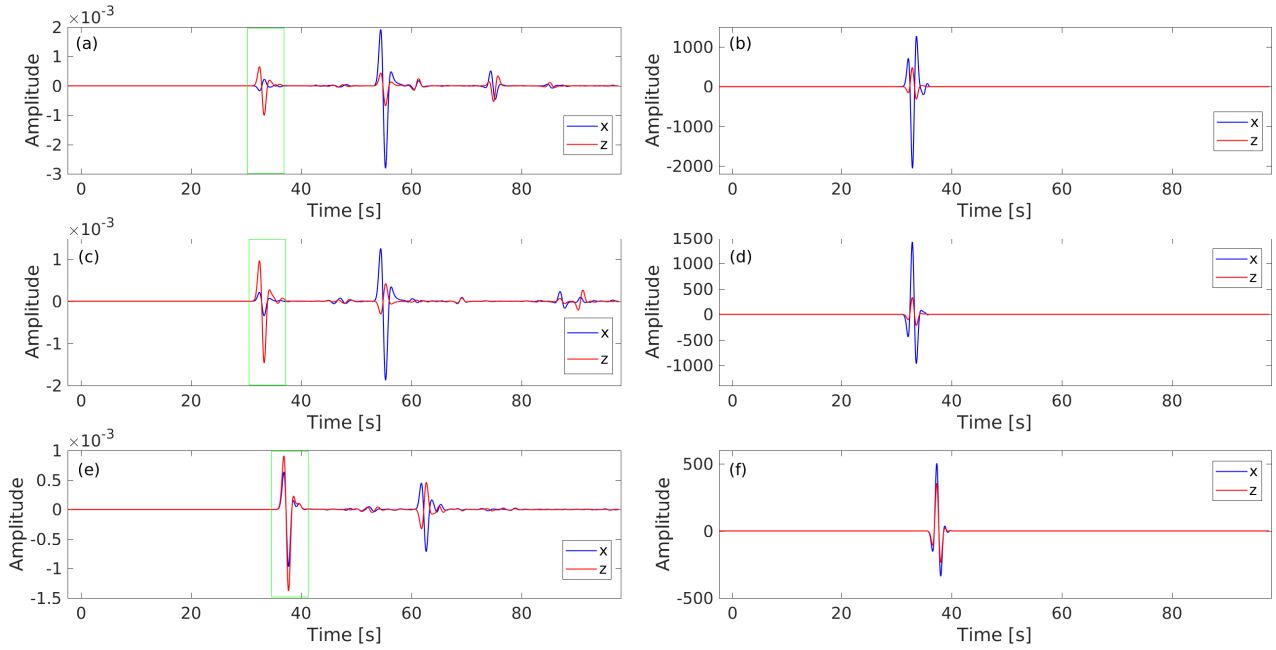

Figure S4. Two-component seismograms registered at the three stations (a,c,e) and their associated adjoint source (b,d,f) computed for the first P wave peak (green rectangles). This example uses the perturbed model with three scatters and the traveltime adjoint source.

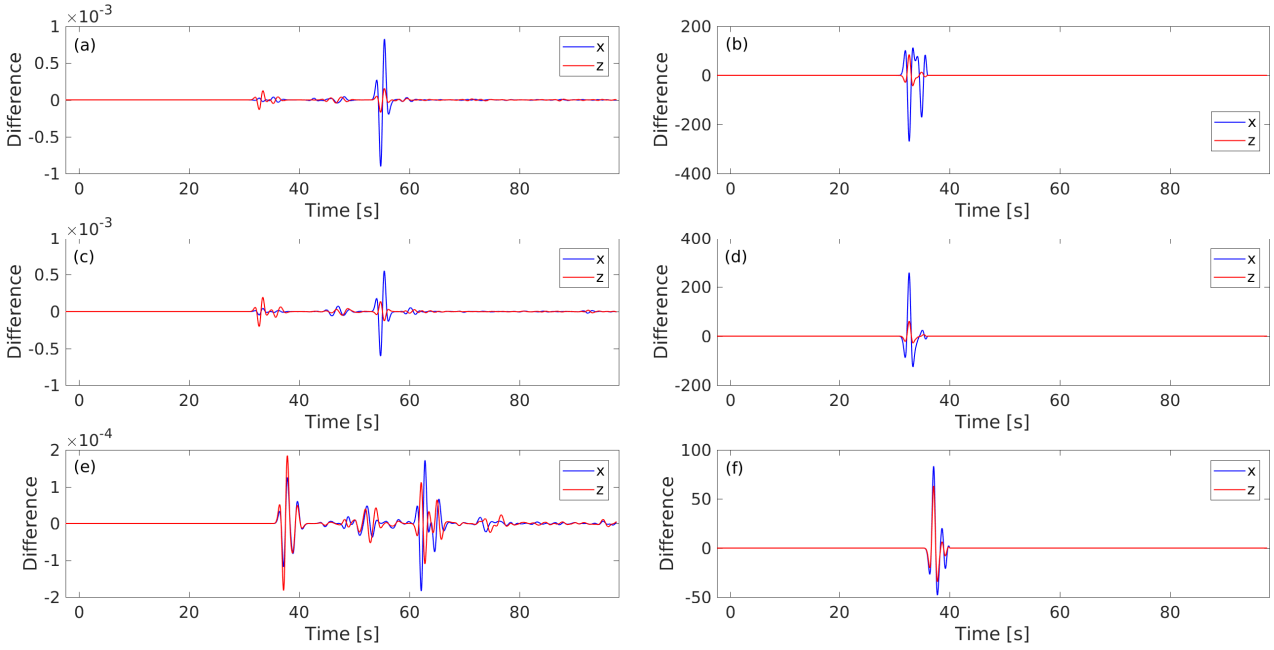

Figure S5. The differences between Figure S4 and Figure S3 (i.e., Figure S4 - Figure S3), which is designed to see the differences in terms of seismograms and adjoint sources due to the perturbation of the model.

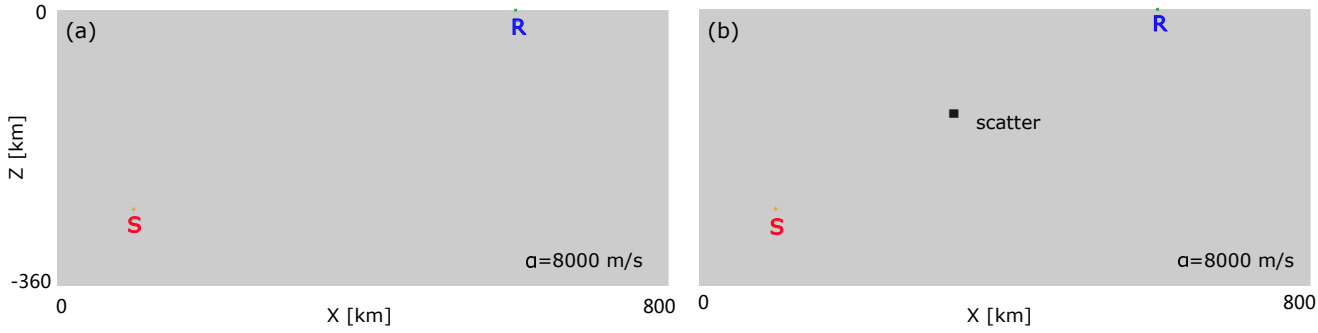

Figure S6. Homogeneous model (a) and the perturbed model with one scatter (b) for compressional wave speed α, where S indicate the source location and R denotes the receiver location. Relative model perturbation for the scatter is set to +10% for the α and β over the homogeneous model.

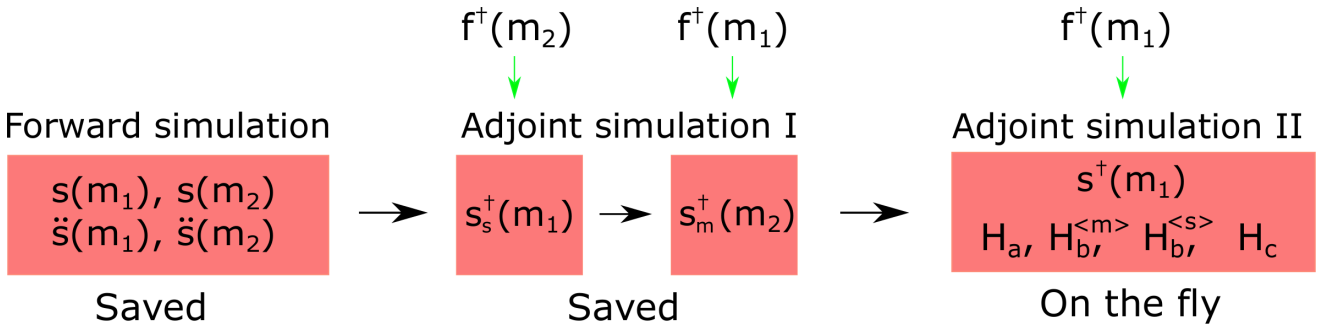

Figure S7. A workflow illuminating the computation of the Hessian kernels by the required forward and adjoint fields. The first step (Forward simulation) is to compute and save the forward fields, the second step (Adjoint simulation I) is to compute and save the two adjoint fields. The last step (Adjoint simulation II) is to compute one adjoint field $\mathbf{s}^\dagger(\mathbf{m}_1)$ on the fly, and read one time step of the saved four or six fields into the temporary memory for the computation of Hessian kernels. The case for the four fields is to compute the Hessian kernels without density perturbation consideration. The $\mathbf{f}^\dagger(\mathbf{m}_1)$ and $\mathbf{f}^\dagger(\mathbf{m}_2)$ denote the two adjoint sources computed from the measurements of the two models, which are used to generate the adjoint fields.

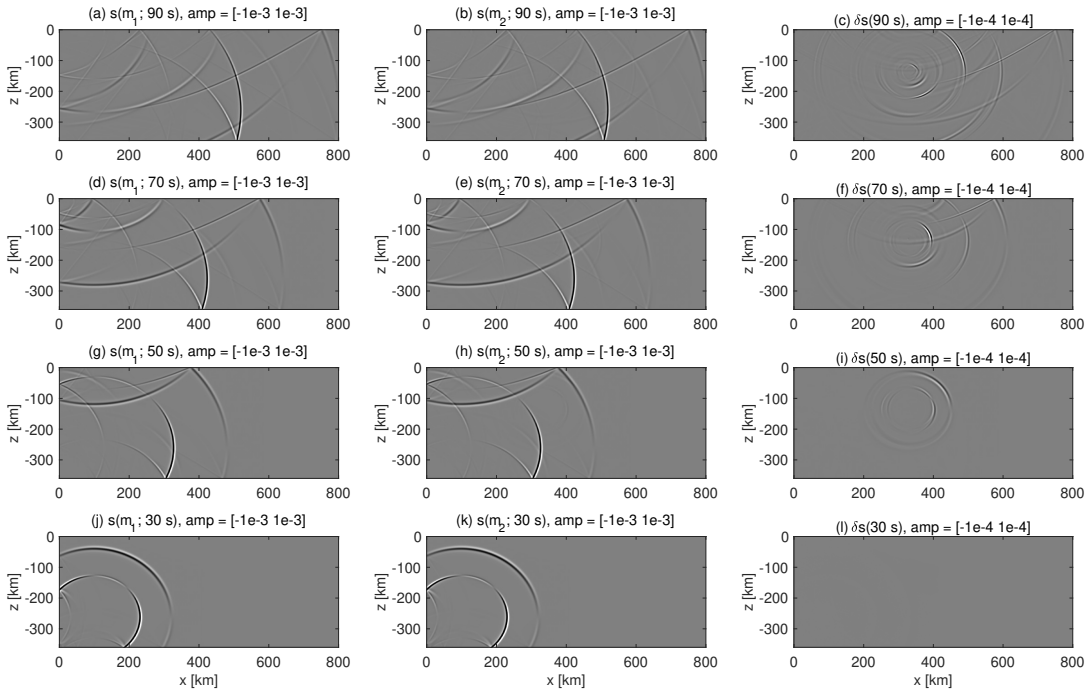

Figure S8. Four time steps of the two forward fields $\mathbf{s}(\mathbf{m}_1)$ and $\mathbf{s}(\mathbf{m}_2)$ and their perturbations $\delta\mathbf{s}$ due to the scatter. The first column shows the forward fields $\mathbf{s}(\mathbf{m}_1)$ for $\mathbf{m}_1$. The second column shows the forward fields $\mathbf{s}(\mathbf{m}_2)$ for $\mathbf{m}_2$. For simplicity, we omit the time dependencies. The perturbed wavefields are computed by using the wavefield subtraction, i.e., $\mathbf{s}(\mathbf{m}_2)$ - $\mathbf{s}(\mathbf{m}_1)$.

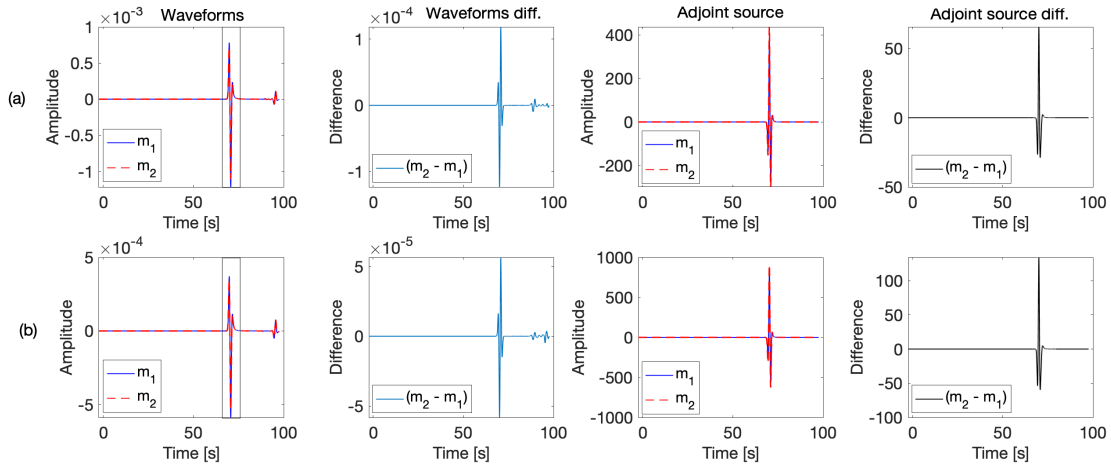

Figure S9. Waveforms and traveltime adjoint sources computed for model $\mathbf{m}_1$ and $\mathbf{m}_2$. Narrow phase-shifted (Ricker) waveforms are observed due to an illumination for the entire time period. The first row (a) shows the x components for the two models. For simplicity, only the P wave (within the time window) is used for computing the adjoint source (see the rectangle window left up). The second row shows the z components for the two models. For the two modes, we also compute the waveform difference (second column) and the adjoint source difference (fourth column) to see the wave difference in magnitude due to the perturbation of the model.

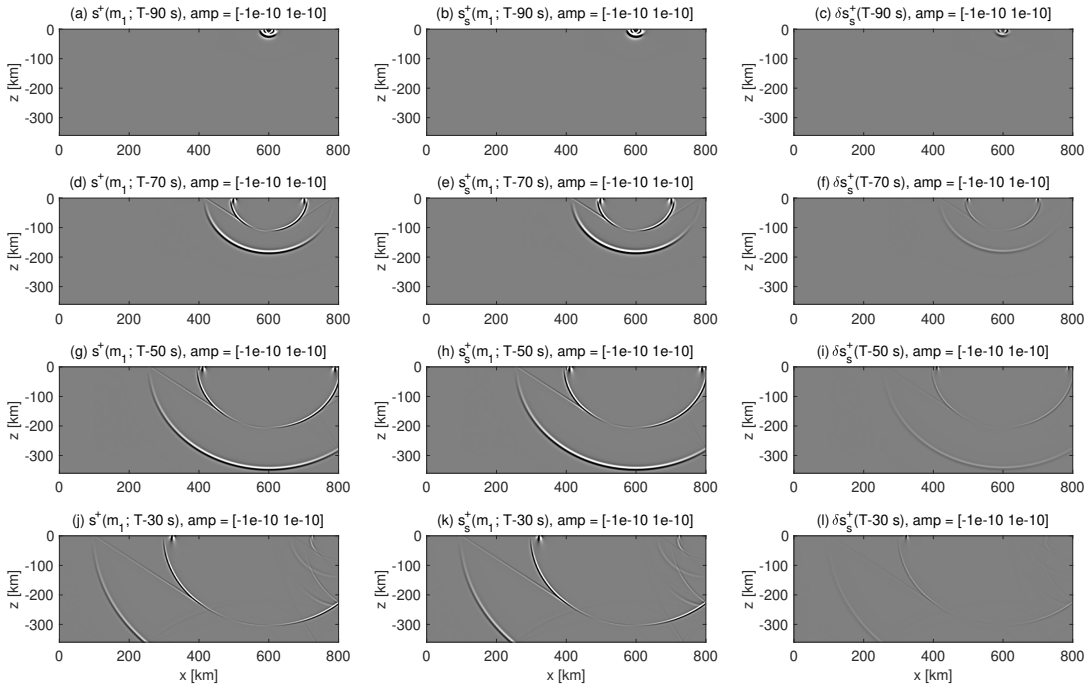

Figure S10. Four time steps of the adjoint fields $\mathbf{s}^\dagger(\mathbf{m}_1)$ and $\mathbf{s}_s^\dagger(\mathbf{m}_1)$ and their perturbations $\delta\mathbf{s}_s^\dagger$. The first column shows the adjoint field $\mathbf{s}^\dagger(\mathbf{m}_1)$ for model $\mathbf{m}_1$. The second column shows the adjoint field $\mathbf{s}_s^\dagger(\mathbf{m}_1)$ for the same model $\mathbf{m}_1$. The third column shows their associated perturbed fields $\delta\mathbf{s}_s^\dagger$ computed by the wavefield subtraction.

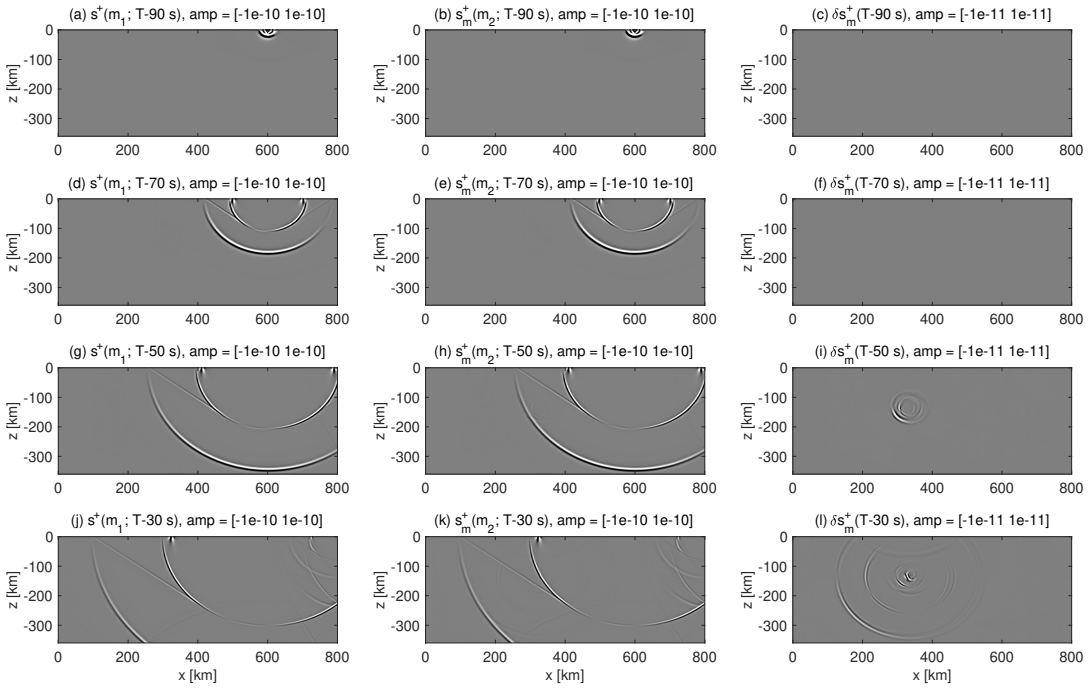

Figure S11. Four time steps of the adjoint fields $\mathbf{s}^\dagger(\mathbf{m}_1)$ and $\mathbf{s}^\dagger_m(\mathbf{m}_2)$ and their perturbations $\delta\mathbf{s}^\dagger_m$. The first column shows the adjoint field $\mathbf{s}^\dagger(\mathbf{m}_1)$ for model $\mathbf{m}_1$. The second column shows the adjoint field $\mathbf{s}^\dagger_m(\mathbf{m}_2)$ for model $\mathbf{m}_2$. The third column shows their perturbed fields $\delta\mathbf{s}^\dagger_m$ computed by the wavefield subtraction.

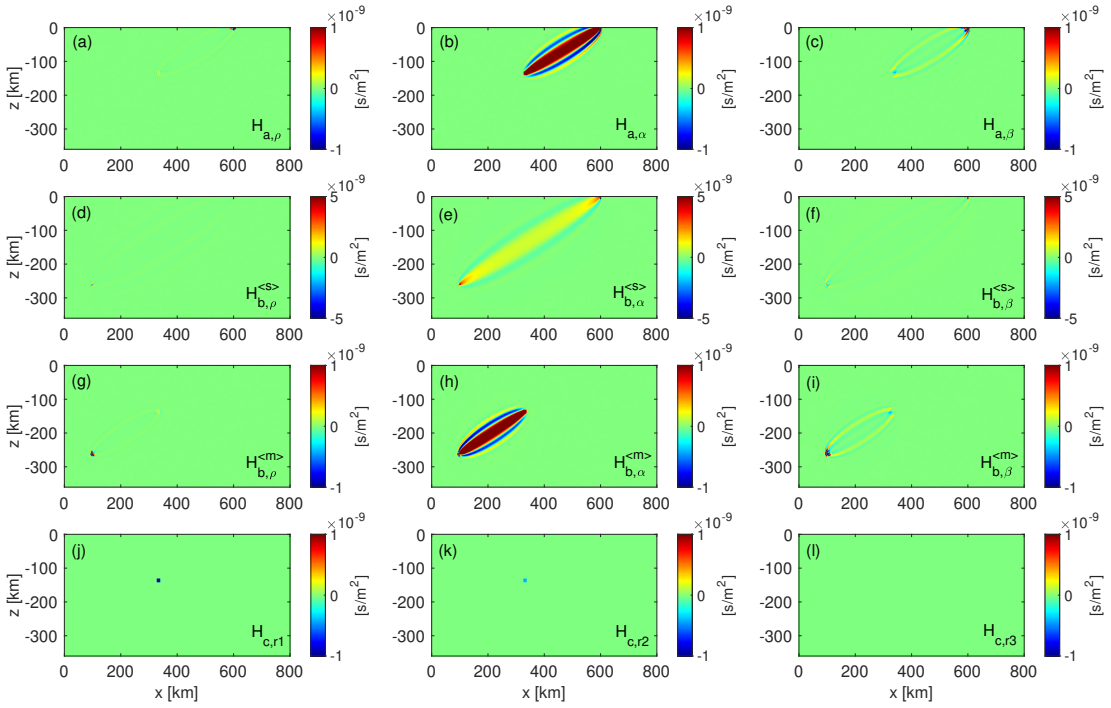

Figure S12. Four components of the Hessian kernels with respect to the model given in ρ, α, and β. The top first row shows the H_a component with respect to the three models parameters. Only the $\mathrm{H}_{a,\alpha}$ is well observed since only the P phase is used for the adjoint source calculation. The second rows shows the $\mathrm{H}_b^{\langle s \rangle}$ component, which is approximate Hessian kernels due to the perturbation of the adjoint source to the adjoint field. The third row shows the $\mathrm{H}_b^{\langle m \rangle}$ component which is due to the perturbation of the model for the adjoint field. The bottom row shows the H_c component. Only the kernels for $\mathrm{H}_{c,r1}$ and $\mathrm{H}_{c,r2}$ are observed since the K_β equals to zero. The ri (where $i = 1, 2, 3$) indicates the three rows in the H_c expression. The full Hessian kernels are obtained by summing the four components together.

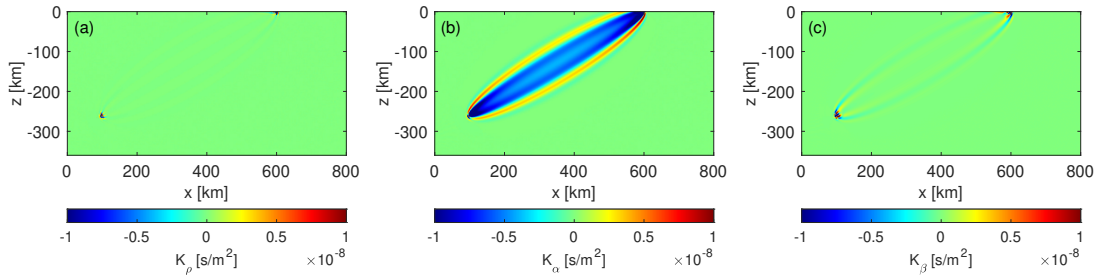

Figure S13. Three components of the Fréchet kernels for the homogeneous model. Only the K_α is well observed since only the P phase is used in the adjoint source calculation. Some artefacts observed near the source and receiver in the K_ρ and K_β components.

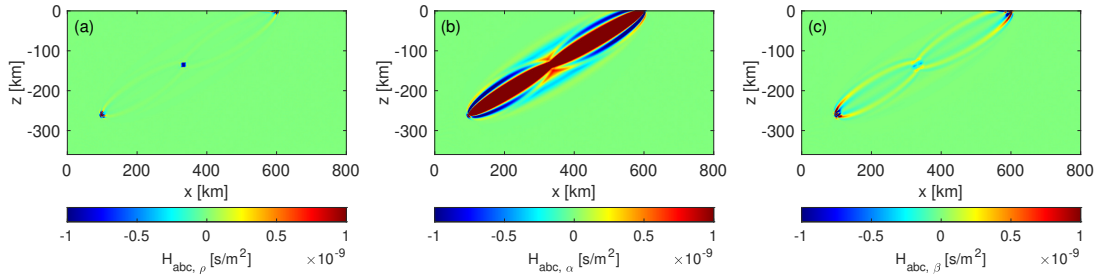

Figure S14. The Hessian kernels with respect to the model parameters ρ, α, and β. The figure is a summation of each row of Figure S12. Significant differences are observed between the full Hessian kernels and the approximate Hessian kernels as well as the Fréchet kernels (see Fig S12 to Fig S14). The different color is due to the minimum and maximum color values set for the kernels.