SMIwiz: An integrated toolbox for multidimensional seismic modelling and imaging

Pengliang Yang[a]

[a] *School of Mathematics, Harbin Institute of Technology, 150001, Harbin, China*

ARTICLE INFO

ABSTRACT

This paper contributes an open source software - SMIwiz, which integrates seismic modelling, reverse time migration (RTM), and full waveform inversion (FWI) into a unified computer implementation. SMIwiz has the machinery to do both 2D and 3D simulation in a consistent manner. The package features a number of computational recipes for efficient calculation of imaging condition and inversion gradient: a dynmaic evolving computing box to limit the simulation cube and a well-designed wavefield reconstruction strategy to reduce the memory consumption when dealing with 3D problems. The modelling in SMIwiz runs independently: each shot corresponds to one processor in a bijective manner to maximize the scalability. A batchwise job scheduling strategy is designed to handle large 3D imaging tasks on computer with limited number of cores. The viability of SMIwiz is demonstrated by a number of applications on benchmark models.

CRediT authorship contribution statement

Pengliang Yang: Conceptualization, methodology, numerical implementation, validation and the preparation of the paper.

1. Introduction

Seismic wave modelling and imaging are fundamental tools to understand the structure of the Earth in complex media. Numerical simulation of wave propagation allows us to predict the response of the Earth to the given media properties. An efficient and memory frugal implementation of wave modelling is also the kernel of seismic imaging algorithms, such as reverse time migration (RTM) (Baysal et al., 1983; McMechan, 1983) and full waveform inversion (FWI) (Tarantola, 1984; Lailly, 1983).

The simulation of seismic waves can be done using different methods (Virieux et al., 2011), i.e., finite difference method (Virieux, 1984, 1986; Levander, 1988), finite element method (Smith, 1975), pseudo-spectral method (Carcione, 2010), spectral element method (Seriani and Priolo, 1994; Komatitsch and Vilotte, 1998) and discontinuous Garlerkin method (Hu et al., 1999; Dumbser and Käser, 2006). Among them, finite difference time-domain (FDTD) method gained its wide popularity due to small memory consumption, simple implementation and superior efficiency in a broad range of applications. The hardware acceleration such as graphics processing units (GPU) and OpenCL is particularly suitable for FDTD approaches to do numeric modelling (Abdelkhalek et al., 2009).

Wave modelling and imaging are computationally intensive tasks. There are many open source codes implementing them in a number of independent ways. Great effort has been spent on efficient implementation of wave simulation. Bohlen (2002) have conducted a parallel 3D viscoelastic seismic modelling using FDTD. Komatitsch and Martin (2007) developed a collection of Fortran90 programs - SEISMIC_CPML - to solve 2D and/or 3D isotropic or anisotropic elastic, viscoelastic or poroelastic wave equation using FDTD combined with convolutional perfectly matched layer (PML) conditions. SOFI2D and SOFI3D (Bohlen et al., 2015) are 2D and 3D seismic modelling coded in C programming language. 3D elastic wave equations are also solved via FDTD method on parallel GPU devices (Michéa and Komatitsch, 2010; Weiss and Shragge, 2013).

Using these modelling engines, two types of imaging are routinely conducted in exploration geophysics: one is the migration of seismic reflections to form an image of the subsurface, the other is building the velocity

ORCID(s):

model according to observed seismograms. RTM has replaced a number of one-way migration techniques and becomes the standard technology for structural imaging (Etgen et al., 2009). RTM assumes a known velocity with sufficient accuracy, which may be acquired using FWI-based velocity model building (Virieux and Operto, 2009).

Inspired by the good convergence of quasi-Newton optimization, Brossier (2011) released TOY2DAC code for 2D frequency domain FWI. To study the influence of model parametrization, Köhn et al. (2012) have developed 2D time-domain isotropic (visco)elastic finite-difference modeling and FWI code DENIS for P/SV-waves. Time-domain RTM and FWI are implemented on GPU devices with an effective boundary saving strategy (Yang et al., 2014, 2015). Unfortunately, these implementations were limited in 2D while the potential of MPI parallelization over distributed memory has not yet been fully exploited. Fabien-Ouellet et al. (2017) have developed a SeisCL software to utilize OpenCL to do time-domain seismic modeling in viscoelastic media for FWI. The modelling was combined with MPI for domain decomposition. The test of SeisCL over different hardware architectures demands device specific optimization for good performance. Elastic RTM and FWI, despite their theoretic importance, continue to be academic exercises, and remains to be explored for commercial usage.

In this work, we focus on developing a FDTD-based open source software `SMIwiz` for multidimensional seismic modelling and inversion. Based on acoustic wave equation, the modelling engine is coded in C, running over a number of parallel processors using distributed memory thanks to MPI-based high performance computing architecture. Each modelling job for independent sources is parallelized with OpenMP for speedup using local shared memory. `SMIwiz` builds a machinery to do both 2D and 3D simulation in a consistent manner. The package features a number of computational recipes for efficient calculation of imaging condition and inversion gradient: a dynamic evolving computing box to limit the simulation cube, a well-designed wavefield reconstruction strategy to reduce the memory consumption typically in 3D, and a batchwise job scheduling strategy giving flexibility to run over heterogeneous computing architectures.

2. Methodology

2.1. Forward modelling

The first order isotropic acoustic wave equation is given by

$$\begin{cases} \rho(\mathbf{x})\partial_t \mathbf{v}(\mathbf{x},t) + \nabla p(\mathbf{x},t) = 0, & \mathbf{x} \in X, t \in [0,T] \\ \partial_t p(\mathbf{x},t) + \kappa(\mathbf{x})\nabla \cdot \mathbf{v}(\mathbf{x},t) = f_p(\mathbf{x},t), & \mathbf{x} \in X, t \in [0,T] \end{cases} \tag{1}$$

where $\mathbf{v} = [v_x, v_y, v_z]^T$ is the particle velocity including three directional components v_x, v_y and v_z; p denotes the acoustic pressure; f_p stands for an explosive source term; ρ is the density; κ is the bulk modulus which can be expressed via density ρ and wave speed V_p, i.e., $\kappa = \rho V_p^2$. Clearly, the pressure and particle velocity fields are functions of both time t and space $\mathbf{x}$, while the medium parameters ρ and κ vary only in space.

To do numerical simulation, we consider the simplest leap-frog finite difference method on staggered grid (Virieux, 1986; Levander, 1988; Graves, 1996). Assume the time are discretized as $t_n = n\Delta t$, while the space variables are also discretized using equal grid spacing, i.e., $x_i = i\Delta x$, $y_j = j\Delta y$ and $z_k = k\Delta z$. As depicted in Figure 1, the pressure sits at integer grid at integer time step, that is, $p(x_i, y_j, z_k, t_n) = p(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$; the particle velocities are defined at half integer time step and shifted half grid along x, y and z directions, i.e., $v_x(x_{i+1/2}, y_j, z_k, t_{n+1/2})$, $v_y(x_i, y_{j+1/2}, z_k, t_{n+1/2})$ and $v_z(x_i, y_j, z_{k+1/2}, t_{n+1/2})$. For brevity, we denote them as follows

$$\begin{cases} p(x_i, y_j, z_k, t_n) & \equiv p|_{i,j,k}^n \\ v_x(x_{i+1/2}, y_j, z_k, t_{n+1/2}) & \equiv v_x|_{i+1/2,j,k}^{n+1/2} \\ v_y(x_i, y_{j+1/2}, z_k, t_{n+1/2}) & \equiv v_y|_{i,j+1/2,k}^{n+1/2} \\ v_z(x_i, y_j, z_{k+1/2}, t_{n+1/2}) & \equiv v_z|_{i,j,k+1/2}^{n+1/2} \end{cases} \tag{2}$$

Based on equation (1) and the homogeneous initial condition, the fields can be advanced from one time

---
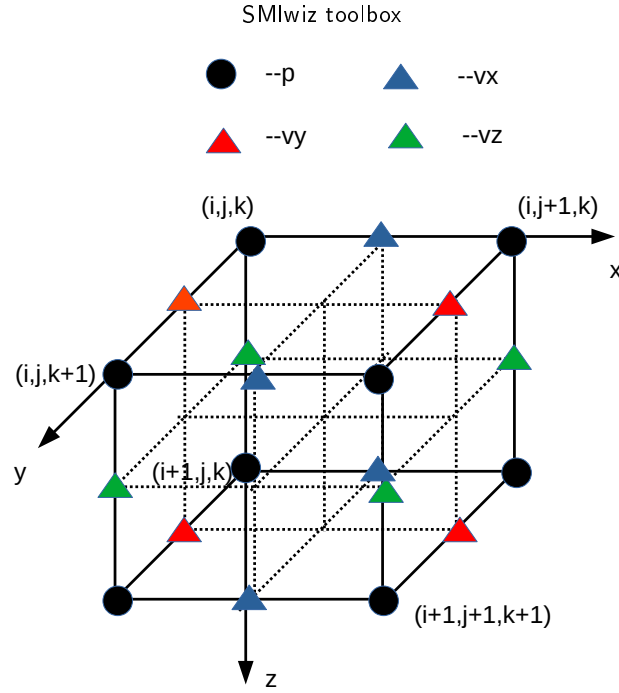
Figure 1: Pressure and particle velocities on the staggered grid

step to the next:

$$
\begin{cases}
v_x|_{i+1/2,j,k}^{n+1/2} & = v_x|_{i+1/2,j,k}^{n-1/2} - \Delta t \rho_{i+1/2,j,k}^{-1} \partial_x p|_{i+1/2,j,k}^{n}, \\
v_y|_{i,j+1/2,k}^{n+1/2} & = v_y|_{i,j+1/2,k}^{n-1/2} - \Delta t \rho_{i,j+1/2,k}^{-1} \partial_x p|_{i,j+1/2,k}^{n}, \\
v_z|_{i,j,k+1/2}^{n+1/2} & = v_z|_{i,j,k+1/2}^{n-1/2} - \Delta t \rho_{i,j,k+1/2}^{-1} \partial_x p|_{i,j,k+1/2}^{n}, \\
p|_{i,j,k}^{n+1} & = p|_{i,j,k}^{n} - \Delta t \kappa_{i,j,k}(\partial_x v_x + \partial_y v_y + \partial_z v_z)|_{i,j,k}^{n+1/2} + \Delta t f_p|_{i,j,k}^{n+1/2},
\end{cases}
\tag{3}
$$

where the first spatial derivatives are approximated using high order finite difference schemes over staggered grid. For x direction, the first derivatives approximated by finite difference of order $O(2N)$ are

$$
\begin{cases}
\partial_x v_x|_{i,j,k} = \sum_{l=1}^{N} a_l(v_x|_{i+(2l-1)/2,j,k} - v_x|_{i-(2l-1)/2,j,k})/\Delta x \\
\partial_x p|_{i+1/2,j,k} = \sum_{l=1}^{N} a_l(p|_{i+l,j,k} - p|_{i-l+1,j,k})/\Delta x
\end{cases}
\tag{4}
$$

where a_l, $l = 1, \cdots, N$ are FD coefficients which can be found in Fornberg (1998). Based on von Neumann analysis, the following Courant-Friedrichs-Lewy (CFL) stability condition must be satisfied

$$
\Delta t V_{p\max} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \sum_{l=1}^{N} |a_l| \leq 1.
\tag{5}
$$

To mimic the waves propagating to infinity, we add 20 perfectly matched layers (PMLs) surrounding the domain of interest (Komatitsch and Martin, 2007). This avoids the artificial reflections due to the truncation of the computing mesh. On the top of the boundary, we have implemented free surface boundary using the image method, by asymmeric mirroring of the pressure and symmetric mirroring of the vertical component of the particle velocity according to Levander (1988). The injection of the seismic source and extraction of the modelled data at arbitrary location is performed by Kaiser-windowed sinc interpolation (Hicks, 2002).

2.2. Nonlinear full waveform inversion

Full waveform inversion aims to iteratively reconstruct the true subsurface parameters by minimizing the error between the observed seismogram d and the synthetic data u predicted according to the wave equation

and the given parameters. The misfit functional for FWI is defined in least-squares sense

$$J(m) = \frac{1}{2} \sum_s \sum_r \int_0^T \left(u(\mathbf{x}_r, t; \mathbf{x}_s) - d(\mathbf{x}_r, t; \mathbf{x}_s) \right)^2 \mathrm{d}t, \tag{6}$$

where the inversion parameter m is a function of ρ and κ; $u(\mathbf{x}_r, t; \mathbf{x}_s)$ and $d(\mathbf{x}_r, t; \mathbf{x}_s)$ are synthetic and observed seismograms recorded at receiver $\mathbf{x}_r$ associated with source $\mathbf{x}_s$. The gradient of the misfit with respect to density and bulk modulus are

$$\begin{cases} \frac{\partial J(m)}{\partial \rho} = \int_0^T \bar{\mathbf{v}}^{\mathrm{H}} \partial_t \mathbf{v} \mathrm{d}t \\ \frac{\partial J(m)}{\partial \kappa} = \frac{1}{\kappa} \int_0^T \bar{p} \nabla \cdot \mathbf{v} \mathrm{d}t \end{cases}, \tag{7}$$

where $(\bar{\mathbf{v}}, \bar{p})$ are the adjoint fields satisfying

$$\begin{cases} \rho \partial_t \bar{\mathbf{v}} + \nabla \bar{p} = -\delta d_{\mathbf{v}}, \\ \partial_t \bar{p} + \kappa \nabla \cdot \bar{\mathbf{v}} = -\kappa \delta d_p, \end{cases} \tag{8}$$

in which $\delta d_{\mathbf{v}}$ and δd_p are data residuals corresponding to particle velocity and pressure components. Since the adjoint equation (8) shares exactly the same structure as the forward state equation (1) except the source terms, we can use the same solver to model $(\bar{\mathbf{v}}, \bar{p})$. The inversion may be carried out using different model parametrizations, for example, (V_p, ρ) or (V_p, I_p) where the P-wave velocity V_p and impedance I_p are related to density via $I_p = \rho V_p$. The misfit gradient with respect to other parameters can easily be deduced from equation (7) thanks to the chain rule. We refer to Yang (2023c) to check out the derivation of the specific expressions and the mathematical details. In order to minimize the misfit $J(m)$, FWI iterates over the model parameter via

$$m^k = m^{k-1} + \alpha \delta m, \tag{9}$$

where α is a proper step length estimated through line search method (Nocedal and Wright, 2006) at the k-th iteration; δm is the descent direction determined based on the gradient of the misfit functional $\nabla J(m)$. To achieve good convergence in nonlinear optimization, δm can be computed by limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm or solving the normal equation using Newton-CG method (Yang et al., 2018). In practice, `SMIwiz` utilizes the two-loop recursion LBFGS implementation to find a descent direction δm (Nocedal, 1980). A pseudo code to implement the nonlinear FWI for model update is shown in Algorithm 1.

Algorithm 1 Nonlinear FWI for updating model parameters m

1: Given initial model: m_0
2: **for** $k = 1, \cdots, niter$ **do**
3:      calculate misfit gradient: ∇J
4:      estimate approximate Hessian $\tilde{H}$ using LBFGS algorithm
5:      estimate descent direction: $\delta m = -\tilde{H}^{-1} \nabla J$
6:      find a good step size α by line search
7:      update model $m_k := m_{k-1} + \alpha \delta m$
8: **end for**

2.3. Reverse time migration

Given a sufficiently accurate background velocity model, one can migrate the seismic events back to its true location where the scattering happens. A defacto technique to do it is the reverse time migration (RTM), which dates back to Baysal et al. (1983); McMechan (1983) four decades ago. With the advance of computer power, RTM has become a mature and standard imaging technique to produce a structural image of the subsurface. RTM overcomes the limitation of one-way imaging methods in handling the turning

waves, high dip and lateral velocity contrast (Zhang and Zhang, 2008). It has a wide range of applications in both exploration geophysics (Zhang and Sun, 2009) and global seismology (Zhang et al., 2023). The RTM image is simply the cross-correlation of the source field $p_s(\mathbf{x}, t; \mathbf{x}_s)$ (computed forward in time with a known source) and the receiver field $p_r(\mathbf{x}, t; \mathbf{x}_s)$ (simulated by simultaneously backpropagating the reflection data from multiple receivers)

$$I(\mathbf{x}) = \sum_s \int_0^T p_r(\mathbf{x}, t; \mathbf{x}_s) p_s(\mathbf{x}, t; \mathbf{x}_s) \mathrm{d}t. \tag{10}$$

To enhance the structural image at depth, one often consider the normalized imaging condition (Yang et al., 2014), in which the images of each source are normalized by source energy/illumination. These imaging conditions easily suffer from low-frequency artifacts (the imprint of the wave path that the total traveltime of different waves coincides). The application of a Laplace filtering helps to eliminate these artifacts (Zhang and Sun, 2009).

Note that the computation of imaging condition in equation (10) follows exactly the same procedure as FWI gradient building, except the injection of the adjoint source is the observed data instead of data residual. Algorithm 2 summarizes the steps for the evaluation of FWI function value and gradient (as well as RTM image). Algorithm 2 includes a number of useful tricks to achieve good efficiency. We will detail them in Sections 3 and 4.

3. Software description

3.1. Generic design

SMIwiz performs 2D and 3D seismic modelling and imaging which runs over single CPU processor as well as multiple CPU processors in parallel. It requires

- FFTW for fast Fourier transform;

- MPICH or OpenMPI for multi-processor parallelization over distributed memory architecture.

It defines a number of pointers to ensure a clean and consistent implementation:

- The pointer of type `acq_t` are specific to acquisition geometry. It reserves the number of source and receiver for each process as well as their coordinates in x-, y- and z- directions.

- The pointer of type `sim_t` is the simulator for forward and adjoint modelling. There are huge number of parameters inside, which will be detailed shortly.

- The pointer of type `fwi_t` is the inversion/imaging data structure, maintaining the number of unknown parameters and their bounds, the choice of parametrization and preconditioner, misfit value etc.

- The pointer of type `opt_t` is dedicated to nonlinear optimization. The relevant parameters for LBFGS, such as the number of iterations, the memory length of LBFGS, the step length of line search, as well as the vectors for storing the inversion parameters and gradients. In the framework of SMIwiz, we have re-implemented LBFGS and Newton-CG algorithms in C programming language following a reverse communication spirit (Métivier and Brossier, 2016).

The objects in these types are coined `acq`, `sim`, `fwi` and `opt` for clarity.

Besides the data pointers listed above, the index and total number of the processors are global variables defined by `iproc` and `nproc`. The header files are sitting in `SMIwiz/include`. The main part of the software is in `SMIwiz/src` and coded in C programming language to achieve high performance computing. We can launch `make` to compile the code following the given `Makefile`, which may be adapted depending on the user's environment and the path of the software packages. The compiled executable will be coined `SMIwiz` in the folder `SMIwiz/bin`. A number of unit tests, with the names of the folder `SMIwiz/run_XXXX`, can be considered as the running templates to be adapted to fit the user's needs of modelling and imaging.

Similar to our 3D electromagnetic modelling and inversion software `libEMMI` (Yang, 2023b,a), SMIwiz automatically parses the argument list based on the parameter specified in the token. It checks out the value

---

Algorithm 2 Evaluation of FWI function and gradient/RTM image

---

```
 1: check_cfl(sim);
 2: cpml_init(sim);
 3: extend_model_init(sim);
 4: fdtd_init(sim, 1);//flag=1, incident field allocation
 5: fdtd_init(sim, 2);//flag=2, adjoint field allocation
 6: fdtd_null(sim, 1);//flag=1, incident field initialization
 7: fdtd_null(sim, 2);//flag=2, adjoint field initialization
 8: decimate_interp_init(sim);
 9: extend_model(sim);
10: computing_box_init(acq, sim, 0);
11: computing_box_init(acq, sim, 1);
```
12: **for** $it = 0, \cdots, nt - 1$(forward modelling) **do**
```
13:     decimate_interp_bndr(sim, 0, it);//interp=0
14:     fdtd_update_v(sim, 1, it, 0);//update fwd v
15:     fdtd_update_p(sim, 1, it, 0);//update fwd p
16:     inject_source(sim, acq, sim->p1, sim->stf[it]);//inject fwd source
17:     extract_wavefield(sim, acq, sim->p1, sim->dcal, it);
```
18: **end for**
```
19: compute adjoint source (and FWI misfit - MPI_Allreduce)
```
20: **for** $it = nt - 1, \cdots, 0$(adjoint modelling) **do**
```
21:     inject_adjoint_source(sim, acq, sim->p2, sim->dres, it);//add adj source
22:     fdtd_update_v(sim, 2, it, 1);//update adj v
23:     fdtd_update_p(sim, 2, it, 1);//update adj p
24:     inject_source(sim, acq, sim->p1, sim->stf[it]);//subtract source
25:     fdtd_update_p(sim, 1, it, 0);//reconstruct fwd p
26:     fdtd_update_v(sim, 1, it, 0);//reconstruct fwd v
27:     decimate_interp_bndr(sim, 1, it);//interp=1
28:     integrate FWI gradient/RTM image
```
29: **end for**
```
30: cpml_close(sim);
31: extend_model_close(sim);
32: fdtd_close(sim, 1);
33: fdtd_close(sim, 2);
34: decimate_interp_close(sim);
35: computing_box_close(sim, 0);
36: computing_box_close(sim, 1);
37: merge all local FWI gradient/RTM image - MPI_Allreduce
```

---

of each keyword after the equal sign =. Multiple values for the same keyword are separated with comma. Different keywords will be separated using the space in the string.

All relevant parameters and the input files in binary or ASCII formats can be set up in a shell script, e.g., run.sh. The first job run.sh does is to write the input parameters in a text file inputpar.txt. These parameters and input files will be easily adapted depending on the user's setup. Then, SMIwiz is activated in run.sh using MPI

```
mpirun -n 4 ../bin/SMIwiz $(cat inputpar.txt)
```

where the number 4 indicates 4 shots of the seismic data are simultaneously simulated and/or inverted. To facilitate the job submission, the modelling and inversion will be launched via

```
bash run.sh
```

3.2. Parallelization

SMIwiz distributes jobs to different CPU cores in a bijective manner using MPI parallelization, so that the modelling for each shot are completely independent, therefore no collective communication among them. The numerical simulation of each shot has been parallelized using OpenMP thanks to local shared memory. Each processor will pick up their own source-receiver geometry from the same acquisition file.

SMIwiz has an argument `shot_idx` to accept a list of shot indices so that specific shots will be modelled or utilized for imaging. For example, `shot_idx=5,13,18,23` indicates shots 5,13,18,23 are modelled/imaged. The total number of MPI processors launched should be 4 to match the number of the shots in `shot_idx`. If `shot_idx` is not specified explicitly, the sources indexed from `0001` to `XXXX` will be consecutively handled by default, where `XXXX` is the number of processors launched.

In an imaging task such as RTM and FWI, we will perform a reduction to collect each piece of gradient/image information from different shots. By doing so, the amount of communication among processors is minimized to maximize the scalability of the imaging computations. The gradient building following this idea is shown in the flowchart in Figure 2.

3.3. Running different jobs

Depending on the value of `mode` parameter, SMIwiz runs different jobs. Setting `mode=0` performs forward modelling based on the given density and velocity parameters. Assuming the observed data in binary are available, one can do FWI (`mode=1`) or output the misfit gradients (`mode=4`). When the seismic source is unknown, one may estimate the source signature before inversion via the method proposed by Pratt (1999) (`mode=5`). Such a design eases the extension of the new functionalities with another different `mode` value for more advanced processing.

3.4. Acquisitions geometry

To specify the survey layout, an acquisition files in ASCII format must be given to parameter `acquifile=` to prescribe the source and receiver positions. With a text file `acquifile=acqui.txt`, the source/receiver file is in the following form:

```
   z          x          y       azimuth      dip       src/rec(0/1)
   5.0        100.       0        0            0         0
   20.0       100.       0        0            0         1
   20.0       120.       0        0            0         1
   20.0       140.       0        0            0         1
   ...
   5.0        200.       0        0            0         0
   20.0       100.       0        0            0         1
   20.0       120.       0        0            0         1
   20.0       140.       0        0            0         1
   ...
```

By default, the first columns gives the z coordinate, while the second and the third give the x and y coordinates. For a 2D simulation, the y coordinate will have no effect. At present, the space location (`x,y,z`) in the unit of meters are active parameters, while the `azimuth` and `dip` parameters are not functional. The last column is the flag of sources and receivers: 0 indicates a source and 1 indicates a receiver. After a source line, all receiver coordinates associated with this source will follow. Once a new source line comes, the following lines direct to the receivers recording waves from this new source. Since all acquisition geometry information will be stored in the same file, such an ASCII file can be large, typically for 3D configurations.

3.5. Binary files for velocity, density and source time function

SMIwiz asks for three binary files for the detailed information on the medium properties and the source time function, which can be specified through the following lines:

```
vpfile=vp
rhofile=rho
stffile=fricker
```
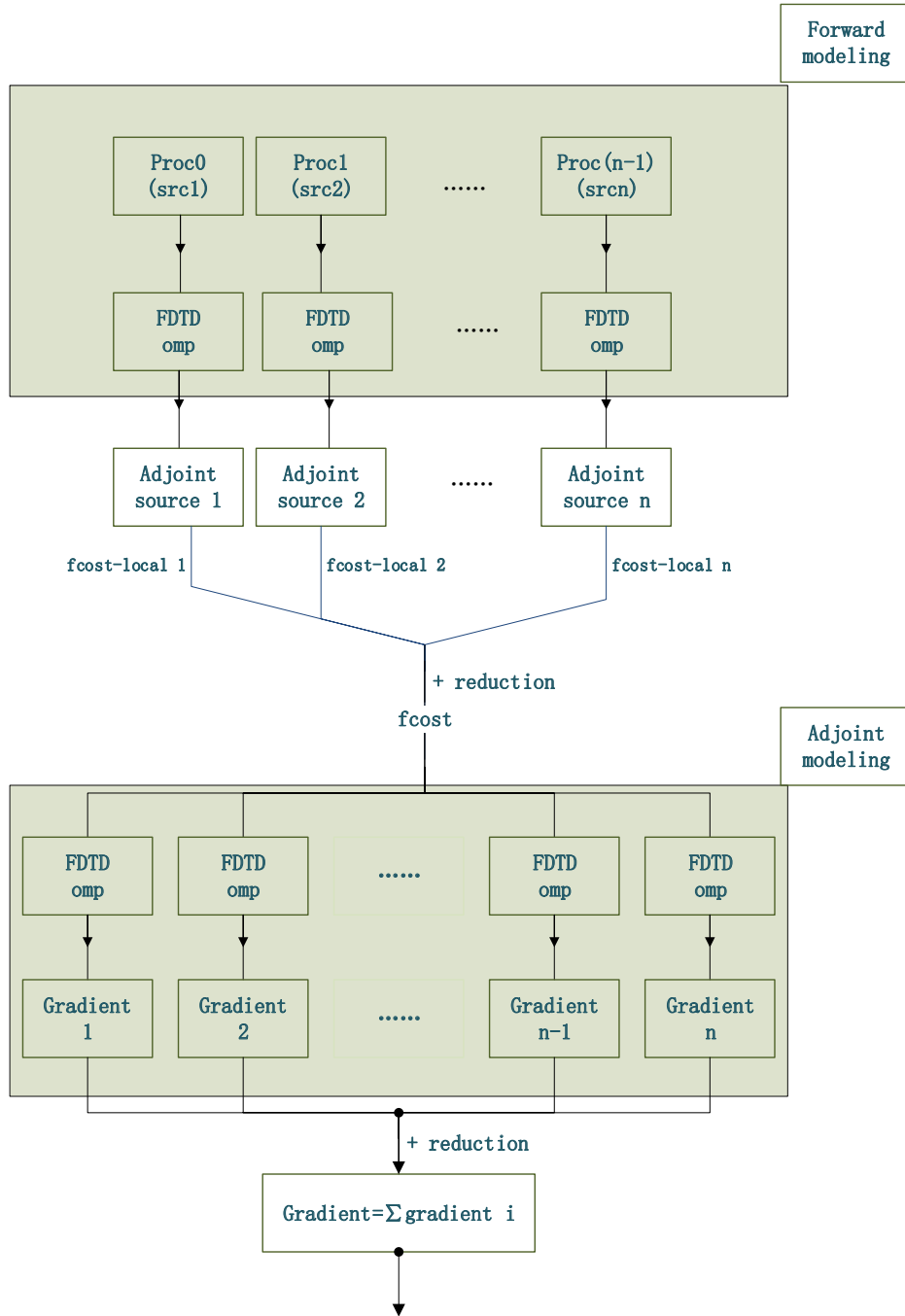
Figure 2: A workflow to build FWI gradient in `SMIwiz`

where the velocity `vp` and the density `rho` of the subsurface must match the dimension of the simulation domain. The argument `stffile` allows the user to work with their own source time function for modelling and inversion. In the above, `fricker` is a Ricker wavelet.

3.6. Parameters in the simulation domain

The size of the finite difference mesh will be specified in z, x and y directions with the dimension parameters `n1`, `n2` and `n3`. The grid spacing are given by the arguments `d1`, `d2` and `d3`. We add `nb` PML layers on each side/face of the model. The simulation time can be determined using the number of time steps `nt` and the temporal interval `dt`. The simulation can be carried out with `freesurf=1` or without free surface `freesurf=0`. The central frequency of the wavelet must also be given. For example, a 3D modelling for 2000 time steps with stepsize $\Delta t = 0.004$ s within a 3D mesh of size (`nx`, `ny`, `nz`)=(81, 401, 401) and grid spacing $\Delta z = \Delta y = \Delta x = 50$ m with free surface using a source with central frequency around 6 Hz in the shell script `run.sh` can be configured via

```
freesurf=1 //free surface boundary condition
fm=6 //centre frequency of the wavelet
nt=2000 //number of time steps
dt=0.004 //temporal sampling
nb=20 //number of layers for absorbing boundaries
n1=81 //size of input FD model nz
n2=401 //size of input FD model nx
n3=401 //size of input FD model ny
d1=50 //grid spacing dz of the input FD model
d2=50 //grid spacing dx of the input FD model
d3=50 //grid spacing dy of the input FD model
```

In case of a 2D simulation, the parameters `n3` and `d3` in the shell script can be removed.

3.7. Inversion setup

Our FWI uses bounded LBFGS algorithm. The number of nonlinear iterations are given by parameter `niter`. At each iteration, it uses line search algorithm to determine the step length. The maximum number of possible line search can be specified by `nls`. The memory length of LBFGS is given by `npair`. By default, no preconditioning is applied (`preco=0`). The depth preconditioner can be turned on by setting `preco=1`, while the pseudo-Hessian preconditioner can be actived using `preco=2`. By default, we activate bounds (`bound=1`) with given minimum and maximum values for velocity (`vpmin` and `vpmax`) and density (`rhomin` and `rhomax`). The parametrizations of $V_p - \rho$ and $V_p - I_p$ are referred to as family 1 and family 2, respectively. The index of parameters are given by `idxpar`. The bounds for inversion parameters will be determined depending on the family of the parametrization considering the log transform of the physical parameters. The following lines set up an inversion of 50 iterations of preconditioned LBFGS algorithm to invert for V_p and I_p: Every update are bounded by the minimum and maximum wave speed 2000 m/s and 6500 m/s, as well as the minimum and maximum density 1000 kg/m^3 and 2800 kg/m^3.

```
niter=50 //number of iterations using LBFGS
nls=20 //number of line search per iteration
npair=5 //memory length in LBFGS
preco=1 //precondition

npar=2  //simultaneously invert 2 parameters - Vp and Ip
idxpar=1,2 //index of parameter to be inverted
bound=1 //bound the inversion parameters
vpmin=2000 //minimum value for wave speed
vpmax=6500 //maximum value for wave speed
rhomin=1000 //minimum value for density
rhomax=2800 //maximum value for density
```

Note that `npar=1` corresponds to monoparameter inversion while `npar=2` corresponds to multiparmeter inversion. A binary file providing bathymetry information may be specified:

```
bathyfile=fbathy
```

This file `fbathy` should contain `n1*n2` values defining the depth of the seafloor. During the inversion, the gradient above the bathymetry will be set to 0 to avoid model update in the sea water. As the sources and receivers are normally placed above the seafloor, this mutes the strong acquisition footprint in the inverted model.

3.8. Accuracy control

The finite difference scheme of orders $O(\Delta t^2, \Delta x^4)$ and $O(\Delta t^2, \Delta x^8)$ can be specified by `order=4` (default value) or `order=8`. To ensure a consistent accuracy, the radius of the Hicks interpolation operator is set to be the same as half of the order `ri=order/2`, to ensure that the same number of points are used for interpolation and finite difference stencil. At the outset of numeric modelling, `SMIwiz` always performs a sanity check on the CFL stability condition in equation (5) and the dispersion requirement for sufficient points per wavelength. `SMIwiz` will be automatically terminated if these conditions are not satisfied.

3.9. Data weighting

We can apply different weights for data with varying offset. The parameter `dxwdat` allows us to configure intervals of offset sampling while different weights can be given by `xwdat`. As shown in Figure 3a, a complete prestack shot profile has been modelled from Marmousi model.As illustrated in Figure 3b, application of offset weighting leads to the dimming of near offset using

```
dxwdat=200 //dx for data weighting
xwdat=0,0.3,0.7,1,1 //weights for dx increment
```

RTM mainly uses reflections to retrieve high resolution structural image of subsurface. Such a purpose can be achieved by muting the front part of the data. Since the main data driving the model update is the first arrivals, transmitted energy including direct waves and refractions are of particular importance in FWI. Figure 3c and 3d examplifies the front and tail muting over the above shot profile. When muting is conducted, we use `ntaper` points to have a smoothing transition between muted and unmuted part of the data. The typical parameters for such data weighting look like

```
ntaper=20 //number of points for tapering
xmute1=0,3306,7534 //front mute line, x coordinate in m
tmute1=0.3,2.3,5.0 //front mute line, t coordinate in s
xmute2=0,3306,7534 //tail mute line, x coordinate in m
tmute2=0.3,2.3,5.0 //tail mute line, t coordinate in s
```

The data weighting gives us great flexibility for preprocessing of the data in different imaging tasks.

3.10. The output from `SMIwiz`

The modelled data will be named as `dat_XXXX` where `XXXX` is the index of the shots. The cross-correlation RTM image and normalized cross-correlation image will be `image_xcorr` and `image_normalized_xcorr`. At each iteration, the updated model parameters and FWI gradient will be printed out, with names `param_final` and `gradient_final` (latest printing will overwrite existing file). In order to track the history of the inversion, all these intermediate gradients and updated parameters are also collected under the file names `param_iter` and `gradient_iter`. All these files are in binary format, implying that the user should be aware of the size of them to check out the results.

4. Judicious implementation

4.1. Computing box

`SMIwiz` aims to do minimum amount of floating point operations to reach an intact wavefield simulation. To do that, a cubic computing box which changes dynamically with the evolution of time has been designed.
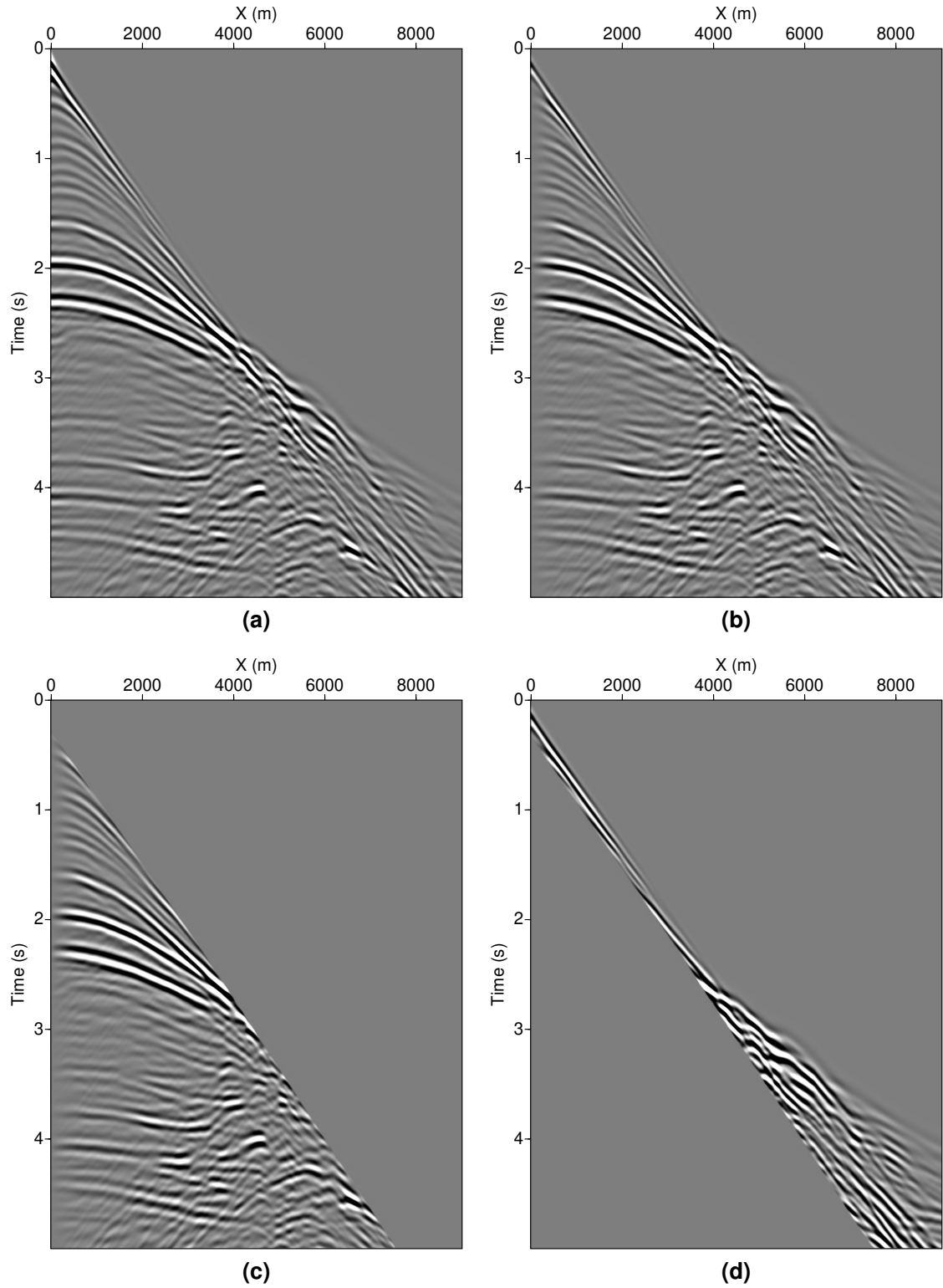
Figure 3: (a) A complete prestack seismic shot profile modelled from Marmousi model; (b) near offset data are tapered (`muteopt=0`, no muting); (c) front muting (`muteopt=1`), leaving only the reflections; (d) tail muting (`muteopt=2`), leaving only the first arrivals and diving waves.

The minimum and maximum bounds for the grid index are determined based on the outermost wavefront, which is calculated based on the evolved number of time steps and maximum wave speed. In a forward modelling mode, the computing box is expanding. At the very beginning, the waves are confined in a small local cube, therefore requiring few gridpoints for finite difference; with the progress the modelling, the waves gradually spread into much large domain, therefore involving more gridpoints for modelling at each time step, as shown in Figure 4. In adjoint simulation, the computing box for the adjoint field is determined based on the outermost receivers, while the computing box for the forward field is shrinking backwards in time.
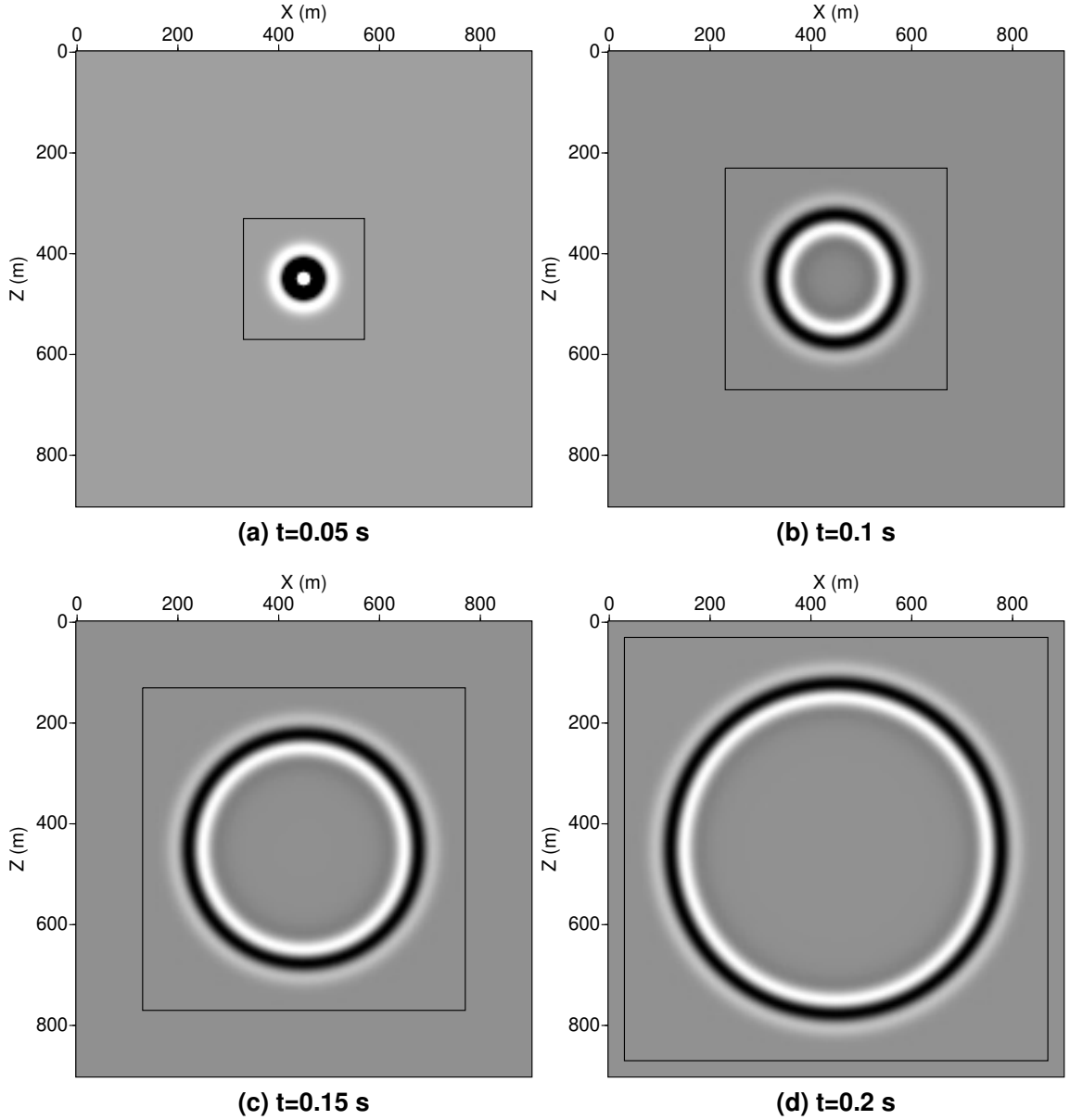
Figure 4: The size of computing box (marked by black box) expands with the evolution of the wavefield at each time step: (a) t=0.05 s; (b) t=0.1 s; (c) t=0.15 s; (d) t=0.2 s.

Let us remark that the efficiency of the application of computing box depends on where the seismic source is located. If the source is closer to one side of the model than other sides, the edge/face (in 2D/3D) of the computing box on this side will hit the boundary of the model first and remain unchanged. The worse case is that the source resides in the center of the model: in this case the edges/faces (2D/3D) of the computing box on each side extends. The efficiency gain due to computing box is more significant when the model is large. The amount of computation time might be only a fraction of the case without applying the box in 3D.

4.2. Source wavefield reconstruction

A common significant issue in nonlinear FWI and RTM is the construction of an inversion gradient or an image, which requires simultaneously accessing forward and adjoint fields at every time step, cf. a detailed analysis of this problem in Yang et al. (2016a). Since the adjoint field has an opposite time order compared to forward field while storing all snapshots of the forward field at every time leads to huge amount of memory consumption and heavy IO traffic, the source wavefield is reconstructed by interpolating the decimated boundaries in backward time order. To drastically reduce the memory consumption, the boundaries of forward field is only stored at every `dr` time step, where the ratio `dr` depends on the velocity contrast with minimum value

$$\mathtt{dr} = \frac{5V_{p\,\max}}{V_{p\,\min}}. \tag{11}$$

Assume $V_{p\,\max} \geq 2V_{p\,\min}$, the default decimation rate is set to be `dr=10`. The method was initially validated in 2D case (Yang et al., 2016b). Despite the idea is no more new, combining decimation and interpolation for significant memory reduction in wavefield reconstruction is critical to work with 3D imaging applications. Here we present the first numerical demonstration of this method in 3D: As shown in Figure 5, a precise reconstruction of the source field in backward time order is feasible, despite the fact that only 1 sample out of 10 time steps of the boundaries is stored).

4.3. Batchwise job scheduling

Up to now, a number of features possessed by `SMIwiz` assumes sufficient number of computer memory and processors, so that seismic shots are processed in bijective fashion. For 3D large scale imaging applications, this can be done as long as a modern cluster is at hand. It becomes challenging to do given only a small workstation. This in fact can still be done in `SMIwiz` thanks to a batchwise job scheduling strategy. Each time, we submit multiple modelling/inversion jobs as a batch, corresponding to a small subset of the target sources. For RTM, we can sequentially loop over the observed seismic data, as all images associated with different sources should be summed together.

However, catching a subset of the shots close to each other in FWI may introduce a strong bias when computing the misfit gradient, which will finally drive the inversion towards an unexpected direction. To get rid of this bias, we introduce the so-called stochastic gradient descent (SGD) method. There is no need to do any modification in the code of `SMIwiz`. Instead, we design a random shuffling algorithm in the shell script to submit inversion jobs in batches. Each batch is responsible for one iteration of steepest descent update of the model, using a specific number of shots determined randomly. Starting from the given initial model, the updated model at each batch will be used as the initial guess for next batch.

Assume there are only 50 CPU cores available in a small workstation while the observed data has 400 shots. Direct submitting 400 shots to 50 processors is infeasible for `SMIwiz` to do FWI/RTM in this case. The following lines implement the batchwise SGD algorithm by first writing the indices (generate via random shuffling algorithm) of the shots into a file, then update the model parameters batch by batch according to the shot indices prescribed in the file. Note that the number of shots do not need to be a multiple of the batchsize. The following code will schedule the last batch according to its actual size.

```
batchsize=50 #each group has less than or equal to 50 shots
ntotal=400 #total number of shots used in the inversion
j=0
k=0
```

(a) forward t=0.72 s

(b) forward t=1 s

(c) reconstructed t=1 s

(d) reconstructed t=0.72 s
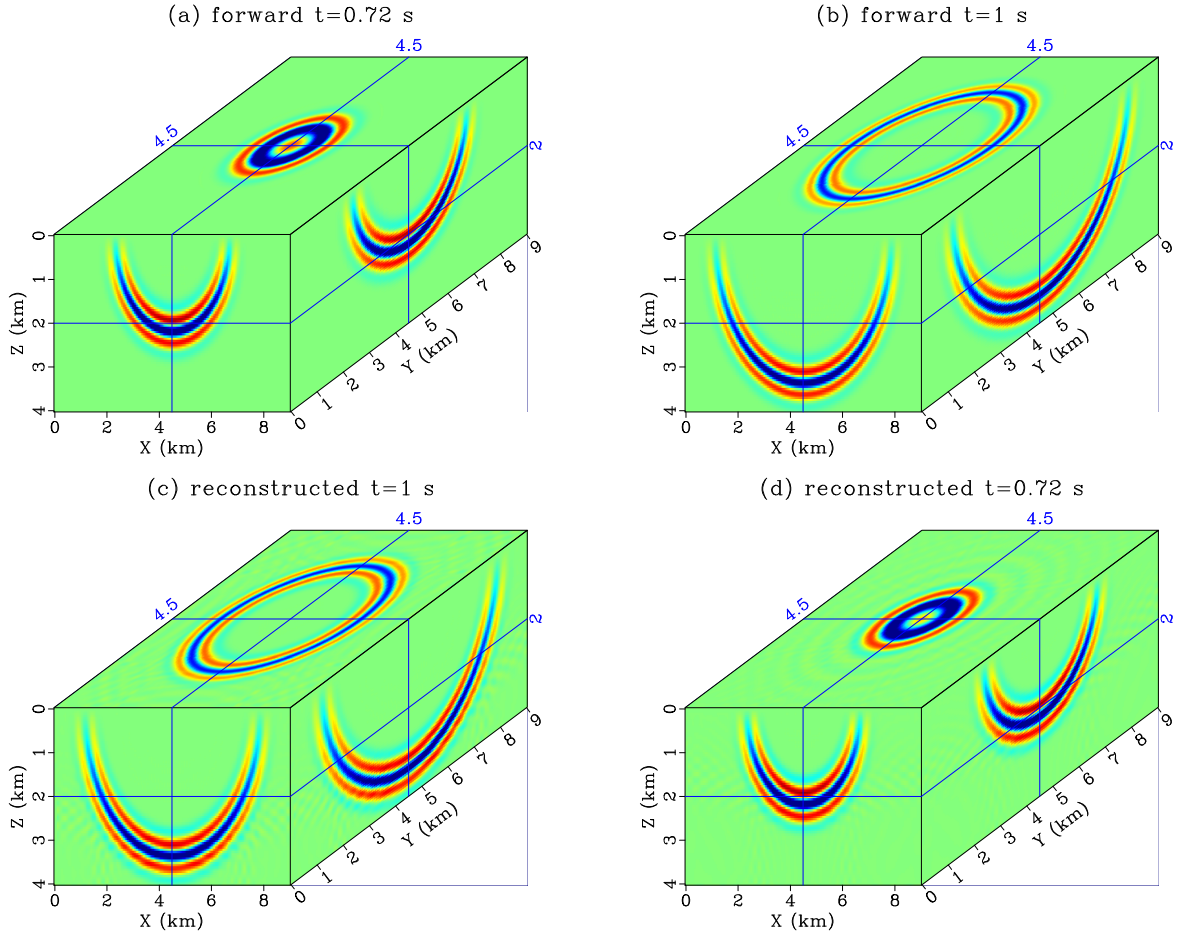
Figure 5: 3D wavefield reconstruction in a constant medium with a source placed in the center of the model: The forward modelling lasts for 2 s with recorded wavefields at (a) t=0.72 s and (b) t=1 s, then backward reconstructed from 2 s to 0 s, with recorded wavefields at (c) t=1 s and (d) t=0.72 s.

```
for i in `shuf -i 1-$ntotal -n $ntotal`; do
    if [ `expr $j % $batchsize` -eq 0 ]; then
        k=$[ $k + 1 ]
echo 'batch' $k
     echo $i > batch$k #write into a new file
    else
     echo $i >> batch$k #append into an existing file
    fi
    j=$[$j + 1]
    echo $i
done
nbatch=$k

for k in `seq 1 $nbatch`; do
    echo "#input parameters ...
" >inputpar$k.txt
    shots=`echo $(echo $(cat batch$k)) | tr ' ' ','`
    echo "shots="$shots >> inputpar$k.txt
```

```
    if [ $k -eq 1 ]; then
echo "vpfile=vp_init">>inputpar$k.txt
    else #use updated vp as initial model
echo "vpfile=param_final">>inputpar$k.txt
    fi

    j=0 #count the actual number of shots in this batch
    for i in $(cat batch$k); do
j=$[$j + 1]
    done
    echo batch$k, 'nshot='$j, 'shot_idx='$shots
    mpirun -n $j ../bin/SMIwiz $(cat inputpar$k.txt)
done
```

In the above script, the keyword `shot_idx` is the key parameter to accept the indices of the shots from random selection.

5. Numerical examples

5.1. Nonlinear multiparameter inversion

We begin our demonstration of `SMIwiz` by nonlinear multiparameter inversion of wave speed and density in classic Marmousi model (3 km × 9.2 km). The model is meshed into $nz \times nx = 151 \times 461$ samples with grid spacing $\Delta x = \Delta z = 20$ m. The source for this experiment is a Ricker wavelet with peak frequency 5 Hz. Starting velocity model is obtained by moderate smoothing of the true model; both the true and initial density models are derived from the velocity using Gardner's law, as shown in Figure 6. In total, 24 shots at 5 m depth are used in the inversion. We consider a fixed spead surface acquisition geometry: 451 receivers with equal spacing are deployed evenly at 20 m depth to record seismic data excited by each shot.
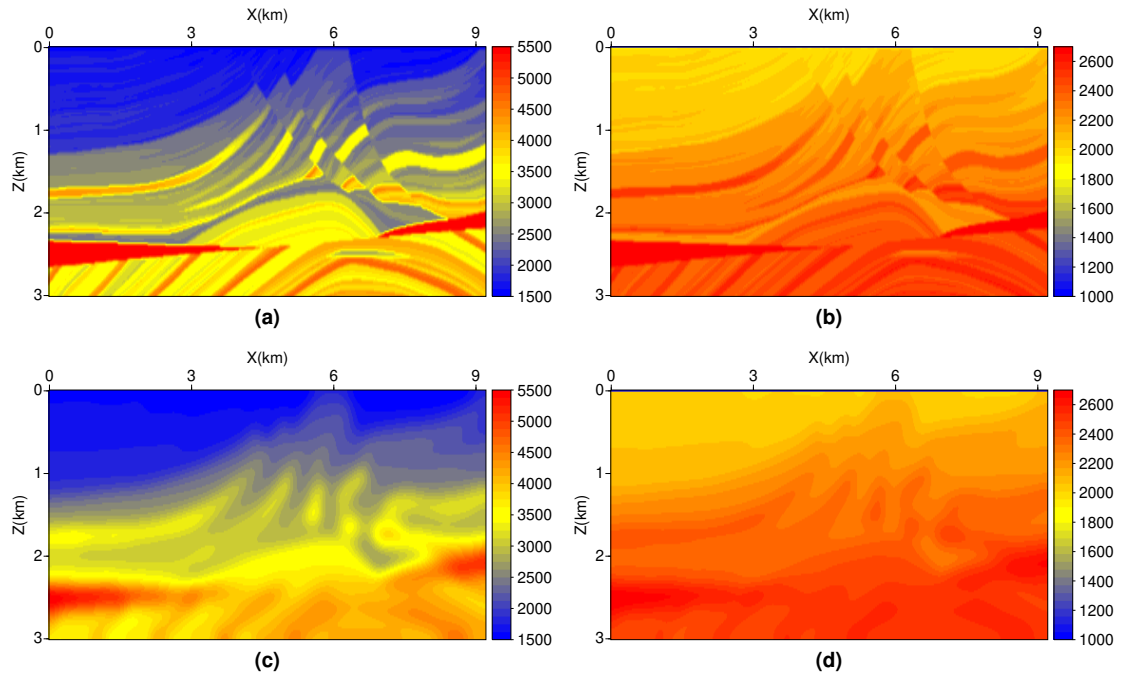
Figure 6: (a) True Marmousi velocity model; (b) Density model derived from true velocity model using Gardner's law; (c) Smoothed velocity as the starting model; (d) Density derived from smoothed velocity model using Gardner's law.

We perform multiparameter inversion for the Marmousi model in both family 1 ($V_p - \rho$ parametrization) and family 2 ($V_p - I_p$ parametrization) using 50 iterations of preconditioned LBFGS algorithm. According to the normalized misfit and the norm of the gradient Figure 7, the inversion under in family 1 and family 2 converges very well. Within limited number of iterations, a slightly faster convergence rate and lower final misfit value in $V_p - \rho$ parametrization have been reached in Figure 7a.
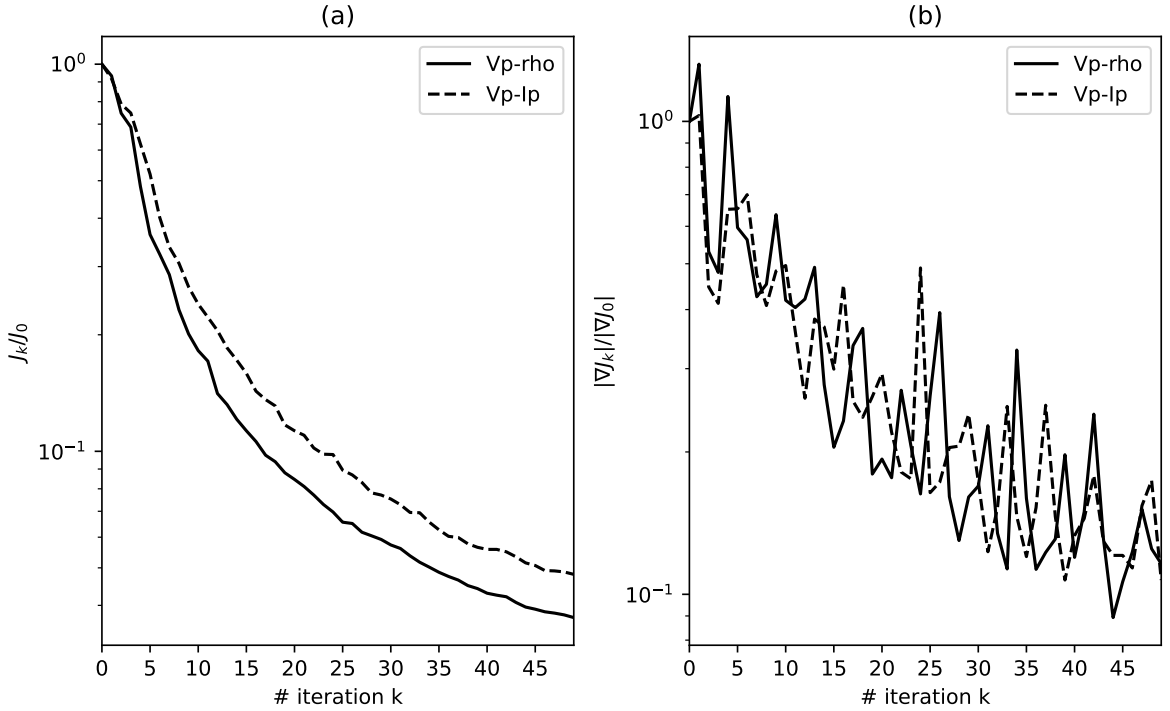
Figure 7: Convergence history of (a) normalized misfit and (b) gradient for two-parameter inversion under family 1 nad family 2.

As can be seen from Figure 8, the inverted velocity and density under $V_p - \rho$ parametrization are closer to the true model, compared with the ones obtained under $V_p - I_p$ parametrization. The inverted density panel under $V_p - I_p$ parametrization presents more high wavenumber oscillations than the one under $V_p - \rho$. This reveals the difference and importance of the choice of an appropriate model parametrization when performing multiparamter inversion. The above result should be understandable based on a radiation pattern analysis, see Yang et al. (2018, Figure 1).

5.2. Reverse time migration

Once a reasonably accurate velocity model is obtained, we can then perform RTM imaging. Here we start from a resampled Marmousi model for RTM test with the grid spacing $\Delta x = \Delta z = 12$ m and the size $nz \times nx = 261 \times 767$. The reason for refined sampling is that RTM generally runs at much higher frequency than FWI (demanding more points per wavelength according to dispersion requirement), to accurately retrieve the interface locations as a structural image. Figure 9a and 9b display the RTM images computed based on cross-correlation imaging condition and normalized cross-correlation imaging condition. The migration artifacts in the shallow part of Figure 9a has been greatly suppressed in Figure 9b by introducing source field normalization; meanwhile, the interface at depth constructed by seismic interference become much more energetic and sharper.

5.3. 3D FWI of Overthrust model

The third example is a 3D FWI test on Overthrust model with interesting geological features such as anticline and the dipping faults. As shown in Figure 10a, the selected part of this 3D model spans over
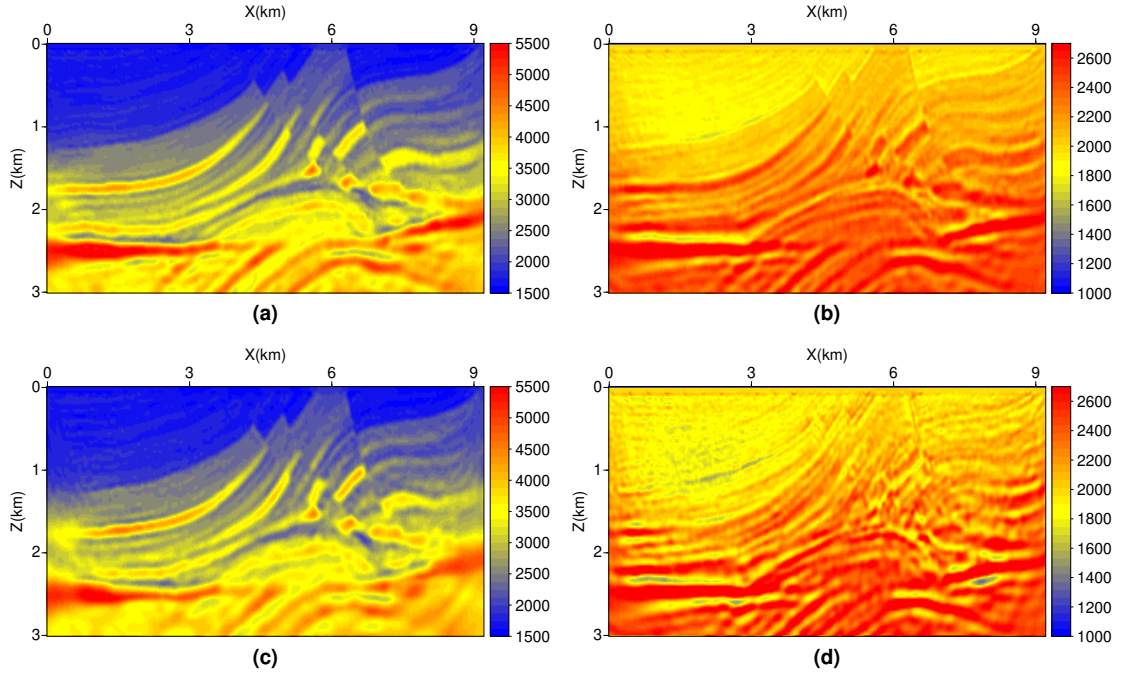
Figure 8: Inverted velocity (a, c) and density (b, d) under parametrizations $V_p - \rho$ (a, b) and $V_p - I_p$ (c, d). The density under $V_p - I_p$ is not a direct inversion parameter, and is derived by $\rho = I_p/V_p$.
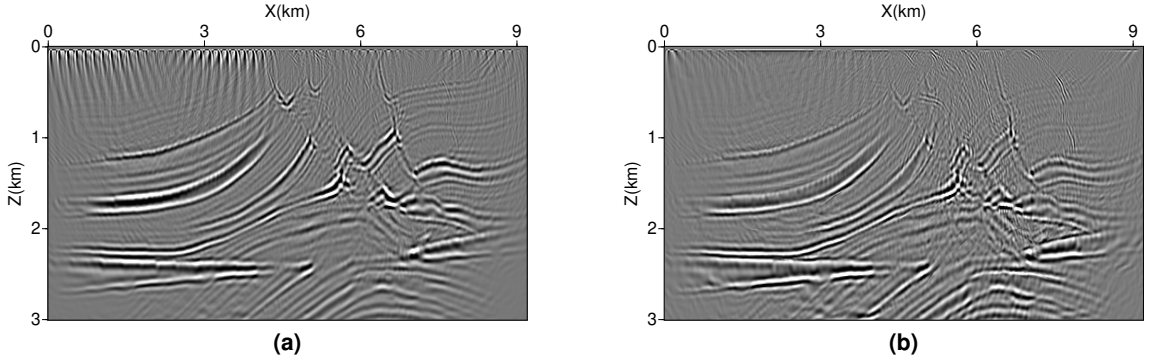

Figure 9: RTM image using (a) cross-correlation imaging condition and (b) normalized cross-correlation imaging condition

10×10 km^2 surface area, extending up to 3 km in depth. The model is uniformly sampled along x-, y- and z- directions: $\Delta x = \Delta y = \Delta z = 50$ m. The initial model in Figure 10b is a strongly smoothed version of the true model. In this example, the density model is again created using Gardner's law, and kept constant. We focus here on monoparameter inversion to retrieve the velocity since it is the first order parameter to explain the phase and amplitude responses of the seismic data. The central frequency of this inversion is around 6 Hz. The modelling lasts for 1500 steps with temporal sampling $\Delta t = 0.004$ s.

As illustrated in Figure 11, we deployed 400 shots evenly distributed in horizontal plane at the depth of 5 m. For each source, we use 10000 receivers at the depth of 10 m to record the seismic data. The source and receiver separations are 500 m and 100 m respectively. Taking into acount 20 PML layers on each side, this

model has a grid size of $nz \times nx \times ny = 101 \times 241 \times 241$. The size of this model and the data set indeed poses a significant challenge for FWI algorithms. To handle such a large scale problem on our small workstation, we employed SGD algorithm to divide all 400 shots into 8 batches, resulting in 50 shots per batch. Figure 12 shows the pattern of the random selection algorithm at each batch using different markers: each time the randomly selected shots do not overlap with other batches. This implies all the data will be touched only once during the inversion. In each batch, we limit the number of iterations to only 3. Figure 13 displays all the updated velocity model after each batch of FWI: the evolution of the model clearly demonstrates the improvement of the resolution. Of course, the resolution can be even higher by repeating more batchwise FWI inversion, which is not pursued since the main feature of the model has been well retrieved.
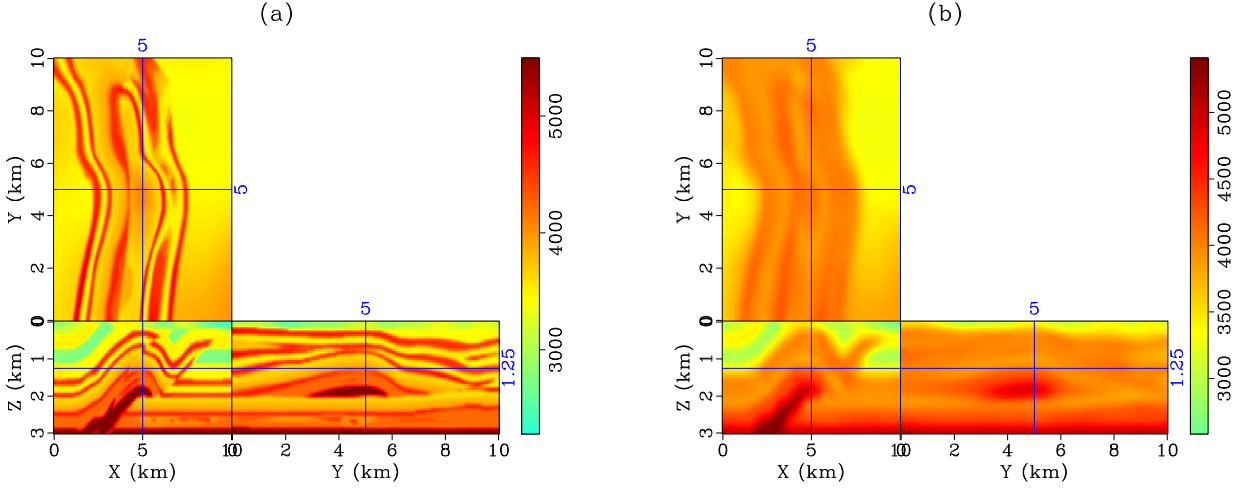

Figure 10: (a) Part of 3D Overthrust velocity model; (b) Smoothed version of (a).

6. Discussion

Let us emphasize that all the computations were done using only a small workstation: it runns with Intel(R) Xeon(R) Gold 6258R CPU @2.7GHz, possessing 28 sockets with 2 cores per socket. When simultaneously launching 50 processes, each shot only consumes 1.5 GB memory. Therefore, it is definitely a remarkable achievement to accomplish such a large size 3D FWI problem using so little computing resources.

Note that building each gradient/imaging condition requires 3 times of modelling, a forward modelling, a backward reconstruction of the forward field and an adjoint modelling. Since we map the modelling shot by shot in a bijective manner, the computing cost will scale with total number of shots linearly. Consequently, we can make a direct accessment on the efficiency of SMIwiz by checking the running time for one modelling job. We now examine the performance of our implementation for one shot simulation in 3D Overthrust model. Figure 14a shows that double the number of OpenMP threads halves the runtime at the beginning, but the efficiency gained by increasing the number of OpenMP threads quickly goes down when the number of OpenMP threads is larger than 4. Figure 14b illustrates that updating particle velocities and pressure takes most of the computing time, while the time cost for injecting source is almost negligible. Because the number of receivers (10000) is pretty large, the extraction of the wavefield at receiver positions also takes a fraction of total runtime.

Note that the acoustic wave equation can also be presented in the 2nd order order form by eliminating the particle velocities in equation (1). We prefer the 1st order system because (1) PML boundaries are much easier to implement in 1st order system than 2nd order one, which can be much more efficient to absorbing the artificial reflections than sponge (Cerjan et al., 1985) and Clayton-Enquist (Clayton and Engquist, 1977) absorbing boundary conditions in truncated mesh; (2) the impact of density can be explicitly expressed in front of fields, which simplifies the derivation of inversion gradient with respect to density; (3)
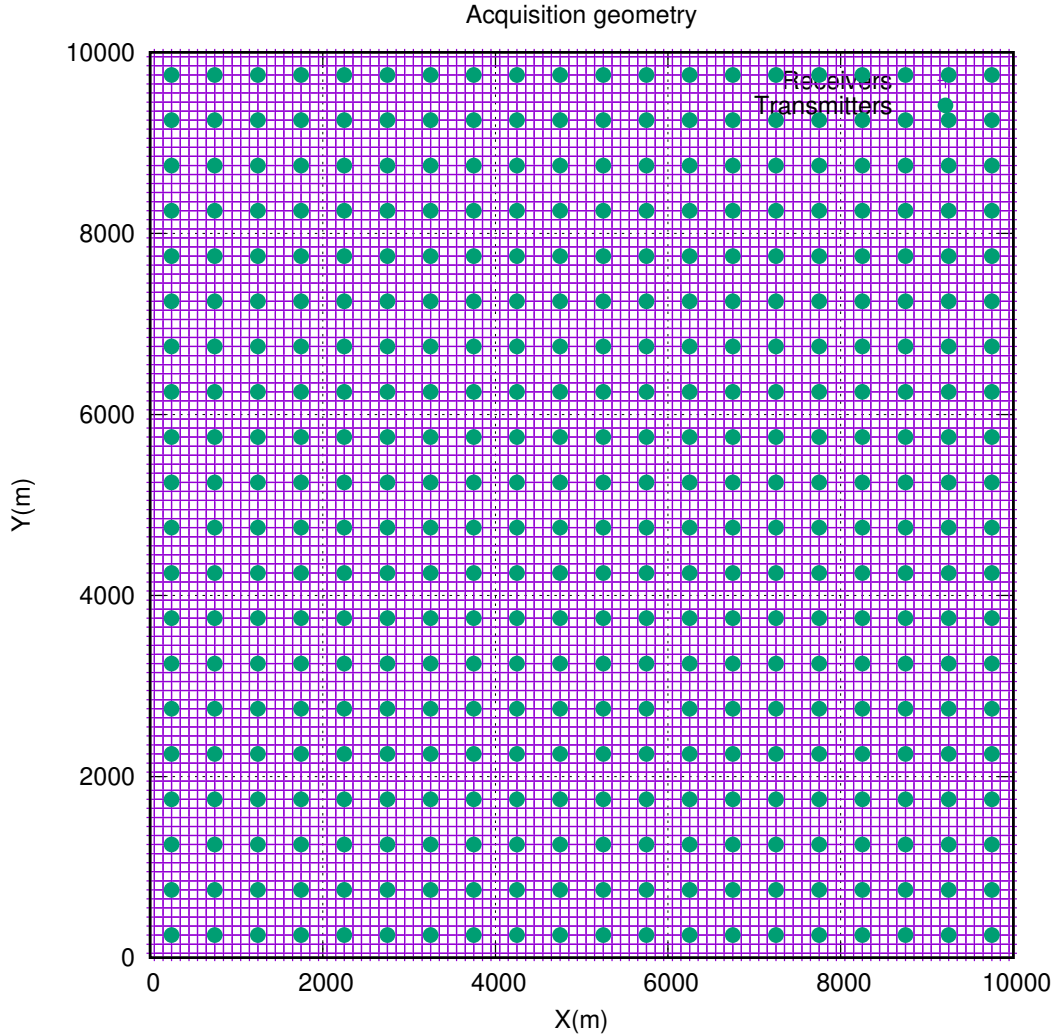
Figure 11: The survey layout sheet of sources (red crossings) and the receivers (green dots)

an accurate injection of different source types can be conveniently injected close to the free surface to extend the functionality of `SMIwiz` in the future.

Instead of reconstructing the source field backwards in time, one may construct the FWI gradient by a mixed domain approach: The modelling of the forward and adjoint wavefields is in the time domain, but their frequency domain counterparts are integrated on the fly using discrete Fourier transform (Nihei and Li, 2007; Sirgue et al., 2007). The FWI gradient can then be computed by multiplication between frequency domain forward and adjoint wavefields. Such a method avoids one modelling in building the FWI gradient, but requires additional computer memory to store the wavefields for every discrete frequency. Unfortunately, this method becomes extraordinarily memory demanding for RTM applications, in which a broad range of frequencies are needed to form a high resolution migration image.

A notoriously difficult problem is the FWI cycle skipping issue. The optimization is often trapped into local minima when FWI is performed from a very poor starting model, using the data without long offset, low-frequency information. A number of solutions to mitigate such difficulties have been proposed, using multiscale strategies from low to high frequencies (Bunks et al., 1995), layer stripping from long to near offset (Bian and Yu, 2011), the modifications of the misfit functions based on cross-correlation (Luo and Schuster, 1991), deconvolution (Warner and Guasch, 2016) or optimal transport distance (Métivier et al.,
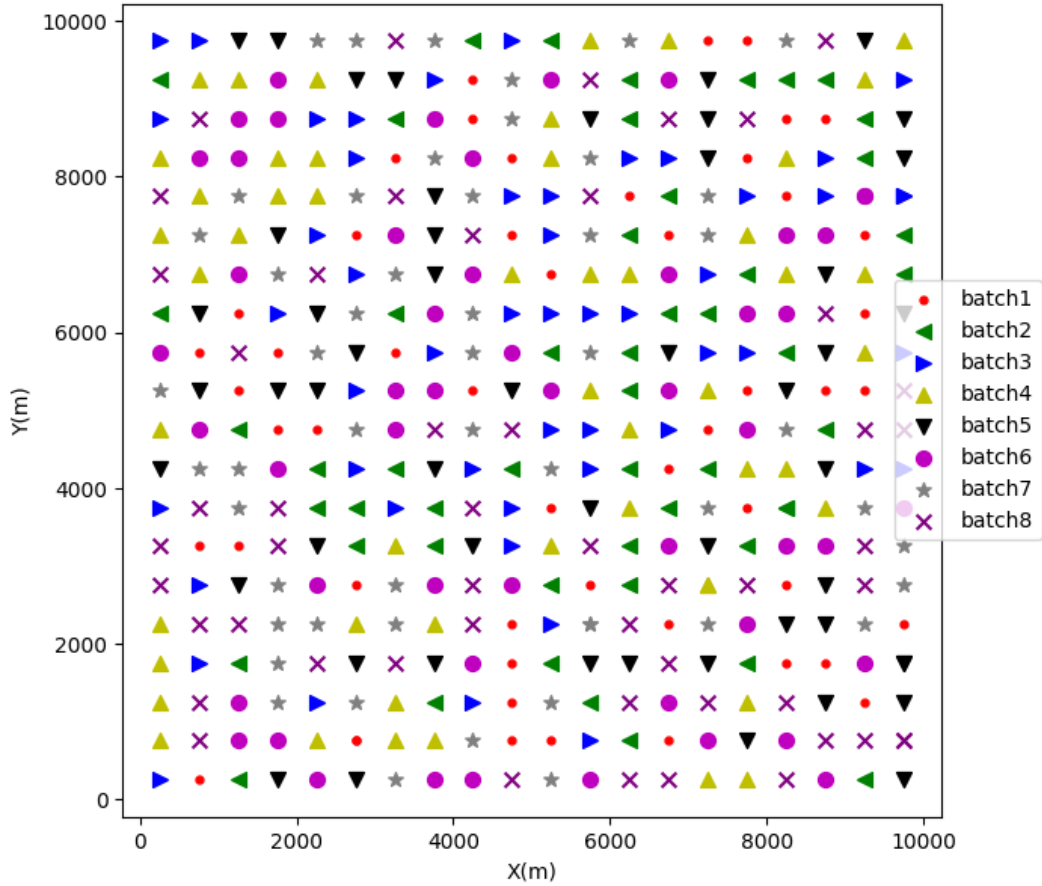
Figure 12: Randomly selected shots for 8 batches of stochastic gradient descent

2016). These ideas can be explored easily within the framework of `SMIwiz`, but out of the scope of this work. We aim to provide a public platform rather than completing all the techniques in the literature.

7. Conclusions

`SMIwiz` serves as a consistent realization of seismic modelling and imaging. It has a mechanism to do 2D/3D imaging flexibly, with some data weighting and preprocessing capabilities.

- The idea of computing box is simple yet powerful to restrict the modelling in a local cubic grid to avoid redundant computations, which significantly speeds up the forward and adjoint modelling.

- To efficiently compute RTM image and the inversion gradient, we have implemented wavefield reconstruction using decimated boundaries, so that the source field can be accurately retrieved backwards in time using interpolated boundaries. It is indeed the key factor for `SMIwiz` to resolve practical 3D large scale imaging problems.

- The batchwise job scheduling gives flexibility to the user to take advantages of the limited computer memory and processors. This can be further explored by the user as long as there are sufficient computer resources accessible in a modern cluster.

We have demonstrated the versatile modelling and imaging functionalities of `SMIwiz` selectively. There are other interesting modules still under development, for example, linearized waveform inversion via Born

---
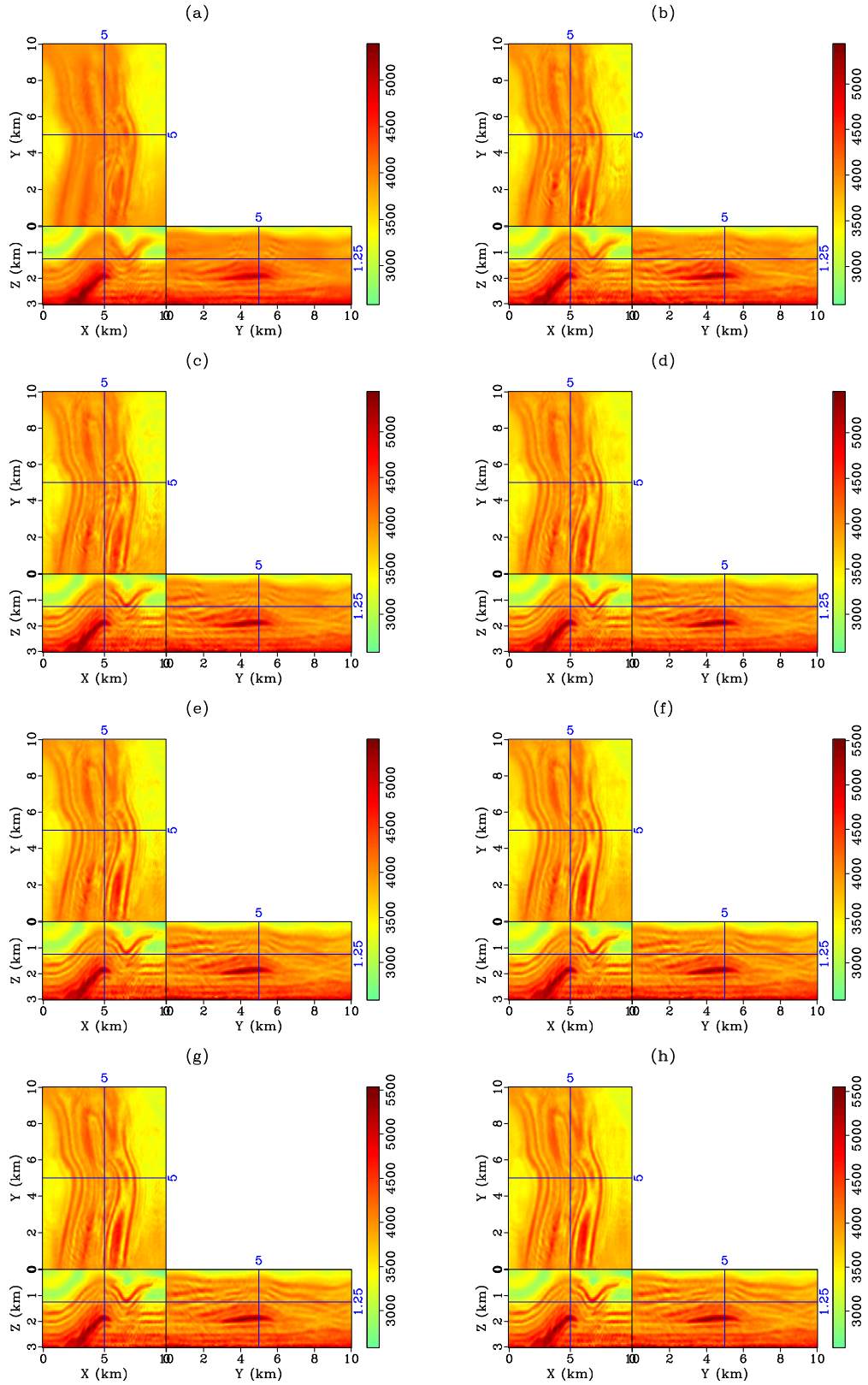
Figure 13: The update velocity model at (a) batch 1; (b) batch 2; (c) batch 3; (d) batch 4; (e) batch 5; (f) batch 6; (g) batch 7; (h) batch 8.
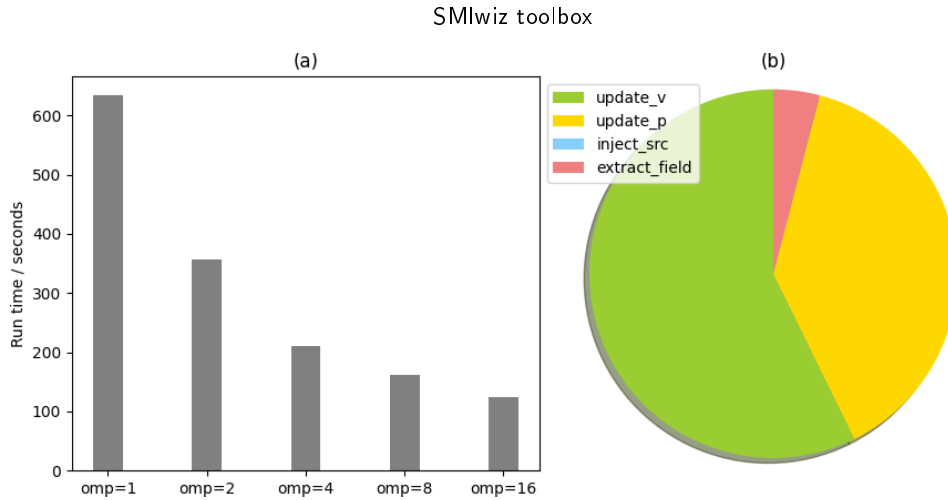
Figure 14: (a) The runtime of our modelling decreases when increasing the number of OpenMP threads; (b) the ratio of the runtime for updating particle velocities, pressure, injecting source and extracting wavefield in one modelling

approximation. The module of modelling has been made self-contained so that it can be extended into elastic regime incorporating anisotropy and attenuation, while the backbone of nonlinear optimization has all necessary components to extend from quasi-Newton approach into a Newton-CG minimization. The software will continue to grow in the future. The efficiency of the modelling engine in `SMIwiz` will be dramatically improved using graphic processing units (GPU).

Acknowledgements

Computer Code Availability

Name of the code: `SMIwiz`
Functionality: An integrated toolbox for multidimensional seismic modelling and imaging
Contact: Pengliang Yang, Harbin Institute of Technology, China
E-mail: ypl.2100@gmail.com
System requirements: Linux OS
Programming language: C, Shell
Result visualization: Madagascar and Seismic Unix
Compilation requirement: gcc compiler, mpicc compiler and make
Software required: FFTW3 (http://fftw.org/)
The code has been released publically via github repository https://github.com/yangpl/SMIwiz.

Data Availability

The data used for the research has been included in the software package - `SMIWiz`.

Declaration of Competing Interest

I declare that I have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

References

Abdelkhalek, R., Calandra, H., Coulaud, O., Roman, J., Latu, G., 2009. Fast seismic modeling and reverse time migration on a gpu cluster, in: 2009 International Conference on High Performance Computing & Simulation, IEEE. pp. 36–43.

Baysal, E., Kosloff, D., Sherwood, J., 1983. Reverse time migration. Geophysics 48, 1514–1524.

Bian, A., Yu, W., 2011. Layer-stripping full waveform inversion with damped seismic reflection data. Journal of Earth Science 22, 241–249.

Bohlen, T., 2002. Parallel 3-D viscoelastic finite-difference seismic modeling. Computers & Geosciences 28, 887–899.

Bohlen, T., Nil, D.D., Jetschny, D.K.S., 2015. SOFI3D - seismic modeling with finite differences 3D - acoustic and viscoelastic version - user guide URL: https://gitlab.kit.edu/kit/gpi/ag/software/sofi3d.

Brossier, R., 2011. Two-dimensional frequency-domain visco-elastic full waveform inversion: Parallel algorithms, optimization and performance. Computers & Geosciences 37, 444 – 455. URL: http://www.sciencedirect.com/science/article/B6V7D-51F7PRV-7/2/bf6e30f7d564d1acb75d009a82610f65, doi:DOI:10.1016/j.cageo.2010.09.013.

Bunks, C., Salek, F.M., Zaleski, S., Chavent, G., 1995. Multiscale seismic waveform inversion. Geophysics 60, 1457–1473.

Carcione, J.M., 2010. A generalization of the fourier pseudospectral method. Geophysics 75, A53–A56.

Cerjan, C., Kosloff, D., Kosloff, R., Reshef, M., 1985. A nonreflecting boundary condition for discrete acoustic and elastic wave equations. Geophysics 50, 2117–2131.

Clayton, R., Engquist, B., 1977. Absorbing boundary conditions for acoustic and elastic wave equations. Bulletin of the Seismological Society of America 67, 1529–1540.

Dumbser, M., Käser, M., 2006. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—II. The three-dimensional isotropic case. Geophysical Journal International 167, 319–336.

Etgen, J., Gray, S.H., Zhang, Y., 2009. An overview of depth imaging in exploration geophysics. Geophysics 74, WCA5–WCA17. URL: http://link.aip.org/link/?GPY/74/WCA5/1, doi:10.1190/1.3223188.

Fabien-Ouellet, G., Gloaguen, E., Giroux, B., 2017. Time-domain seismic modeling in viscoelastic media for full waveform inversion on heterogeneous computing platforms with opencl. Computers & Geosciences 100, 142–155.

Fornberg, B., 1998. Classroom note: Calculation of weights in finite difference formulas. SIAM review 40, 685–691.

Graves, R., 1996. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences. Bulletin of the Seismological Society of America 86, 1091–1106.

Hicks, G.J., 2002. Arbitrary source and receiver positioning in finite-difference schemes using Kaiser windowed sinc functions. Geophysics 67, 156–166.

Hu, F.Q., Hussaini, M.Y., Rasetarinera, P., 1999. An analysis of the discontinuous galerkin method for wave propagation problems. Journal of Computational Physics 151, 921–946.

Köhn, D., De Nil, D., Kurzmann, A., Przebindowska, A., Bohlen, T., 2012. On the influence of model parametrization in elastic full waveform tomography. Geophysical Journal International 191, 325–345.

Komatitsch, D., Martin, R., 2007. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. Geophysics 72, SM155–SM167.

Komatitsch, D., Vilotte, J.P., 1998. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. Bulletin of the Seismological Society of America 88, 368–392.

Lailly, P., 1983. The seismic inverse problem as a sequence of before stack migrations, in: Bednar, R., Weglein (Eds.), Conference on Inverse Scattering, Theory and application, Society for Industrial and Applied Mathematics, Philadelphia, pp. 206–220.

Levander, A.R., 1988. Fourth-order finite-difference P-SV seismograms. Geophysics 53, 1425–1436.

Luo, Y., Schuster, G.T., 1991. Wave-equation traveltime inversion. Geophysics 56, 645–653.

McMechan, G., 1983. Migration by extrapolation of time-dependent boundary values. Geophysical Prospecting 31, 413–420.

Métivier, L., Brossier, R., 2016. The SEISCOPE optimization toolbox: A large-scale nonlinear optimization library based on reverse communication. Geophysics 81, F11–F25.

Métivier, L., Brossier, R., Mérigot, Q., Oudet, E., Virieux, J., 2016. An optimal transport approach for seismic tomography: Application to 3D full waveform inversion. Inverse Problems 32, 115008.

Michéa, D., Komatitsch, D., 2010. Accelerating a 3D finite-difference wave propagation code using GPU graphics cards. Geophysical Journal International 182, 389–402.

Nihei, K.T., Li, X., 2007. Frequency response modelling of seismic waves using finite difference time domain with phase sensitive detection (TD-PSD). Geophysical Journal International 169, 1069–1078.

Nocedal, J., 1980. Updating Quasi-Newton Matrices With Limited Storage. Mathematics of Computation 35, 773–782.

Nocedal, J., Wright, S.J., 2006. Numerical Optimization. 2nd ed., Springer.

Pratt, R.G., 1999. Seismic waveform inversion in the frequency domain, Part I: theory and verification in a physical scale model. Geophysics 64, 888–901.

Seriani, G., Priolo, E., 1994. Spectral element method for acoustic wave simulation in heterogeneous media. Finite elements in analysis and design 16, 337–348.

Sirgue, L., Etgen, T.J., Albertin, U., Brandsberg-Dahl, S., 2007. System and method for 3D frequency-domain waveform inversion based on 3D time-domain forward modeling. US Patent Application Publication US2007/0282535 A1.

Smith, W.D., 1975. The application of finite element analysis to body wave propagation problems. Geophysical Journal International 42, 747–768.

Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation. Geophysics 49, 1259–1266.

Virieux, J., 1984. SH wave propagation in heterogeneous media: Velocity-stress finite difference method. Geophysics 49, 1259–1266.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media: Velocity stress finite difference method. Geophysics 51, 889–901.

Virieux, J., Calandra, H., Plessix, R.E., 2011. A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging. Geophysical Prospecting 59, 794–813.

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

Warner, M., Guasch, L., 2016. Adaptive waveform inversion: Theory. Geophysics 81, R429–R445.

Weiss, R.M., Shragge, J., 2013. Solving 3d anisotropic elastic wave equations on parallel gpu devices. Geophysics 78, F7–F15.

Yang, P., 2023a. 3D fictitious wave domain CSEM inversion by adjoint source estimation. Computers & Geosciences 180, 105441. doi:https://doi.org/10.1016/j.cageo.2023.105441.

Yang, P., 2023b. libEMM: A fictious wave domain 3D CSEM modelling library bridging sequential and parallel GPU implementation. Computer Physics Communications 288, 108745. doi:https://doi.org/10.1016/j.cpc.2023.108745.

Yang, P., 2023c. A unified formulation of nonlinear, linearized and reflection waveform inversion. Bulletin of the Seismological Society of America , under review.

Yang, P., Brossier, R., Métivier, L., Virieux, J., 2016a. Wavefield reconstruction in attenuating media: A checkpointing-assisted reverse-forward simulation method. Geophysics 81, R349–R362. doi:10.1190/geo2016-0082.1.

Yang, P., Brossier, R., Métivier, L., Virieux, J., Zhou, W., 2018. A Time-Domain Preconditioned Truncated Newton Approach to Visco-acoustic Multiparameter Full Waveform Inversion. SIAM Journal on Scientific Computing 40, B1101–B1130.

Yang, P., Brossier, R., Virieux, J., 2016b. Wavefield reconstruction from significantly decimated boundaries. Geophysics 80, T197–T209. doi:10.1190/GEO2015-0711.1.

Yang, P., Gao, J., Wang, B., 2014. RTM using effective boundary saving: A staggered grid GPU implementation. Computers & Geosciences 68, 64–72.

Yang, P., Gao, J., Wang, B., 2015. A graphics processing unit implementation of time-domain full-waveform inversion. Geophysics 80, F31–F39.

Zhang, H.J., Zhang, Y., 2008. Reverse time migration in 3D heterogeneous TTI media. SEG Technical Program Expanded Abstracts 27, 2196–2200.

Zhang, Y., Sun, J., 2009. Practical issues in reverse time migration: true amplitude gathers, noise removal and harmonic source encoding. First break 27, 53–59.

Zhang, Z., Irving, J., Simons, F., Alkhalifah, T., 2023. Seismic evidence for a 1000 km mantle discontinuity under the pacific. Nature Communications 14. doi:10.1038/s41467-023-37067-x.