Bessel smoothing filter for spectral-element mesh

Phuong-Thu Trinh, Romain Brossier, Ludovic Métivier, Jean Virieux, Paul Wellington

Bessel smoothing filter for spectral-element mesh

P.T. Trinh,[1] R. Brossier,[1] L. Métivier,[1,2] J. Virieux[1] and P. Wellington[1]

[1]*Univ. Grenoble Alpes, ISTerre, F-38000 Grenoble, France. E-mail: phuong-thu.trinh@univ-grenoble-alpes.fr*
[2]*Univ. Grenoble Alpes, CNRS, LJK, F-38000 Grenoble, France*

SUMMARY

Smoothing filters are extremely important tools in seismic imaging and inversion, such as for traveltime tomography, migration and waveform inversion. For efficiency, and as they can be used a number of times during inversion, it is important that these filters can easily incorporate prior information on the geological structure of the investigated medium, through variable coherent lengths and orientation. In this study, we promote the use of the Bessel filter to achieve these purposes. Instead of considering the direct application of the filter, we demonstrate that we can rely on the equation associated with its inverse filter, which amounts to the solution of an elliptic partial differential equation. This enhances the efficiency of the filter application, and also its flexibility. We apply this strategy within a spectral-element-based elastic full waveform inversion framework. Taking advantage of this formulation, we apply the Bessel filter by solving the associated partial differential equation directly on the spectral-element mesh through the standard weak formulation. This avoids cumbersome projection operators between the spectral-element mesh and a regular Cartesian grid, or expensive explicit windowed convolution on the finite-element mesh, which is often used for applying smoothing operators. The associated linear system is solved efficiently through a parallel conjugate gradient algorithm, in which the matrix vector product is factorized and highly optimized with vectorized computation. Significant scaling behaviour is obtained when comparing this strategy with the explicit convolution method. The theoretical numerical complexity of this approach increases linearly with the coherent length, whereas a sublinear relationship is observed practically. Numerical illustrations are provided here for schematic examples, and for a more realistic elastic full waveform inversion gradient smoothing on the SEAM II benchmark model. These examples illustrate well the efficiency and flexibility of the approach proposed.

Key words: Image processing; Inverse theory; Numerical approximations and analysis; Controlled source seismology; Seismic tomography.

1 INTRODUCTION

Full waveform inversion (FWI) offers the possibility to extract high-resolution quantitative multi-parameters of the subsurface from seismic data. The majority of these applications are carried out under finite-difference (FD) approximation, due to the numerical efficiency of this method and its ease of implementation. Standard formulations of this approach are, however, limited on regular grids, which require significant extra effort in terms of design and computational cost in the presence of surface topography or important geological interfaces (Robertsson 1996; Bohlen & Saenger 2006; Fuji *et al.* 2016; Huiskes *et al.* 2016). Finite-element (FE) methods have become popular for regional and global problems, especially spectral-element methods (SEMs), where complex geometry can be handled with accurate numerical calculation of wavefields (Komatitsch & Tromp 1999). In dealing with 3-D elastic FWI, we would like to develop a complete inversion numerical workflow using a spectral FE scheme while including an efficient smoothing filter implemented directly on the FE mesh.

In most geophysical applications, FWI is introduced as an iterative local optimization problem that attempts to minimize the least-squares residuals between the observed and the calculated data at the receiver location. This inverse problem is mathematically ill-posed, which thus leads to the non-uniqueness of the solution. For seismic imaging using either traveltimes or wavefields, the inversion often needs to be stabilized by applying regularization. This can be performed through model-driven regularization (Menke 1984; Tarantola 2005) or by preconditioning the gradient through filtering operators (Guitton *et al.* 2012). This data-driven strategy is sometimes called model preconditioning (Fomel & Claerbout 2003). These regularization techniques normally assume the particular properties or structure of the model, such as smoothness or

geological features, to guide the problem toward the desired solution. In 3-D elastic FWI, the computational cost remains one of the main challenges (Virieux & Operto 2009), therefore decimation of the data volume by reducing the number of computed shots, random subsampling strategies, and/or source-encoding becomes mandatory (Capdeville *et al.* 2005; Herrmann *et al.* 2009; Krebs *et al.* 2009; Warner *et al.* 2013; Castellanos *et al.* 2015). This implies a decrease in the data coverage and the presence of cross-talk and artefacts, which makes the inversion problem even more ill-posed. In this context, the role of regularization at each iteration becomes crucial and needs to be performed efficiently.

Dealing with complex structures or geological heterogeneity might also require adequate regularization strategies. These issues can be addressed by attenuation of the wave-number content in a particular direction, such as plane-wave destruction filters (Claerbout 1992; Fomel 2002), or by imposition of expected structures through directional Laplacian preconditioning in the model space (Hale 2007; Guitton *et al.* 2012). This latter highlights the importance of non-stationary filters in complex geology; for example, the Laplace filter can smooth or drastically reduce local planar events according to a local dip field.

These non-stationary preconditioning filters are expressed as convolution operators, such as Gaussian or Laplacian filters, through

$$\mathbf{s}(\mathbf{x}) = \int C_{nD}(\mathbf{x} - \mathbf{x}')\mathbf{m}(\mathbf{x}')d\mathbf{x}'. \tag{1}$$

The vector $\mathbf{m}(\mathbf{x})$ is transformed into a new vector $\mathbf{s}(\mathbf{x})$ through the convolution operator C_{nD}, where n is the dimension of the problem ($n = 2$ or $n = 3$). In the FD grid, the Gaussian filter (without local rotation) can be applied efficiently to any model or gradient vector, due to the tensorial property of the function. For long filters, recursive implementation can be used to improve the computing performance (Deriche 1992; Van Vliet *et al.* 1998). These approaches can be extended to FE methods by including a projection between the Cartesian mesh, where the parameters to be reconstructed are located, and the FE mesh. However, this can be limited by the accuracy of the back and forth projections. Filtering can also be applied as a windowed convolution of the filter and the vector (Tape *et al.* 2010; Peter *et al.* 2011), as for the SPECFEM open-source package

$$\mathbf{s}(\mathbf{x}) \approx \int_{-\alpha L}^{\alpha L} C_{nD}(\mathbf{x} - \mathbf{x}')\mathbf{m}(\mathbf{x}')d\mathbf{x}', \tag{2}$$

where L is the coherent length associated with the filter: as the kernel of the filter is decaying, the integration can be limited over the finite domain expressed by the effective radius αL from the position $\mathbf{x}$. This convolution approach can be relatively computer demanding, because for each input point to be filtered, contributions of other points of the medium are required in the surrounding volume, which leads to significant computer manipulation, especially for functions with long tails, such as the Laplace filter.

In this report, we highlight that the inverse of given operators C_{nD} can be relatively sparse. Therefore, it is interesting to consider them to obtain the contribution of the smoothing operator (Wellington 2016). Instead of performing the convolution in eq. (1), we can consider solving the following eq. (3), relying on the inverse operator

$$\int C_{nD}^{-1}(\mathbf{x} - \mathbf{x}')\mathbf{s}(\mathbf{x}')d\mathbf{x}' = \mathbf{m}(\mathbf{x}), \tag{3}$$

which requires knowledge of the inverse operator C_{nD}^{-1}. This leads to the definition of Bessel filters, which are defined by the modified Bessel functions (Abramowitz & Stegun 1972) and their sparse inverse operators. As the filter is defined through a non-homogeneous elliptic partial differential equation (PDE) with delta source function, its inverse operator can be expressed as a distribution function. The integral eq. (3) can then be efficiently solved using any FD or FE method. In 3-D, applying the Bessel filter twice provides an excellent approximation of the Laplace operator, with negligible mismatch at the origin. This approximation implies that the Bessel filter can be applied either once as a smoothing filter, or twice to reproduce the decay of the Laplace filter.

This paper is organized as follows. In Section 2, we first define the Bessel filter through a PDE with specific boundary conditions, and the approximation of the Laplace filter by Bessel functions in 2-D and 3-D geometries. In Section 3, the application of the Bessel sparse inverse operator on a vector is described by a system of PDEs. The weak formulation of these equations yields a sparse linear system, which is symmetric positive even for variable coherent lengths, dip and azimuth angles. In Section 4, several numerical illustrations and an example from the synthetic 3-D SEAM Phase II foothills model are provided of such smoothing processes. In Section 5, efficient numerical schemes in a high-performance computer environment are presented. We show how a parallel conjugate gradient (CG) iterative solver can be implemented in a matrix-free fashion. The convergence of this CG iterative solver is analysed with respect to several parameters. Significant scaling behaviour is also obtained when this strategy is compared with the explicit convolution method. Conclusions and perspectives are given in Section 6. We highlight that the choice of applying the Bessel filter once or twice depends on the specific application, and this decision will be case dependent.

2 METHODOLOGY

Before moving to the mathematical development, we would like to specify some definitions regarding forward and inverse filters. Let us consider the kernel $C(\mathbf{x})$ as a 2-D or 3-D smoothing filter. The application of this filter to a function $g(\mathbf{x})$ is defined by the convolution operator, which is denoted by the symbol '*', through

$$f(\mathbf{x}) = C(\mathbf{x}) * g(\mathbf{x}), \quad \text{which is equivalent to} \quad f(\mathbf{x}) = \int C(\mathbf{x} - \mathbf{x}')g(\mathbf{x}')d\mathbf{x}'. \tag{4}$$

The inverse filter of $C(\mathbf{x})$, namely $C(\mathbf{x})^{-1}$, is defined through the relation

$$C^{-1}(\mathbf{x}) * C(\mathbf{x}) = \delta(\mathbf{x}). \tag{5}$$

2.1 Definition of Bessel filters and their sparse inverse operators

Depending on the space dimension, we introduce the normalized Bessel filters

$$B_{3D}(z, x, y) = \frac{1}{(2\pi)^{3/2} L_z L_x L_y} r^{-1/2} K_{1/2}(r) \quad \text{where} \quad r = \sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2} + \frac{y^2}{L_y^2}}, \tag{6}$$

$$B_{2D}(z, x) = \frac{1}{2\pi L_z L_x} K_0 \left(\sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2}} \right), \tag{7}$$

where L_x, L_y and L_z are coherent lengths in the x, y and z directions, and K_ν is the modified Bessel function of the second kind, as presented in Appendix A. While the 2-D Bessel filter is the normalized modified Bessel function $K_\nu(r)$ with $\nu = 0$, the 3-D Bessel filter is the modified spherical Bessel function $z^{-1/2} K_{\nu + 1/2}(r)$ with $\nu = 0$ (Abramowitz & Stegun 1972). When coherent lengths are uniform over space, these Bessel filters are unique solutions of the following PDEs

$$B_{3D}(z, x, y) - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} + L_y^2 \frac{\partial^2}{\partial y^2} \right) B_{3D}(z, x, y) = \delta(z, x, y), \tag{8}$$

$$B_{2D}(z, x) - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} \right) B_{2D}(z, x) = \delta(z, x), \tag{9}$$

with radiative boundary conditions $B_{nD}(\mathbf{x}) \to 0$ as $\|\mathbf{x}\| \to \infty$, and the additional bounded constraint on the integral through $0 < \int B_{nD}(\mathbf{x}) d\mathbf{x} < \infty$. The construction of these normalized Bessel filters is developed in Appendix B.

According to eqs (8) and (9) and the definition of the inverse filter in eq. (5), the associated sparse inverse of the Bessel filters can be defined as

$$B_{3D}^{-1}(z, x, y) = \delta^0(z, x, y) - \left(L_z^2 \delta^2(z) \delta^0(x) \delta^0(y) + L_x^2 \delta^2(x) \delta^0(y) \delta^0(z) + L_y^2 \delta^2(y) \delta^0(x) \delta^0(z) \right) \tag{10}$$

$$B_{2D}^{-1}(z, x) = \delta^0(z, x) - \left(L_z^2 \delta^2(z) \delta^0(x) + L_x^2 \delta^2(x) \delta^0(z) \right), \tag{11}$$

where δ^n is defined as

$$\int \delta^n(x - x') f(x') dx' = (-1)^n \frac{d^n f(x)}{dx^n}. \tag{12}$$

It is important to note that Bessel filters only depend on the radial part (Appendix B), which favours the transformation from Cartesian coordinates to polar coordinates (r, θ) in 2-D, or to spherical coordinates (r, θ, ϕ) in 3-D

$$(z, x) \underset{2D}{\to} \sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2}}; \quad (z, x, y) \underset{3D}{\to} \sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2} + \frac{y^2}{L_y^2}}. \tag{13}$$

Let us consider an original vector $\mathbf{m}$ in the model space Ω that we want to smooth. The vector $\mathbf{s}$ will be the smoothed vector obtained by applying the Bessel filter $B_{3D}(z, x, y)$. We can equivalently consider the application of the inverse Bessel filter, which leads to the choice between the solutions of the following two problems:

$$\mathbf{s} = B_{3D} * \mathbf{m} \quad \text{or} \quad \mathbf{m} = B_{3D}^{-1} * \mathbf{s}. \tag{14}$$

Both systems are convolution over space. The first one embeds a relatively broad kernel shape, where the computation can be demanding. Therefore, we are interested in the second equation. According to the definition of the inverse operator B_{3D}^{-1} (eq. 10), this convolution can be translated into the following PDE over the domain Ω

$$\mathbf{s}(z, x, y) - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} + L_y^2 \frac{\partial^2}{\partial y^2} \right) \mathbf{s}(z, x, y) = \mathbf{m}(z, x, y), \tag{15}$$

in which the vector $\mathbf{m}$ appears on the right-hand side. Once discretized by any FD or FE method, this system leads to a sparse operator and can be solved efficiently, to obtain the smoothed vector $\mathbf{s}$. Our approach is similar in some ways to the structure-oriented smoothing filter proposed by Williamson *et al.* (2011), which was derived from the diffusion equation as proposed by Fehmers & Höcker (2003). This smoothing process is controlled by a symmetric diffusion tensor, without knowing the kernel shape of the forward filter.
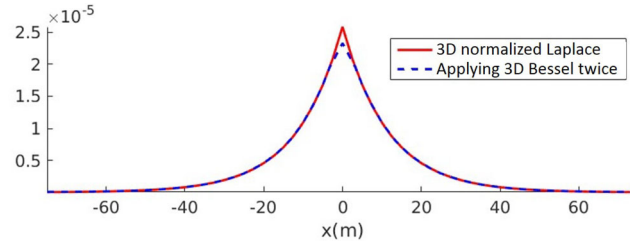
Figure 1. Comparison between the behaviour of the 3-D normalized Laplace filter defined in eq. (20) and the convolution of two 3-D Bessel filters for $L_x = L_y = L_z = 20$ m, obtained from the iterative solving of eq. (8), which defines the 3-D Bessel filter.

2.2 Approximation of the Laplace filter by the Bessel operator

The convolution of two Bessel filters can provide an excellent approximation of the Laplace filter in 2-D and 3-D. We first illustrate this for the 3-D case, as this is our main interest. The double operator $B_{3D}^{-1} * B_{3D}^{-1}$ is the inverse filter of the operator $B_{3D} * B_{3D}$, which is the solution of the following equation

$$\left[B_{3D}^{-1} * B_{3D}^{-1}(z, x, y) \right] * f(z, x, y) = \delta(z, x, y), \tag{16}$$

where the function f satisfies the following PDE

$$\left[1 - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} + L_y^2 \frac{\partial^2}{\partial y^2} \right) \right]^2 f(z, x, y) = \delta(z, x, y). \tag{17}$$

Changing parameters from Cartesian coordinates to spherical coordinates, where

$$r = \sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2} + \frac{y^2}{L_y^2}}, \tag{18}$$

and knowing that the expected solution, the filter $B_{3D} * B_{3D}$, only depends on the radial part, eq. (17) now becomes

$$\left[1 - \frac{\partial^2}{\partial r^2} \right] f(r) = \frac{\delta(r)}{4\pi r^2} + \underbrace{\left[\frac{\partial^2}{\partial r^2} + \frac{4}{r} \frac{\partial}{\partial r} - \frac{\partial^4}{\partial r^4} - \frac{4}{r} \frac{\partial^3}{\partial r^3} - \frac{4}{r^2} \frac{\partial^2}{\partial r^2} \right] f(r)}_{M_{3D}[f(r)]}. \tag{19}$$

The normalized 3-D Laplace filter $L_{3D}(z, x, y)$ is given by

$$L_{3D}(z, x, y) = \frac{1}{8\pi L_z L_x L_y} e^{-\sqrt{\frac{z^2}{L_z^2} + \frac{x^2}{L_x^2} + \frac{y^2}{L_y^2}}} \quad \text{or} \quad L_{3D}(r) = \frac{1}{8\pi L_z L_x L_y} e^{-r}. \tag{20}$$

This is the solution of eq. (19), for $r > 0$, provided the residual term $M_{3D}[L_{3D}(r)]$ can be ignored, as

$$L_{3D}(r) - \frac{\partial^2}{\partial r^2} L_{3D}(r) = 0. \tag{21}$$

The residual term M_{3D} for the 3-D Laplace filter satisfies

$$\left\| \frac{M_{3D}[L_{3D}(r)]}{L_{3D}} \right\| = \frac{4}{r^2}, \tag{22}$$

which implies that the residual term becomes smaller when the distance r increases. Fig. 1 presents an excellent match between the 3-D normalized Laplace filter defined in eq. (20) and the convolution of the two 3-D Bessel filters for $L_x = L_y = L_z = 20$ m, obtained from solving the PDE defining the 3-D Bessel filter. The mismatch at zero origin comes from the residual terms $M_{3D}[L_{3D}(r)]$, which are more important at small distances r. This analysis is also coherent with the 3-D inverse operator of the Laplace filter proposed by Tarantola (2005).

In 2-D, by applying the same workflow, we can find a similar approximation. However, we would like to highlight the flexibility of this approximation by introducing a scaling parameter a into the definition of the 2-D normalized Laplace filter

$$L_{2D}(z, x, a) = \frac{1}{2\pi a^2 L_x L_z} e^{-\sqrt{\frac{z^2}{a^2 L_z^2} + \frac{x^2}{a^2 L_x^2}}} \quad \text{or} \quad L_{2D}(r) = \frac{1}{2\pi a^2 L_x L_z} e^{-r}. \tag{23}$$

This 2-D Laplace operator can be well approximated by the application of two 2-D Bessel filters, as shown in Fig. 2, except at the origin where there is a singularity. By using different values of the scaling factor a, we have different approximations of the Laplace filter, which mitigates the discrepancies for both the amplitude and the decay.

3 WEAK FORMULATION IN FE METHODS

Following the weak formulation, as usual for FE methods, we can discretize eq. (15) and deduce a sparse linear system that can be efficiently solved using an iterative linear solver. First, we start by introduction of the weak formulation in SEM for homogeneous coherent lengths
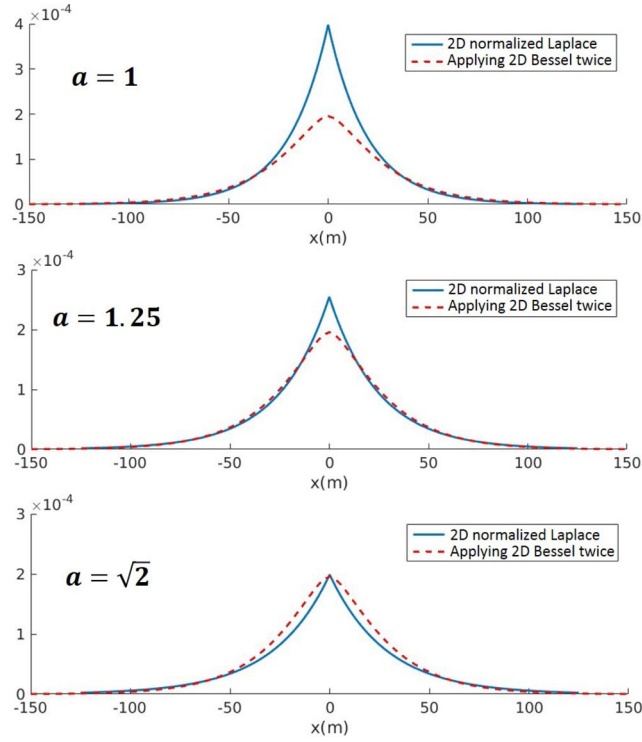
Figure 2. Comparison between the 2-D normalized Laplace filter defined in eq. (23) and the application of two 2-D Bessel filters for $L_x = L_z = 20$ m at different values of the scaling parameter a. The Bessel filters are obtained from the iterative solving of eq. (9), which defines the 2-D Bessel filter.

without dip and azimuth, which naturally leads to a symmetric linear system. We then introduce an efficient method to incorporate variable coherent lengths, and dip and azimuth angles into our linear system, while preserving the symmetry of the left-hand-side matrix. Although the following was developed for SEM, it can be extended to other FE formulations, and also for FD methods (Wellington 2016).

3.1 Weak formulation of the filter for homogeneous coherent lengths without dip and azimuth

We do not develop the weak formulation of eq. (15) in Cartesian space, but in the dimensionless coordinates in the domain $\tilde{\Omega}$, as defined by the following expressions

$$\tilde{z} = \frac{z}{L_z}, \quad \tilde{x} = \frac{x}{L_x}, \quad \tilde{y} = \frac{y}{L_y}. \tag{24}$$

As coherent lengths L_z, L_x and L_y are homogeneous, the relationship of eq. (24) is an one-to-one projection from (z, x, y) to $(\tilde{z}, \tilde{x}, \tilde{y})$, which implies that solving eq. (15) in dimensionless or Cartesian coordinates should provide identical results. In the dimensionless coordinates, eq. (15) can be written in its weak form as

$$\iiint_{\tilde{\Omega}} \left[\mathbf{s} - \nabla^2_{\tilde{z}, \tilde{x}, \tilde{y}} \mathbf{s} \right] v \, \mathrm{d}\tilde{z}\mathrm{d}\tilde{x}\mathrm{d}\tilde{y} = \iiint_{\tilde{\Omega}} \mathbf{m} v \, \mathrm{d}\tilde{z}\mathrm{d}\tilde{x}\mathrm{d}\tilde{y}, \tag{25}$$

leading to the weak formulation

$$\iiint_{\tilde{\Omega}} \mathbf{s} v \, \mathrm{d}\tilde{z}\mathrm{d}\tilde{x}\mathrm{d}\tilde{y} + \iiint_{\tilde{\Omega}} \nabla_{\tilde{z}, \tilde{x}, \tilde{y}} \mathbf{s} \nabla_{\tilde{z}, \tilde{x}, \tilde{y}} v \, \mathrm{d}\tilde{z}\mathrm{d}\tilde{x}\mathrm{d}\tilde{y} = \iiint_{\tilde{\Omega}} \mathbf{m} v \, \mathrm{d}\tilde{z}\mathrm{d}\tilde{x}\mathrm{d}\tilde{y}, \tag{26}$$

where the test function $v(\tilde{z}, \tilde{x}, \tilde{y})$ is chosen such that it satisfies the homogeneous Dirichlet boundary conditions, which implies that the boundary terms have vanished.

We introduce the notation of the division of two vectors $(x_1, x_2, x_3)^t$ and $(y_1, y_2, y_3)^t$ (where 't' stands for the transpose operator)

$$\mathbf{B} = \frac{(y_1, y_2, y_3)^t}{(x_1, x_2, x_3)^t}, \tag{27}$$

such that $\mathbf{B}$ satisfies the relationship

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathbf{B} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{28}$$

This matrix **B** is actually the transformation matrix when the coordinate transformation is performed from the system (x_1, x_2, x_3) to (y_1, y_2, y_3).

In SEM, the physical domain is decomposed into a set of non-overlapping hexahedral elements. Each element is further discretized by $(N + 1)^3$ Gauss–Lobatto–Legendre (GLL) points $(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3})$ in the reference space, $k_1, k_2, k_3 = 0, \dots, N$. The mapping from the reference space (ξ, η, ζ) to the Cartesian space (z, x, y) is expressed by the Jacobi matrix, according to the notation in eq. (27),

$$\mathbf{J} = \frac{(\partial z, \partial x, \partial y)^t}{(\partial \xi, \partial \eta, \partial \zeta)^t} \quad \text{and} \quad J_e = \det(\mathbf{J}). \tag{29}$$

Similarly, the relationship between the reference space and the dimensionless coordinate is defined through

$$\tilde{\mathbf{J}} = \frac{(\partial \tilde{z}, \partial \tilde{x}, \partial \tilde{y})^t}{(\partial \xi, \partial \eta, \partial \zeta)^t} \quad \text{and} \quad \tilde{J}_e = \det(\tilde{\mathbf{J}}). \tag{30}$$

In the reference space, the basis functions are defined as Lagrange polynomials over the GLL points. The integrals are then numerically approximated by GLL quadrature (more details on SEM are given in Appendix C). Due to these ingredients and the property of the Lagrange polynomials that have values at the GLL nodes of either 0 or 1, the weak formulation in eq. (26) can be written as

$$\underbrace{(\mathbf{M} + \mathbf{K})}_{\mathbf{A}}\mathbf{s} = \mathbf{Mm}, \tag{31}$$

in which the mass matrix **M** is diagonal, and the stiffness matrix **K** is symmetric. The impedance matrix $\mathbf{A} = \mathbf{M} + \mathbf{K}$ is then symmetric. For the mass matrix, its components are given by

$$M_{\hat{k}\hat{k}} = w_{k_1} w_{k_2} w_{k_3} \tilde{J}_e(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3}) \quad \hat{k} \text{ stands for the triple of indexes } \{k_1, k_2, k_3\}, \tag{32}$$

and, for the stiffness matrix, by

$$K_{\hat{k}\hat{h}} = \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \tilde{J}_e(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}) \left[\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} \left(\sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \right) \frac{\partial v_{\hat{h}}}{\partial r_j} \right], \tag{33}$$

where

$$(p_1, p_2, p_3) := (\tilde{z}, \tilde{x}, \tilde{y}) \quad \text{and} \quad (r_1, r_2, r_3) := (\xi, \eta, \zeta).$$

We introduce here the geometric factors G_{ij} associated with the projection between the dimensionless coordinates and the reference coordinates

$$G_{ij}(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}) = \sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \tilde{J}_e(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}), \tag{34}$$

which simplifies the expression of the stiffness matrix to

$$K_{\hat{k}\hat{h}} = \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \left[\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} G_{ij} \frac{\partial v_{\hat{h}}}{\partial r_j} \right]. \tag{35}$$

These expressions of eqs (32), (34) and (35) are used for the implementation of the matrices **M** and to compute the product of the matrix **A** with a given vector in the dimensionless coordinates. To do so, the evaluation of the geometric factors G_{ij} is critical. Note that the linear system of eq. (31) is constructed and solved in the dimensionless coordinates and not in the Cartesian coordinates. The results are not affected by this coordinate transformation, due to the bijective correspondence between the two coordinates systems (eq. 24).

When wave-propagation simulation is performed by SEM, all of the elements of the inverse Jacobi matrix $\mathbf{J}^{-1}$ and the volumetric Jacobian J_e associated to this projection are available at no extra cost. It is of great interest to incorporate these ingredients into the construction of matrices **M** and geometric factors G_{ij}. According to the definition of the relationship between dimensionless coordinates and Cartesian coordinates (eq. 24), the determinant $\tilde{J}_e$ of the Jacobi matrix $\tilde{\mathbf{J}}$ can be estimated from the volumetric Jacobian J_e of the Jacobi matrix **J** through

$$\tilde{J}_e(\xi, \eta, \zeta) = \left| \frac{(\partial \tilde{z}, \partial \tilde{x}, \partial \tilde{y})^t}{(\partial \xi, \partial \eta, \partial \zeta)^t} \right| = \frac{1}{L_z L_x L_y} \left| \frac{(\partial z, \partial x, \partial y)^t}{(\partial \xi, \partial \eta, \partial \zeta)^t} \right| = \frac{1}{L_z L_x L_y} J_e(\xi, \eta, \zeta). \tag{36}$$

The inverse matrix $\tilde{\mathbf{J}}^{-1}$ can be deduced from the elements of the inverse Jacobi matrix $\mathbf{J}^{-1}$ through the identity

$$\begin{bmatrix} \partial_{\tilde{z}}\xi & \partial_{\tilde{x}}\xi & \partial_{\tilde{y}}\xi \\ \partial_{\tilde{z}}\eta & \partial_{\tilde{x}}\eta & \partial_{\tilde{y}}\eta \\ \partial_{\tilde{z}}\zeta & \partial_{\tilde{x}}\zeta & \partial_{\tilde{y}}\zeta \end{bmatrix} = \begin{bmatrix} L_z \partial_z\xi & L_x \partial_x\xi & L_y \partial_y\xi \\ L_z \partial_z\eta & L_x \partial_x\eta & L_y \partial_y\eta \\ L_z \partial_z\zeta & L_x \partial_x\zeta & L_y \partial_y\zeta \end{bmatrix}. \tag{37}$$

Following this framework, the application of the Bessel filter to a model vector, through the solving of the linear system (eq. 31), can be efficiently achieved in the dimensionless coordinate system, whereas the wave-propagation simulation is still performed in Cartesian coordinates.

The symmetric matrix $\mathbf{A}$ is referred to as the discrete Helmholtz operator, which has been shown to be positive definite (Deville *et al.* 2002). The CG method is thus the method of choice to iteratively solve the linear system. In these numerical experiments, the condition number of the matrix $\mathbf{A}$ ranges from 10^2 to 10^4, which depends on the value of the coherent lengths. Therefore, the iterative solver converges rapidly toward the desired solution, as we will see with the numerical implementation.

3.2 Variable coherent lengths, dip and azimuth angles

In the previous section, the weak formulation of the Bessel filter in a dimensionless coordinate system was shown to naturally yield a symmetric stiffness matrix. When introducing variable parameters—coherent lengths, dip and azimuth angle—we want to preserve the symmetry of the matrix $\mathbf{A}$, and therefore we develop the weak formulation in a rotated dimensionless coordinate system $(\tilde{v}, \tilde{u}, \tilde{w})$. Before defining these coordinates, let us provide the definition of the azimuth θ and dip φ angles for a given vector. The azimuth is the horizontal angle measured from the North, and the dip is the angle that the vector has with the horizontal (Sheriff 2002). Their values range such that $\theta \in [-\pi, \pi]$ and $\varphi \in [-\pi/2, \pi/2]$. In 3-D space, a rotation R_{3D} with azimuth θ and dip φ transforms the Cartesian coordinates (z, x, y) into the rotated coordinates (v, u, w), where v is vertical direction, perpendicular to the bedding planes, and the two horizontal directions u and w define the plane of the geological structure. Considering the previous definitions, we have the following rotational operator

$$R_{3D}(\theta, \varphi) = \frac{(v, u, w)^t}{(z, x, y)^t} = \begin{bmatrix} \cos\varphi & -\cos\theta\sin\varphi & \sin\theta\sin\varphi \\ \sin\varphi & \cos\theta\cos\varphi & -\sin\theta\cos\varphi \\ 0 & \sin\theta & \cos\theta \end{bmatrix}. \tag{38}$$

Similar to the last section, we would like to develop the weak formulation of the PDEs associated with the application of the sparse inverse Bessel filter in the rotated dimensionless coordinates system $(\tilde{v}, \tilde{u}, \tilde{w})$, defined by the following relationships

$$\tilde{v} = \frac{v}{L_v(v, u, w)}; \qquad \tilde{u} = \frac{u}{L_u(v, u, w)}; \qquad \tilde{w} = \frac{v}{L_w(v, u, w)}, \tag{39}$$

in which L_v, L_u and L_w are coherent lengths in the v, u and w directions, respectively. In this rotated dimensionless coordinate system, the Bessel filter is defined by the PDE

$$B(\tilde{v}, \tilde{u}, \tilde{w}) - \left(\frac{\partial^2}{\partial \tilde{v}^2} + \frac{\partial^2}{\partial \tilde{u}^2} + \frac{\partial^2}{\partial \tilde{w}^2} \right) B(\tilde{v}, \tilde{u}, \tilde{w}) = \delta(\tilde{v}, \tilde{u}, \tilde{w}). \tag{40}$$

The rotated dimensionless coordinates are mapped to the Cartesian coordinates through the relationship

$$\frac{(\tilde{v}, \tilde{u}, \tilde{w})^t}{(z, x, y)^t} = \frac{(\tilde{v}, \tilde{u}, \tilde{w})^t}{(v, u, w)^t} \times \frac{(v, u, w)^t}{(z, x, y)^t} = \begin{bmatrix} \frac{1}{L_v} & 0 & 0 \\ 0 & \frac{1}{L_u} & 0 \\ 0 & 0 & \frac{1}{L_w} \end{bmatrix} \times R_{3D}(\theta, \varphi). \tag{41}$$

As coherent lengths, dip and azimuth (i.e. the parameters of the filter) are non-stationary, eq. (40) does not provide the same structure as eq. (8) when expressed in the Cartesian coordinate system. The chain rules for derivative estimation introduce spatial derivative terms that are related to the parameter variations in the PDE, when moving from one to the other set of coordinates, through the relationship of eq. (41). However, if the filter parameters vary slowly in space, their spatial derivatives can be ignored, which leads to the following approximation

$$\frac{(\partial\tilde{v}, \partial\tilde{u}, \partial\tilde{w})^t}{(\partial z, \partial x, \partial y)^t} \approx \frac{(\tilde{v}, \tilde{u}, \tilde{w})^t}{(z, x, y)^t} = \begin{bmatrix} \frac{1}{L_v} & 0 & 0 \\ 0 & \frac{1}{L_u} & 0 \\ 0 & 0 & \frac{1}{L_w} \end{bmatrix} \times R_{3D}(\theta, \varphi). \tag{42}$$

Within this approximation, the Bessel filters defined in the rotated dimensionless coordinates and in the Cartesian coordinates are almost identical. In other words, performing the smoothing operation in the dimensionless system or the Cartesian system should provide approximately the same results. This argument has an important role in this implementation, because it allows the weak formulation to be developed in dimensionless coordinates, to maintain the symmetry of matrices $\mathbf{K}$ and $\mathbf{A}$; this is a key point for numerical efficiency. It should be noted that this approximation will systematically introduce an error into the amplitude of the filtering operator when rapid variations of the filter parameters occur. The associated error analysis will be discussed in Section 4.1.

In the rotated dimensionless coordinate system, the differential relationship between the original vector $\mathbf{m}$ and the smoothed vector $\mathbf{s}$ that was obtained by filtering with the Bessel filter $B_{3D}(\tilde{v}, \tilde{u}, \tilde{w})$ can be written as

$$\mathbf{s}(\tilde{v}, \tilde{u}, \tilde{w}) - \nabla^2_{\tilde{v}, \tilde{u}, \tilde{w}} \mathbf{s}(\tilde{v}, \tilde{u}, \tilde{w}) = \mathbf{m}(\tilde{v}, \tilde{u}, \tilde{w}). \tag{43}$$

Ignoring the derivatives related to variations in the filter parameters allows the same workflow to be applied as for homogeneous parameters. Let us consider the mapping from reference space (ξ, η, ζ) to the rotated dimensionless coordinates $(\tilde{v}, \tilde{u}, \tilde{w})$, defined by the Jacobi matrix $\tilde{\mathbf{J}}^{\text{rot}}$

$$\tilde{\mathbf{J}}^{\text{rot}} = \frac{(\partial \tilde{v}, \partial \tilde{u}, \partial \tilde{w})^t}{(\partial \xi, \partial \eta, \partial \zeta)^t}, \tag{44}$$

and the volumetric Jacobian $\tilde{J}_e^{\text{rot}}$

$$\tilde{J}_e^{\text{rot}} = \det(\tilde{\mathbf{J}}^{\text{rot}}). \tag{45}$$

The weak form of eq. (43) can again be discretized into the linear system of eq. (31), where the mass matrix $\mathbf{M}$ is diagonal and the stiffness matrix $\mathbf{K}$ remains symmetric through relations

$$M_{\hat{k}\hat{k}} = w_{k_1} w_{k_2} w_{k_3} \tilde{J}_e^{\text{rot}}(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3}) \quad \text{where } \hat{k} \text{ stands for the triple of indexes } \{k_1, k_2, k_3\}, \tag{46}$$

and

$$K_{\hat{k}\hat{h}} = \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \left[\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} G_{ij} \frac{\partial v_{\hat{h}}}{\partial r_j} \right], \tag{47}$$

where

$$(p_1, p_2, p_3) := (\tilde{v}, \tilde{u}, \tilde{w}) \quad \text{and} \quad (r_1, r_2, r_3) := (\xi, \eta, \zeta).$$

The geometric factors G_{ij} associated with the projection between the rotated dimensionless coordinates and the reference coordinates becomes

$$G_{ij}(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}) = \sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \tilde{J}_e^{\text{rot}}(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}). \tag{48}$$

The numerical implementation is identical to the case of the homogeneous parameters in eqs (32), (34) and (35), where the linear system of eq. (31) is again constructed and solved in the reduced coordinates system. However, we need new expressions of the volumetric Jacobian $\tilde{J}_e^{\text{rot}}$, and the elements of the inverse Jacobi matrix $(\tilde{\mathbf{J}}^{\text{rot}})^{-1}$. These quantities can be computed from elements in the mapping between the reference space and the Cartesian space, knowing that

$$\tilde{\mathbf{J}}^{\text{rot}} = \frac{(\partial \tilde{v}, \partial \tilde{u}, \partial \tilde{w})^t}{(\partial \xi, \partial \eta, \partial \zeta)^t} = \frac{(\partial \tilde{v}, \partial \tilde{u}, \partial \tilde{w})^t}{(\partial z, \partial x, \partial y)^t} \times \frac{(\partial z, \partial x, \partial y)^t}{(\partial \xi, \partial \eta, \partial \zeta)^t}. \tag{49}$$

Combining eq. (49) with the approximation of eq. (42), we have

$$\tilde{\mathbf{J}}^{\text{rot}} \approx \begin{bmatrix} \frac{1}{L_v} & 0 & 0 \\ 0 & \frac{1}{L_u} & 0 \\ 0 & 0 & \frac{1}{L_w} \end{bmatrix} \times R_{3D}(\theta, \varphi) \times \mathbf{J}. \tag{50}$$

As $\det[R_{3D}(\theta, \varphi)] = 1$, the determinant of the Jacobi matrix $\mathbf{J}^{\text{rot}}$ is

$$\tilde{J}_e^{\text{rot}} = \frac{\det(\mathbf{J})}{L_v L_u L_w} = \frac{J_e}{L_v L_u L_w}. \tag{51}$$

The inverse of the Jacobi matrix $\mathbf{J}^{\text{rot}}$ is

$$(\mathbf{J}^{\text{rot}})^{-1} = \mathbf{J}^{-1} \times R_{3D}(\theta, \varphi)^{-1} \times \begin{bmatrix} L_v & 0 & 0 \\ 0 & L_u & 0 \\ 0 & 0 & L_w \end{bmatrix}, \tag{52}$$

or

$$(\mathbf{J}^{\text{rot}})^{-1} = \mathbf{J}^{-1} \times \begin{bmatrix} L_v \cos\varphi & L_u \sin\varphi & 0 \\ -L_v \cos\theta \sin\varphi & L_u \cos\theta \cos\varphi & L_w \sin\theta \\ L_v \sin\theta \sin\varphi & -L_u \sin\theta \cos\varphi & L_w \cos\theta \end{bmatrix}. \tag{53}$$

The volumetric Jacobian $\tilde{J}_e^{\text{rot}}$ and elements in the inverse Jacobian matrix $\tilde{\mathbf{J}}^{\text{rot}}$ can be computed from eqs (51) and (53), which completes the construction of matrix $\mathbf{M}$ and geometric factors G_{ij} in the dimensionless coordinates. If the azimuth and dip are zero, and if the coherent lengths are homogeneous, eqs (51) and (53) are identical to eqs (36) and (37). Finally, we end up with the linear system

$$\mathbf{As} = \mathbf{Mm}. \tag{54}$$

4 NUMERICAL ILLUSTRATIONS

Various 3-D examples that illustrate the numerical efficiency of the workflow are proposed in this section. We will show that the decay of Laplace filters can be mimicked by using Bessel operators. However, it should be emphasized that there is no obligation to use a Laplace filter; for example, the software SPECFEM3D uses a 3-D Gaussian smoothing filter (Peter *et al.* 2011). We will also quantify the numerical approximation for variable coherent length, dip and azimuth. We first consider an estimation of the Bessel kernel by considering an original vector **m** as a delta function at the origin. Then, we will consider the impact of the filter when the input vector only contains random noise, without any predefined structures. Finally, we show the application of the filter to a gradient obtained from a subset of the synthetic 3-D SEAM Phase II foothills model (Oristaglio 2012).

4.1 Spike test

When the source term, that is, the original vector $\mathbf{m}_0$, is set as a delta function at location $(0, 0, 0)$, the linear system

$$\mathbf{A} \, \mathbf{s} = \mathbf{M} \, \mathbf{m}_0, \tag{55}$$

should provide a solution that is identical to the Bessel filter (eq. 6). When we solve the linear system of eq. (55) twice through the following sequential system

$$\mathbf{A} \, \mathbf{s}^* = \mathbf{M} \, \mathbf{m}_0$$

$$\mathbf{A} \, \mathbf{s} = \mathbf{M} \, \mathbf{s}^*, \tag{56}$$

we should obtain a response similar to the Laplace filter. Let us consider a model of size $1.5 \, \text{km} \times 1.5 \, \text{km} \times 1.5 \, \text{km}$. A delta function is located at the centre of the model, as shown in Fig. 3.

In this example, the coherent lengths in the x and y directions are homogeneous $L_x = L_y = 80$ m and the dip and azimuth angle are set to zero, as there is no specific orientation. The coherent length L_z is either constant or variable in the z direction, as indicated in Figs 3A(a), B(a) and C(a). In Figs 3A(b), B(b) and C(b), the output of the linear system (blue dashed line) and the theoretical Laplace function (red line) in the z directions are superimposed. Fig. 3A(b) illustrates that the double application of the Bessel filter provides an excellent approximation of the Laplace filter for homogeneous coherent lengths, although still with the negligible singularity at the origin. Under the slow variation of the coherent lengths, $\partial_z L_z = 0.1$ in Fig. 3B(b), this method can correctly follow the shape of the Laplace filter, which implies that ignoring the spatial derivatives of the coherent lengths is acceptable. This conclusion is further supported by the identical shapes of the zx cross-sections of the 3-D Laplace filter and the output of the linear system of eq. (55) in Figs 3B(c) and (d). When this variation becomes more important, as $\partial_z L_z = 0.2$, this approximation cannot exactly reproduce the amplitude of the theoretical filter, but the influence of this approximation still appears acceptable for the smoothing effect of the model vector **m** that we consider.

The implementation of constant dip and azimuth is illustrated by the spike test in Fig. 4. The delta function is again located at the centre of the model. Homogeneous different coherent lengths are used in all directions, but the filters are highly anisotropic, with $L_v = 50$ m, $L_u = 100$ m, and $L_w = 200$ m. Due to this design, the 3-D filter is a tri-axial ellipsoid with distinct semi-axis length, as indicated in the zx, xy, and zy cross-sections in Fig. 4(A). After applying the rotation with azimuth $\theta = 60°$ and dip $\varphi = 30°$, the comparison between Figs 4(B) and (A) shows that the ellipsoid is tilted $30°$ from the x axis in the zx section, and it is also rotated $60°$ from the y axis in the xy cross-section.

When variable dip and azimuth are introduced into the filter, similar amplitude errors as for the variable-coherent-lengths study in Fig. 3(C) are expected. Furthermore, the amplitude error introduced by the 3-D rotation can be mitigated by careful design of the coherent lengths, as the rotation has no impact in an isotropic filter. For FWI applications, which are generally limited at low frequency, the approximation of the slow variation of the filter parameters still appears acceptable for the smoothing effect.

4.2 Random noise tests

The filtering operator normally assumes the particular properties of the structures, which implies that the shape of the smoothed gradient/model is driven by the imposed variation of coherent lengths, dip, and azimuth. This argument is illustrated in the following examples, when the input vector **m** only contains random noise, with no pre-defined structures. Fig. 5 focuses on homogeneous coherent lengths, dip, and azimuth filters on a model of size $1.5 \, \text{km} \times 1.5 \, \text{km} \times 1.5 \, \text{km}$. The zx, xy, and zy cross-sections of the input vector are shown in Fig. 5(A), which only contains high frequency variations of random noise. Fig. 5(B) shows the smoothed vector obtained with coherent lengths $L_z = L_v = 25$ m, $L_x = L_u = 100$ m, and $L_y = L_w = 25$ m, without dip and azimuth. As the filter is strongly anisotropic, with the longest semi-axis length in the x direction, the zx and xy sections both contain features aligned in the x direction. The patterns in the zy section have no specific orientation as $L_z = L_y$, that is, the 3-D filter is isotropic in the zy plane. In Fig. 5(C), after applying a rotation with $45°$ dip, the aligned features are tilted $45°$ from the x axis in the zx section. Note that the apparent coherent length L_x is now almost identical to L_y, with no specific trend in the xy cross-section. Meanwhile, the apparent L_z becomes greater than L_y, which induces the alignment along the z direction in the zy cross-section. A similar interpretation can be applied to Fig. 5(D), which shows the application of a filter with azimuth $45°$ to smooth the initial vector. Compared to Fig. 5(B), the features in the xy section are rotated $45°$ from the y axis, and the patterns presented in the zx cross-section are lost due to the apparent coherent length in the x direction.
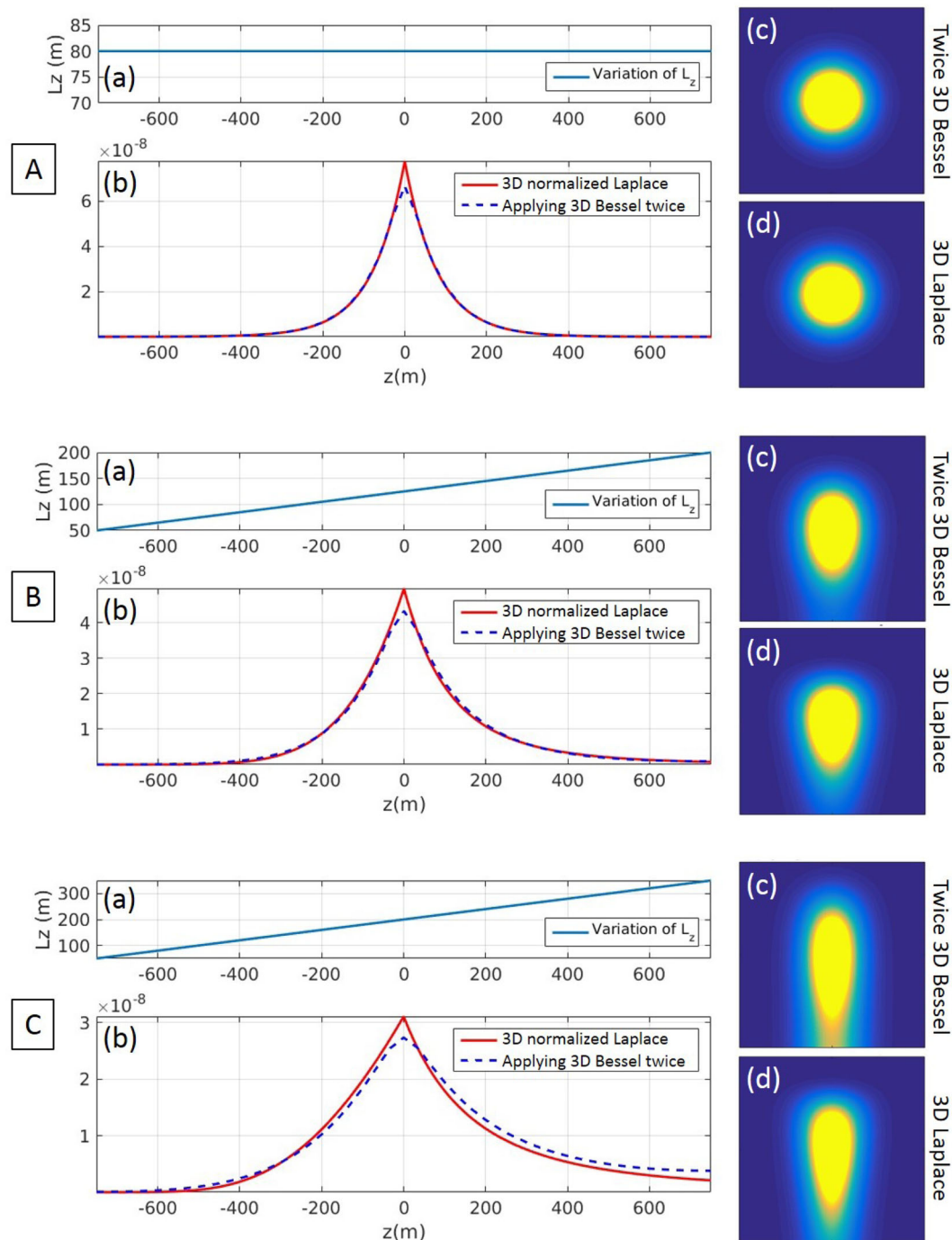
Figure 3. Comparison of the normalized Laplace filter with the application of two Bessel filters through the spike tests, with homogeneous coherent lengths in the x and y directions $L_x = L_y = 80$ m. Coherent length in the z direction: (A) Constant L_z; (B) L_z varies from 50 m to 200 m; (C) L_z varies from 50 to 350 m. In each panel, (a) shows the variation of L_z in the z direction, and (b) compares the output of applying the 3-D Bessel filter twice (blue dashed line), with the theoretical Laplace filter (red line). The respective zx cross-section of these filters are shown in (c) and (d).

The example shown in Fig. 6 is a longer gradient/ model with 3 km length in the x direction that is used to illustrate the implementation of variable coherent lengths, dip and azimuth. We use simple sine variation for these geological properties. Again, the input vector (Fig. 6A) contains random noise without any predefined structure. In Fig. 6(B), the coherent lengths in the z and y directions are homogeneous $L_z = L_y = 25$ m, whereas L_x varies as a sine function in the x direction, from 25 m to 85 m. The zx section of the smoothed vector (Fig. 6B, left) correctly follows the variation of the coherent length L_x (Fig. 6B, right). Fig. 6(C) shows the zx and xy cross-sections of the smoothed vector with homogeneous coherent lengths $L_z = L_v = 25$ m, $L_x = L_u = 250$ m, and $L_y = L_w = 25$ m, and no dip and azimuth. An extremely long L_u was intentionally designed so that the smoothed vector has a layered structure, in parallel with the x direction. Fig. 6(C) will be used as the reference to compare this with the variable dip and azimuth filters in the next examples. In Fig. 6(D), similar values of homogeneous
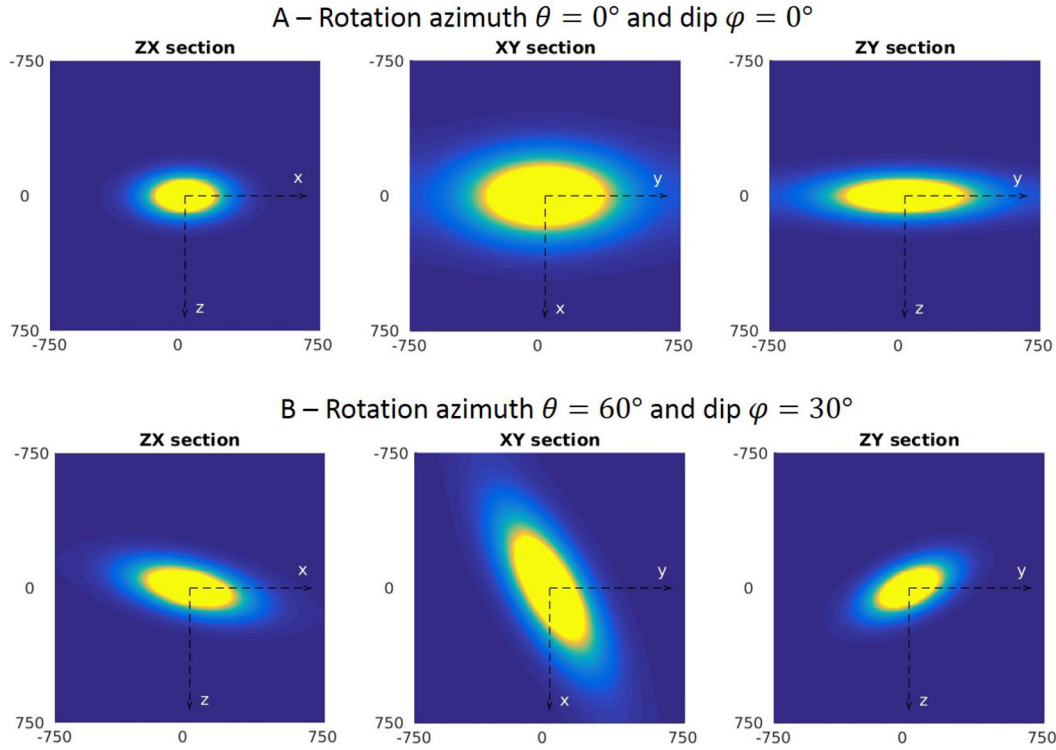
Figure 4. Spike test to illustrate the kernel of the 3-D Bessel filter with homogeneous coherent lengths $L_v = L_z = 50$ m, $L_x = L_u = 100$ m, and $L_y = L_w = 200$ m. (A) *zx*, *xy* and *zy* cross-sections of the 3-D Bessel filter without dip and spike. (B) The filter consists of a 3-D rotation with 60° azimuth and 30° dip.

coherent lengths were use as those for Fig. 6(C). The dip (Fig. 6D, right) varies as a sine function in the *x* direction from −45° to 45°, which leads to folding structures in the *zx* section (Fig. 6D, left). Similar variation in the *xy* cross-section in Fig. 6(E) is obtained when the azimuth varies as a sine function in the *x* direction, from −60° to 60°.

4.3 FWI gradient smoothing

This section illustrates the application of a non-stationary Laplace filter to a realistic gradient vector from FWI, as obtained from a subset of the 3-D SEAM Phase II foothills model (Oristaglio 2012). Surface acquisition is used with a line of 20 sources, with 350 m between adjacent sources. The receivers are located in the whole surface, with 12.5 m between receivers. A Ricker wavelet centred at 3 Hz is used as the source signal. The 2-D cross-section of the shear-wave velocity (V_s) model underneath the source line is shown in Fig. 7(A).

The topography variation is significant in this model, with maximal vertical elevation of 900 m. SEM is used for both forward and inversion problems. The initial V_s model is shown in Fig. 7(B), which is a smoothed version of the true model. Compared with the true model in Fig. 7(A), this initial V_s is overestimated at the near surface and underestimated at greater depths. The dip field is extracted from the true velocity model by manual picking, as illustrated in Fig. 7(C). Fig. 7(D) shows the first scaled gradient without any smoothing filter, which contains a significant acquisition footprint at the near surface. The horizontal oscillation of the features at greater depths might come from high wavenumber artefacts.

An anisotropic non-stationary Laplace filter with coherent lengths $L_z = 0.05\lambda_s$ and $L_x = L_y = 0.15\lambda_s$ is applied, where λ_s is the shear wavelength at each spatial position. The true dip field, as shown in Fig. 7(C), and the zero azimuth angle are used for 3-D rotation. It should be noted again here that the Laplace filter is efficiently approximated by application of two Bessel filters. The filters help to remove the near-surface artefacts due to the acquisition footprint, without degrading the deeper structures. The continuity of the features at greater depths is actually enhanced because the horizontal-oscillation artefacts are attenuated. In addition, the gradient correctly determines the update direction of the model: it reduces the V_s at the near surface and increases it at greater depths (knowing that FWI updates the model following the negative gradient).

In summary, we have illustrated the property of the Bessel filter through various numerical examples in spike and random-noise tests. The approximation of the Laplace filter by Bessel operators is shown, which indicates prospective applications of this filter, either as a smoothing filter, or to efficiently mimic the decay of the Laplace filter. We also illustrate the robustness of this method for variable coherent lengths, dip and azimuth. The numerical example on a realistic gradient highlights the potential application of the filter for FWI.
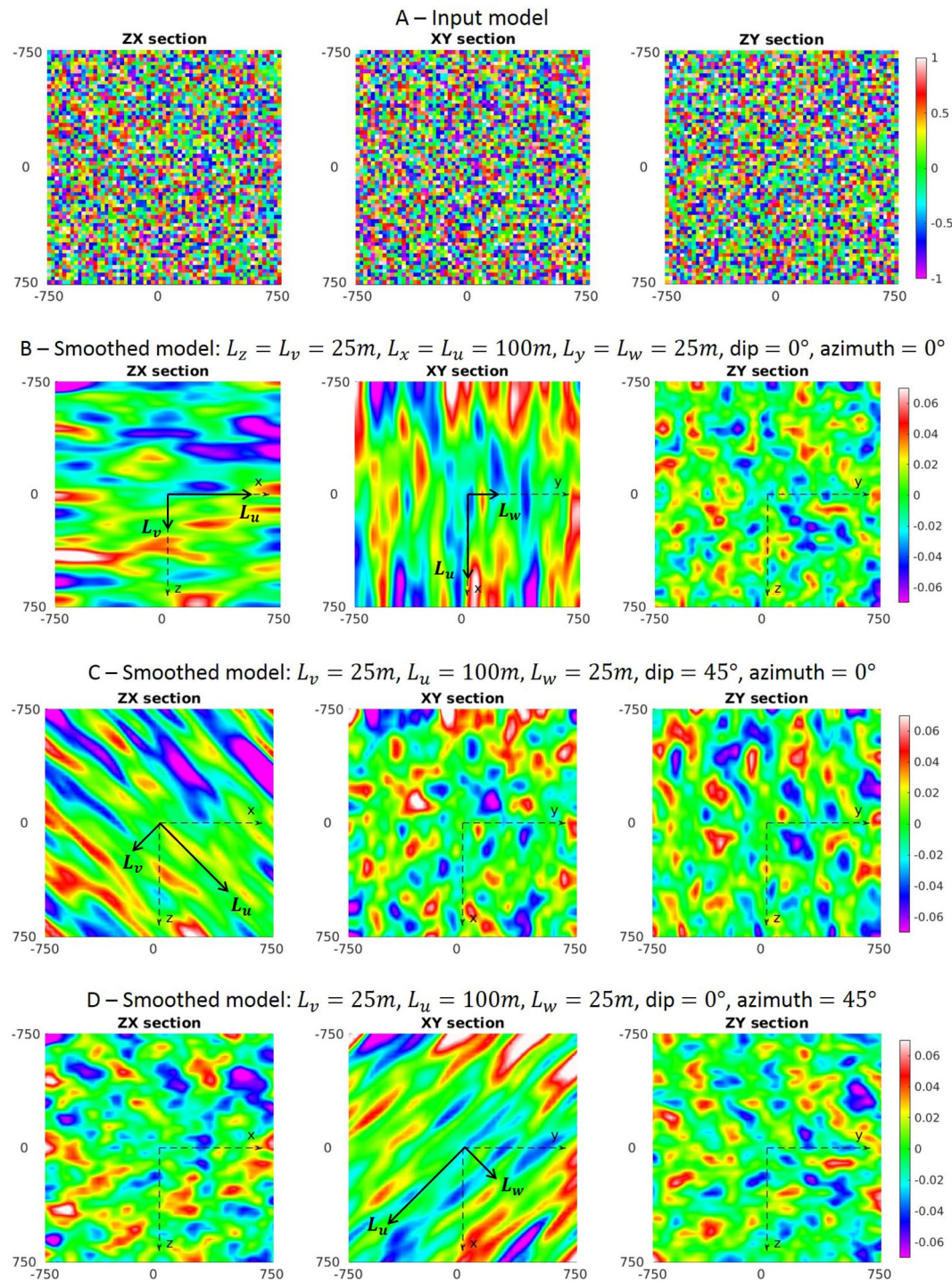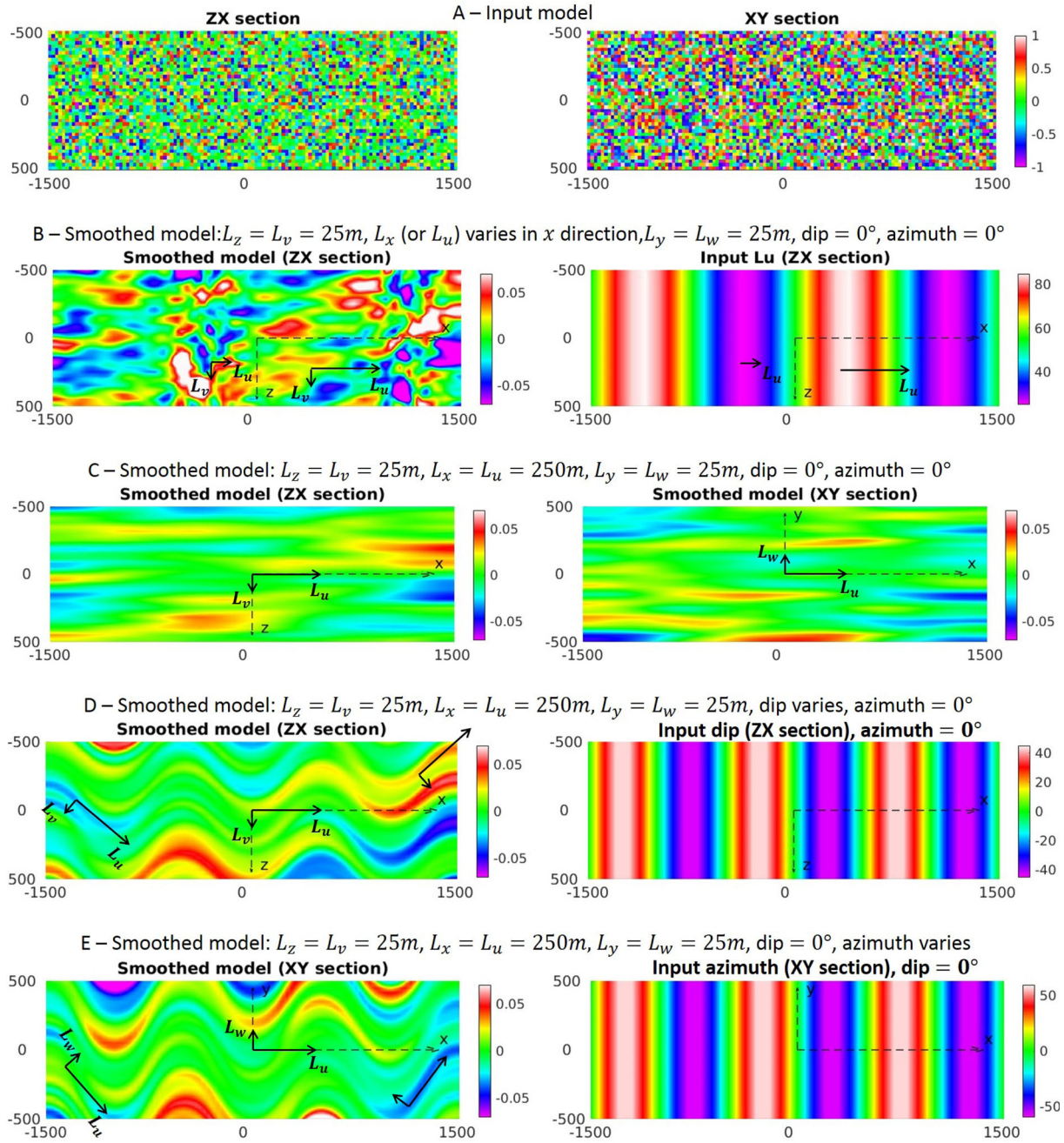
Figure 5. Random noise test with a stationary filter to illustrate the implementation of homogeneous coherent lengths, dip and azimuth. (A) *zx*, *xy* and *zy* cross-sections of the input model. (B) Smoothed model obtained from applying an anisotropic filter, $L_v = 25$ m, $L_u = 100$ m, and $L_w = 25$ m, without dip and azimuth. (C) Smoothed model when $45°$ dip and $0°$ azimuth are introduced into the filter. (D) Smoothed model when $0°$ dip and $45°$ azimuth are introduced into the filter. Model size, $1.5\,\text{km} \times 1.5\,\text{km} \times 1.5\,\text{km}$.

5 NUMERICAL IMPLEMENTATION IN A PARALLEL SEM SCHEME

In FWI, the gradient vector of the misfit function is computed from the correlation of the forward and backward propagation wavefields (as one per source) at each iteration, due to the adjoint-state method (Plessix 2006). However, the modelling mesh can sometimes be quite dense compared to the resolution that can be expected from the inversion, which leads to high wave-number noise in many applications. Consequently, the gradient/ model vector must be smoothed or regularized on this forward-modelling mesh. By doing so, the filtering operation (i.e. the Bessel operator here) has to be directly and efficiently implemented on the modelling mesh, which can be described by a domain

Figure 6. Random noise test with non-stationary filter to illustrate the implementation of variable coherent lengths, dip and azimuth. The size of the model is 1 km in the *z* and *y* directions, and 3 km in the *x* direction. (A) *zx* and *xy* cross-section of the input model. (B) Smoothed model (left) obtained from a filter with variable coherent length L_x (right). (C) *zx* and *xy* cross-section of the smoothed model obtained from a highly anisotropic stationary filter. (D) Smoothed model (left) obtained from a filter with variable dip (right). (E) Smoothed model (left) obtained from a filter with variable azimuth (right).

decomposition for parallel computation. As the application of the Bessel filter is related to a PDE, this can proceed in a similar way as for the wave-propagation simulation.

Following an analogue of the framework to that designed for the SEM for wave simulation, the linear system associated with the Bessel filters is constructed and solved by a parallel CG iterative solver

$$\underbrace{(\mathbf{M} + \mathbf{K})}_{\mathbf{A}} \mathbf{s} = \mathbf{M}\,\mathbf{m}. \tag{57}$$

According to Saad (2003), only the matrix-vector product and the inner product of two vectors are required in the CG method. Thus, the most expensive operator is the product of the matrix **A** with a given vector. Each subdomain computes its part of the product in parallel, and the communications between subdomains during the CG iterations share the same strategies as that of wave simulation, which requires no extra
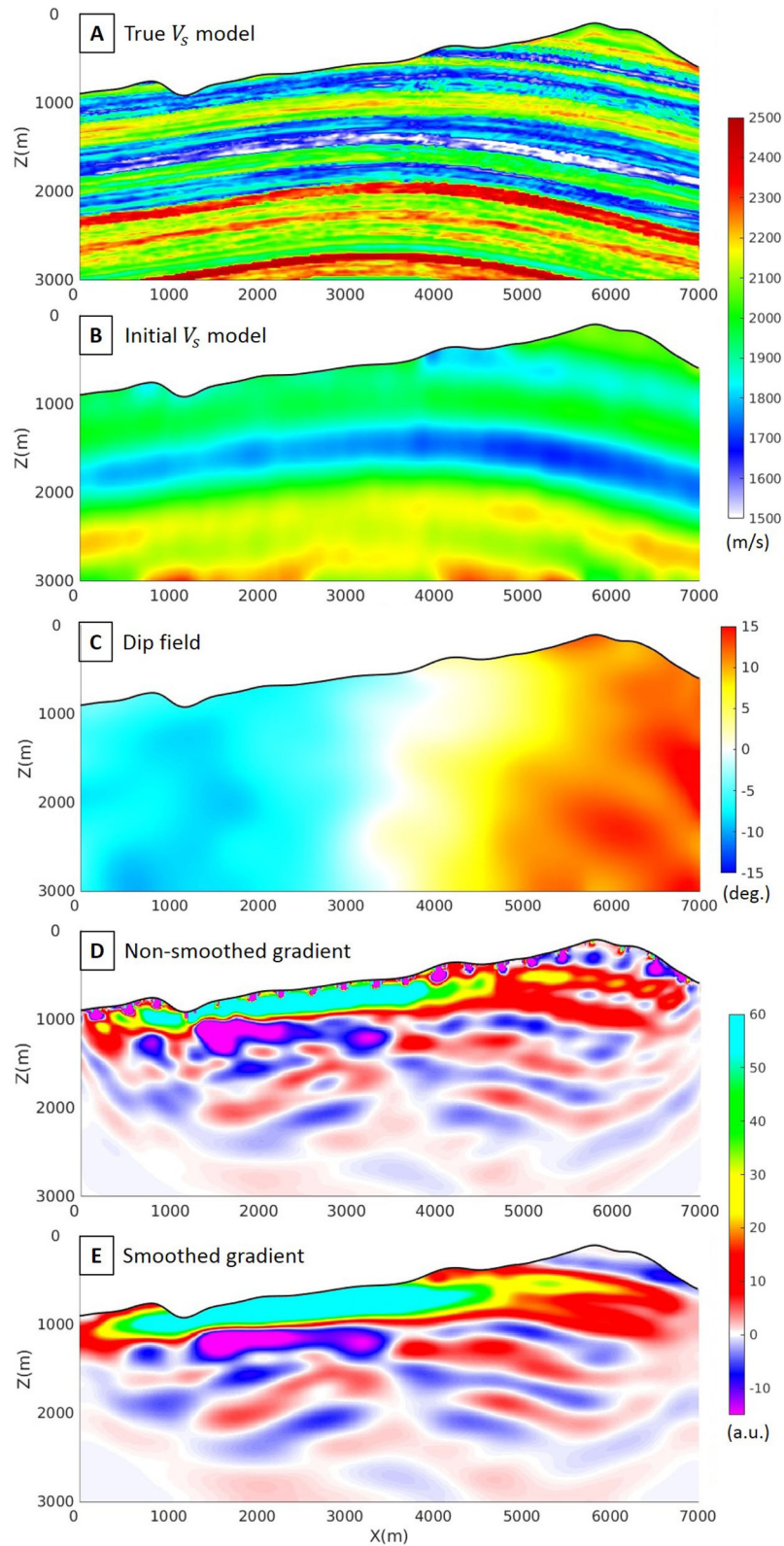
Figure 7. Example of the filtering operator on the FWI gradient, from a subset of the 3-D SEAM Phase II foothills model. (A) True shear wave velocity model. (B) Initial shear wave velocity model. (C) Dip field. (D) Original scaled gradient without any smoothing. (E) Smoothed gradient, with an anisotropic non-stationary Laplace filter (approximated by the application of two Bessel filters) for the dip field as presented in (C). $L_z = 0.05\lambda_s$ and $L_x = L_y = 0.15\lambda_s$, where λ_s is the local shear wavelength.

effort to manage the parallelism. This section presents this implementation and highlights the efficiency of this approach, compared with the standard 3-D convolution method. The convergence rate of the CG solver for several parameters are then analysed.

5.1 Linear system construction and matrix-vector product evaluation

The mass matrix $\mathbf{M}$ is diagonal, and the number of non-zero elements (NNZs) of this matrix is identical to the size of the input gradient/model vector,

$$NNZ(\mathbf{M}) = SIZE(\mathbf{m}). \tag{58}$$

It is then stored in the same form as the input vector. The dimensions of the matrices $\mathbf{K}$ and $\mathbf{A}$ are $[SIZE(\mathbf{m})]^2$; thus the full storage is not reasonable for realistic application. Assuming that the same order of interpolation N is used in each direction, and the inverse of Jacobi matrix $\check{\mathbf{J}}^{rot}$ is full of non-zero elements, the NNZs in the matrix $\mathbf{A}$ can be estimated from the size of the vector $\mathbf{m}$, as

$$\frac{NNZ(\mathbf{A})}{SIZE(\mathbf{m})} \longrightarrow (3(N+1)^2 + 3(N+1) + 1). \tag{59}$$

For example, when 4th order interpolation is used (i.e. each hexahedral element is discretized by $5 \times 5 \times 5$ GLL points)

$$NNZ(\mathbf{A}) \approx 91 \times SIZE(\mathbf{m}). \tag{60}$$

This estimation illustrates that the stiffness matrix is extremely sparse. Therefore, the non-zero elements of this matrix can either be stored by some efficient compressed storage technique or the matrix-vector product can be directly evaluated without storing the matrix. These two implementation approaches are discussed in this section.

5.1.1 Coordinate list storage of non-zero elements of matrix A

As the impedance matrix $\mathbf{A}$ is sparse, only the non-zero elements of this matrix need to be computed and stored. At each CG iteration, these non-zero elements are used to evaluate the product of matrix $\mathbf{A}$ with a given vector.

The matrix $\mathbf{A}$ is computed from the stiffness matrix $\mathbf{K}$. According to eq. (47), the element at row $\hat{k}$th, column $\hat{h}$th of matrix $\mathbf{K}$ can be computed through a triple loop over all of the degrees of freedom (q_1, q_2, q_3) inside the current element, which is computationally expensive. This component can be further developed as a sum of six terms, as shown in eq. (C17) in Appendix C. This development allows the computation of the left-hand side matrix $\mathbf{A}$, while minimizing the loops over all of the degrees of freedom inside the current element.

The matrix $\mathbf{A}$ can thus be stored using the compressed storage techniques coordinate list (COO) format (Golub 1996), where a list of row and column indices and the associated values of the non-zero elements are stored. In addition, as the matrix $\mathbf{A}$ is symmetric, only the upper triangular part has to be stored; that is, the elements A_{ji} where $i \geq j$. We only store half of the diagonal terms $\frac{1}{2}A_{jj}$. This helps to simplify the matrix-vector product in the CG algorithm. By applying this triangular storage strategy, and based on several tests carried out with this implementation on realistic size problems, the required memory for storage of matrix $\mathbf{A}$ is reduced by 49 per cent, and the computation time of the CG by 33 per cent, .

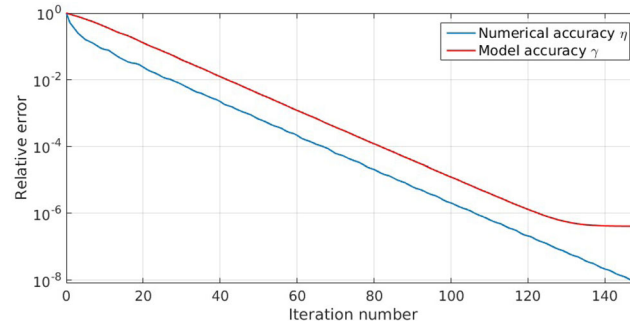#### 5.1.2 Matrix-free implementation of the matrix-vector product Au

Instead of building explicitly the impedance matrix $\mathbf{A}$ by storing its non-zero elements, its product with a given vector $\mathbf{u}$ can be directly estimated inside each CG iteration in a matrix-free fashion. As the product of a vector with the diagonal mass matrix $\mathbf{M}$ is trivial, the challenge is the stiffness matrix-vector product $\mathbf{Ku}$. According to the development of the weak formulation of eq. (26) in Appendix C and the definition of the geometric factors G_{ij} in eq. (48), this matrix-vector product can be written as

$$\sum_{\hat{h}} K_{\hat{k}\hat{h}} u_{\hat{h}} = \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \sum_{\hat{h}=0}^{N} \left[\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} G_{ij} \frac{\partial v_{\hat{h}}}{\partial r_j} \right] u_{\hat{h}}, \tag{61}$$

or under the factorized version, as

$$\sum_{\hat{h}} K_{\hat{k}\hat{h}} u_{\hat{h}} = \underbrace{\sum_{i=1}^{3} \left\{ \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \frac{\partial v_{\hat{k}}}{\partial r_i} \overbrace{\left[\sum_{j=1}^{3} G_{ij} \underbrace{\left(\sum_{\hat{h}=0}^{N} \frac{\partial v_{\hat{h}}}{\partial r_j} u_{\hat{h}} \right)}_{\mathbf{Du}} \right]}^{\mathbf{GDu}} \right\}}_{\mathbf{D}_w \mathbf{GDu}} . \tag{62}$$

Figure 8. Relative numerical error modulus η and model error modulus γ for smoothing a model vector containing $80 \times 80 \times 80$ elements; that is, 30×10^6 degrees of freedom. The coherent lengths are homogeneous $L_z = L_x = L_y = 80$ m, without dip and azimuth.

Eq. (62) implies that the product stiffness matrix-vector can be factorized as

$$\mathbf{Ku} = \mathbf{D}_w\mathbf{GDu}, \tag{63}$$

where the operator $\mathbf{D}$ evaluates the spatial derivatives of the vector $\mathbf{u}$ in the reference space, $\mathbf{G}$ is the geometric matrix, and $\mathbf{D}_w$ is equivalent to a weighted spatial derivatives operator.

The derivatives operators $\mathbf{Du}$ and the weighted spatial derivatives $\mathbf{D}_w(\mathbf{GDu})$ can be efficiently estimated using the highly efficient algorithms developed by Deville *et al.* (2002), which benefit from the tensorial properties of SEM and the optimization of cache usage by manual loop unrolling. For each degree of freedom, the geometric matrix $\mathbf{G}$ is symmetric and only depends on the projection between the rotated dimensionless coordinates and the reference coordinates (eq. 48). Therefore, only six vectors, as G_{11}, G_{12}, G_{13}, G_{22}, G_{23}, G_{33}, are stored in the memory, and they are computed outside the CG loops. Due to the factorization in eq. (62) and the vectorized computation for the stiffness/vector product, each CG iteration costs about 0.46 times a time-step of wave propagation for deformed elements. Compared with the explicit matrix-vector product with the COO format of the previous section, this matrix-free implementation decreases the computation cost by a factor of around 7, and reduces the memory requirement by $>$20-fold. These numbers have been estimated on the same high-performance computing architecture for the two implementations, on a model with 33×10^6 degrees of freedom, with decomposition on 64 subdomains (64 computing cores with infiniband interconnections).

5.2 Parallel conjugate gradient iterative solver

The parallel CG solver introduced in this section is similar to the standard algorithm, except that the matrix-vector and vector-vector operators are computed in parallel. As mentioned before, the matrices $\mathbf{M}$ and $\mathbf{A}$ are evaluated in parallel without assembly over the boundaries between subdomains. After each matrix-vector product, some point-to-point communications are required to assemble the values on the shared degrees of freedom.

The modulus of the relative numerical error

$$\eta = \frac{\|\mathbf{M.m} - \mathbf{A.s}^k\|}{\|\mathbf{M.m}\|} \tag{64}$$

is used for the convergence criteria, where $\mathbf{s}^k$ is the smoothed model obtained at the kth iteration. In the next paragraphs, we study the properties of the linear system (eq. 57), as well as the convergence rate based on the values of the error modulus η.

5.2.1 Linear system behaviour

First, we build a reference system with smoothed vector $\mathbf{s}_{\text{ref}}$ and input $\mathbf{m}_{\text{ref}}$, which is computed as

$$\mathbf{m}_{\text{ref}} = \mathbf{M}^{-1}.\mathbf{A.s}_{\text{ref}}. \tag{65}$$

The linear system

$$\mathbf{A.s} = \mathbf{M.m}_{\text{ref}} \tag{66}$$

is then solved iteratively by the CG solver. At each iteration, the modulus of the relative model error γ is computed

$$\gamma = \frac{\|\mathbf{s}_{\text{ref}} - \mathbf{s}^k\|}{\|\mathbf{s}_{\text{ref}}\|}. \tag{67}$$

The numerical error η is defined in eq. (64), and it is compared to this model error γ to characterize the behaviour of the linear system of eq. (57). An example of this comparison for a model vector that contains 30×10^6 degrees of freedom is shown in Fig. 8. The filter is defined with homogeneous coherent lengths $L_z = L_x = L_y = 80$ m, without dip and azimuth. It should be noted that the numerical and model error
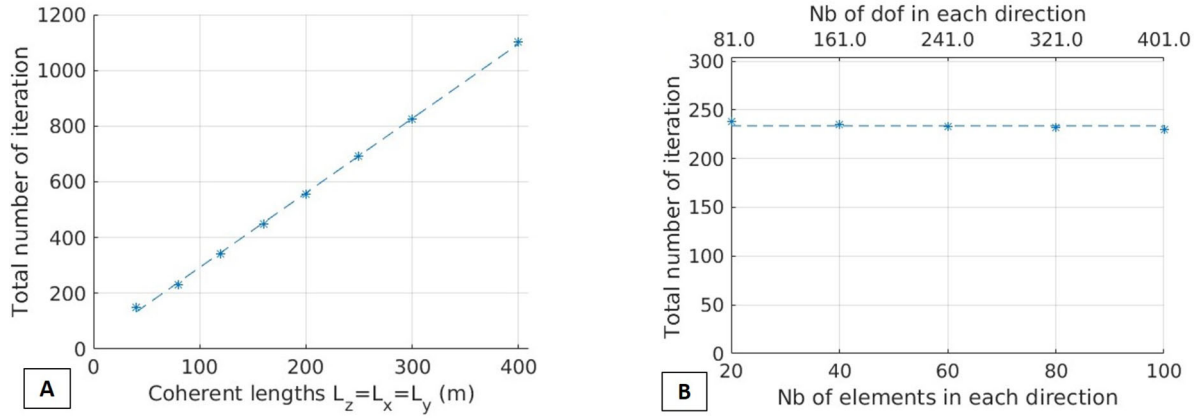
Figure 9. (A) Dependence of the total number of iterations (obtained from consecutively solving two linear systems associated with the Bessel filter) on coherent length. The model contains $80 \times 80 \times 80$ elements; that is, 30×10^6 degrees of freedom. (B) Dependence of the total number of iterations on the size of the problem. For both images, the stopping criterion is $\eta = 10^{-6}$.

modulus both decrease with the same speed, which implies that the linear system associated with the Bessel filter is well conditioned. The system converges relatively rapidly, and attains the numerical limit of 10^{-6} after 110 iterations, using FORTRAN single-precision arithmetics. Beyond this limit, the relative model accuracy remains unchanged, so the stopping criteria can be set as $\eta = 10^{-6}$. In FWI applications, to optimize the numerical cost, the CG solver could be stopped before the numerical limit is attained, because the model parameters will be repetitively modified. Evaluation of this tolerance η should be carefully studied, according to the objective of the application.

5.2.2 Convergence properties

After fixing the stopping criteria as $\eta = 10^{-6}$, we investigate the dependence of the convergence rate of the CG solver on several parameters, including coherent length, size of model vector, and presence of dip and azimuth. As we are approximating the Laplace filter by the Bessel operators, the linear system associated with the Bessel filter is solved twice to obtain the smoothed vector. The number of iterations shown in Fig. 9 is then estimated by consecutively solving two linear systems. Obviously, the same conclusions should be draw when solving one linear system, which corresponds to the direct application of the Bessel filter.

Fig. 9(A) indicates that for a given model the number of iterations required for solving two linear systems increases linearly with the values of the coherent length. This plot is obtained from a model of size $80 \times 80 \times 80$ elements; that is, 30×10^6 degrees of freedom. Each point in the plot is obtained from filtering with the isotropic Bessel operator, where the coherent lengths are homogeneous. We carry out a similar study when the coherent lengths are fixed, and the dip and azimuth angles are introduced into the filter. These parameters have no influence on the computation time. Therefore, these data are not shown here. In Fig. 9(B), the filter shape is kept unchanged, with $L_z = L_x = L_y = 80$ m, but the size of the model vector increases gradually from $20 \times 20 \times 20$ elements to $100 \times 100 \times 100$ elements. The corresponding number of degrees of freedom in each direction is indicated by the axis at the top of Fig. 9(B). For a pre-defined filter, the total number of iterations required for smoothing a model vector is independent of the model size.

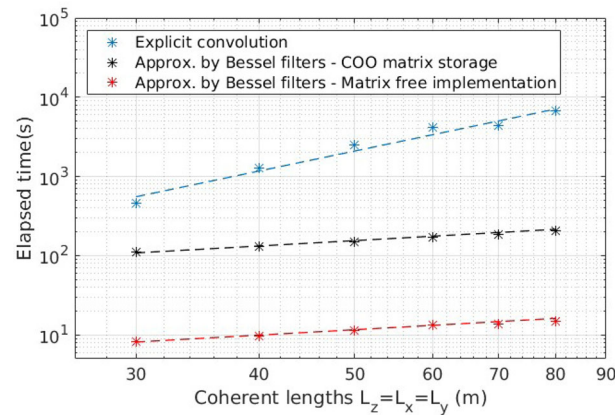### 5.2.3 Numerical performance versus explicit 3-D convolution filtering

As Bessel filters can be efficiently used to approximate the Laplace filter, the efficiency of this method needs to be evaluated with respect to the 3-D explicit convolution approach. We limit this study to an isotropic and homogeneous Laplace filter

$$L_{3D}(z, x, y) = \frac{1}{8\pi L^3} e^{-\sqrt{\frac{x^2+y^2+z^2}{L^2}}}. \tag{68}$$

The filter theoretically has infinite tail, but it can be implemented as a truncated 3-D convolution of the filter and the model/ gradient vector. We then define a sphere $\Omega_{\alpha L}$ centred at the origin, with radius αL, such that

$$\iiint_{\Omega_{\alpha L}} L_{3D}(z, x, y) \mathrm{d}z \mathrm{d}x \mathrm{d}y = 0.99, \tag{69}$$

which provides $\alpha \approx 8.4$ (Fig. 11). This condition is equivalent to 99 per cent energy of the filter. Fig. 10 compares the computation cost of our approach with the 3-D explicit convolution in a model with 1.77×10^6 degrees of freedom and where the coherent length increases from 30 to 80 m. The 3-D explicit convolution is the most expensive method, but the comparison is partially biased as our approaches are highly optimized. However, it should be noted that the numerical cost of the 3-D explicit convolution method is driven by the number of points within the effective sphere $\Omega_{\alpha L}$, thus with the complexity $\mathcal{O}(L^3)$. Whereas, for a given model size the computational complexity of the approximation of the Laplace filter by Bessel operators only depends on the number of CG iterations, which increases linearly with the

Figure 10. Comparison of the numerical performance of this method. The COO storage of non-zero elements in matrix **A** (black) and the matrix-free implementation of the product matrix-vector **Au** (red) are compared to the truncated explicit 3-D convolution of the Laplace filter (blue) for a model with 1.77×10^6 degrees of freedom, with decomposition for two subdomains.
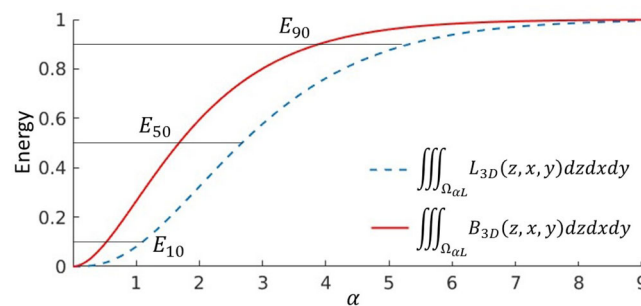
Figure 11. Energy analysis for isotropic Laplace and Bessel filters within the sphere $\Omega_{\alpha L}$ centred at the origin with the radius αL (L is the coherent length in the z, x and y directions). E_{10}, E_{50} and E_{90} indicate the 10 per cent, 50 per cent and 90 per cent energy.

coherent length ($\propto \mathcal{O}(L^1)$), as shown in Fig. 9(A). In Fig. 10, the observed numerical complexity of the explicit convolution method *versus* the increase in the coherent length is $\mathcal{O}(L^{2.6})$ due to the limited model size. We obtain a sublinear relationship between the numerical cost of our approximation and the coherent length ($\mathcal{O}(L^{0.7})$), for the two implementation approaches mentioned in Section 5.1: COO-storage and matrix-free implementation. This is due to some calculations that are independent of the coherent length. This property is encouraging for the application of this approach to large problems.

6 CONCLUSION AND DISCUSSION

The application of a filter on a given vector is usually described through convolution of the vector by a filter that is a linear operation. Smoothing will require filters with decaying tails. Thus, the convolution operator might be expensive if it is computed explicitly. We promote the idea of applying the filter through its inverse operator, which is expected to be a sparse operator. The smooth output is obtained as the solution of a new linear system, in which the sparse inverse of the filter has been constructed. We introduce the Bessel filters and their corresponding diffusion-like PDEs to be solved numerically in 2-D and 3-D geometries. We analytically explain why the iterative application of two Bessel filters provides an excellent approximation of the Laplace operator for smoothing the model, which offers the choice of either using Bessel filters as smoothing operators or to mimic the decay of the Laplace filter. This argument is well supported by the numerical illustration of spike tests, where the PDE defining the Bessel filter is solved with a delta source function.

These PDEs can be discretized through a weak formulation that involves SEM, which leads to a symmetric linear system that can be efficiently solved through a CG method. There are no specific requirements for the mesh, which can be chosen in a suitable way for the application being considered.

We present a robust and efficient approach to incorporate variable coherent lengths, dip and azimuth into the Bessel filter, without losing the symmetry of the left-hand-side term of the linear system. To do so, we develop the weak formulation of the PDE associated with the sparse inverse filter in the rotated dimensionless coordinate system. The construction of this system naturally takes advantage of the available ingredients in the underlying mesh that are associated to the study problem. All of the information about the geological variations of the medium is preserved in the Jacobi matrix and its inverse, which does not break the symmetry of the stiffness matrix. The well-behaving performance of this approximation is illustrated by spike tests, which closely reproduce the shape of analytical Laplace filter when assuming slow variations of these geological properties. In more complex tests, when random noise is used as the input model, the output vector correctly

follows the imposed shape that is described by the filter parameters. When the non-stationary filter is applied to a realistic FWI gradient, it effectively removes near-surface artefacts due to the acquisition footprint, and further enhances the continuity of the deeper structures.

We propose efficient implementation of the application of the Bessel filter, where the linear system is solved with a parallel CG following the same domain decomposition as the wave simulation. This parallel computation of the matrix-vector product at each CG iteration can be achieved either by using explicit building of the local matrix, or in a matrix-free fashion. For the matrix-free implementation, the stiffness matrix-vector product can be factorized and highly optimized, due to the vectorized computation and the optimizing cache usage developed by Deville *et al.* (2002). This strategy significantly reduces the memory requirement and computation time for the CG solver. After each matrix-vector or vector-vector product, the vectors obtained are spatially communicated to assemble the values on the subdomain boundaries. Consequently, any projection back and forth from the modelling mesh and the inversion mesh is avoided, as well as any explicit windowed convolution.

The sparse inverse linear system associated with the Bessel filter is well conditioned, due to the structure of the PDE. The CG solver applied to this system converges relatively rapidly. The convergence depends on many factors, but it increases linearly with the coherent length. Other factors, such as the number of elements in the medium and the presence of dip and azimuth angles, appear to have no influence on the number of iterations required for the system to converge. The approximation of the Laplace filter by the Bessel filter proposed here is also compared with the windowed explicit 3-D convolution method. The approach here is significantly faster, with a sublinear relationship between the numerical cost and the coherent length, which promises general applications to various large problems.

One question might be whether we should apply one Bessel filter with longer coherent length or two Bessel filters, to reproduce the decay of the Laplace filter. Choosing between Bessel and Laplace filters is case dependent, and this should be decided based on the energy distribution of the filters and the numerical cost. Knowing that the energy distribution of these filters are different (as shown in Fig. 11), they should result in different smoothing effects. To have the same energy as the Laplace filter at 90 per cent, 50 per cent or 10 per cent energy, the coherent length in the Bessel filter should be increased by a factor of 1.4, 1.6 or 2.1, respectively. These different scaling factors also imply that it is not possible to mimic the Laplace filter by simply modifying the coherent length in the Bessel filter. For the same coherent length, the contributions of the neighbors surrounding the origin are more important in the Bessel kernel, which is indicated by the 10 per cent energy level. It should also be noted that increasing the coherent length will eventually increase the numerical cost of the iterative solving of the linear system associated with the Bessel filter.

Only the numerical implementation for SEM is presented here, but the method can be extended to other FE techniques, and also to FD techniques (Wellington 2016). This study illustrates the potential applications for model preconditioning, but this approach of applying a filter (or a covariance matrix) can be considered for regularization and for prior-model strategies. As the filter has a robust shape, even with dip, azimuth and coherent length variations, it can be used to amplify expected geological features, while attenuating features that come from numerical artefacts in other areas. In this way, the null-space dimension is reduced, to converge to more meaningful models while honouring the data fit.

REFERENCES

Abramowitz, M. & Stegun, I.A., 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,* Dover publications, Inc.

Bohlen, T. & Saenger, E.H., 2006. Accuracy of heterogeneous staggered-grid finite-difference modeling of Rayleigh waves, *Geophysics,* **71,** 109–115.

Capdeville, Y., Gung, Y. & Romanowicz, B., 2005. Towards global earth tomography using the spectral element method: a technique based on source stacking, *Geophys. J. Int.,* **162,** 541–554.

Castellanos, C., Métivier, L., Operto, S., Brossier, R. & Virieux, J., 2015. Fast full waveform inversion with source encoding and second-order optimization methods, *Geophys. J. Int.,* **200**(2), 720–744.

Claerbout, J.F., 1992. *Earth Souding Analysis,* Blackwell Scientific Publications.

Deriche, R., 1992. Recursively implementing the gaussian and its derivatives, in *Proceedings of the 2nd International Conference on Image Processing,* pp. 263–267, Singapore.

Deville, M., Fischer, P. & Mund, E., 2002. *High Order Methods for Incompressible Fluid Flow,* Cambridge Univ. Press.

Fehmers, G.C. & Höcker, C.F.W., 2003. Fast structural interpretation with structure-oriented filtering, *Geophysics,* **68**(4), 1286–1293.

Fomel, S., 2002. Applications of plane-wave destruction filters, *Geophysics,* **67**(6), 1946–1960.

Fomel, S. & Claerbout, J.F., 2003. Mutlidimensional recursive filter preconditioning in geophysical estimation problems, *Geophysics,* **68**(2), 1–12.

Fuji, N., Ovcharenko, O., Martin, R. & Ovcharenko, O., 2016. Simple and accurate operators based on Taylor expansion for 2D elastic seismogram calculation under geological discontinuities with regular Cartesian grids, in *Methods and Challenges of Seismic Wave Modelling for Seismic Imaging Workshop, 78th Annual EAGE Meeting (Vienna)*, doi:10.3997/2214-4609.201601665.

Golub, G.H., 1996. *Matrix Computation* 3rd edn, Johns Hopkins Studies in Mathematical Sciences.

Guitton, A., Ayeni, G. & Díaz, E., 2012. Constrained full-waveform inversion by model reparameterization, *Geophysics,* **77**(2), R117–R127.

Hale, D., 2007. Local dip filtering with directional Laplacians, Tech. rep., Centre for Wave Phenomena, Colorado School of Mines.

Herrmann, F.J., Erlangga, Y.A. & Lin, T.T.Y., 2009. Compressive simultaneous full-waveform simulation, *Geophysics,* **74(4),** A35–A40.

Huiskes, M., Plessix, R. & Mulder, W., 2016. A fast3D free-surface topography method for acoustic full-waveform inversion, in *Methods and Challenges of Seismic Wave Modelling for Seismic Imaging Workshop, 78th Annual EAGE Meeting (Vienna)*, doi:10.3997/2214-4609.201601664.

Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral element method for three-dimensional seismic wave propagation, *Geophys. J. Int.,* **139**(3), 806–822.

Krebs, J., Anderson, J., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A. & Lacasse, M.D., 2009. Fast full-wavefield seismic inversion using encoded sources, *Geophysics,* **74(6),** WCC105–WCC116.

Menke, W., 1984. *Geophysical Data Analysis: Discrete Inverse Theory,* Academic Press, Inc.

Oristaglio, M., 2012. Seam phase II: Land seismic challenges, *Leading Edge,* **31**(3), 264–266.

Peter, D. *et al.*, 2011. Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes, *Geophys. J. Int.,* **186**(2), 721–739.

Plessix, R.E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, *Geophys. J. Int.,* **167**(2), 495–503.

Polyanin, A.D. & Nazaikinskii, V.E., 2002. *Handbook of Linear Partial Differential Equations for Engineers and Scientists,* CRC Press.

Robertsson, J.O.A., 1996. A numerical free-surface condition for elastic/viscoelastic finite-difference modeling in the presence of topography, *Geophysics,* **61,** 1921–1934.

Saad, Y., 2003. *Iterative Methods for Sparse Linear Systems,* SIAM.

Sheriff, R.E., 2002. Encyclopedic dictionary of applied geophysics, in *Geophysical References Series,* ed. Scherrer, E.F., Society of Exploration Geophysicists.

Tape, C., Liu, Q., Maggi, A. & Tromp, J., 2010. Seismic tomography of the southern California crust based on spectral-element and adjoint methods, *Geophys. J. Int.,* **180,** 433–462.

Tarantola, A., 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation,* SIAM.

Van Vliet, L. J., Young, I.T. & Verbeek, P.W., 1998. Recursive gaussian derivative filters, in *14th International Conference on Pattern Recognition,* vol. 1, pp. 509–514, IEEE.

Virieux, J. & Operto, S., 2009. An overview of full waveform inversion in exploration geophysics, *Geophysics,* **74**(6), WCC1–WCC26.

Warner, M. *et al.*, 2013. Anisotropic 3D full-waveform inversion, *Geophysics,* **78**(2), R59–R80.

Wellington, P., 2016. Efficient 1D, 2D and 3D geostatistical constraints and their application to full waveform inversion, *PhD thesis,* Univ. Grenoble Alpes.

Williamson, P., Atle, A., Fei, W. & Hale, D., 2011. Regularization of wave-equation migration velocity analysis by structure-oriented smoothing, in *2011 SEG Annual Meeting,* Society of Exploration Geophysicists.

APPENDIX A: MODIFIED BESSEL FUNCTIONS AND THEIR INTEGRAL REPRESENTATION

In this Appendix, we provide the definition of the modified Bessel function of the second kind $K_\nu(x)$ for $\nu \in \mathbf{R}$. This function is defined from the modified Bessel function of the first kind $I_\nu(x)$ (Abramowitz & Stegun 1972), written in the power series expansion

$$I_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{\Gamma(k+1)\Gamma(\nu+k+1)} \left(\frac{x}{2}\right)^{2k}, \tag{A1}$$

where the gamma function $\Gamma(x)$ is defined as

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t. \tag{A2}$$

When ν is an integer, we have the following property of the function $I_\nu(x)$

$$I_\nu(x) = I_{-\nu}(x). \tag{A3}$$

According to the definition of the function $I_\nu(x)$ in eq. (A1), we describe how it behaves when $x \to 0$:

$$\lim_{x \to 0} I_\nu(x) = \begin{cases} 0 & Re(\nu) > 0 \\ 1 & \nu = 0 \\ \pm\infty & Re(\nu) < 0 \end{cases}. \tag{A4}$$

This implies that $I_\nu(x)$ and $I_{-\nu(x)}$ are linearly independent if ν is not an integer. The modified Bessel function of the second kind $K_\nu(x)$ is then defined as

$$K_\nu(x) = \frac{\pi}{2} \frac{I_{-\nu} - I_\nu}{\sin \pi \nu}, \tag{A5}$$

for $\nu \notin \mathbf{Z}$. For $n \in \mathbf{Z}$, the function $K_n(x)$ is defined as

$$K_n(x) := \lim_{\nu \to n} K_\nu(x). \tag{A6}$$

APPENDIX B: DEFINITION OF BESSEL FILTERS AND THEIR NORMALIZED FACTORS

In this Appendix, we demonstrate that 2-D and 3-D Bessel filters are unique solutions of the PDEs (8) and (9), respectively, when considering the radiative boundary condition

$$B_{nD}(\mathbf{x}) \to 0 \quad \text{as} \quad \|\mathbf{x}\| \to \infty \tag{B1}$$

and the boundness property

$$0 < \int B_{nD}(\mathbf{x})d\mathbf{x} < \infty \tag{B2}$$

B1 The 2-D case

We look for all possible analytical solutions of eq. (9):

$$C(z, x) - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} \right) C(z, x) = \delta(z, x). \tag{B3}$$

Changing from Cartesian to dimensionless coordinates

$$x \to \tilde{x} = \frac{x}{L_x}; \quad z \to \tilde{z} = \frac{z}{L_z}, \tag{B4}$$

knowing that L_x and L_z are homogeneous, we obtain

$$C(\tilde{z}, \tilde{x}) - \left(\frac{\partial^2}{\partial \tilde{x}^2} C(\tilde{z}, \tilde{x}) + \frac{\partial^2}{\partial \tilde{z}^2} C(\tilde{z}, \tilde{x}) \right) = \delta(\tilde{z}, \tilde{x}). \tag{B5}$$

We first solve the homogeneous form of eq. (B5) for $(\tilde{z}, \tilde{x}) \neq (0, 0)$. Again, we move from dimensionless coordinates $(\tilde{z}, \tilde{x})$ to polar coordinates (r, θ), and assume that the solution is separated; that is, $C(r, \theta) = R(r)A(\theta)$. The homogeneous form of eq. (B5) now becomes

$$R(r)A(\theta) - \left(R''(r)A(\theta) + \frac{1}{r} R'(r)A(\theta) + \frac{1}{r^2} R(r)A''(\theta) \right) = 0. \tag{B6}$$

Dividing both sides of this equation by $R(r)A(\theta)r^2$:

$$\frac{r^2 R''(r)}{R(r)} + \frac{r R'(r)}{R(r)} - r^2 = -\frac{A''(\theta)}{A(\theta)}. \tag{B7}$$

As $R(r)$ and $A(\theta)$ are independent, eq. (B7) can be set as a constant V, for $r > 0$. Note that valid solutions must satisfy the periodicity in the angular domain; that is, $A(\theta) = A(\theta + 2\pi)$, which leads to a non-negative value of V.

By considering the angular part of the solution:

$$\frac{A''(\theta)}{A(\theta)} = -V \iff A''(\theta) + V.A(\theta) = 0. \tag{B8}$$

Two independent real solutions of this equation are

$$A_1(\theta) = \cos(\sqrt{V}\theta); \quad A_2(\theta) = \sin(\sqrt{V}\theta). \tag{B9}$$

As $A(\theta) = A(\theta + 2\pi)$, V must satisfy $V = \nu^2$, where ν is an integer. The radial part now becomes

$$r^2 R''(r) + r R'(r) - (r^2 + \nu^2)R(r) = 0, \tag{B10}$$

which is actually the modified Bessel differential equation (Polyanin & Nazaikinskii 2002; Abramowitz & Stegun 1972). Two linearly independent solutions of this equation are the modified Bessel function of the first kind $I_\nu(r)$, singular at ∞, and the second kind $K_\nu(r)$, singular at zero, where ν is a real number.

As we are interested in building a smoothing filter, it must vanish at infinity (a condition of eq. B1):

$$R(r) \to 0 \quad \text{as} \quad r \to \infty, \tag{B11}$$

which implies that the modified Bessel function of the first kind I_ν can be excluded, as

$$\lim_{r \to +\infty} I_\nu(r) = +\infty. \tag{B12}$$

In the polar coordinates (r, θ), the condition of eq. (B2) can be re-written as

$$0 < \int_{-\pi}^{\pi} d\theta \int_0^\infty r dr C(r, \theta) < \infty. \tag{B13}$$

As our solution has the form $C(r, \theta) = R(r)A(\theta)$, this condition becomes

$$0 < \int_{-\pi}^{\pi} d\theta A(\theta) \int_0^\infty r dr R(r) < \infty. \tag{B14}$$

The angular part of the forward filter has the form

$$A_\nu(\theta) = c_1 \cos(\nu\theta) + c_2 \sin(\nu\theta) \text{ where } \nu = 0, \pm 1, \pm 2... \tag{B15}$$

and c_1, c_2 are constant, which leads to

$$\int_{-\pi}^{\pi} d\theta\, A_\nu(\theta) = \begin{cases} c_1 \sin(\nu\theta)\big|_{-\pi}^{\pi} - c_2 \cos(\nu\theta)\big|_{-\pi}^{\pi} = 0 & \text{if } \nu = \pm 1, \pm 2... \\ 2\pi(c_1 + c_2) = \text{Cst} > 0 & \text{if } \nu = 0. \end{cases} \tag{B16}$$

Consequently, the modified Bessel function of the second kind K_ν with $\nu > 0$ cannot be our expected filter. In addition,

$$\int_0^\infty r K_0(r) dr = 1, \tag{B17}$$

which clearly satisfies the condition

$$0 < \int_{-\pi}^{\pi} d\theta\, A_0(\theta) \int_0^\infty r dr\, K_0(r) < \infty. \tag{B18}$$

The boundary conditions at infinity imply that K_0 is the only possible solution of the homogeneous form of eq. (B5). As the function $K_0(r)$ is singular at zero, the normalized function $\frac{1}{2\pi L_x L_z} K_0\left(\sqrt{\frac{x^2}{L_x^2} + \frac{z^2}{L_z^2}}\right)$ is also the solution of the non-homogeneous eq. (B3).

B2 The 3-D case

Similar to the 2-D case, we look for all possible solution of eq. (8):

$$C(z, x, y) - \left(L_z^2 \frac{\partial^2}{\partial z^2} + L_x^2 \frac{\partial^2}{\partial x^2} + L_y^2 \frac{\partial^2}{\partial y^2} \right) C(z, x, y) = \delta(z, x, y). \tag{B19}$$

By repeating the same workflow as for the 2-D case, moving from Cartesian coordinates to spherical coordinates, and assuming again that the solution is separated $C(r, \theta, \phi) = R(r)A(\theta)B(\phi)$, where $0 < r < \infty$, $0 \le \theta \le \pi$, and $-\pi \le \phi \le \pi$, the homogeneous form of eq. (B19) becomes

$$R(r)A(\theta)B(\phi) - \left(\frac{\partial^2 R(r)}{\partial r^2} + \frac{2}{r} \frac{\partial R(r)}{\partial r} \right) A(\theta)B(\phi) - \frac{1}{r^2} \left(\frac{\cos\theta}{\sin\theta} \frac{\partial A(\theta)}{\partial\theta} + \frac{\partial^2 A(\theta)}{\partial\theta^2} \right) R(r)B(\phi)$$

$$- \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 B(\phi)}{\partial\phi^2} R(r)A(\theta) = 0. \tag{B20}$$

Multiplying both sides of this equation by $\frac{r^2 \sin\theta}{R(r)A(\theta)B(\phi)}$, assuming it does not vanish, we obtain

$$\sin^2\theta \left(r^2 - r^2 \frac{R''(r)}{R(r)} - 2r \frac{R'(r)}{R(r)} \right) - \left(\cos\theta \sin\theta \frac{A'(\theta)}{A(\theta)} + \sin^2\theta \frac{A''(\theta)}{A(\theta)} \right) - \frac{B''(\phi)}{B(\phi)} = 0, \tag{B21}$$

which should be valid for all different sets of values of (r, θ, ϕ). By choosing $\theta = 0$, eq. (B21) should hold for any values of r and ϕ, leading to

$$\frac{B''(\phi)}{B(\phi)} = 0 \implies B(\phi) = \text{Constant} \tag{B22}$$

(because this function is periodic $B(\phi) = B(\phi + 2\pi)$). Replacing this expression back into eq. (B21), and dividing both sides of the equation by $\sin\theta$, we obtain

$$\sin\theta \left(r^2 - r^2 \frac{R''(r)}{R(r)} - 2r \frac{R'(r)}{R(r)} \right) - \left(\cos\theta \frac{A'(\theta)}{A(\theta)} + \sin\theta \frac{A''(\theta)}{A(\theta)} \right) = 0. \tag{B23}$$

Again, this equation should hold for any values of r and $\theta = 0$, which leads to

$$\frac{A'(\theta)}{A(\theta)} = 0 \implies A(\theta) = \text{Constant}. \tag{B24}$$

Replacing this expression back into eq. (B23), we get

$$r^2 R''(r) + 2r R'(r) - r^2 R(r) = 0. \tag{B25}$$

According to Abramowitz & Stegun (1972), this is the modified spherical Bessel function, which has two particular solutions:

$$\sqrt{\frac{1}{r}} I_{\pm 1/2}(r) \quad \text{and} \quad \sqrt{\frac{1}{r}} K_{1/2}(r). \tag{B26}$$

Again, we expect that the solution vanishes at infinity, which means that $r^{-1/2} I_{\pm 1/2}(r)$ can be excluded. The function $r^{-1/2} K_{1/2}(r)$ satisfies the normalizable condition of eq. (B2) in 3-D as

$$\int_{-\pi}^{\pi} d\phi \int_0^\pi \sin\theta d\theta \int_0^\infty r^2 \left[r^{-1/2} K_{1/2}(r) \right] dr = (2\pi)^{3/2}. \tag{B27}$$

Again, as the function $K_0(r)$ is singular at zero, the normalized function $\frac{1}{(2\pi)^{3/2} L_x L_y L_z} r^{-1/2} K_{1/2}(r)$ is also the solution of the non-homogeneous eq. (B19).

APPENDIX C: WEAK FORMULATION DEVELOPMENT

This Appendix clarifies the definition of the mesh used in SEM, which consists of Gauss–Lobatto–Legendre (GLL) points, defined in the reference space. The base functions are Lagrange polynomials. GLL quadrature is used to approximate analytical integrals over collocation points. Let us recall the weak formulation of eq. (26)

$$\iiint_{\tilde{\Omega}} \mathbf{s}\, v \, d\tilde{z}d\tilde{x}d\tilde{y} + \iiint_{\tilde{\Omega}} \nabla_{\tilde{z},\tilde{x},\tilde{y}}\mathbf{s}\, \nabla_{\tilde{z},\tilde{x},\tilde{y}}v \, d\tilde{z}d\tilde{x}d\tilde{y} = \iiint_{\tilde{\Omega}} \mathbf{m}\, v \, d\tilde{z}d\tilde{x}d\tilde{y}, \tag{C1}$$

where $v(\tilde{z}, \tilde{x}, \tilde{y})$ is a test function, $\mathbf{m}$ and $\mathbf{s}$ are the original and the smoothed vectors, respectively.

In SEM, the physical domain Ω is decomposed into a set of non-overlapping hexahedral elements Ω_e. This implies that the dimensionless space $\tilde{\Omega}$ is also represented by a set of elements $\tilde{\Omega}_e$, which is related to the physical element Ω_e through the one-to-one mapping (eq. 24). Each reduced coordinate element $\tilde{\Omega}_e$ can be mapped to the unitary reference space of the GLL points, where the cube $[-1, 1] \otimes [-1, 1] \otimes [-1, 1]$ is discretized into $(N + 1) \times (N + 1) \times (N + 1)$ GLL points $(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3})$; $k_1, k_2, k_3 = 0, \ldots, N$. These collocation points define $(N + 1) \times (N + 1) \times (N + 1)$ base functions, which are triple products of Lagrange polynomials of degree N, over the element $\tilde{\Omega}_e$

$$v_{\hat{k}}(\xi, \eta, \zeta) = \ell_{k_1}(\xi)\ell_{k_2}(\eta)\ell_{k_3}(\zeta) \quad \hat{k} \text{ stands for the triple of indexes } \{k_1, k_2, k_3\}. \tag{C2}$$

Lagrange polynomials have the interesting property that their values at GLL nodes are either 0 or 1

$$\ell_j(\xi) = \prod_{\substack{i=0 \\ i \neq j}}^{N} \frac{\xi - \xi_i}{\xi_j - \xi_i}; \quad \ell_j(\xi_i) = \delta_{ji}. \tag{C3}$$

An element of volume $d\tilde{z}d\tilde{x}d\tilde{y}$ is related to an element of volume $d\xi d\eta d\zeta$ in the reference cube by

$$d\tilde{z}d\tilde{x}d\tilde{y} = \tilde{J}_e d\xi d\eta d\zeta, \tag{C4}$$

where the volumetric Jacobian $\tilde{J}_e$ is the determinant of the Jacobian matrix $\tilde{\mathbf{J}}$

$$\tilde{\mathbf{J}} = \begin{bmatrix} \partial_\xi \tilde{z} & \partial_\eta \tilde{z} & \partial_\zeta \tilde{z} \\ \partial_\xi \tilde{x} & \partial_\eta \tilde{x} & \partial_\zeta \tilde{x} \\ \partial_\xi \tilde{y} & \partial_\eta \tilde{y} & \partial_\zeta \tilde{y} \end{bmatrix}. \tag{C5}$$

We introduce the following notations of the coordinates of the dimensionless physical space and the reference space

$$(p_1, p_2, p_3) := (\tilde{z}, \tilde{x}, \tilde{y}) \quad \text{and} \quad (r_1, r_2, r_3) := (\xi, \eta, \zeta), \tag{C6}$$

to develop the weak formulation in eq. (C1) as

$$\iiint_{\tilde{\Omega}} \mathbf{s}\, v_{\hat{k}} d\tilde{z}d\tilde{x}d\tilde{y} + \iiint_{\tilde{\Omega}} \sum_{d=1}^{3} \frac{\partial v_{\hat{k}}}{\partial p_d} \frac{\partial \mathbf{s}}{\partial p_d} d\tilde{z}d\tilde{x}d\tilde{y} = \iiint_{\tilde{\Omega}} \mathbf{m}\, v_{\hat{k}} d\tilde{z}d\tilde{x}d\tilde{y}. \tag{C7}$$

When moving to the reference space, according to the chain rule, this equation becomes

$$\iiint_{-1}^{1} \mathbf{s}\, v_{\hat{k}}\tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta + \iiint_{-1}^{1} \sum_{d=1}^{3} \left[\sum_{i=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} \frac{\partial r_i}{\partial p_d} \right] \left[\sum_{j=1}^{3} \frac{\partial \mathbf{s}}{\partial r_j} \frac{\partial r_j}{\partial p_d} \right] \tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta = \iiint_{-1}^{1} \mathbf{m}\, v_{\hat{k}}\tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta. \tag{C8}$$

or

$$\iiint_{-1}^{1} \mathbf{s}\, v_{\hat{k}}\tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta + \iiint_{-1}^{1} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} \left[\sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \right] \frac{\partial \mathbf{s}}{\partial r_j} \tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta = \iiint_{-1}^{1} \mathbf{m}\, v_{\hat{k}}\tilde{J}_e(\xi, \eta, \zeta) d\xi d\eta d\zeta. \tag{C9}$$

In the following step, we use the approximation of a field $\mathbf{u}$ over the element $[-1, 1] \otimes [-1, 1] \otimes [-1, 1]$, as follows

$$\mathbf{u}(\xi, \eta, \zeta) \approx \sum_{h_1=0}^{N} \sum_{h_2=0}^{N} \sum_{h_3=0}^{N} u_{\hat{h}}\ell_{h_1}(\xi)\ell_{h_2}(\eta)\ell_{h_2}(\zeta) =: \sum_{\hat{h}=0}^{N} u_{\hat{h}}v_{\hat{h}}(\xi, \eta, \zeta), \tag{C10}$$

where $u_{\hat{h}} = \mathbf{u}(\xi_{h_1}, \eta_{h_2}, \zeta_{h_3})$ for vectors $\mathbf{m}$ and $\mathbf{s}$. Therefore, eq. (C9) becomes

$$\sum_{\hat{h}=0}^{N} \iiint_{-1}^{1} v_{\hat{k}} v_{\hat{h}} s_{\hat{h}} \tilde{J}_e \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta + \sum_{\hat{h}=0}^{N} \iiint_{-1}^{1} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} \left[\sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \right] \frac{\partial v_{\hat{h}}}{\partial r_j} s_{\hat{h}} \tilde{J}_e \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta = \sum_{\hat{h}=0}^{N} \iiint_{-1}^{1} v_{\hat{k}} v_{\hat{h}} m_{\hat{h}} \tilde{J}_e \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta. \tag{C11}$$

The integral over the reference cube can be approximated by the GLL quadrature

$$\iiint_{-1}^{1} f(\xi, \eta, \zeta) \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \approx \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} f(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}), \tag{C12}$$

where w_{q_1}, w_{q_2}, w_{q_3} are quadrature weights. As the test function $v_{\hat{k}}$ is the triple product of Lagrange polynomials (eq. C2), which is 1 at node $(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3})$, and 0 at other nodes (eq. C3), the weak formulation developed in eq. (C11) can be transformed into a linear system

$$\mathbf{Ms} + \mathbf{Ks} = \mathbf{Mm}. \tag{C13}$$

The mass matrix $\mathbf{M}$ is diagonal with

$$M_{\hat{k}\hat{k}} = w_{k_1} w_{k_2} w_{k_3} \tilde{J}_e(\xi_{k_1}, \eta_{k_2}, \zeta_{k_3}) \tag{C14}$$

and the stiffness matrix $\mathbf{K}$ is symmetric and sparse

$$K_{\hat{k}\hat{h}} = \sum_{\hat{q}=0}^{N} w_{q_1} w_{q_2} w_{q_3} \tilde{J}_e(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}) \left[\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial v_{\hat{k}}}{\partial r_i} \left(\sum_{d=1}^{3} \frac{\partial r_i}{\partial p_d} \frac{\partial r_j}{\partial p_d} \right) \frac{\partial v_{\hat{h}}}{\partial r_j} \right]. \tag{C15}$$

The property mentioned in eq. (C3) of the test functions can be used again to simplified the spatial derivatives of the test functions as

$$\frac{\partial v_{\hat{k}}}{\partial \xi}(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3}) = \ell'_{k_1}(\xi_{q_1}) \delta_{k_2, q_2} \delta_{k_3, q_3}, \tag{C16}$$

similar expressions can be written for $\partial v_{\hat{k}}/\partial \eta$ and $\partial v_{\hat{k}}/\partial \zeta$ at $(\xi_{q_1}, \eta_{q_2}, \zeta_{q_3})$. Therefore, the element at row $\hat{k}$th, column $\hat{h}$th of the matrix $\mathbf{K}$ can be explicitly computed as

$$\begin{aligned}
K_{\hat{k}\hat{h}} =\ & \sum_{q_1=0}^{N} w_{q_1} w_{k_2} w_{k_3} J_e(\xi_{q_1}, \eta_{k_2}, \zeta_{k_3}) \ell'_{h_1}(\xi_{q_1}) \ell'_{k_1}(\xi_{q_1}) \left[\sum_{d=1}^{3} (\partial_{p_d} \xi(\xi_{q_1}, \eta_{k_2}, \zeta_{k_3})^2 \right] \\
& + \sum_{q_2=0}^{N} w_{k_1} w_{q_2} w_{k_3} J_e(\xi_{k_1}, \eta_{q_2}, \zeta_{k_3}) \ell'_{h_2}(\eta_{q_2}) \ell'_{k_2}(\eta_{q_2}) \left[\sum_{d=1}^{3} (\partial_{p_d} \eta(\xi_{k_1}, \eta_{q_2}, \zeta_{k_3}))^2 \right] \\
& + \sum_{q_3=0}^{N} w_{k_1} w_{k_2} w_{q_3} J_e(\xi_{k_1}, \eta_{k_2}, \zeta_{q_3}) \ell'_{h_3}(\zeta_{q_3}) \ell'_{k_3}(\zeta_{q_3}) \left[\sum_{d=1}^{3} (\partial_{p_d} \zeta(\xi_{k_1}, \eta_{k_2}, \zeta_{q_3}))^2 \right] \\
& + w_{k_1} w_{h_2} w_{k_3} J_e(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \ell'_{h_1}(\xi_{k_1}) \ell'_{k_2}(\eta_{h_2}) \left[\sum_{d=1}^{3} \partial_{p_d} \xi(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \partial_{p_d} \eta(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \right] \\
& + w_{h_1} w_{k_2} w_{k_3} J_e(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \ell'_{h_2}(\eta_{k_2}) \ell'_{k_1}(\xi_{h_1}) \left[\sum_{d=1}^{3} \partial_{p_d} \xi(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \partial_{p_d} \eta(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \right] \\
& + w_{k_1} w_{k_2} w_{h_3} J_e(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \ell'_{h_1}(\xi_{k_1}) \ell'_{k_3}(\zeta_{h_3}) \left[\sum_{d=1}^{3} \partial_{p_d} \xi(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \partial_{p_d} \zeta(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \right] \\
& + w_{h_1} w_{k_2} w_{k_3} J_e(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \ell'_{h_3}(\zeta_{k_3}) \ell'_{k_1}(\xi_{h_1}) \left[\sum_{d=1}^{3} \partial_{p_d} \xi(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \partial_{p_d} \zeta(\xi_{h_1}, \eta_{k_2}, \zeta_{k_3}) \right] \\
& + w_{k_1} w_{k_2} w_{h_3} J_e(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \ell'_{h_2}(\eta_{k_2}) \ell'_{k_3}(\zeta_{h_3}) \left[\sum_{d=1}^{3} \partial_{p_d} \eta(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \partial_{p_d} \zeta(\xi_{k_1}, \eta_{k_2}, \zeta_{h_3}) \right] \\
& + w_{k_1} w_{h_2} w_{k_3} J_e(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \ell'_{h_3}(\zeta_{k_3}) \ell'_{k_2}(\eta_{h_2}) \left[\sum_{d=1}^{3} \partial_{p_d} \eta(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \partial_{p_d} \zeta(\xi_{k_1}, \eta_{h_2}, \zeta_{k_3}) \right].
\end{aligned} \tag{C17}$$