GEO-2015-0653

Wavefield Compression for Adjoint Methods in Full-Waveform Inversion

Christian Boehm, Mauricio Hanzich†, Josep de la Puente†, Andreas Fichtner**

ABSTRACT

Adjoint methods are a key ingredient of gradient-based full-waveform inversion schemes. While being conceptually elegant, they face the challenge of massive memory requirements caused by the opposite time directions of forward and adjoint simulations and the necessity to access both wavefields simultaneously for the computation of the sensitivity kernel. To overcome this bottleneck, we present lossy compression techniques that significantly reduce the memory requirements with only a small computational overhead. Our approach is tailored to adjoint methods and utilizes the fact that the computation of a sufficiently accurate sensitivity kernel does not require the fully-resolved forward wavefield. The collection of methods comprises re-interpolation with a coarse temporal grid as well as adaptively chosen polynomial degree and floating-point precision to represent spatial snapshots of the forward wavefield on hierarchical grids. Furthermore, the first arrivals of adjoint waves are used to identify "shadow zones" that do not contribute to the sensitivity kernel. Numerical experiments show the high potential of this approach achieving an effective compression factor of three orders of magnitude with only a minor reduction in the rate of convergence. Moreover, it is computationally cheap and straightforward to integrate in finite-element wave propagation codes with possible extensions to finite-difference methods.

INTRODUCTION

Adjoint methods offer a powerful tool to solve problems in full-waveform inversion (FWI). They are essential to efficiently compute the first and second derivatives of the misfit functional and have been applied successfully to solve tomography problems on many different scales, see, for instance, Igel et al. (1996); Pratt et al. (1998); Fichtner et al. (2009); Virieux and Operto (2009); Tape et al. (2010); Zhu et al. (2012); Rickers et al. (2013); Métivier et al. (2013). However, the time-reversed nature of the adjoint simulations introduces a severe bottleneck by the necessity to access forward and adjoint wavefields simultaneously during the computation of sensitivity kernels. Hence, in addition to requiring a huge amount of computational resources, FWI on 3D data sets also demands massive storage capabilities.

In general, there exist two opposing strategies to deal with this challenge that show a tradeoff between memory requirements and increasing the computational cost. On the one hand, saving the whole forward wavefield to the disk yields a significant overhead to store and process the four-dimensional space-time evolution of the seismic wavefield. Moreover, since this can easily exceed tens or hundreds of terabytes, it necessitates to use the slow levels in memory hierarchy and limitations of the memory bandwidth degrade the performance significantly, especially on GPU clusters. On the other hand, checkpointing techniques (Griewank, 1992; Walther and Griewank, 2004) can trade an almost arbitrary amount of memory requirements for additional computations. While the optimal placement of the checkpoints has been studied and fine-tuned to its application in reverse time migration and seismic tomography (Symes, 2007; Anderson et al., 2012), this approach yields a significant computational overhead, which is typically in the order of one additional forward simulation and can only be reduced at the cost of increasing the memory requirements. A related strategy was proposed by Tromp et al. (2008) and requires to store only the boundary values and the final state of the forward wavefield. In a second step, an auxiliary PDE - the so-called backward wave equation - is solved during the adjoint run to propagate the saved forward wavefield in reversed time direction. While its computational overhead is comparable to checkpointing techniques, this approach is limited to symmetric time stepping schemes and when attenuation in the Earth is weak.

In summary, storing the whole forward wavefield is prohibitively expensive for large data sets and potential remedies like checkpointing introduce a significant computational overhead. Hence, the amount of auxiliary data that has to be transferred to and from disk and related input/output (I/O) operations are key challenges for full-waveform inversion.

In this paper we propose an alternative approach that strikes a balance between memory requirements and the need for additional computations. This is based on techniques for the lossy compression of the forward wavefield that fulfill the following important requirements:

- The error in the inexact gradients can be controlled such that the lossy compression does not significantly affect the inverted results.
- The I/O overhead is reduced substantially without the need for extra simulations.
- The computational cost for compressing/decompressing the wavefield is cheap compared to solving the elastic wave equation.
- Using approximate sensitivity kernels resulting from the compressed wavefield does not significantly slow down the rate of convergence of an iterative minimization scheme to solve the inverse problem.

Our method combines (i) storing coarse temporal snapshots and re-interpolation with a sliding window cubic spline, (ii) spatial compression based on an adaptively chosen floating-point precision and local representation of the wavefield by lower-dimensional polynomials on hierarchical grids and (iii) identification of "shadow zones", where forward and adjoint wavefield do not overlap.

Related ideas have been proposed in previous work by Unat et al. (2009); Weiser and Götschel (2012); Hanzich et al. (2013) and Götschel and Weiser (2014), achieving compression factors between 8 and 213 for different applications. However, we can obtain significantly higher compression factors by tailoring the methods to the computation of sensitivity kernels in high-order finite-element methods for time-domain FWI.

Alternative compression strategies based on the discrete cosine transform have been proposed and studied by Hanzich et al. (2014) and Unat et al. (2009). A rigorous comparison with wavelet-based compression methods is beyond the scope of this paper. Intermediate results indicate, however, that wavelet-based methods tend to achieve a slightly higher compression factor given a fixed accuracy. On the other hand, the methods presented in this paper have a significantly smaller computational overhead for compression and decompression, which makes them a particularly favorable alternative to checkpointing techniques.

This paper is organized as follows. First, we develop error estimates for inexact sensitivity kernels that result from a (de-)compressed forward wavefield. Then we describe the compression techniques in detail and conclude with numerical examples in the last section.

ITERATIVE INVERSION SCHEMES WITH INEXACT GRADIENTS

In this section we briefly recall nonlinear minimization methods to solve inverse problems and introduce important notation. A classical Newton-type line-search method iteratively updates the model m by setting

$$m^{k+1} = m^k + \alpha_k s^k, \tag{1}$$

with a search direction s^k, along which the misfit decreases, and a suitable step-length $\alpha_k > 0$, until the sequence m^k converged to a (local) minimum. s^k is obtained by minimizing a quadratic approximation of the misfit functional χ around the current model m^k, which gives

$$s^k = -B_k g^k, \tag{2}$$

where g^k denotes the gradient of χ evaluated at m^k and B_k is a positive definite matrix that approximates the inverse Hessian of χ at m^k. Typical choices for B_k are given by the BFGS or L-BFGS method (Nocedal and Wright, 2006) and for the particular case of using the identity operator as B_k, we obtain the steepest descent method.

In order to compute search directions to update the model, we require the Fréchet derivative of χ with respect to m. This sensitivity can be efficiently computed using adjoints. Here, it is important to note that all derivatives with respect to different structural parameters like density, bulk or shear moduli, have a similar structure that involves multiplication of forward and adjoint velocity fields or strains, respectively, and integration in time. For instance, in case of the elastic wave equation for isotropic media, the Fréchet derivatives with respect to density ρ and shear modulus μ are given by

$$K_\rho(\mathbf{x}) = -\int_0^T u_t(\mathbf{x},t) \cdot u_t^\dagger(\mathbf{x},t)\, dt, \tag{3}$$

$$K_\mu(\mathbf{x}) = \int_0^T \varepsilon(u)(\mathbf{x},t) : \varepsilon(u^\dagger)(\mathbf{x},t)\, dt, \tag{4}$$

where $u^\dagger$ is the adjoint wavefield and $\varepsilon(u)$ and $\varepsilon(u^\dagger)$ denote the strain tensors of forward and adjoint field, respectively. More generally, the Fréchet derivatives with respect to structural parameters have the following form

$$K(\mathbf{x}) = \int_0^T (\mathcal{D}\, u)(\mathbf{x},t) \cdot (\mathcal{D}\, u^\dagger)(\mathbf{x},t)\, dt, \tag{5}$$

where $\mathcal{D}$ is a first-order differential operator that extracts spatial or temporal derivatives of the wavefield. Transforming K to the model space then yields the gradient g. For further examples see, e.g., Fichtner (2011) or Tromp et al. (2005). As mentioned before, the opposite time directions of forward and adjoint simulations cause the main challenge of computing the Fréchet derivatives.

In what follows, the key idea is to compress $\mathcal{D}u$ during the forward simulation and to decompress it again during the adjoint run, i.e., we replace $\mathcal{D}u$ in equation 5 by an approximate field $\widetilde{\mathcal{D}u} \approx \mathcal{D}u$. This yields an approximate gradient $\tilde{g}^k$ and - in case of a quasi-Newton method - also a modified $\tilde{B}_k$ in equation 2. Although only the forward wavefield is compressed, all derived quantities that involve $\widetilde{\mathcal{D}u}$ will be denoted with $\tilde{\cdot}$ to indicate the inexactness.

Note that we substitute $\widetilde{\mathcal{D}u}$ for $\mathcal{D}u$ only during the computation of the sensitivity kernel, i.e., after the forward run. Hence, the compression neither affects the calculation of the misfits nor the adjoint sources. This implies, in particular, that also the adjoint state $u^\dagger$ has the full accuracy and does not introduce an additional compression error.

As pointed out above, we aim for a good approximation of the sensitivity kernel rather than an accurate reconstruction of the forward wavefield itself. In the next step, we show that by carefully controlling the approximation error, an inexactly computed sensitivity kernel is sufficient to ensure convergence of an iterative minimization method to solve the inverse problem. Therefore, we have to ensure that $\tilde{g}^k \approx g^k$ and also $\tilde{s}^k \approx s^k$. In particular, we require that all $\tilde{s}^k$ are directions along which χ decreases. Using the structure of the sensitivity kernels as indicated in equation 5, we can bound the absolute error in the gradient for each structural parameter field by the estimate

$$\|\tilde{g} - g\|_{L^\infty} \leq const \cdot \int_0^T \|(\widetilde{\mathcal{D}u})(\mathbf{x}, t) - (\mathcal{D}\,u)(\mathbf{x}, t)\|_{L^2} \cdot \|(\mathcal{D}\,u^\dagger)(\mathbf{x}, t)\|_{L^2}\, dt. \tag{6}$$

Here, the particular choice of the norm depends on the finite-element discretization and would typically correspond to some discrete Sobolev space norm, see Boehm and Ulbrich (2015) for details. From inequality 6 we derive that the error in the gradient is bounded by the approximation error of the compressed forward wavefield $\|(\widetilde{\mathcal{D}u})(\mathbf{x}, t) - (\mathcal{D}\,u)(\mathbf{x}, t)\|$. Controlling the latter at the time of compression is key to our method. An estimate of the adjoint state would additionally allow for time-dependent compression thresholds, but such estimate is not yet available during the forward simulation.

Intuitively, we would expect similar model updates if the error $\|\tilde{g} - g\|$ is small. More precisely, we can ensure convergence of the iterative minimization scheme if the inexactly computed search directions $\tilde{s}^k$ satisfy the so-called angle condition, i.e.,

$$(g^k, \tilde{s}^k) \leq -\beta\, \|g^k\| \cdot \|\tilde{s}^k\| \tag{7}$$

holds with some $\beta > 0$ for all k, cf. Nocedal and Wright (2006). The geometric interpretation of inequality 7 is that the search directions $\tilde{s}^k$ enclose an angle of strictly less than 90 degrees with the exact negative gradient g^k at the current iterate. This ensures, in particular, that all $\tilde{s}^k$ are directions of descent.

For the steepest descent method, we have $\tilde{s}^k = \tilde{g}^k$ and it can easily be shown that the angle condition is satisfied if the relative error of the inexact gradient is smaller than 0.5. Indeed, if $\|\tilde{g}^k - g^k\| < \delta\|\tilde{g}^k\|$ for some $\delta \in [0, 0.5)$, we obtain

$$\begin{aligned} (g^k, \tilde{s}^k) = (g^k, -\tilde{g}^k) &\leq \|\tilde{g}^k - g^k\| \cdot \|g^k\| - \|g^k\|^2 \\ &\leq \delta\|\tilde{g}^k\| \cdot \|g^k\| - (1 - \delta)\|\tilde{g}^k\| \cdot \|g^k\|\,, \end{aligned} \tag{8}$$

which gives the angle condition, inequality 7, for the steepest descent method with $\beta = 1 - 2\delta$. Now, we can utilize the estimate from inequality 6 to ensure that the error in the approximate gradient is sufficiently small, i.e., $\|\tilde{g}^k - g^k\| < \delta\|\tilde{g}^k\|$.

In the case of a quasi-Newton method like (L)-BFGS, the inexact search direction $\tilde{s}^k$ computed with the approximations $\widetilde{B}_k$ and $\tilde{g}^k$ can be interpreted as a solution to a perturbed version of the linear system in equation 2. Here, the Dennis-Moré conditions (Dennis and Moré, 1977) have to be satisfied to ensure a superlinear rate of convergence.

Before introducing the compression methods, we should discuss three main challenges when using inexact gradients in an iterative descent scheme. First, convergence to the same local minimum as with exact gradients can not be guaranteed. Secondly, the rate of convergence might be reduced, i.e., it might take more iterations to obtain the same reduction in the objective function. Furthermore, depending on the level of inexactness, the approximated gradient might not be a direction of descent and iterations may get stuck.

With the above derivation using the angle condition and the error bound from inequality 6, convergence of the inexact LBFGS-method to a local minimizer can be guaranteed provided that the inexactness of the gradient is sufficiently small. Of course, however, without any further assumptions on the problem structure or the initial model, it is in general not possible to ensure convergence towards a global minimizer or even towards the same local minimizer. This is a common limitation of non-convex optimization problems, which applies to all iterative descent schemes. On the other hand, there is no guarantee the iterations using exact gradients will converge to a better minimum.

To address the second and third challenge, it is important to ensure that more accurate gradients are used once a local minimizer is approached. This is achieved by adaptively choosing the compression criteria, which will be introduced in the next session. Thereby, we can guarantee that the accuracy of the approximate gradient improves when its norm decreases. With carefully chosen compression thresholds, we expect that all variants of descent methods tailored to full-waveform inversion can also be applied successfully with inexact gradients.

LOSSY COMPRESSION TECHNIQUES

In this section we present several methods for lossy compression of the forward wavefield. All of the following methods require suitable criteria to steer the quality of the compression. For the spatial compression, we recall from inequality 6 that the error in the gradient can be bounded by $\|(\widetilde{\mathcal{D}u})(\mathbf{x},t) - (\mathcal{D}\,u)(\mathbf{x},t)\|$, which gives a localized quantity that can be computed at the time of compression. Hence, we propose using thresholds on the absolute and relative point-wise difference between the decompressed and the fully-resolved wavefield. More specifically, for given tolerances $\varepsilon_{\mathrm{abs}_2} > \varepsilon_{\mathrm{abs}_1} > 0$ and $\varepsilon_{\mathrm{rel}}$, we will ensure that one of the following conditions holds for all snapshots of the decompressed wavefield:

(C-1) The maximum absolute error is smaller than $\varepsilon_{\mathrm{abs}_1}$.

(C-2) The maximum relative error is smaller than $\varepsilon_{\mathrm{rel}}$ and the maximum absolute error is smaller than $\varepsilon_{\mathrm{abs}_2}$.

Specific configurations of these thresholds can be found in the section on numerical results.

In the following, we use the term “forward wavefield” in a generic way, which is motivated by the fact that the techniques apply to all kinds of vector fields, e.g., displacements, velocities, strains, pressure or velocity divergence. Furthermore, we denote the wavefield by u instead of $\mathcal{D}u$ to simplify the notation.

Field re-quantization

As a first method for the spatial compression of the wavefield, we propose to use an adaptive number of bits to represent the values at the grid points; see Hanzich et al. (2013) for prior work in this field. The main idea is to adapt the floating-point precision for storing the wavefield to the local amplitudes. This enables us to represent values by fewer bits in areas with a small range of amplitudes and, consequently, to reduce the memory requirements. We borrow the term *quantization* from image processing, which refers to a transfer function that maps a continuous interval of values $[v_{\min}, v_{\max}]$ onto a finite set $\{v_0, v_1, v_2, \ldots, v_k\}$ with increments of $\Delta v = v_{i+1} - v_i$. Since all entities are already quantized in our case - for instance, in single precision floating-point format - and only the number of discrete values will change, we use the term *re-quantization*.

Compressing and decompressing the wavefield means to transform the higher-precision representation into the lower-precision one and vice versa, which is summarized in Algorithms 1 and 2, respectively. In addition to storing the values at all grid points with the reduced number of bits, the offset value u^o and the spacing $s = \Delta v$ need to be stored. Due to this overhead, we cannot pick the floating-point precision individually for every grid point. Instead, we divide the computational domain Ω into smaller subdomains Ω_i, $i = 1, \ldots, K$. Within every subdomain we use a constant number of bits to store the values of its grid points and, thus, only two additional values for offset and spacing are required per subdomain. Hence, the size of the subdomains should be small enough to take advantage of the locally varying amplitudes of the wavefield, but also large enough such that the memory overhead introduced by offset and spacing is not significant. For instance, in a spectral-element code the finite elements - with 125 grid points for fourth-order shape functions in 3D - can serve as partitioning, but larger subdomains can be used as well.

Algorithm 1 Re-quantization, compression

```
Require: uncompressed raw wavefield $u$, subdomains $\Omega_i$, vector of bit-resolutions $b$
for all $\Omega_i$ do
    Set offset $u_i^o = \min\{u(x) : x \in \Omega_i\}$.
    Set spacing $s_i = \frac{1}{2^{b_i}-1} \cdot \big(\max\{u(x) : x \in \Omega_i\} - u_i^o\big)$.
    for all $\mathbf{x}_j \in \Omega_i$ do
        Set $\bar{u}_j = \left\lfloor \frac{(u_j - u_i^o)}{s_i} + 0.5 \right\rfloor$.
    end for
end for
Ensure: compressed wavefield $\bar{u}$, vector of offsets $u^o$ and spacings $s$.
```

Algorithm 2 Re-quantization, decompression

```
Require: Compressed wavefield $\bar{u}$, offsets $u^o$, spacings $s$.
for all $\Omega_i$ do
    for all $\mathbf{x}_j \in \Omega_i$ do
        $\tilde{u}_j = u_i^o + s_i \cdot \bar{u}_j$
    end for
end for
Ensure: Decompressed wavefield $\tilde{u}$.
```

From Algorithms 1 and 2 we directly deduce that the point-wise compression error is bounded by

$$|\tilde{u}_j - u_j| = |u_i^o + s_i \cdot \bar{u}_j - u_j| = \left| u_i^o - u_j + s_i \left\lfloor \frac{(u_j - u_i^o)}{s_i} + 0.5 \right\rfloor \right| \leq 0.5 s_i. \tag{9}$$

Hence, it is straightforward to apply the criteria (C-1) and (C-2) presented at the beginning of this section to determine the required number of bits in every subdomain.

p-Coarsening

This second technique for the spatial compression of the wavefield is tailored specifically to spectral-element methods, which are widely used in numerical wave propagation codes like SPECFEM3D (Peter et al., 2011) or SES3D (Gokhberg and Fichtner, 2016). Here, the wavefield is spatially represented by higher-order shape functions, most commonly with polynomials of order 4. Note that we restrict the following presentation to fourth-order polynomials, but the method can easily be extended to other high-order finite-element methods. A straightforward way to approximate the wavefield with fewer degrees of freedom is to adaptively reduce the polynomial degree within the spectral elements. This requires to downsample the wavefield locally onto a lower-dimensional space. Based on similar concepts for adaptive mesh refinement in finite-element methods, we call this approach p-coarsening.

Spectral-element methods represent the wavefield on a single element by the Galerkin projection of the following form

$$u(\mathbf{x}, t) = \sum_{i,j,k=0}^{4} u_4^{ijk}(t) \psi^{ijk}(\mathbf{x}), \tag{10}$$

with shape functions ψ^{ijk} and time-dependent coefficients $u_4^{ijk}(t)$ that approximate the wavefield at the grid points. Here, ψ^{ijk} are the tensorized Lagrange polynomials of degree 4 with control points given by the Gauss-Lobatto-Legendre (GLL) quadrature rule. As the number of collocation points quickly increase from 8 (degree 1) to 27 (degree 2) and 125 (degree 4), a lower polynomial degree reduces the memory requirements significantly.

In the following, we denote the fourth-order coefficients by u_4^{ijk} and the lower-dimensional representation of u using polynomials of order $p \in \{0, 1, 2, 3, 4\}$ by u_p. Here, u_0 is a constant function defined by the average value of the wavefield at the collocation points of $p = 1$. The projection of u_4 onto the lower-dimensional subspace can be computed by solving a least-squares problem, which requires the solution of a linear system with $(p+1)^3$ unknowns.

As a cheaper alternative, we can determine u_p by simply evaluating u_4 at the lower-order collocation points. Moreover, we can exploit the hierarchical structure of the GLL points - see Figure 1 - which means that the projection requires only memory access to the corresponding indices of u_4, but no interpolation for $p \in \{0, 1, 2, 4\}$. Since this approach does not incorporate information from the higher-order collocation points, this approximation is slightly worse compared to the least-squares projection, but it proved to be acceptable in our numerical tests.

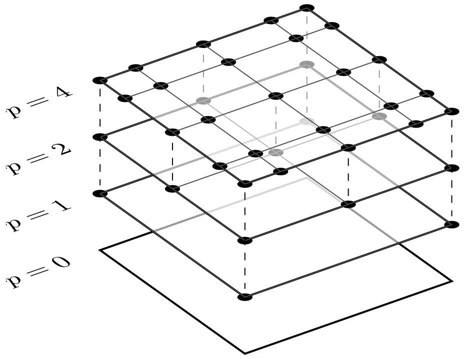


Figure 1: Hierarchical order of the GLL points in 2D (same structure as in 3D). For degrees 1, 2 and 4, every collocation point of the lower-order polynomial is also a collocation point of the higher-order polynomials.

The local polynomial degree is determined based on the criteria (C-1) and (C-2) introduced at the beginning of this section. In order to estimate the point-wise errors and to decompress the wavefield, we simply need to plug in u_p into equation 10 and to evaluate u at the fourth-order collocation points.

Re-quantization and p-coarsening on hierarchical grids

While re-quantization and p-coarsening can be used independently, a combination of both achieves the highest compression rate. The main idea is to use a hierarchy of grids on which the lower-order information acts as a predictor for the values of the next finer grid level. More specifically, on each level we compute the difference between the values predicted by the lower-order shape functions and the actual values of the wavefield and re-quantize these residuals. This is conceptually similar to MPEG video compression, see Sullivan and Wiegand (2005). For spectral-element methods we exploit again the hierarchy of the GLL points and denote the set of collocation points on level l as $P_{p[l]}$ with $p = (0, 1, 2, 4)$. Hence, we have $P_{p[l]} \subset P_{p[l+1]}$ and $P_0 = \emptyset$. Now, we use Lagrange polynomials of order $p[l]$ and interpolating $P_{p[l]}$, to predict the values at the points in $P_{p[l+1]}$. Algorithm 3 describes the compression in detail. The error bound given by inequality 9 determines the number of bits that are required to re-quantize the residuals at the collocation points of the next higher level such that criteria (C-1) and (C-2) are satisfied, cf. line 5 in Algorithm 3. This number is determined independently for each level. Note that we use the approximated values $\tilde{u}^{ijk}_{p[l]}$ from the previous level instead of the actual wavefield $u^{ijk}_{p[l]}$ in line 7 in Algorithm 3. Hence, we only use information that is available at the time of decompression. This ensures that the approximation error does not increase if we move to higher levels and we indeed have control over point-wise absolute and relative errors after decompression.

```
Algorithm 3 Combined re-quantization and $p$-coarsening
Require: uncompressed raw wavefield on a subdomain, thresholds $\varepsilon_{\text{abs}_1}$, $\varepsilon_{\text{abs}_2}$ and $\varepsilon_{\text{rel}}$.
Set $\tilde{u}_0 = u_0$.
for $l = 0, 1, 2$ do
    for all $\mathbf{x}^{ijk} \in P_{p[l+1]} \setminus P_{p[l]}$ do
        Set $r^{ijk} = u_4(\mathbf{x}^{ijk}) - \tilde{u}_{p[l]}(\mathbf{x}^{ijk})$.
        Determine number of bits $b_l$ to re-quantize $r^{ijk}$ based on $\varepsilon_{\text{abs}_1}$, $\varepsilon_{\text{abs}_2}$ and $\varepsilon_{\text{rel}}$.
        Apply Algorithm 1 to re-quantize $r^{ijk}$ and obtain $\bar{r}^{ijk}$ and $s_l$.
        Set $\tilde{u}^{ijk}_{p[l+1]} = \tilde{u}^{ijk}_{p[l]} + \bar{r}^{ijk}$.
    end for
end for
Set $\bar{u} = \bar{r}$.
Ensure: compressed wavefield $\bar{u}$, spacings $s_l$ and number of bits $b_l$ for every level.
```

The procedure is sketched in Figure 2 for a 2D element. Note that in 3D the number of collocation points rapidly increases from 8 ($P_1 \setminus P_0$) to 19 ($P_2 \setminus P_1$) and 98 ($P_4 \setminus P_2$). Furthermore, the magnitude of the residuals typically decreases from level to level, which yields high compression factors.

While this approach is straightforward to use with finite-element methods, it is important to note that the patch-wise partitioning of the mesh can be done independently of the discretization scheme. We will comment on possible extensions to finite-difference methods in the discussion.

Temporal Compression

Explicit time stepping schemes, which are widely used in seismic wave propagation codes, require much shorter time steps due to the CFL condition than the sample rate predicted by the Nyquist-Shannon theorem. This motivates using a coarser temporal grid for storing the wavefield and FWI codes typically save only every kth time step with k in the order of 10 to 30; see Fichtner et al. (2009) for a derivation of the sampling rate.

Instead of using a piecewise constant extrapolation of the forward wavefield during the adjoint run, however, we propose a cubic spline interpolation to capture the smoothness of the signal in time. This comes at low computational cost, but improves the approximation significantly and enables us to reduce the sampling rate of the snapshots. Since it would be prohibitively expensive to retrieve and access all snapshots simultaneously, we suggest to use a sliding window of 4 consecutive snapshots to sequentially compute the spline interpolant. This procedure is illustrated in Figure 3. The wavefield in the central subinterval between snapshots 2 and 3 is approximated with the cubic spline. Afterwards, the oldest snapshot is released and the next one is retrieved from disk to continue with the reconstruction in the next subinterval. Here, spatial decompression is carried out in a first step. Afterwards, the spline coefficients are determined by solving a tridiagonal linear system, which can be done analytically, since only 2 unknowns remain after applying natural boundary conditions. Although every grid point requires its own cubic spline and spatial dependencies are not

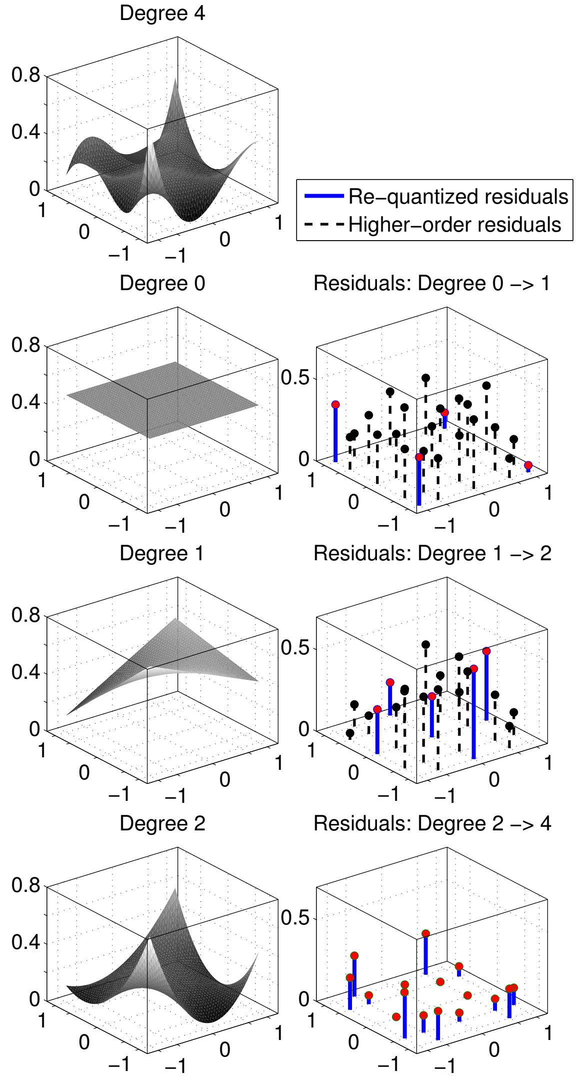


Figure 2: 2D sketch of re-quantization for a reference element of $[-1, 1] \times [-1, 1]$ on a hierarchy of grids. The current interpolant predicts the values at the collocation points of the higher polynomial degree. Only the residuals of the next level are re-quantized (blue lines). These residuals are then used together with the current interpolant to construct the prediction on the next higher level. This procedure is repeated for degrees 0, 1 and 2.

taken into account, the tridiagonal matrix only depends on the time increments between the snapshots, which makes the computation of the spline coefficients comparatively cheap. The evaluation of the spline at intermediate time steps requires 3 summations and 3 multiplications per grid point. Note that we can further reduce the computational overhead by evaluating the spline only every second or third time step when calculating the sensitivity kernel.

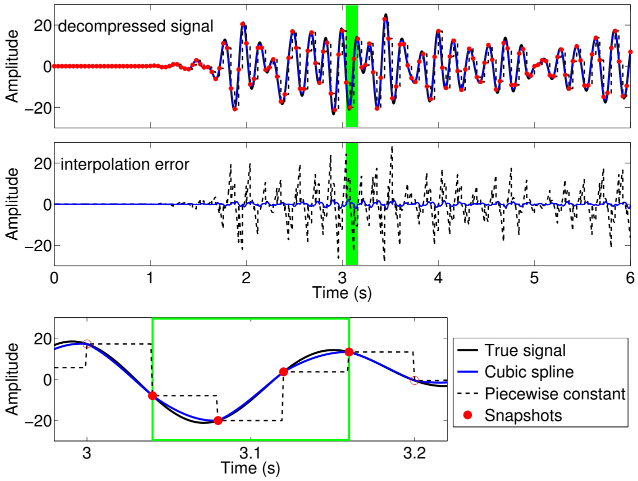


Figure 3: Cubic spline interpolation using a sliding window with 4 consecutive snapshots. Top row: reconstructed signals using snapshots at the red markers with either a piecewise constant extrapolation (dashed line) or a cubic spline (blue line). Middle row: interpolation errors for both methods. Bottom row: zoom to sliding window. Snapshots at the 4 time steps with the filled red markers determine a cubic spline that we use to interpolate the signal in the central subinterval.

Adjoint Shadow Zones

The Fréchet derivatives with respect to different structural parameters have in common that the forward wavefield, or its temporal or spatial derivatives, respectively, is correlated with the corresponding adjoint field, cf. equation 5. Hence, only time steps at which forward and adjoint waves locally overlap contribute to the sensitivity kernel. In particular, there is no need to store any information locally at time steps prior to the first arrival of the forward wavefield or later than the first arrival of the adjoint waves. This observation is remarkably useful in the latter case, since it works independently of the energy and the amplitudes of the forward wavefield. Thus, it enables us to disregard parts of the forward wavefield, which would not have been identified by the previously discussed methods. Moreover, this compression is loss-less with regards to the sensitivity kernel. Figure 4 visualizes the evolution of the shadow zones for a single source-receiver pair in a 2D domain.

As has been pointed out above, the shadow zones require lower bounds on the arrival times of forward and adjoint waves for every subdomain. This can be done in a preprocessing step prior to the inversion, which computes the respective first and last time steps for each of the subdomains. As a rough estimate for the last time step at which the forward wavefield needs to be stored locally, we compute the minimum distance of the subdomain

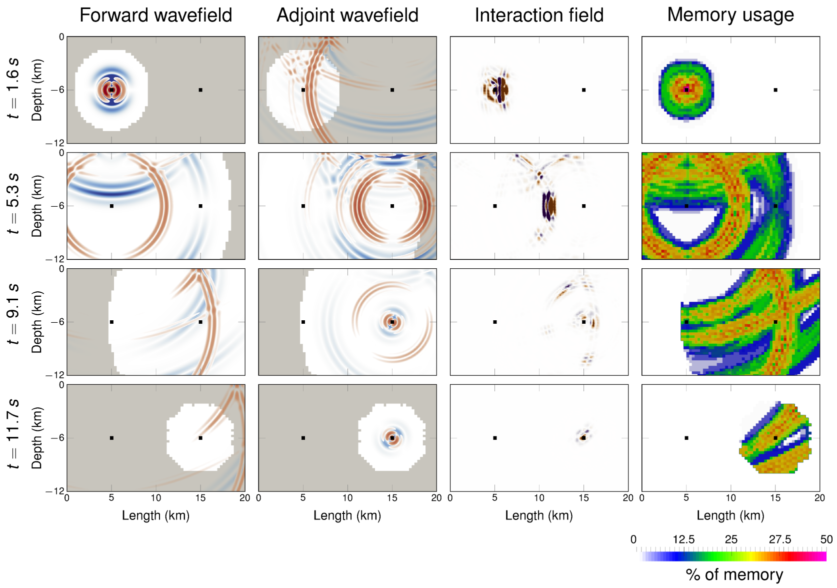


Figure 4: Time evolution of adjoint shadow zones for a single source-receiver pair on a 12 km × 20 km domain. The rows depict the curl (red) and the divergence (blue) of the forward and the adjoint wavefield as well as the interaction field, i.e., the product of both wavefields, which gives the instantaneous contribution to the sensitivity kernel. The gray-shaded areas indicate the shadow zones, where either the forward or the adjoint wavefield is zero. Obviously, also the interaction field is zero in these areas and there is no need to store the wavefield. The right column shows the current memory usage of the compressed wavefield relative to storing all values with full precision.

to any seismic station and divide it by the maximum P-wave velocity. Of course, more sophisticated methods can be applied to estimate the travel times. For instance, we could run a single adjoint simulation and track the first arrivals of the adjoint waves to obtain a more accurate prediction. The same technique can also be applied for the first arrivals of the forward wavefield.

RESULTS

In this section we present numerical examples for the proposed compression methods for a single parameter gradient and compress the strain field during the forward run. All numerical tests have been carried out on Piz Daint at the Swiss National Supercomputing Center (CSCS). Details on the implementation of the spectral-element code can be found in Boehm (2015). All computations are in double-precision floating-point arithmetic.

Clearly, the actual compression factor depends on the geometry of the domain, the locations of sources and receivers, and on the underlying velocity model. However, most of them share the common property of sources and receivers located at or near the surface and a velocity profile that increases with depth. Hence, the results presented in this section provide a good indication of the capabilities of the method. For simplicity, we always use the L^2-norm as misfit functional, but, of course, more sophisticated measures can be used as well. Furthermore, we apply a free-surface condition at the top boundary and dashpot absorbing boundary conditions (Epanomeritakis et al., 2008) on all other faces of the computational domain.

We use two different criteria to assess the quality of the inexactly computed gradient using the compressed wavefield. On the one hand, we consider the angular difference θ, which denotes the angle enclosed by exact and inexactly computed gradients, cf. the condition given by inequality 7,

$$\cos\theta = \frac{(\tilde{g}, g)}{\|\tilde{g}\| \cdot \|g\|} \tag{11}$$

If the angular difference is small, the gradient computed with compressed information points into a similar direction as the fully-resolved gradient and thus, the model update will be similar, too. Furthermore, we measure the difference between both gradients by the structural similarity index (SSIM) developed by Wang et al. (2004), which is widely used in image processing. The SSIM rates the perceived similarity of images with values between zero and one with one being the best score. It is important to note that all examples use the consistent discrete gradient as result of the adjoint simulation without any modification (e.g., clipping or smoothing). In particular, even higher compression factors can be achieved if we compare smoothed gradients.

Example 1: Spherical Wave Problem

As a first example, we consider a simple spherical wave problem on a 3D domain of 20 km × 20 km × 18 km with a homogeneous elastic material $\rho = 2000$ kg/m^3, $v_p = 2500$ m/s and a Poisson ratio of 0.25. The simulation time is 15 s. Furthermore, we use a single source-receiver pair at 6 km depth with a horizontal distance of 10 km. The source-time function is a Ricker wavelet with a dominant frequency of 3 Hz and we re-inject the recorded displacements as adjoint source.

In order to assess the quality of the proposed methods, we compare sensitivity kernels that result from different configurations of the compression thresholds with the exact kernel. A detailed quantitative analysis is given in Table 1. Here, we indicate the effective compression factor, which we define as the ratio of the total memory required for the uncompressed wavefield and the total memory required by the compressed wavefield. Furthermore, we show the minimum instantaneous spatial compression factor, which varies between 8 and 16 in all configurations and mainly depends on $\varepsilon_{\text{abs}_1}$. The quality of the approximation is measured by the angular difference and the SSIM. In addition, Figure 5 shows vertical slices through the center of the domain.

The memory requirements can be reduced by three orders of magnitude without hardly any differences in the visual appearance or the direction of the gradient. As expected, the quality of the approximated gradient declines with an increasing compression factor. However, even with a factor of 4714, we obtain an approximate kernel that is at least qualitatively similar and, more importantly, that is still a direction of descent with an angular difference of roughly 46 degrees. Note that `gzip` (version 1.3.12 with option -9 for highest compression) as a standard black-box loss-less compression software yields only a compression factor of 1.24 for this particular wavefield. This shows the necessity and also the great benefit of tailoring lossy compression methods to their application in FWI.

	$\varepsilon_{\text{abs}_1}$	$\varepsilon_{\text{abs}_2}$	ε_{rel}	sr	**cf**	miscf	θ	SSIM$_x$	SSIM$_y$	SSIM$_z$	ohd
(i)	0.001	0.1	0.005	30	**544**	8.6	3.1	0.9991	0.9983	0.9998	9.26%
(ii)	0.01	0.1	0.02	30	**1043**	16.0	3.1	0.9991	0.9983	0.9998	9.09%
(iii)	0.01	0.5	0.1	50	**1708**	15.8	19.7	0.9497	0.9638	0.9164	5.95%
(iv)	0.001	1	0.005	125	**2100**	8.2	25.2	0.8967	0.9099	0.8499	2.64%
(v)	0.01	1	0.1	75	**2506**	15.6	33.0	0.8976	0.8015	0.9400	4.27%
(vi)	0.01	10	0.3	100	**3277**	15.5	39.0	0.8843	0.6808	0.9191	3.28%
(vii)	0.01	10	0.3	150	**4714**	15.1	46.4	0.5975	0.8909	0.5334	2.28%

Table 1: Statistics for various configurations of the compression algorithm. $\varepsilon_{\text{abs}_1}$, $\varepsilon_{\text{abs}_2}$ and ε_{rel} refer to the spatial compression thresholds and sr denotes the sampling rate of the spline snapshots. cf is the overall compression factor, miscf the minimum instantaneous spatial compression factor, θ the angular difference between exact and approximated gradient. Columns 9-11 indicate the SSIM for slices through the center of the domain and orthogonal to each of the coordinate axis and the last column states the overhead in CPU time - relative to a forward simulation without I/O - for spatial compression and decompression and the computation of the spline coefficients.

Figure 6 shows the temporal evolution of the spatial compression. Throughout the simulation we achieve at least an instantaneous spatial compression factor of 15.8. Furthermore, the increase of the compression factor towards the end of the simulation indicates the effect of the adjoint shadows.

Table 2 states the overhead in CPU time that is required to compress the forward strain field and to decompress it again during the adjoint run. The total overhead adds only about 2-10% to the cost of the forward simulation itself. Hence, the compression methods proposed in this paper are much less expensive than checkpointing techniques or solving the

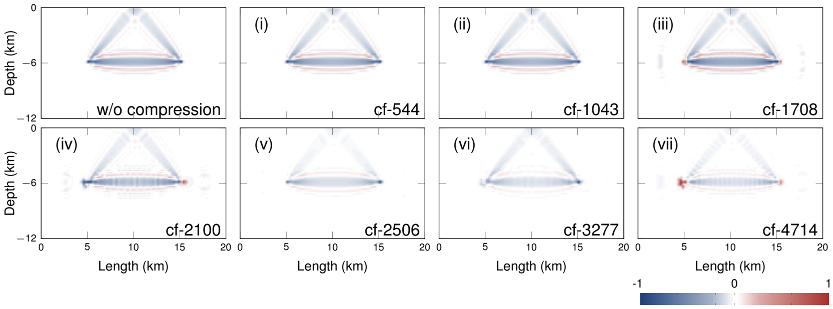


Figure 5: Gallery of normalized sensitivity kernels with a compressed forward wavefield for the configurations of the thresholds indicated in Table 1. The top left image shows the sensitivity kernel without compressing the forward wavefield; the numbers in the lower right corner indicate the compression factor. All images have been truncated at 12 km depth and show a vertical slice through the center of the domain, which corresponds to SSIM$_x$.

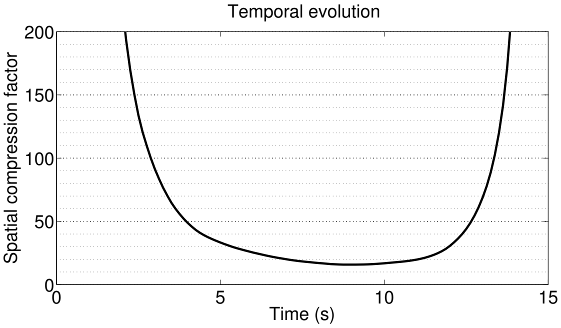


Figure 6: Evolution of the spatial compression factor during the simulation for scenario (iii) in Table 1. Note that the vertical axis has been truncated to a factor of 200. The spatial compression factor tends to infinity towards the beginning and the end of the simulation due to the shadow zones. The minimum instantaneous spatial compression factor is 15.8.

backward wave equation, which would typically require additional computations in the order of one forward simulation. Moreover, the actual overhead in wall-clock time that is needed for compression and decompression is typically even less, as these tasks can be carried out in separate threads overlapping with the simulation. Furthermore, Table 2 shows that the compression factor and the computational overhead are inversely proportional. This is due to the fact, that a higher sampling rate requires fewer spline coefficients and fewer snapshots to be spatially compressed and decompressed. In addition, higher compression thresholds not only increase the compression factor, but also allow for skipping more steps in the prediction-correction phase.

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)
spatial compression	3.80%	3.66%	2.43%	1.10%	1.74%	1.35%	0.93%
spatial decompression	1.63%	1.60%	1.02%	0.44%	0.73%	0.56%	0.38%
spline coefficients	3.83%	3.83%	2.49%	1.10%	1.80%	1.38%	0.96%
total overhead (ohd)	9.26%	9.09%	5.95%	2.64%	4.27%	3.28%	2.28%

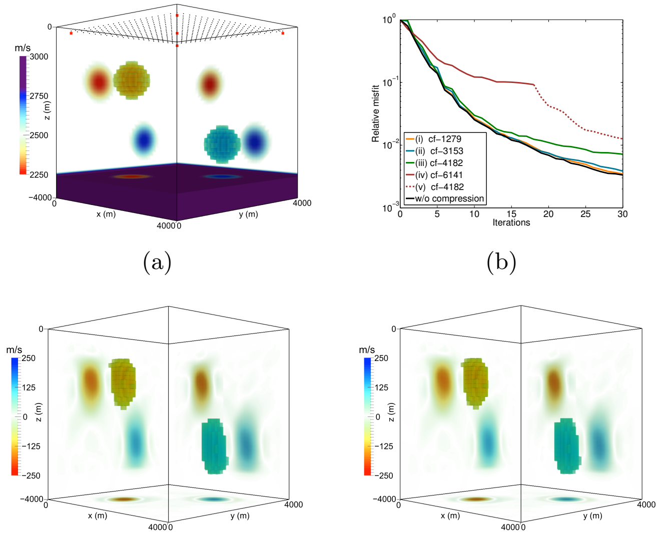
Table 2: CPU time of the different components of the compression methods relative to one forward simulation. Results are shown for the different configurations of the compression thresholds from Table 1.

Example 2: 3D Inversion in a Cube

We consider a cubic domain of 4 km × 4 km × 4 km with a homogeneous background medium with $\rho = 2000$ kg/m^3 and $v_p = 2500$ m/s, and a strong reflector near the bottom with $v_p = 3000$ m/s. Again, we assume a constant Poisson ratio of 0.25 and invert only for v_p. The true model contains two ball-shaped anomalies with ±10% deviations from the background model. Synthetic test data is generated by five point sources using a Ricker wavelet with a dominant frequency of 5 Hz. An array of 441 equidistantly aligned receivers is placed below the surface. Figure 7(a) shows the geometry of the setup as well as the true velocity model.

The initial model considered for the inversion contains the two layers, but not the ball-shaped anomalies. We run LBFGS several times using either the fully-resolved forward wavefield or different settings of the compression thresholds. Two reconstructions and the evolution of the misfit for several configurations are shown in Figure 7. Even with an average compression factor of more than 3000 per source, the iterations look very similar and converge to the same model. The models from scenario (i) and (ii) after 30 iterations of LBFGS are visually indistinguishable from the reconstruction without compression. As expected, the performance of LBFGS declines when too much information is lost during the compression. However, this can be circumvented by adaptively lowering the compression thresholds and restarting the iterations. This is the case in scenario (iv) where after 18 iterations the inexact LBFGS search direction is not a direction of descent, see Figure 7(b) for details. After lowering the compression thresholds during the iterations for scenarios (iii) and (iv), all tests eventually converge to the same model.

In summary, LBFGS performs well using inexact gradients that result from compressing and decompressing the wavefield. In particular, neither the rate of convergence nor the



(a) (b)

(c) (d)

Figure 7: (a) Computational setup with the true model for v_p, receiver locations marked as black dots and the source locations indicated by red squares. Slices through the center of the ball-shaped anomalies are projected to the rear and bottom. (b) Misfit reduction with and without compression. The numbers cf-X indicate the average compression factor per source. Note that the thresholds were lowered in scenario (iv) after 18 iterations. (c) Reconstruction without compression illustrating the v_p deviation from the initial model. (d) Reconstruction of scenario (i) using the compressed forward wavefield.

reconstruction changes significantly when the memory requirements are reduced by three orders of magnitude.

Example 3: SEG Overthrust Model

In this last example, we consider a more complex geology and use a subset of the SEG overthrust model (Aminzadeh et al., 1996). Here, we use a domain of 8 km × 8 km × 3.2 km, where we extract the central parts of the model in x- and y-direction. We consider an array of 6561 receivers equidistantly distributed on a plane 8 m beneath the surface with a lateral and longitudinal spacing of 100 m. We use two configurations for the source. Setup 1 (S1) is a single point source in the center of the x-y-plane at 40 m depth. Setup 2 (S2) is an encoded source with 6400 simultaneous sources equidistantly aligned on an array with a spacing of 100 m at 40 m depth. The encoding weights are chosen as independent samples of Rademacher's distribution, i.e., either $+1$ or -1 with equal probability. In both cases, we use a Ricker wavelet with a dominant frequency of 10 Hz as source time function and a simulation time of 6 s.

We generate synthetic data with the SEG overthrust model and evaluate the gradient using the L^2-norm as misfit functional for two different models. Model A (MA) is a 1D-model that is constant in the x-y-plane with the average value of the true model at that depth. For Model B (MB) we apply a Gaussian smoothing filter to the true model, which yields a model that is already close to the true model. Figure 8 shows vertical slices of the true model, as well as Model A and Model B.

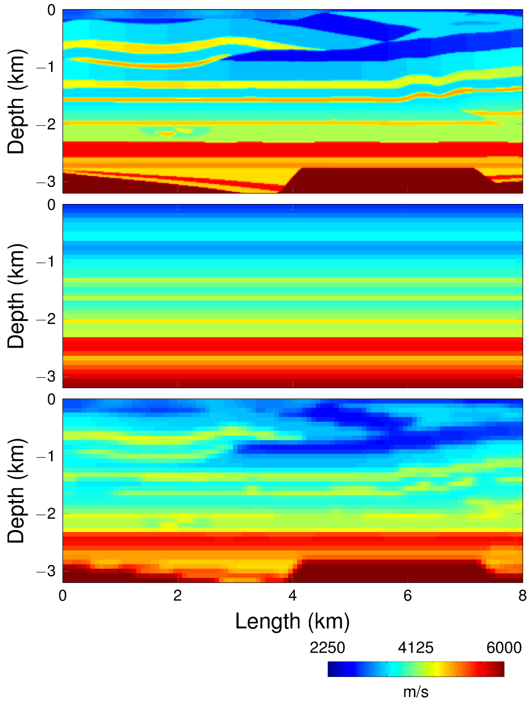


Figure 8: Vertical slices of the models to compare the gradients. Top: True model that is used to generate synthetic data. Middle: 1D-model with average value of the x-y-plane (Model A). Bottom: Gaussian smoothing filter applied to the true model (Model B). All images show the P-wave velocity.

Table 3 compares the accuracy of the gradient with respect to both Lamé coefficients for both source setups and both models. Similar to the previous examples, we achieve a compression factor of three orders of magnitude with only a small angular difference for the point source. The 6400 simultaneously firing point sources in Setup 2 create a more complicated wavefield. Hence, for fixed error thresholds the achievable compression factor is smaller than for a single point source. However, for an angular difference of 20 - 32 degrees the memory requirements can still be reduced by three orders of magnitude.

Furthermore, Figure 9 shows that the inexact gradient still preserves fine-scale information if the compression thresholds are chosen carefully. Here, we show slices of the inexact gradient with respect to the shear modulus accumulated from 20 encoded sources with an average compression factor of 1364. Although the level of detail decreases with increasing depth, some fine-scale structures remain visible in the inexact gradient. This confirms that the compression methods are applicable to models with complex geology. With carefully chosen compression thresholds the inexact gradients can be used with any iterative descent method. We will comment on this in more detail in the following section.

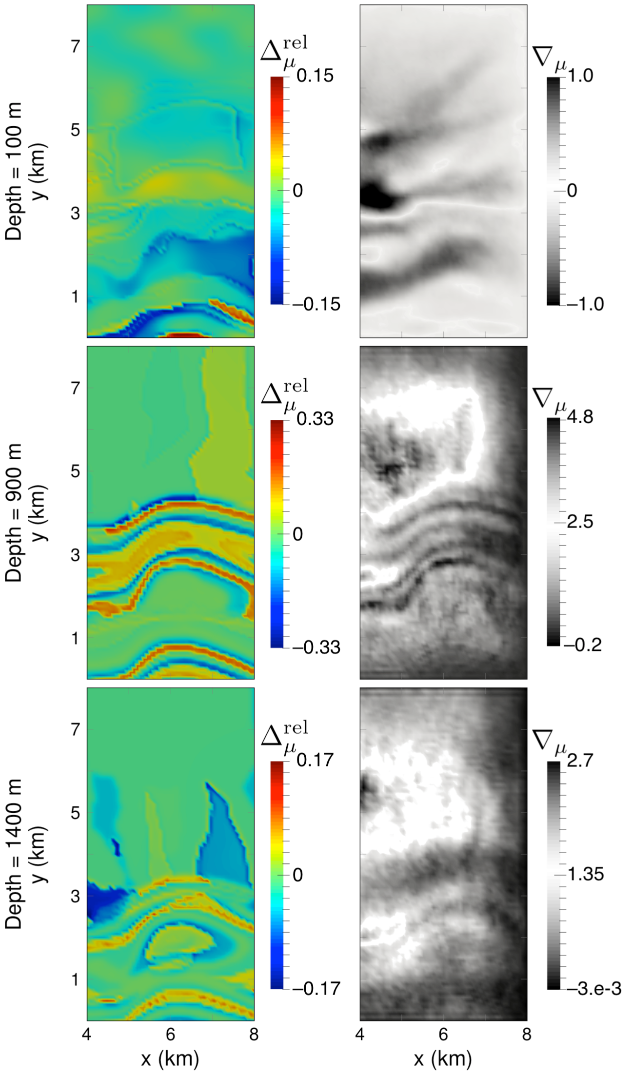


Figure 9: Sections of the accumulated gradient from 20 encoded sources at horizontal slices through the domain for Model B. The left column shows the relative deviation from the true shear modulus and the right column the normalized gradient w.r.t. the shear modulus for each slice.

	$\varepsilon_{\text{abs}_1}$	$\varepsilon_{\text{abs}_2}$	ε_{rel}	sr	**cf**	miscf	θ	ohd
S1-MA	0.001	0.01	0.05	50	**4462**	44.3	7.8	6.62%
	0.001	0.01	0.05	100	**8895**	44.3	46.2	3.36%
S1-MB	0.001	0.01	0.05	50	**4025**	41.0	3.1	6.75%
	0.001	0.01	0.05	100	**8020**	41.0	21.4	3.36%
S2-MA	0.0001	0.01	0.05	50	**306**	5.0	4.6	6.73%
	0.01	0.1	0.1	100	**1371**	9.7	32.3	3.48%
S2-MB	0.0001	0.01	0.05	50	**302**	5.0	3.4	6.75%
	0.01	0.1	0.1	100	**1350**	9.7	20.8	3.45%

Table 3: Statistics for various tests with the SEG overthrust model. The first column indicates the source type, i.e., S1 for a single point source or S2 for an encoded source, and the model for which the gradient is computed, i.e., MA for the 1D-model and MB for the smoothed model.

DISCUSSION

The compression methods proposed in this paper offer a promising alternative to checkpointing techniques. As indicated in the previous section, sufficiently accurate approximations of the sensitivity kernel can be obtained with only a small computational overhead, adding only about 2-10% percent to the CPU time of a forward simulation, which is significantly less than checkpointing methods even if they are tailored to FWI (Anderson et al., 2012). Furthermore, the practical performance is significantly better than with black-box loss-less compression tools like `gzip` that are not tailored to the application in FWI. Note that even higher compression factors can be achieved if the sensitivity kernels are smoothed before the model is updated.

A notable feature of the compressed kernels displayed in Figure 5 is the successive disappearance of the higher Fresnel zones. With increasing compression factor, kernels resemble "fat rays", proposed as a compromise between ray theory and accurate finite-frequency kernels that are more expensive to compute, cf. Husen and Kissling (2001); Yoshizawa and Kennett (2002). Higher Fresnel zones are relevant only when tomographic resolution is sufficient to ensure that structure as small as their width can indeed be recovered, see, for instance, Van Der Hilst and De Hoop (2005). This implies that compression factors may be further increased in regions of lower coverage where only the first Fresnel zone contributes significantly to resolution.

With an increasing resolution of the numerical mesh even the reduced memory requirements through compression might become prohibitively expensive at some point. In this case, compression can complement checkpointing techniques by spatially compressing the snapshots. This allows us to store more checkpoints and, thus, it reduces the computational overhead of checkpointing techniques.

The analysis of iterative inversion schemes using inexact gradient information shows that convergence to a (local) minimum can be guaranteed. It is, in general, not possible to ensure that the iterates with and without compression converge to the same model, although we did not encounter such problems in our numerical tests. This common limitation of non-convex problems can be partially mitigated by choosing the compression thresholds adaptively based on the norm of the gradient, thus gradually including more information. Similar strategies in the context of stochastic gradient methods have been applied successfully in full-waveform inversion (Li et al., 2012; Moghaddam et al., 2013), where randomly chosen subsets of sources introduce inexactness in the gradient. It has been observed by van Leeuwen and Herrmann (2013, 2014) that the level of inexactness can be quite high during the first iterations and only needs to be refined once a local minimum is approached.

In the remainder of this section, we comment on several aspects of the implementation and possible extensions of the methods.

A multi-parameter inversion requires strains for the sensitivity kernels with respect to the Lamé coefficients as well as the velocity field for the density kernel, cf. equations 3 and 4. The error thresholds steering the compression can be chosen independently for each field. Compressed strains are stored element-wise, which is necessary as strains are discontinuous across element boundaries in a spectral-element discretization. On the other hand, velocities are continuous throughout the whole domain and we can store the velocity field using global vectors. This reduces the memory requirements by the number of grid points that are shared among two or more elements. Note that storing only the displacement field instead of strains and velocities is disadvantageous for two reasons. First, re-computing temporal and spatial derivatives of the wavefield during the adjoint run would require significant computing time. Secondly, a small compression error in the displacement does not necessarily yield a small difference in the derivatives, which makes it very difficult to control the accuracy of the approximation for the strains.

All of the compression techniques are spatially localized and do not require MPI communication. Furthermore, compression, decompression and I/O operations, can be carried out asynchronously to the simulation and in multiple threads, which further reduces the computational overhead. Due to the significantly reduced memory requirements, using a local scratch file system is feasible and highly desirable in case it is available on the HPC architecture.

Possible extensions of the compression methods can combine $p-$ and h-coarsening in a similar fashion as in hp-adaptivity (Demkowicz et al., 2002). This would enable us to adaptively merge neighboring elements and to further reduce the memory requirements, but also introduce an additional computational overhead to transform the mesh.

While this paper primarily targets finite-element methods, the same ideas for wavefield compression can be applied for finite-difference methods as well. First of all, the temporal compression of the wavefield and re-interpolation with cubic splines works completely independently of the spatial discretization as long as the spatial grid does not change with time. Likewise, re-quantization requires only a partitioning of the spatial domain in smaller subsets on which the same number of bits is used, but no additional geometric information. The main requirement to apply re-quantization in combination with p-coarsening is a hierarchical structure of the grid. Although this may not be given naturally for a finite-difference discretization, we can easily split up any rectilinear grid into small patches with 2^d grid points per dimension and compress the wavefield on this subdomain using our proposed method. While for high-order finite-element methods the partitioning of the mesh is intrinsically tied to the discretization, locally interpolating the wavefield at the grid points by Lagrange polynomials is possible for arbitrary rectilinear grids. This hierarchical subdivision of the domain appears in many other compression approaches as well, for instance, MPEG or wavelet-based techniques, and is not limited to finite-element methods.

This paper focuses on adjoint methods to compute sensitivity kernels, but the techniques for wavefield compression are useful in other areas as well, e.g., in the context of scattering-integral methods (Chen et al., 2007). Furthermore, we currently investigate extensions for the adjoint-based computation of Hessian-vector products that are required, for instance, in a Newton-CG method (Götschel and Weiser, 2014; Boehm and Ulbrich, 2015). While it is conceptually straightforward to apply the same compression methods to the forward and adjoint wavefields for the computation of Hessian-vector products, the two main challenges are error propagation and accuracy. Since the decompressed wavefields also appear as a distributed source term in the two auxiliary wave equations, errors are propagated to the perturbed wavefields as well. On the other hand, the inexact Newton steps resulting from the decompressed wavefields must still yield significantly better updates than the LBFGS step to compensate for the additional simulations required for the CG iterations. We carried out some preliminary tests applying the previously discussed compression techniques to the forward and adjoint wavefield in a Trust-Region Newton-PCG (Boehm and Ulbrich, 2015) method. Obtaining descent directions was not a problem in our tests, however, the inexact Newton steps provided only a small improvement compared to LBFGS updates. Of course, more accurate steps can be obtained by reducing the compression thresholds, but this significantly lowers the achievable compression factor. Tailoring the compression methods to Hessian-vector products is subject of future research.

CONCLUSION

In this paper we proposed compression techniques tailored to FWI, where the forward wavefield is compressed and retrieved during the adjoint run to compute an approximate gradient. All of the presented methods are easy to implement and computationally cheap, adding an overhead of only a few percent to a simulation.

Numerical examples show that the methods are capable of reducing the memory requirements by three orders of magnitude without affecting the iterations to solve the inverse problem. Although the methods are tailored to spectral-element methods in the time domain, similar ideas can be applied for other numerical techniques as well.

ACKNOWLEDGMENTS

The authors would like to thank Editor John Etgen, Associate Editor Anatoly Baumstein and three anonymous reviewers for their valuable comments and excellent suggestions to improve the paper.

We gratefully acknowledge support by the Swiss National Supercomputing Centre (CSCS) in the form of CHRONOS project ch1 and PASC project GeoScale. The research leading to these results has received funding from the European Union's Horizon 2020 Programme (2014-2020) and from Brazilian Ministry of Science, Technology and Innovation through Rede Nacional de Pesquisa (RNP) under the HPC4E Project (www.hpc4e.eu), grant agreement n° 689772. Furthermore, the first author acknowledges funding from Shell within the project "Boosting full-waveform inversion".

REFERENCES

Aminzadeh, F., Burkhard, N., Long, J., Kunz, T., and Duclos, P., 1996, Three dimensional SEG/EAEG models - an update: The Leading Edge, **15**, no. 2, 131–134.

Anderson, J. E., Tan, L., and Wang, D., 2012, Time-reversal checkpointing methods for RTM and FWI: Geophysics, **77**, no. 4, S93–S103.

Boehm, C., and Ulbrich, M., 2015, A semismooth Newton-CG method for constrained parameter identification in seismic tomography: SIAM Journal on Scientific Computing, **37**, no. 5, S334–S364.

Boehm, C., 2015, Efficient inversion methods for constrained parameter identification in full-waveform seismic tomography: Ph.D. thesis, Technische Universität München.

Chen, P., Jordan, T. H., and Zhao, L., 2007, Full three-dimensional tomography: a comparison between the scattering-integral and adjoint-wavefield methods: Geophysical Journal International, **170**, no. 1, 175–181.

Demkowicz, L., Rachowicz, W., and Devloo, P., 2002, A fully automatic hp-adaptivity: Journal of Scientific Computing, **17**, no. 1-4, 117–142.

DennisJr., J., and Moré, J., 1977, Quasi-Newton methods, motivation and theory: SIAM Review, **19**, no. 1, 46–89.

Epanomeritakis, I., Akçelik, V., Ghattas, O., and Bielak, J., 2008, A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion: Inverse Problems, **24**, no. 3, 034015.

Fichtner, A., Kennett, B. L. N., Igel, H., and Bunge, H.-P., 2009, Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods: Geophysical Journal International, **179**, no. 3, 1703–1725.

Fichtner, A., 2011, Full seismic waveform modelling and inversion: Springer, Berlin Heidelberg.

Gokhberg, A., and Fichtner, A., 2016, Full-waveform inversion on heterogeneous HPC systems: Computers & Geosciences, **89**, 260–268.

Götschel, S., and Weiser, M., 2014, Lossy compression for PDE-constrained optimization: adaptive error control: Computational Optimization and Applications, pages 1–25.

Griewank, A., 1992, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation: Optimization Methods and Software, **1**, no. 1, 35–54.

Hanzich, M., Rubio, F., Aguilar, G., and Gutierrez, N., 2013, Efficient lossy compression for seismic processing: 75th EAGE Conference & Exhibition, Th–16–08.

Hanzich, M., Rubio, F., de la Puente, J., and Gutierrez, N., 2014, Lossy data compression with dct transforms: Workshop on High Performance Computing for Upstream, 7th-10th September, Crete, Greece, HPC30.

Husen, S., and Kissling, E., 2001, Local earthquake tomography between rays and waves: fat ray tomography: Physics of the earth and Planetary Interiors, **123**, no. 2, 127–147.

Igel, H., Djikpesse, H., and Tarantola, A., 1996, Waveform inversion of marine reflection seismograms for P-impedance and Poisson's ratio: Geophysical Journal International, **124**, no. 2, 363–371.

Li, X., Aravkin, A. Y., van Leeuwen, T., and Herrmann, F. J., 2012, Fast randomized full-waveform inversion with compressive sensing: Geophysics, **77**, no. 3, A13–A17.

Métivier, L., Brossier, R., Virieux, J., and Operto, S., 2013, Full waveform inversion and the truncated Newton method: SIAM Journal on Scientific Computing, **35**, no. 2, B401–B437.

Moghaddam, P., Keers, H., Herrmann, F., and Mulder, W., 2013, A new optimization approach for source-encoding full-waveform inversion: Geophysics, **78**, no. 3, R125–R132.

Nocedal, J., and Wright, S., 2006, Numerical optimization: Springer, New York, NY.

Peter, D., Komatitsch, D., Luo, Y., Martin, R., Le Goff, N., Casarotti, E., Le Loher, P., Magnoni, F., Liu, Q., Blitz, C., Nissen-Meyer, T., Basini, P., and Tromp, J., 2011, Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes: Geophysical Journal International, **186**, no. 2, 721–739.

Pratt, R. G., Shin, C., and Hick, G. J., 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, no. 2, 341–362.

Rickers, F., Fichtner, A., and Trampert, J., 2013, The Iceland–Jan Mayen plume system and its impact on mantle dynamics in the North Atlantic region: Evidence from full-waveform inversion: Earth and Planetary Science Letters, **367**, 39–51.

Sullivan, G., and Wiegand, T., Jan 2005, Video compression - from concepts to the H.264/AVC standard: Proceedings of the IEEE, **93**, no. 1, 18–31.

Symes, W. W., 2007, Reverse time migration with optimal checkpointing: Geophysics, **72**, no. 5, SM213–SM221.

Tape, C., Liu, Q., Maggi, A., and Tromp, J., 2010, Seismic tomography of the southern California crust based on spectral-element and adjoint methods: Geophysical Journal International, **180**, no. 1, 433–462.

Tromp, J., Tape, C., and Liu, Q., 2005, Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels: Geophysical Journal International, **160**, no. 1, 195–216.

Tromp, J., Komatitsch, D., and Liu, Q., 2008, Spectral-element and adjoint methods in seismology: Communications in Computational Physics, **3**, no. 1, 1–32.

Unat, D., Hromadka, T., and Baden, S. B., 2009, An adaptive sub-sampling method for in-memory compression of scientific data: An adaptive sub-sampling method for in-memory compression of scientific data:, Data Compression Conference, 2009. DCC'09., 262–271.

Van Der Hilst, R. D., and De Hoop, M. V., 2005, Banana-doughnut kernels and mantle tomography: Geophysical Journal International, **163**, no. 3, 956–961.

van Leeuwen, T., and Herrmann, F. J., 2013, Fast waveform inversion without source-encoding: Geophysical Prospecting, **61**, 10–19.

van Leeuwen, T., and Herrmann, F. J., 2014, 3D frequency-domain seismic inversion with controlled sloppiness: SIAM Journal on Scientific Computing, **36**, no. 5, S192–S217.

Virieux, J., and Operto, S., 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, no. 6, WCC1–WCC26.

Walther, A., and Griewank, A. Advantages of binomial checkpointing for memory-reduced adjoint calculations.:. Feistauer, M. (ed.) et al., Numerical mathematics and advanced applications. Proceedings of ENUMATH 2003, the 5th European conference on numerical mathematics and advanced applications, Prague, Czech Republic, August 18-22, 2003. Berlin-Springer. 834-843 (2004)., 2004.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E., April 2004, Image quality assessment: from error visibility to structural similarity: Image Processing, IEEE Transactions on, **13**, no. 4, 600–612.

Weiser, M., and Götschel, S., 2012, State trajectory compression for optimal control with parabolic PDEs: SIAM Journal on Scientific Computing, **34**, no. 1, A161–A184.

Yoshizawa, K., and Kennett, B. L. N., 2002, Determination of the influence zone for surface wave paths: Geophysical Journal International, **149**, no. 2, 440–453.

Zhu, H., Bozdağ, E., Peter, D., and Tromp, J., 2012, Structure of the European upper mantle revealed by adjoint tomography: Nature Geoscience, **5**, no. 7, 493–498.