CREWES

www.crewes.org

Instruction for C++ package of visco-elastic multiparameter FWI in the frequency domain

Jinji Li*, Scott Keating, Daniel Trad, and Kris Innanen

li.jinji@ucalgary.ca

ABSTRACT

The frequency domain full-waveform inversion (FWI) is a nonlinear optimization problem where large matrices such as the Helmholtz matrix and many derivatives are saved and utilized. In many inversion tests involving multiple variables and larger-scale models, the computational cost and consumption of computer resources should be considered. Matlab is presumably the best platform for solving large linear systems, manipulating matrices, and intuitively analyzing the intermediate results. However, its massive memory occupation and limited computational effectiveness can hamper going to 3-D problems or some larger 2-D FWI. Motivated by this, we developed a C++ version of the current 2-D frequency domain multi-parameter visco-elastic FWI, which can be carried out on relatively larger models. The reduced cost also allows us to go to the 3-dimensional inverse problems or the probabilistic approaches. Essential tools and concepts for working within the C++ ecosystem are covered in this instructive report, with examples of how to use this package.

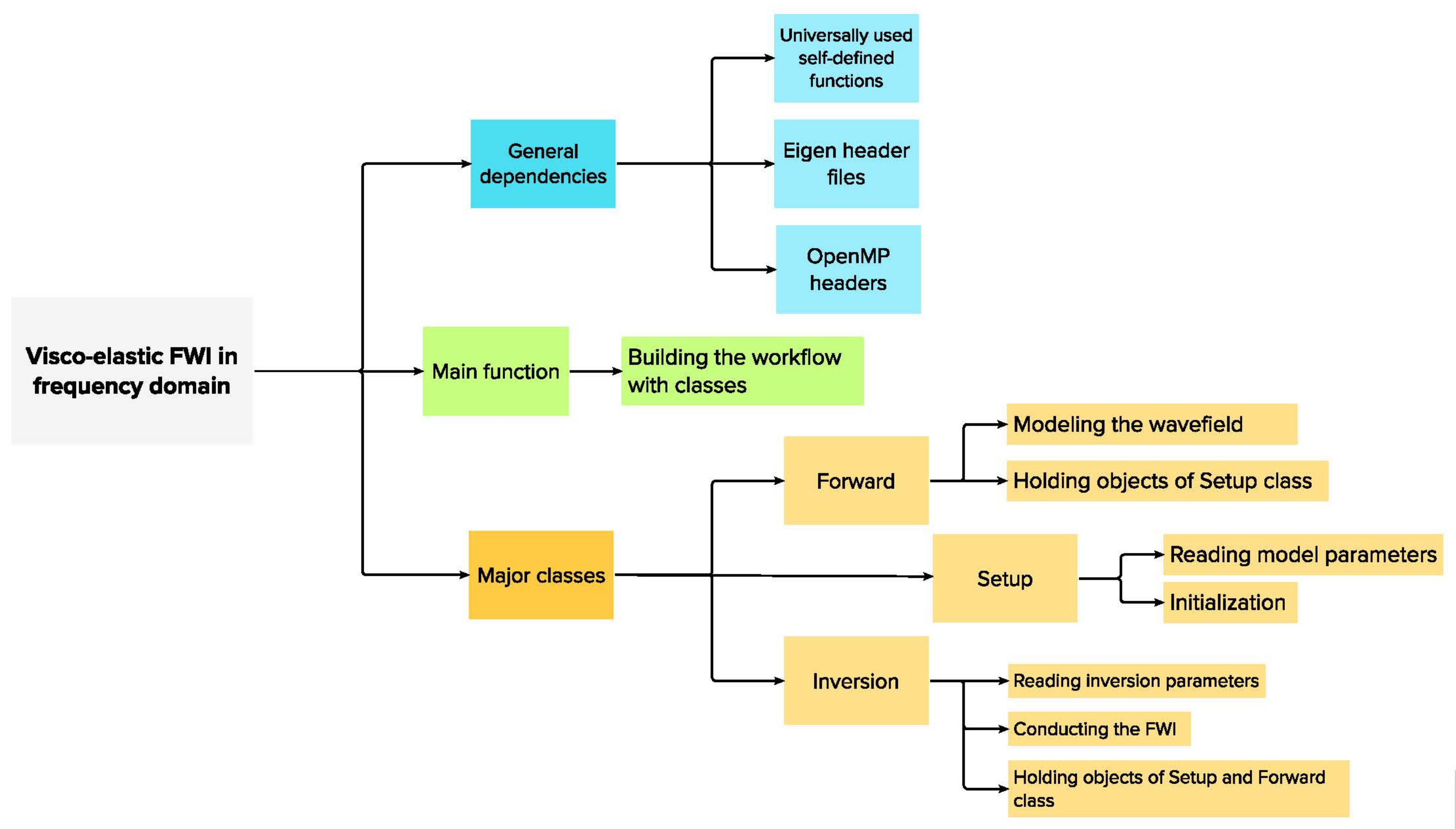

Visco-elastic FWI in frequency domain
General dependencies
Universally used self-defined functions
Eigen header files
OpenMP headers
Main function
Building the workflow with classes
Major classes
Forward
Modeling the wavefield
Holding objects of Setup class
Setup
Reading model parameters
Initialization
Inversion
Reading inversion parameters
Conducting the FWI
Holding objects of Setup and Forward class

Figure 1. Basic structure of the VEFWI package.

Advantages:

1. Endogenously faster
2. Easy to couple with industry
3. Multi-platform
4. Memory manipulation
5. Open source

Prerequisites:

- Eigen
- OpenMP
- Basic Linear Algebra Subprograms (BLAS)
- Linear Algebra package (LAPACK)

COMPUTATIONAL CONSIDERATIONS ON MATRIX VIEW

The 2-D frequency domain elastic wave equation can be represented by matrix multiplications, thus making it an implicit finite-difference approach denoted by a system of linear equations shown below:

$$\boldsymbol{S}(\omega)\boldsymbol{u}(\omega) = \boldsymbol{f}(\omega), \qquad (1)$$

where $\boldsymbol{S}$ is the impedance matrix, $\boldsymbol{u}$ is the wavefield, and $\boldsymbol{f}$ is the source term. In the 2-D space, the dimensionality of $\boldsymbol{S}$ will be 2 × nx × nz by 2 × nx × nz, and the nonzero values are 9 × nx × nz. The resolution of this system is also expensive because of the memory for saving LU factors.

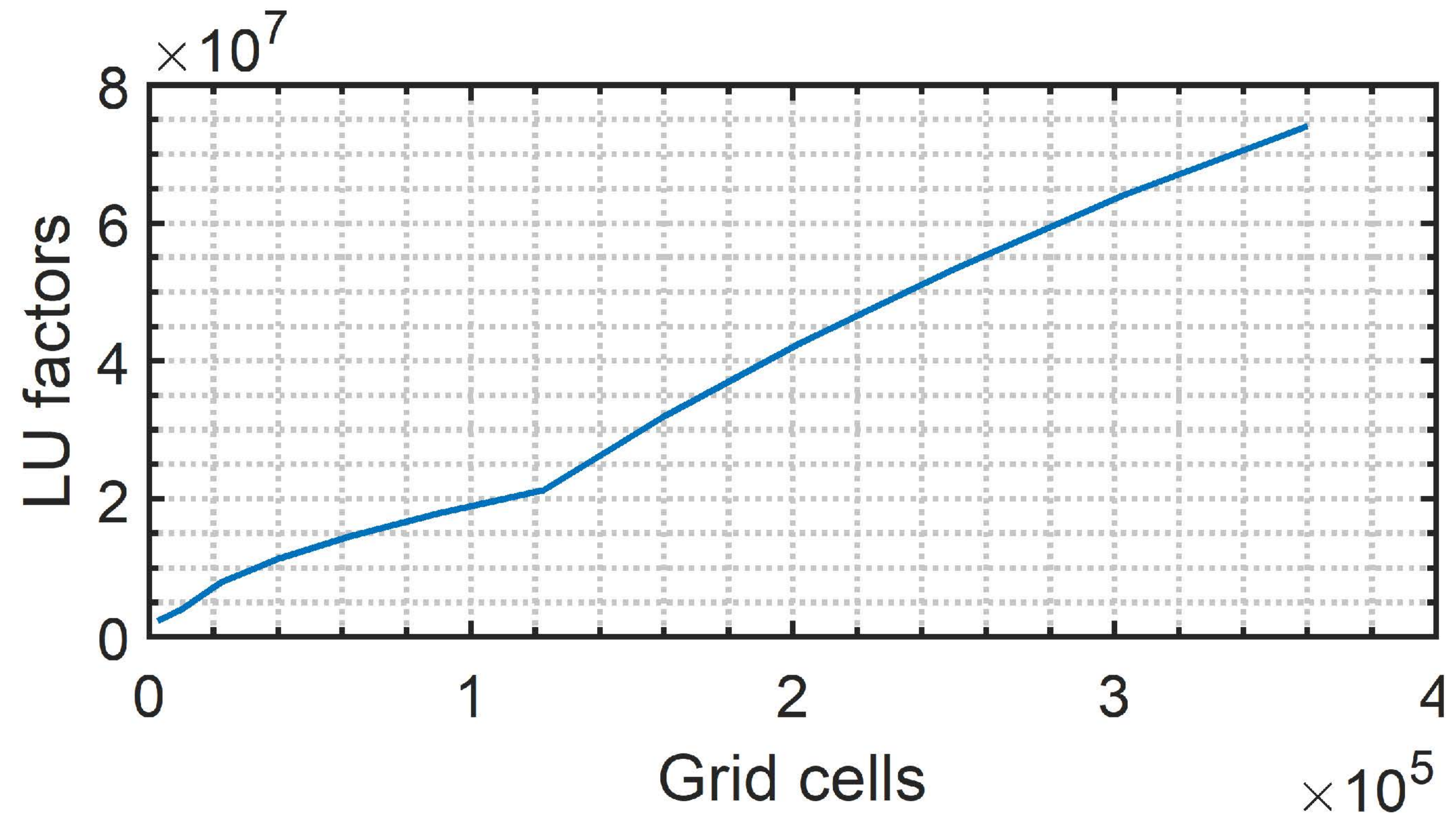


Figure 2. Amount of LU factors as a function of the grid cell number.

ALGORITHM IMPLEMENTATION

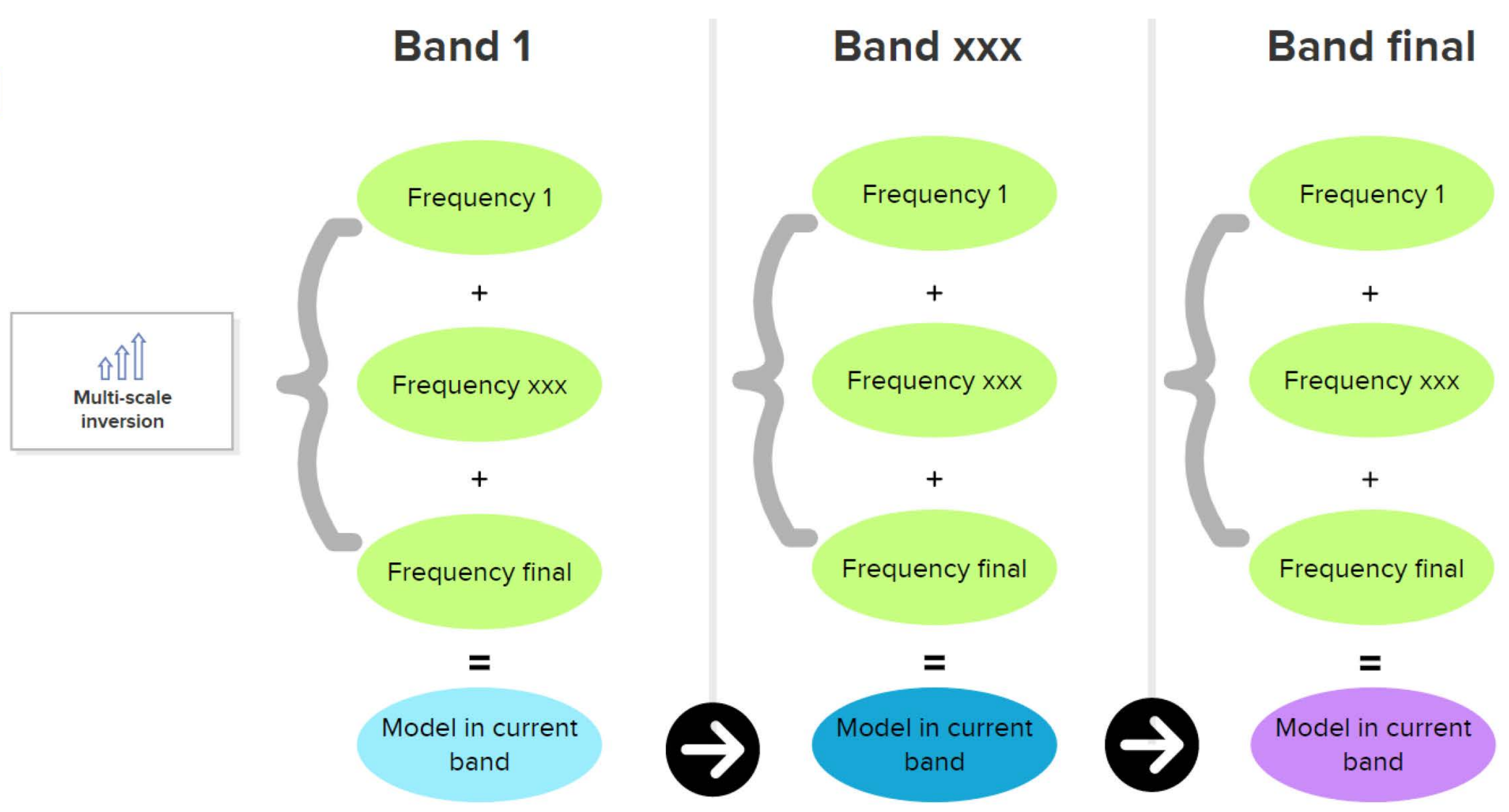
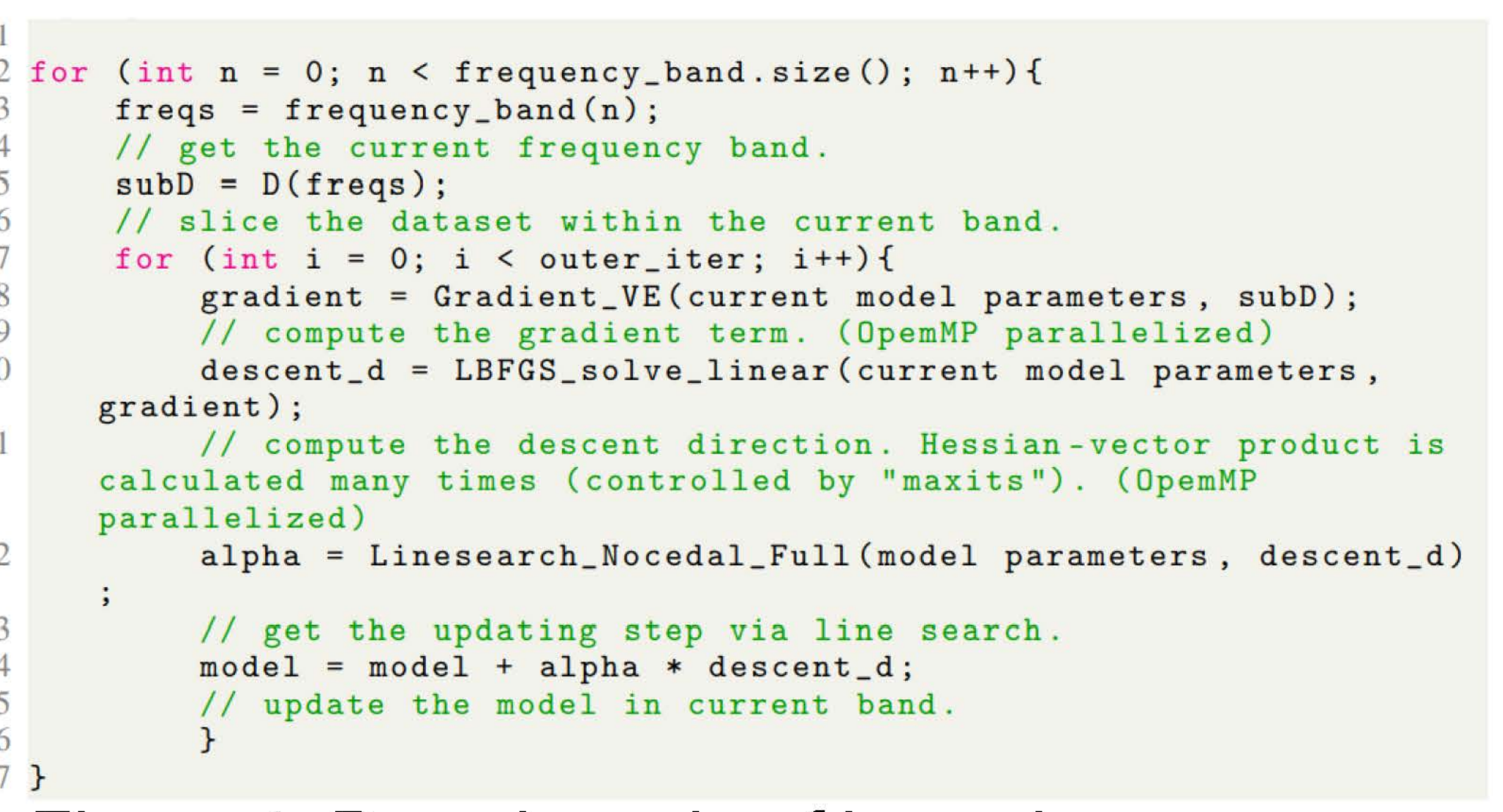


Figure 3. Multi-scale inversion logic.



```
for (int n = 0; n < frequency_band.size(); n++){
    freqs = frequency_band(n);
    // get the current frequency band.
    subD = D(freqs);
    // slice the dataset within the current band.
    for (int i = 0; i < outer_iter; i++){
        gradient = Gradient_VE(current model parameters, subD);
        // compute the gradient term. (OpemMP parallelized)
        descent_d = LBFGS_solve_linear(current model parameters,
  gradient);
        // compute the descent direction. Hessian-vector product is
  calculated many times (controlled by "maxits"). (OpemMP
  parallelized)
        alpha = Linesearch_Nocedal_Full(model parameters, descent_d)
  ;
        // get the updating step via line search.
        model = model + alpha * descent_d;
        // update the model in current band.
        }
}
```

Figure 4. Pseudo code of inversion process.

EXPERIMENTS

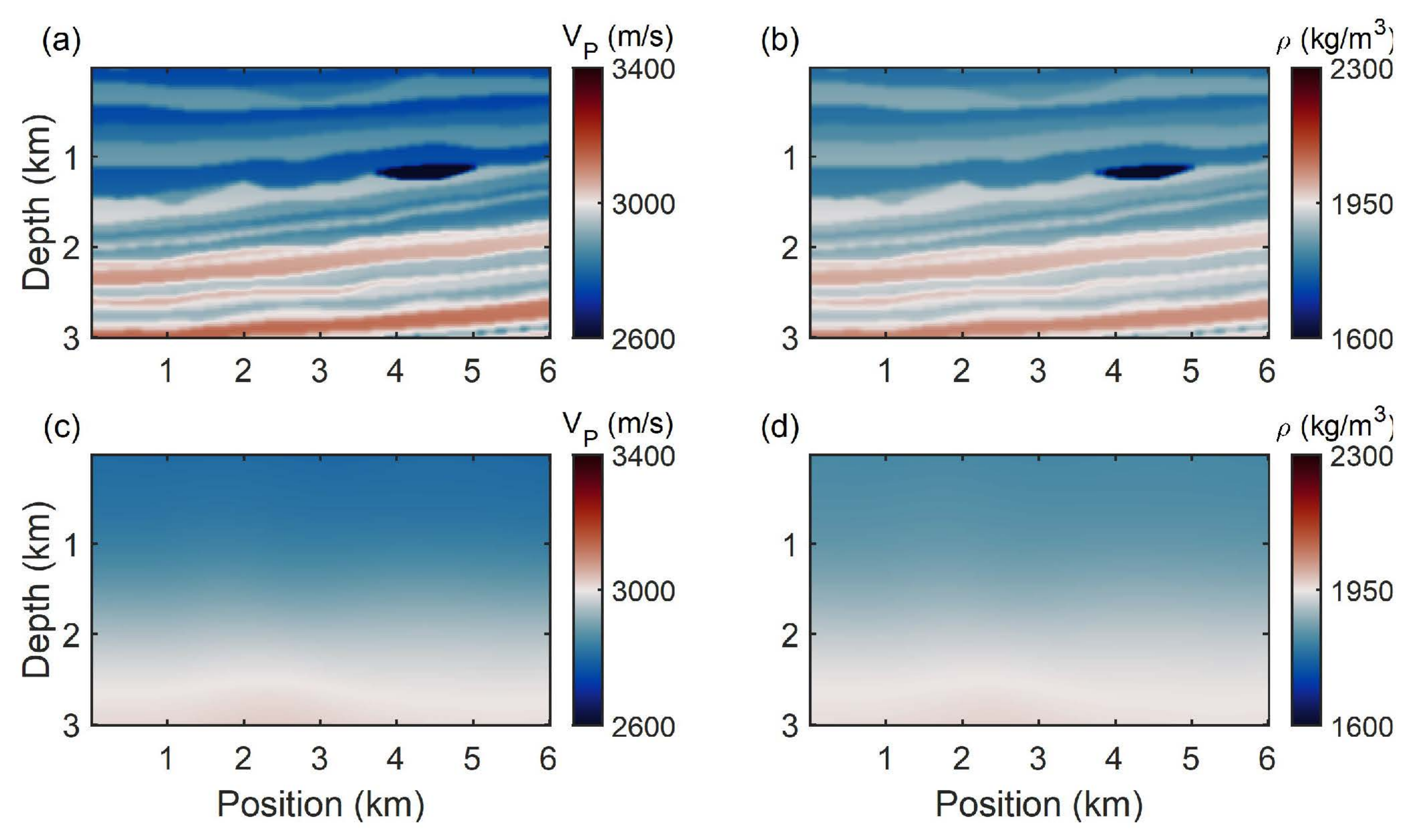


Figure 5. True and initial models.

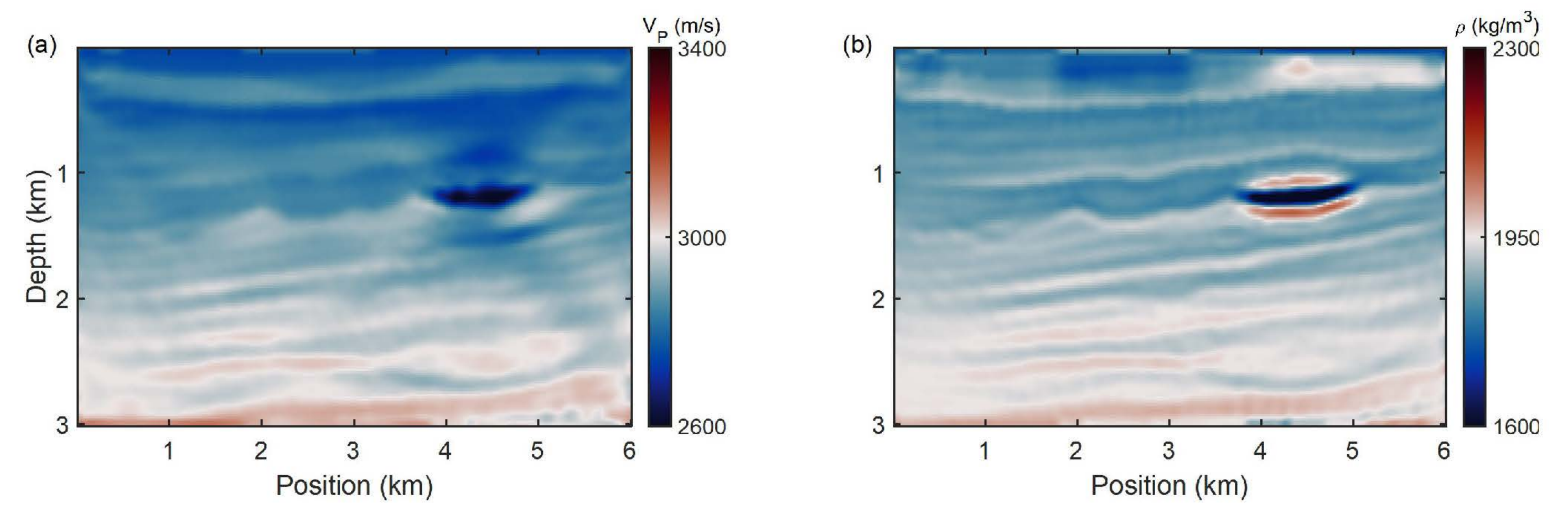


Figure 6. Estimated models.

Table 1: runtime measurement

Model size	Synthetic modeling	1 band inversion	Total runtime
50 × 50	6.23 s	58.1 s	581.3 s
100 × 50	15.31 s	162.93 s	1589.14 s
100 × 100	28.5 s	308.97 s	2097.9 s
150 × 100	57.59 s	655.6 s	6322.1 s
200 × 100	98.23 s	979.6 s	9701.43 s
250 × 100	143.76 s	1649 s	16351.6 s
300 × 100	215.45 s	1649 s	16351.6 s
300 × 150	371.71 s	3813.3 s	37017.1 s

Table 2: memory consumption

Model size	Synthetic modeling	Inversion
50 × 50	0.59 GB	1.47 GB
100 × 50	1.36 GB	2.52 GB
100 × 100	2.35 GB	4.51 GB
150 × 100	3.81 GB	8.25 GB
200 × 100	6.23 GB	11.9 GB
250 × 100	8.85 GB	15.1 GB
300 × 100	10.2 GB	20.8 GB
300 × 150	15.4 GB	26.2 GB

ACKNOWLEDGEMENTS

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19.

NSERC CRSNG

UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience