修士論文

High-Fidelity GPU-Accelerated Computation of Long-term Viscoelastic Crustal Deformation for Improving Earthquake and Tsunami Source Models

地震・津波モデル高度化のための、GPU を活用した長期粘弾性地殻変動の高詳細解析

Palacios Espinoza Julian Miller
Student ID: 37-216998

Signature	Date
Advisor: 藤田 航平	2023/07/21
Co-advisor: 堀 高峰	2023/07/20

Department of Civil Engineering
Graduate School of Engineering
The University of Tokyo

2023 年 7 月

Abstract

Improvement in the estimation of slip-deficit rates and coseismic slips of past earthquakes in plate subduction zones is expected to improve the estimation of seismic and tsunami source models of future megathrust earthquakes. Such improvement is expected to be realized using viscoelastic models reflecting the 3D crustal structure and assimilating with geodetic data obtained from onshore and seafloor observation networks. However, the computation of long-term viscoelastic deformation for a 3D crustal structure requires extensive computation; thus, simplified models are often used in previous research to analyze slip-deficit rates.

In this study, we developed a GPU-accelerated viscoelastic analysis method to enable the computation of many long-term viscoelastic Green's functions required to conduct inversion analyses of slip-deficit and coseismic slip distributions of past earthquakes. Here, an Aggregate-based Correction (ABC) and an Adaptive Time-stepping (ATS) method were incorporated into the highly tuned GPU-based multigrid finite-element solver to realize fast computation. The GPU-based solver with ABC and ATS attained 1.22-fold and 6.02-fold speedup from the state-of-art multigrid solver, respectively, which in total led to a 22.6-fold speedup from a GPU-based solver with a standard multigrid algorithm.

As an application example of the developed method, we computed 80 long-term viscoelastic Green's functions on a large-scale 3D crustal model of the Nankai region with 4.2×10^9 degrees of freedom. These functions were used to estimate slip-deficit and historical coseismic slip on a hypothetical observation dataset. Results show improvement in the estimation of slip-deficit when compared to results obtained using Green's functions under completely relaxed state assumption.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Kohei Fujita from the Earthquake Research Institute (ERI), for his support, guidance, and invaluable feedback throughout the research process. Professor Fujita's deep knowledge and passion for high-performance computing have paved the way for my immersion into the world of the most advanced computational technologies. I am incredibly grateful for his exceptional care and consideration towards me as an international student with a different cultural background conducting research in Japan.

I am also profoundly grateful to Dr. Takane Hori from Japan Agency for Marine-Earth Science and Technology (JAMSTEC) for his valuable contribution as my co-supervisor. His expertise and guidance from the perspective of seismology and earthquake estimation have greatly enriched this research. Additionally, I would like to thank Professor Tsuyoshi Ichimura and Professor Lalith Wijerathne for their advice and support during our laboratory meetings.

I want to thank Mr. Sota Murakami, a doctoral student who helped me in many ways. His comments contributed to implementing numerical methods and facilitated the successful execution of simulations in supercomputer environments. Furthermore, I thank my fellow Japanese and international students who provided an enriching academic environment. Their constructive discussions and shared experiences contributed to this research. I would also like to thank Ms. Yumiko Nagasaki, an academic staff member, for her support in handling administrative processes crucial in my academic life.

I am grateful to Dr. Carlos Zavala, Dr. Miguel Diaz, and Msc. Jorge Morales from the Japan-Peru Center for Earthquake Engineering and Disaster Mitigation (CISMID) for their trust and belief in my abilities, granting me the invaluable opportunity to pursue this master's degree. I want to express my deepest thanks to the Japan International Cooperation Agency (JICA) for their generous financial support, which made my stay in Japan possible. I would also like to thank Ms. Yenifer Taipicuri and Mr. Andre Munoz, both dear friends and former colleagues at CISMID. Their warm welcome and introduction to the academic life at ERI have made my experience enjoyable.

Last, I would like to acknowledge the support and understanding of my family, fiancee, and friends. Their encouragement and belief in my abilities have sustained me throughout this journey. This thesis would not have been possible without the guidance, support, and contributions of the individuals and organizations mentioned above. Thank you all.

Contents

List of Figures

List of Tables

1 Introduction

Assessing ground motion and tsunami hazards is crucial for mitigating the impact of future megaquakes. The source models of future megathrust earthquakes have been estimated by analyzing the slip-deficit, slip locked in the plate interface, and coseismic slips of past earthquakes in plate subduction zones. Seismologists use onshore and seafloor geodetic observation networks to estimate slip-deficit regions in plate subduction zones where megathrust earthquakes occur. However, such slip-deficit estimations rely on simplified crustal models and assume an elastic or completely relaxed state (a long time after an earthquake) of the viscoelastic asthenosphere. Improving slip estimation requires accurate simulations that faithfully replicate the viscoelastic crustal deformation phenomenon. Therefore, it is crucial to conduct simulations of time-dependent viscoelastic response due to slip motion on a high-fidelity crustal model.

In a previous study, the viscoelastic crustal deformation analysis was computed using the Finite Element (FE) method to consider a detailed 3D model of the crustal structure (Ichimura et al., 2016). To estimate slip distributions, large-scale viscoelastic analyses are required for a considerable number of cases, around 10^{2-3}. Huge analysis cost is involved in $\sim 10^{2-3}$ cases of analysis for solving large linear systems of $\sim 10^{9-10}$ degrees of freedom (DOF) and $\sim 10^{2-4}$ time steps with $dt \sim 10$ days. Since the analysis cost of a 3D high-fidelity model is high, viscoelastic response after an earthquake has been performed using simplified crustal models (Noda et al., 2018; Sherrill and Johnson, 2021). In this regard, 3D crustal deformation analysis of large-scale models was enabled by advancements in computational methods (Fujita et al., 2017) and improvements in the capabilities of high-performance computing (HPC) systems. For current HPC systems, Yamaguchi et al. (2017) developed a fast solver that offloads a large amount of crustal deformation computation to graphics processing units (GPUs), further improving the computational performance. Recent studies have significantly improved the computation of viscoelastic crustal deformation by predicting the numerical solution from previous simulation steps (Fujita et al., 2022; Murakami et al., 2023).

Despite significant efforts, conducting numerical simulations of high-fidelity crustal deformation for long-term viscoelastic response remains computationally extensive. Therefore, in this study, we further accelerate the computation of long-term viscoelastic deformation by incorporating an Aggregation-based Correction (ABC) and an Adaptive Time-Stepping (ATS) method. The ABC enhances the high-accuracy initial guess in solving the large linear system. On the other hand, the ATS method reduces the number of simulation steps by dynamically adjusting the time increment considering the viscoelastic relaxation phenomenon. We measured the performance of the proposed methods by computing viscoelastic deformation on a large-scale 3D crustal model of the Nankai Trough subduction zone. This region experienced recurring megathrust earthquakes and is expected to experience a megaquake in the near future (Fukushima et al., 2023; Central Disaster Management Council). As a result of the performance measurements, the GPU-based solver with ABC attained a 1.22-fold speedup from the state-of-art solver, while the ATS method achieved a simulation speedup of 6.02-fold.

As an application example of the developed method, we computed 80 long-term viscoelastic Green's functions for the large-scale 3D crustal model of the Nankai region with 4.2×10^9 degrees of freedom using 56 A100 GPUs. Such results were used to estimate the slip-deficit rate and historical coseismic slip distribution given synthetic observation data. Slip-deficit estimation results show that long-term viscoelastic inversion outperforms the

inversion assuming a completely relaxed state.

The following is the structure of this paper. Section 2 describes the target viscoelastic crustal deformation problem and inversion analysis. Section 3 describes the development of the GPU-based solver with the proposed methods, and Section 4 describes the performance improvement by these methods. Section 5 describes an application example of the developed method computing long-term viscoelastic Green's functions for the Nankai Trough subduction zone. Finally, Section 6 summarizes this study.

2 Target problem

Our target problem is to estimate the elastic/viscoelastic crustal response on a detailed 3D crustal model. Section 2.1 shows the details of the Finite Element (FE) formulation for coseismic and post-seismic crustal deformation based on previous studies. The FE method was selected because of its ability to deal with complex geometries, such as the three-dimensional crustal structure. At the end of Section 2.1, we present an algorithm for calculating the time-history viscoelastic crustal deformation by iteratively solving large-scale linear systems derived from the FE model.

Section 2.2 describes the construction of a high-fidelity FE model designed for complex crustal geometries. This FE model results in a large linear system of 10^{9-10} DOF, which is solved iteratively in 10^{2-4} time steps. However, conventional computers cannot perform such a large-scale computation. Therefore, Section 3 introduces a method to solve large-scale linear systems efficiently.

The viscoelastic response due to a fault slip distribution can be estimated based on the finite-element formulation. However, in real applications, the viscoelastic response is known at certain observation stations on the crustal surface, but fault slip distribution is unknown. Therefore, conducting an inversion analysis becomes necessary to estimate the slip distribution along the plate interface and improve our understanding of future earthquake potential. In that sense, Section 2.3 shows the formulation of inversion analyses to estimate the distribution of coseismic slip and slip-deficit rate.

2.1 Finite element formulation

2.1.1 Coseismic crustal deformation

Given the time scale of coseismic crustal deformation due to a fault slip, which typically takes a few minutes, we can consider the crust (including the lithosphere and asthenosphere) as a linear elastic solid. Here, we analyze the static elastic response considering a prescribed slip distribution on the plate interface. The governing equations for this analysis are provided below,

$$\sigma_{ij,j} + f_i = 0 \tag{1}$$

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \tag{2}$$

$$\epsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$

where σ_{ij} and f_i are stress tensor, and force due to a fault slip. $\delta_{ij}, \epsilon_{ij}, u_i$ are Kronecker delta, strain tensor and displacement, respectively. λ and μ are Lame's constants. We used the FE method to solve the above governing equations. By discretizing Eq. (1) with quadratic tetrahedral elements, we obtain,

$$\mathbf{K}\mathbf{u}^0 = \mathbf{f}^0$$

where

$$\mathbf{K} = \sum_e \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega_e$$

Here, $\mathbf{K}$ is a symmetric positive definite matrix, and $\mathbf{u}^0$ and $\mathbf{f}^0$ are an unknown displacement vector and a known force vector. $\sum_e$ denotes an "assembly operator" that adds element contributions to the appropriate locations in $\mathbf{K}$. $\mathbf{D}$ is the elasticity matrix for isotropic material and $\mathbf{B}$ is strain-displacement transformation matrix. $\mathbf{D}$ is defined in a matrix form of Eq. (2) as,

$$\mathbf{D} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

In the case of quadratic tetrahedral elements, the strain-displacement transformation matrix is defined as $\mathbf{B} = [\mathbf{B}_1 \mathbf{B}_2 ... \mathbf{B}_{10}]$, with

$$\mathbf{B_i} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \\ N_9 \\ N_{10} \end{bmatrix} = \begin{bmatrix} r_1(2r_1-1) \\ r_2(2r_2-1) \\ r_3(2r_3-1) \\ r_4(2r_4-1) \\ 4r_1r_2 \\ 4r_2r_3 \\ 4r_1r_3 \\ 4r_1r_4 \\ 4r_2r_4 \\ 4r_3r_4 \end{bmatrix}, \quad r_4 = 1 - r_1 - r_2 - r_3$$

Here x, y, z and r_1, r_2, r_3 are global and natural coordinate systems, respectively. The global derivatives in matrix $\mathbf{B}_i$ can be obtained by utilizing Jacobian matrix $\mathbf{J}$ as follows:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial r_1} \\ \frac{\partial N_i}{\partial r_2} \\ \frac{\partial N_i}{\partial r_3} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r_1} & \frac{\partial y}{\partial r_1} & \frac{\partial z}{\partial r_1} \\ \frac{\partial x}{\partial r_2} & \frac{\partial y}{\partial r_2} & \frac{\partial z}{\partial r_2} \\ \frac{\partial x}{\partial r_3} & \frac{\partial y}{\partial r_3} & \frac{\partial z}{\partial r_3} \end{bmatrix}$$

Regarding the force vector $\mathbf{f}^0$ due to a fault slip, it can be defined using the split-node technique (Melosh and Raefsky, 1981) as:

$$\mathbf{f}^0 = -\left[\mathbf{K}^{upper} \left(\frac{\delta \mathbf{u}^f}{2} \right) + \mathbf{K}^{lower} \left(-\frac{\delta \mathbf{u}^f}{2} \right) \right]$$

where $\delta \mathbf{u}^f$, $\mathbf{K}^{upper}$ and $\mathbf{K}^{lower}$ are the slip on the plate interface and stiffness matrices consisting of the element stiffness matrices in the upper/lower sides of the plate interface.

The coseismic crustal deformation is obtained by solving $\mathbf{K}\mathbf{u}^0 = \mathbf{f}^0$. Then, the stress due to fault slip forces is obtained by $\boldsymbol{\sigma}^0 = \mathbf{D}\mathbf{B}\mathbf{u}^0$, which is evaluated at Gaussian points in each element.

2.1.2 Post-seismic crustal deformation

Linear viscoelastic models have been widely used to simulate the post-seismic crustal relaxation phenomenon. Although nonlinear models may offer improved accuracy in certain cases, linear viscoelasticity remains valuable in geodetic inversion due to its superposition property. This study adopts the Maxwell body as a preferred linear viscoelastic model. However, alternative linear viscoelastic models can be implemented following a similar formulation.

The Maxwell model is used for simulating the gradual relaxation of stresses in the Earth's crust following a fault slip as:

$$\begin{aligned} \sigma_{ij,j} + f_i &= 0 \\ \dot{\sigma}_{ij} = \lambda \dot{\epsilon}_{kk}\delta_{ij} + 2\mu\dot{\epsilon}_{ij} - \tfrac{\mu}{\eta}\left(\sigma_{ij} - \tfrac{1}{3}\sigma_{kk}\delta_{ij}\right) \\ \epsilon_{ij} &= \tfrac{1}{2}\left(u_{i,j} + u_{j,i}\right) \end{aligned} \tag{3}$$

where η is viscosity. To solve the above governing equations, we employ the same algorithm as the Geophysical Finite Element Simulation Tool, a FE package designed for geophysical and other applications provided by the NASA QuakeSim project (Parker et al., 2008).

The post-seismic crustal deformation at i-th time step is obtained by solving $\mathbf{K}^v \delta\mathbf{u}^i = \delta\mathbf{f}^i$, with viscoelastic linear system $\mathbf{K}^v$ defined as:

$$\begin{aligned} \mathbf{K}^v &= \sum_e \int_{\Omega_e} \mathbf{B}^T \mathbf{D}^v \mathbf{B} d\Omega_e \\ \mathbf{D}^v &= (\mathbf{D}^{-1} + \alpha dt \boldsymbol{\beta}')^{-1} \\ \boldsymbol{\beta}^i &= \mathbf{D}^{-1}\mathbf{A}\boldsymbol{\sigma}^i \end{aligned} \tag{4}$$

where $\boldsymbol{\sigma}^i$ is the stress at i-th time step, which is evaluated at Gaussian points in each element and α is a parameter ranging from 0 to 1 ($\alpha = 1$ for full implicit analysis). $\boldsymbol{\beta}^i$ represents the Maxwell viscoelastic strain rate, and $\boldsymbol{\beta}'$ its Jacobian matrix. In the case of linear viscosity, $\mathbf{A}$ is defined as follows:

$$\mathbf{A} = \frac{\mu}{3\eta}\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

The right-hand side vector $\delta\mathbf{f}^i$ is defined in the following manner:

$$\delta\mathbf{f}^i = \mathbf{f}^0 + \rho g \sum_e \int_{\delta\Omega_e} \mathbf{N}^T(\mathbf{u}^i, \hat{\mathbf{n}}_g)\hat{\mathbf{n}}_g d\delta\Omega_e + \sum_e \int_{\Omega_e} \mathbf{B}^T(dt\mathbf{D}^v\boldsymbol{\beta}^i - \boldsymbol{\sigma}^i)d\Omega_e \tag{5}$$

where $\rho, g, \hat{\mathbf{n}}_g$ and $(,)$ are density, magnitude of gravity, the normal vector of gravity force, and dot product. Now, we can define Algorithm 1 based on the above formulations to illustrate the crustal deformation analysis.

Algorithm 1 Analysis of viscoelastic crustal deformation due to a fault slip. The computation is conducted for N^t time steps. Superscript $()^i$ denotes a variable in the i-th time step. In each time step, we solve a linear system with the initial solution $\delta\mathbf{u}^i$, which is set using the data-driven predictor (detailed in Section 3).

1. Coseismic crustal deformation analysis

   set $i = 0$
   compute $\mathbf{f}^0$ from given fault slip $\delta\mathbf{u}^f$ ▷ use split-node technique
   solve $\mathbf{K}\mathbf{u}^0 = \mathbf{f}^0$ ▷ use Algorithm 2
   $\boldsymbol{\sigma}^0 \leftarrow \mathbf{D}\mathbf{B}\mathbf{u}^0$

2. Post-seismic crustal deformation analysis

   while $i < N^t$ **do**
      compute $\delta\mathbf{f}^i$ ▷ use Eq. (5)
      predict solution $\delta\mathbf{u}^i$ based on previous steps
      solve $\mathbf{K}^v\delta\mathbf{u}^i = \delta\mathbf{f}^i$ ▷ use Algorithm 2
      $\mathbf{u}^{i+1} \leftarrow \mathbf{u}^i + \delta\mathbf{u}^i$ ▷ update displacement field
      $\boldsymbol{\sigma}^{i+1} \leftarrow \boldsymbol{\sigma}^i + \mathbf{D}^v(\mathbf{B}\delta\mathbf{u}^i - dt\boldsymbol{\beta}^i)$ ▷ update stress field
      $i \leftarrow i + 1$
   end while

2.2 High-fidelity finite element model

The FE model for the Nankai Trough subduction zone was constructed using the Japan integrated velocity structure model version 1 (JIVSM) (Koketsu et al., 2009, 2012), which has been proposed for earthquake hazard assessments conducted by the Japanese government. To be more specific, the high-fidelity FE model was constructed from the layer boundaries of the subsurface structure model in JIVSM, along with the viscoelastic oceanic mantle and viscoelastic mantle wedge set following Sherrill and Johnson (2021) (all other layers are considered to be elastic).

In viscoelastic response analysis, higher-order elements should be used to accurately evaluate the stress in a crustal structure with complex geometry. In the following lines, we will elaborate on efficiently meshing complex geometries using higher-order elements and the application of fault slip in the constructed model.

2.2.1 Finite element mesh construction for parallel computing

The construction of the finite element mesh is conducted in two stages. First, a linear tetrahedral mesh is created. Second, the mesh is partitioned into domains for parallel computing, and a quadratic tetrahedral mesh is constructed for each domain. In the first stage, we define a background mesh composed of voxel elements for the target region (see Fig. 1a). Then, the layer boundary geometry is approximated to avoid generating poor aspect-ratio elements (see Fig. 1b). Although this figure shows voxel elements of the same size, the inefficiently small elements can be enlarged (e.g., merging four 2D voxel elements into a new element or merging eight voxel elements in 3D). At the end of this stage, linear elements are generated from the background mesh and the approximated ground surface (or layer boundary) using Delaunay triangulation (see Fig. 1c).

In the second stage, we use METIS (Karypis and Kumar, 1998) to perform domain decomposition for parallel computation, with processes working concurrently and communicating, to solve a system of equations by the FE method. We utilize Message Passing

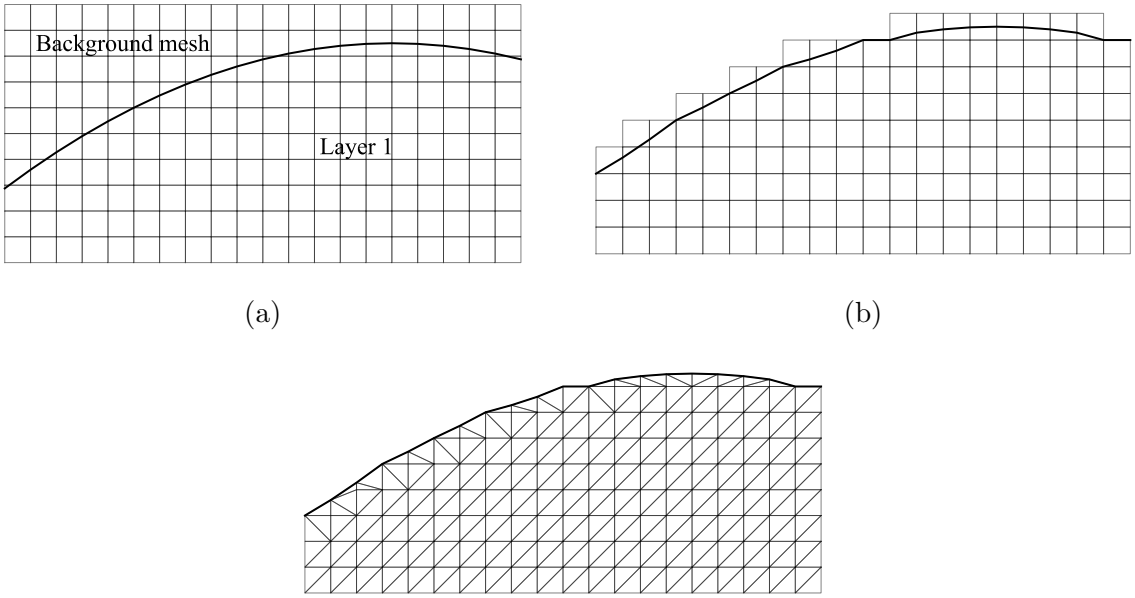


Fig. 1: Schematic view depicting the first stage of FE mesh generation. (a) Target domain is covered by a uniform background mesh. (b) The boundary geometries are slightly simplified to maintain the mesh quality. (c) Linear tetrahedral elements are generated from background mesh and layer boundary.

Interface (MPI) for communication between parallel processes. Then, we use the linear tetrahedral elements (defined for the coarse model) to generate quadratic tetrahedral elements for the fine target model.

Finally, the information of coarse and fine models, boundary conditions, process communication, and mapping between fine-coarse models is stored in Hierarchical Data Format 5 (HDF5, 2017). The model partitions are stored in HDF5 format because it is specifically designed to support parallel I/O operations, enabling efficient reading and writing of large datasets in parallel computing environments.

2.2.2 Application of fault slip in crustal model

The fault slips are expected to occur at the boundary between the top of the subducting oceanic crust and the bottom of the continental crust in JIVSM, particularly in the context of subduction earthquakes. Although fault slips do not always occur at this boundary, we assume that the relative motion between two plates is accommodated on this plate interface. Fig. 2 shows a 3D visualization of the plate interface and the input slip applied at some points on this boundary.

Note that the specific fault slip is defined in a single direction. However, if we aim to consider the actual motion, we can input a fault slip that is perpendicular to this defined direction. In cases where the fault slip or part of its distribution extends beyond the trench or trough, the slip is forcibly set to zero. However, it is crucial to ensure continuity in the distribution of fault slips in all other cases to prevent the accumulation of stress over time.

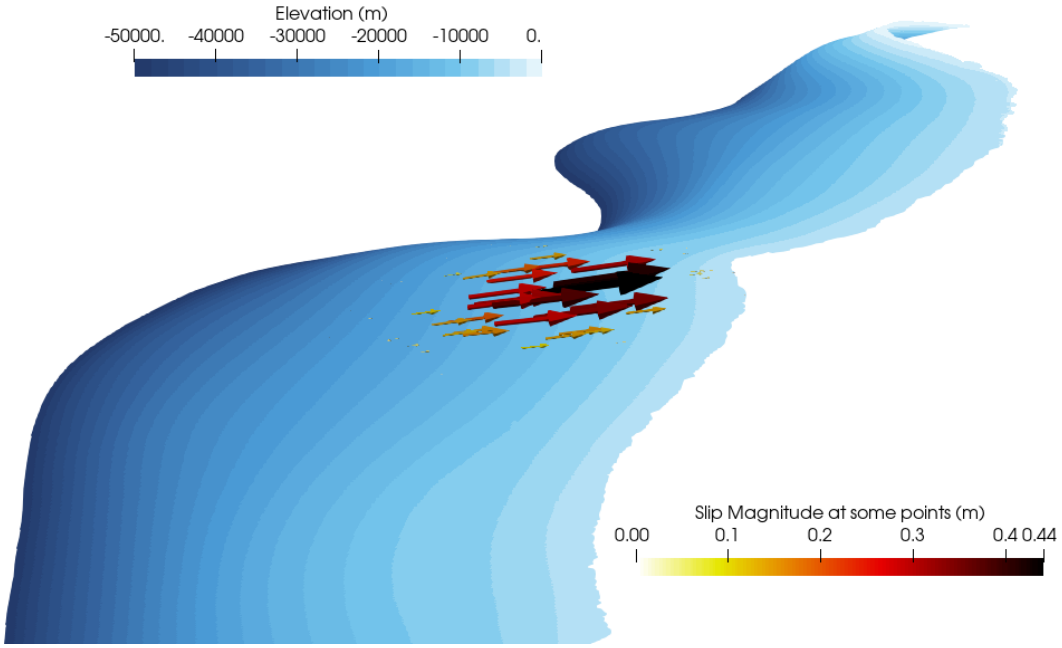


Fig. 2: 3D visualization of the plate interface (top of subducting oceanic crust) and application of a specific fault slip at some points.

2.3 Inversion analysis formulation

Inversion analysis methods are crucial in estimating slip distributions on a plate interface from surface crustal deformation. This section elaborates on two inversion methods: coseismic inversion and viscoelastic inversion.

Coseismic inversion focuses on determining the slip distribution that occurred in an earthquake, assuming an elastic response of the Earth's crust. The viscoelastic inversion method offers a more comprehensive approach by considering the time-dependent response of the Earth's crust. Unlike coseismic inversion, viscoelastic inversion accounts for the post-seismic relaxation and inter-seismic slip-deficit over longer time scales. This method leads to a more accurate estimation of slip distributions and a better understanding of the crust's response to seismic events (Li et al., 2015).

Completely relaxed inversion is a specific case of viscoelastic inversion, where the time after the earthquake is assumed to be infinite. This approach uses crustal surface deformation to estimate inter-seismic slip-deficit rates assuming a completely relaxed state of the crust.

2.3.1 Coseismic inversion method

In order to estimate the distribution of coseismic slip, the plate interface is divided into "m" unit faults, each represented by a basis function that defines its shape. For the basis function H_j (Heaviside function defined for j-th unit slip), we use the normalized third-order bicubic B-splines (Yabuki and Matsu'ura, 1992). These B-spline-shaped unit slips are applied on the plate boundary along a structured grid (see Fig. 9 of Section 5.1). The slip direction is defined opposite to subduction based on previous studies (Hori et al., 2021; Murakami et al., 2022). Subsequently, the surface responses (Green's functions) due to unit faults are computed. Using a combination of "m" basis functions, the spatial

slip distribution on the plate interface can be represented as:

$$w(\xi) = \sum_{j=1}^{m} H_j(\xi) a_j \tag{6}$$

Using Green's functions G_{ij}^e (elastic surface displacement on observation point i for unit slip H_j), we can define the observation equation by arranging displacements at all stations in a vector form:

$$\mathbf{d} = \mathbf{G^e a} + \mathbf{e}$$

where $\mathbf{d}$ is crustal deformation data at observation points on the surface and $\mathbf{a}$ is a set of model parameters. Here, we assume that error $\mathbf{e}$ obeys Gaussian distribution with a mean value of 0 and covariance matrix $\mathbf{E}$.

The inversion process may encounter instability due to the limited number of observation data compared to the number of unit faults and potential errors in the observation data and crustal model. Regularization techniques are employed to ensure stability and obtain reliable solutions by incorporating a priori information on the slip distribution (smoothness). By imposing a smooth constraint on the slip distribution, we ensure a realistic and smooth slip variation along the fault. In this way, the model parameters, $\mathbf{a}$, can be determined by minimizing the objective function:

$$s(\mathbf{a}) = (\mathbf{d} - \mathbf{Ga})^T \mathbf{E}^{-1} (\mathbf{d} - \mathbf{Ga}) + \alpha^2 \mathbf{a}^T \mathbf{L}^T \mathbf{La} \tag{7}$$

where $\mathbf{L}$ is a matrix representing the smoothness constraint on the slip distribution, and α represents a parameter to control the smoothness constraint (Yabuki and Matsu'ura, 1992).

Based on Tomita et al. (2020) and Bayes' theorem, the prior constraint can be represented in the form of a probability density function as follows:

$$p(\mathbf{a}; \alpha^2 | \mathbf{d}) = c(2\pi\sigma^2)^{-\frac{N+M}{2}} |\mathbf{E}|^{-\frac{1}{2}} \left|\alpha^2 \mathbf{L}^T \mathbf{L}\right|^{\frac{1}{2}} \exp\left[-\frac{1}{2\sigma^2} s(\mathbf{a})\right]$$

where α, c, and σ^2 are the hyper-parameter adjusting the spatial smoothing, a normalizing factor independent of the unknowns, and a scale factor for the covariance matrix $\mathbf{E}$. The ABIC value is obtained by the following equation:

$$\mathrm{ABIC}(\alpha^2) = N \log s(\mathbf{a}^*) - \log \left|\alpha^2 \mathbf{L}^T \mathbf{L}\right| + \log \left|\mathbf{G}^T \mathbf{E}^{-1} \mathbf{G} + \alpha^2 \mathbf{L}^T \mathbf{L}\right| + C$$

with

$$\mathbf{a}^* = (\mathbf{G}^T \mathbf{E}^{-1} \mathbf{G} + \alpha^2 \mathbf{L}^T \mathbf{L})^{-1} \mathbf{G}^T \mathbf{E}^{-1} \mathbf{d}$$

where N and C are the number of observation data and a constant term independent of α^2. To find the optimal value of α^2, we can employ a conjugate direction method to iteratively search the minimum ABIC value.

Once the value of α^2 is found, we can obtain the best estimate of $\mathbf{a}$ by minimizing the objective function described in Eq. (7). In this optimization process, we can consider the coseismic slip as purely opposite to the subduction direction by setting non-negative constraints in the objective function minimization ($a_i \geq 0$).

2.3.2 Viscoelastic inversion method

In general, the viscoelastic surface displacement u caused by a slip motion on a plate interface Σ at the rate $\dot{w}$ that started at time $t = t_k$ can be written with the following equation (Fukahata et al., 2004):

$$u_i(\mathbf{x}, t) = \int_{t_k}^{t} \int_{\Sigma} G_i(\mathbf{x}, t-\tau; \boldsymbol{\xi}, 0)\dot{w}(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \tag{8}$$

where $G_i(\mathbf{x}, t; \boldsymbol{\xi}, \tau)$ denotes the i-th component of the viscoelastic Green's function and indicates viscoelastic displacement at a point $\mathbf{x}$ at a time t on the surface, due to a fault slip at point $\boldsymbol{\xi}$ at time τ on the plate interface Σ, and t_k is timing of fault slip.

In this study, we considered the occurrence of historical earthquakes or previous coseismic slip distributions ($w(\boldsymbol{\xi}, \tau) \neq 0$ if $t < t_k, k > 1$). The slip motion w_k of k-th earthquake can be decomposed into slip motion due to a coseismic slip w_k^c and due to a steady slip motion $\dot{w}_k^s$ after k-th earthquake (Noda et al., 2018):

$$w(\boldsymbol{\xi}, \tau) = w^c(\boldsymbol{\xi}, \tau) + w^s(\boldsymbol{\xi}, \tau), \quad \tau > t_{N_{eq}}$$

$$w^c(\boldsymbol{\xi}, \tau) = \sum_{k=1}^{N_{eq}} H(\tau - t_k) a_k^c(\boldsymbol{\xi}), \quad w^s(\boldsymbol{\xi}, \tau) = (\tau - t_{N_{eq}})\dot{a}_{N_{eq}}^s(\boldsymbol{\xi}) + \sum_{k=1}^{N_{eq}-1} (t_{k+1} - t_k)\dot{a}_k^s(\boldsymbol{\xi})$$

where H represents a Heaviside function and N_{eq} is the number of earthquakes to consider in viscoelastic inversion. According to this decomposition, we can rewrite Eq. (8) as follows:

$$u_i(\mathbf{x}, t) = \sum_{k=1}^{N_{eq}} \int_{\Sigma} G_i(\mathbf{x}, t - t_k; \boldsymbol{\xi}, 0) a_k^c(\boldsymbol{\xi}) d\boldsymbol{\xi} + \int_{t_1}^{t} \int_{\Sigma} G_i(\mathbf{x}, t-\tau; \boldsymbol{\xi}, 0)\dot{a}_k^s(\boldsymbol{\xi}) d\boldsymbol{\xi} d\tau, \quad t > t_{N_{eq}} \tag{9}$$

Note that we consider that the steady slip motion $\dot{a}_k^s(\boldsymbol{\xi})$ changes after an earthquake due to the redistribution of stresses and the release of accumulated strain. Therefore, the temporal integral in Eq. (9) must be integrated by intervals of $[t_k, t_{k+1}]$ for $k = (1, 2, ..., N_{eq} - 1)$ and the last interval $[t_{N_{eq}}, t]$. Then, we can define the viscoelastic Green's function related to $a_k^c(\boldsymbol{\xi})$ and $\dot{a}_k^s(\boldsymbol{\xi})$ as follows:

$$G_k^c(\mathbf{x}, t; \boldsymbol{\xi}, 0) = \begin{cases} 0, & t < t_k \\ G_i(\mathbf{x}, t - t_k; \boldsymbol{\xi}, 0), & t > t_k \end{cases}$$

$$G_k^s(\mathbf{x}, t; \boldsymbol{\xi}, 0) = \begin{cases} 0, & t < t_k \\ \int_{t_k}^{t} G_i(\mathbf{x}, t-\tau; \boldsymbol{\xi}, 0) d\tau, & t_k < t < t_{k+1} \\ c_k = \int_{t_k}^{t_{k+1}} G_i(\mathbf{x}, t-\tau; \boldsymbol{\xi}, 0) d\tau, & t > t_{k+1} \end{cases}$$

Using viscoelastic Green's functions $\mathbf{G}^v$, the observation equation for the coseismic slip and steady slip motion can be expressed in a vector form:

$$\mathbf{d} = \mathbf{G}^v \mathbf{a} + \mathbf{e} = \begin{bmatrix} \mathbf{G}_1^c & \mathbf{G}_1^s & \cdots & \mathbf{G}_{N_{eq}}^c & \mathbf{G}_{N_{eq}}^s \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^c \\ \dot{\mathbf{a}}_1^s \\ \vdots \\ \mathbf{a}_{N_{eq}}^c \\ \dot{\mathbf{a}}_{N_{eq}}^s \end{bmatrix} + \mathbf{e} \tag{10}$$

Like the coseismic inversion method, we impose spatial smoothing to the slip distributions and minimize the objective function by Eq. (7). Note that $\mathbf{G}_k^s$ coefficients are typically large, while $\dot{\mathbf{a}}_k^s$ parameters are relatively small compared to coseismic array values. Although Eq. (10) was derived considering absolute displacements relative to the time just before the first coseismic slip, we can change the reference time by subtracting the relative displacement and the corresponding Green's functions at the time when observation data becomes available.

3 Development of GPU-accelerated methods

In the previous section, we detailed the construction of a FE model and defined slips on a plate interface to conduct crustal deformation simulation. This simulation plays a crucial role in seismic inversion analysis. However, the computation involves huge analysis cost reflecting heterogeneous 3D structure for solving a large linear system of $\sim 10^{9-10}$ DOF for $\sim 10^{2-3}$ fault slips and $\sim 10^{2-4}$ time steps with $dt \sim 10$ days. The high computational cost highlights the importance of employing efficient algorithms and parallel computing strategies to address this challenge effectively.

Iterative solvers are preferred over direct solvers in current high-performance computing (HPC) environments due to their scalability and ability to distribute computational tasks across multiple processors efficiently. They leverage parallel computing to achieve faster convergence, making them well-suited for solving large-scale linear systems in complex and computationally demanding simulations.

In an iterative solver, there are two approaches to reduce the number of iterations or the solver time. The first approach is to provide a high-accuracy initial guess that brings the solution closer to convergence. The second approach is to employ an efficient preconditioner that improves the convergence rate, resulting in faster computation and reduced solver time.

To improve the initial guess, a Data-Driven Predictor (DDP) was proposed by Fujita et al. (2022) for estimating a high-accuracy initial solution, achieving a 3.19-fold speedup. Regarding preconditioners, the Multigrid preconditioner is effective because it reduces the number of iterations by estimating the solution at a coarser grid, where the computation cost is cheaper. The Data-Driven Predictor and Multigrid solver will be detailed in sections 3.1 and 3.2, respectively.

We propose two methods to enhance the computation speedup: Aggregation-based Correction (ABC) and Adaptive Time-stepping (ATS). The ABC method enhances the high-accuracy initial guess by eliminating the low-frequency error inherent in the DDP approach. On the other hand, the ATS method reduces the number of simulation steps, denoted as N^t, by increasing the time step dt when the viscoelastic relaxation decays. Section 3.3 and 3.4 will elaborate on the ABC and ATS methods, respectively.

3.1 Data-driven predictor

In post-seismic crustal deformation analysis, we solve a system over N^t time steps using an initial solution, as shown in Algorithm 1. To enhance convergence in the iterative solver, we implement a predictor that utilizes previous steps' solutions to estimate the initial solution for the current time step. For example, a backward finite difference ap-

proximation can be employed as an initial predictor using three previous steps:

$$\delta \mathbf{u}^i_{p3} = 2\mathbf{u}^{i-1} - 3\mathbf{u}^{i-2} + \mathbf{u}^{i-3}$$

To enhance the accuracy of initial guess, Fujita et al. (2022) have introduced the concept of dynamic mode decomposition (DMD) to estimate the solution at the current time step. DMD defines an operator that captures the system's time evolution utilizing information from s previous time steps.

By denoting the difference of initial predictor from the correct solution $\delta \mathbf{u}^i$ as $\mathbf{x}^i = \delta \mathbf{u}^i - \delta \mathbf{u}^i_{p3}$, and using the data from the previous s steps, we construct $\mathbf{X}^{i-1} = [\mathbf{x}^{i-1}\mathbf{x}^{i-2} \dots \mathbf{x}^{i-s}]$. Utilizing this matrix, the time-evolution operator $\mathbf{C}$ is defined as $\mathbf{X}^{i-1} = \mathbf{C}\mathbf{X}^{i-2}$ (in general $\mathbf{Y} = \mathbf{C}\mathbf{X}$). The initial solution $\delta \mathbf{u}^i_{DDP}$ that considers the modes estimated in local space (or block) and local time is computed as,

$$\mathbf{x}^i = \mathbf{C}\mathbf{x}^{i-1} \quad \text{(in general } \mathbf{y} = \mathbf{C}\mathbf{x}\text{)}$$
$$\delta \mathbf{u}^i_{DDP} = \delta \mathbf{u}^i_{p3} + \mathbf{C}(\delta \mathbf{u}^{i-1} - \delta \mathbf{u}^{i-1}_{p3})$$

In this data-driven method, given data sets $\mathbf{X}, \mathbf{Y}$ of sizes $m \times s$ (the number of DOF in a block $\times$ previous time steps) where $\mathbf{X}$ is the input and $\mathbf{Y}$ is the corresponding output, the response $\mathbf{y}$ to an input $\mathbf{x}$ is computed by

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \quad \mathbf{C} = \mathbf{Y}\mathbf{R}^{-1}\mathbf{Q}^{\mathbf{T}}$$

In contrast to DMD, $\mathbf{C}\mathbf{x}$ is calculated by the modified Gram-Schmidt orthonormalization. To provide more detail, $\mathbf{Q}$ and $\mathbf{R}$ are obtained from orthogonalization of $\mathbf{X} = \mathbf{Q}\mathbf{R}$ by Gram-Schmidt method ($\mathbf{Q}$ and $\mathbf{R}$ sizes are $m \times s$ and $s \times s$, respectively). Replacing this expression in $\mathbf{Y} = \mathbf{C}\mathbf{X}$ and isolating $\mathbf{C} = \mathbf{Y}\mathbf{R}^{-1}\mathbf{Q}^{\mathbf{T}}$. However, the direct implementation of this method is not suitable for GPUs with small memory capacity because it requires keeping matrices $\mathbf{X}, \mathbf{Y}$ and another temporary arrays on memory during orthogonalization. Therefore, Murakami et al. (2023) proposed a random matrix $\hat{\mathbf{R}}$ of size $n \times m$ ($n \ll m$) to reduce the size of the given data set $\mathbf{X}$. This modification makes the data-driven predictor suitable for GPU computing, reducing computational cost and memory usage.

As described above, the data-driven method is computed in local space (small domains or blocks). This approach allows us to avoid global communication, as we can independently estimate the displacement increment for each small domain using time-history data. Here, we partition the domain in each MPI process into nonoverlapping small domains using METIS (Karypis and Kumar, 1998). The partitioning can be seen in Fig. 3.

3.2 Fast and Scalable Finite Element solver

In crustal deformation analysis, most of the computation time is dedicated to solving the large system outlined in Algorithm 1. To address this, a fast and scalable solver that can effectively leverage the capabilities of HPC systems is required. Since multigrid-based iterative solvers satisfy these requirements, a conjugate gradient solver with a geometric multigrid preconditioner proved effective in viscoelastic analysis (Ichimura et al., 2016).

In that study, a single coarse grid level was constructed geometrically using linear tetrahedral elements (the same second-order elements' shape but without the edge nodes).

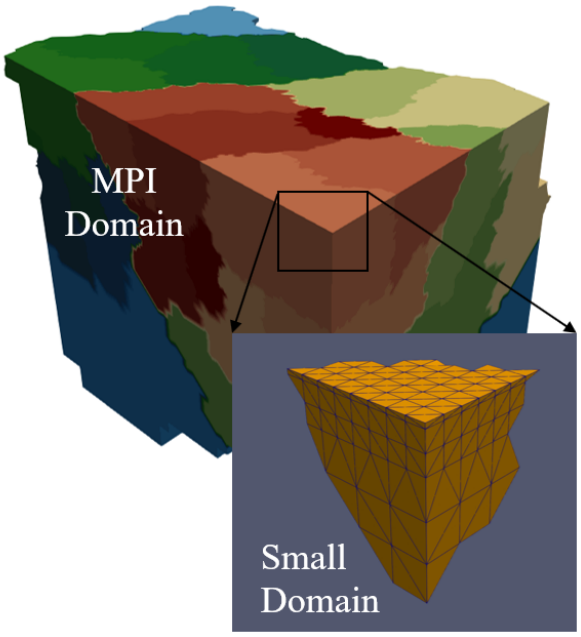


Fig. 3: MPI domain divided into small domains (blocks).

With structured grids, adding geometric coarse grid levels (i.e., equally spaced coarsening) and solving at this level often leads to an approximate solution close to the exact solution. However, for 3D unstructured grids, another geometric grid level cannot be used (i.e., there is no concept of equally spaced coarsening). Therefore, Fujita et al. (2017) proposed a solver with an integrated geometric/algebraic multigrid preconditioner, which resulted in a significant speedup of 2.18-fold compared to using only the geometric multigrid approach. This solver is shown in Algorithm 2.

The main loop of the iterative solver begins at line 3 (outer loop). In order to improve the convergence, a standard preconditioner is utilized at line 4 by transforming the original linear system into an equivalent one that is easier to solve. Here the equivalent system $\mathbf{B}$ is computed from the block diagonal of $\mathbf{K}$ (3x3 block Jacobi preconditioner).

Lines 6-12 correspond to the efficient three-level multigrid preconditioner, which involves operations of single-precision arrays. The integration of this preconditioner in a solver attained an 11.1-fold speedup compared to using only a standard preconditioner (Fujita et al., 2017). Like the standard preconditioner, an equivalent system is constructed with three levels: fine level, first coarse level, and second coarse level.

The fine-level system is obtained from the model with quadratic tetrahedral elements. In the first coarse level, the same model is used without the edge nodes, which is the model with linear tetrahedral elements. The second coarse level system is constructed from the first coarse level using the algebraic multigrid method. Restriction $\bar{\mathbf{R}}_j$ and prolongation $\bar{\mathbf{P}}_j$ operators are defined to map arrays between levels (e.g., $\bar{\mathbf{z}}_\mathbf{1} \Leftarrow \bar{\mathbf{R}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{0}$ and $\bar{\mathbf{z}}_\mathbf{0} \Leftarrow \bar{\mathbf{P}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{1}$ to map from level 0 to 1 and vice versa). Similarly, the stiffness matrix of a different level is related to another level using these operators (e.g., $\bar{\mathbf{K}}_\mathbf{1} = \bar{\mathbf{R}}_\mathbf{1}\bar{\mathbf{K}}_\mathbf{0}\bar{\mathbf{P}}_\mathbf{1}$ and $\bar{\mathbf{K}}_\mathbf{2} = \bar{\mathbf{R}}_\mathbf{2}\bar{\mathbf{K}}_\mathbf{1}\bar{\mathbf{P}}_\mathbf{2}$).

In line 8, the system of the second coarse level $\bar{\mathbf{K}}_\mathbf{2}\bar{\mathbf{z}}_\mathbf{2} = \bar{\mathbf{r}}_\mathbf{2}$ is solved using a conjugate gradient iteration. We refer to this iteration as inner loop level 2. Then, we prolongate the solution to the first coarse level and use it as an initial guess to solve $\bar{\mathbf{K}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{1} = \bar{\mathbf{r}}_\mathbf{1}$ by a conjugate gradient iteration, we refer to this iteration as inner loop level 1. Finally,

Algorithm 2 Fast and scalable iterative solver to obtain solution $\delta\mathbf{u}$. $(^{-})$ represents FP32 variables, while the others are in FP64. The input variables are $\mathbf{K}, \bar{\mathbf{K}}_j, \bar{\mathbf{R}}_j, \bar{\mathbf{P}}_j, \delta\mathbf{u}, \mathbf{f}$ and ϵ. Here, $\bar{\mathbf{K}}_j, \bar{\mathbf{R}}_j$ and $\bar{\mathbf{P}}_j$ are global stiffness matrix, restriction and prolongation operator related to multigrid level j. All computation steps in this solver, except MPI synchronization and scalar coefficient operations, are performed on GPUs.

1: $\mathbf{r} \Leftarrow \mathbf{f} - \mathbf{K}\delta\mathbf{u}$ ▷ synchronize $\mathbf{r}$ by MPI communication
2: $\beta \Leftarrow 0, i \Leftarrow 1$
3: **while** $||\mathbf{r}||_2/||\mathbf{f}||_2 > \epsilon$ **do** ▷ outer loop
4: $\quad \mathbf{z} \Leftarrow \mathbf{B}^{-1}\mathbf{r}$
5: $\quad \bar{\mathbf{r}}_\mathbf{0} \Leftarrow \mathbf{r}, \bar{\mathbf{z}}_\mathbf{0} \Leftarrow \mathbf{z}$ ▷ to single precision
6: $\quad \bar{\mathbf{r}}_\mathbf{1}, \bar{\mathbf{z}}_\mathbf{1} \Leftarrow \bar{\mathbf{R}}_\mathbf{1}\bar{\mathbf{r}}_\mathbf{0}, \bar{\mathbf{R}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{0}$ ▷ restrict to level 1
7: $\quad \bar{\mathbf{r}}_\mathbf{2}, \bar{\mathbf{z}}_\mathbf{2} \Leftarrow \bar{\mathbf{R}}_\mathbf{2}\bar{\mathbf{r}}_\mathbf{1}, \bar{\mathbf{R}}_\mathbf{2}\bar{\mathbf{z}}_\mathbf{1}$ ▷ restrict to level 2
8: $\quad$ Solve $\bar{\mathbf{K}}_\mathbf{2}\bar{\mathbf{z}}_\mathbf{2} = \bar{\mathbf{r}}_\mathbf{2}$ ▷ inner loop level 2, solve with Algorithm 3
9: $\quad \bar{\mathbf{z}}_\mathbf{1} \Leftarrow \bar{\mathbf{P}}_\mathbf{2}\bar{\mathbf{z}}_\mathbf{2}$ ▷ prolongate to level 1
10: $\quad$ Solve $\bar{\mathbf{K}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{1} = \bar{\mathbf{r}}_\mathbf{1}$ ▷ inner loop level 1, solve with Algorithm 3
11: $\quad \bar{\mathbf{z}}_\mathbf{0} \Leftarrow \bar{\mathbf{P}}_\mathbf{1}\bar{\mathbf{z}}_\mathbf{1}$ ▷ prolongate to level 0
12: $\quad$ Solve $\bar{\mathbf{K}}_\mathbf{0}\bar{\mathbf{z}}_\mathbf{0} = \bar{\mathbf{r}}_\mathbf{0}$ ▷ inner loop level 0, solve with Algorithm 3
13: $\quad \mathbf{z} \Leftarrow \bar{\mathbf{z}}_\mathbf{0}$ ▷ to double precision
14: $\quad$ **if** $i > 1$ **then**
15: $\quad\quad \gamma \Leftarrow (\mathbf{z}, \mathbf{q})$ ▷ synchronize γ by MPI communication
16: $\quad\quad \beta \Leftarrow \gamma/\rho$
17: $\quad$ **end if**
18: $\quad \mathbf{p} \Leftarrow \mathbf{z} + \beta\mathbf{p}$
19: $\quad \mathbf{q} \Leftarrow \mathbf{K}\mathbf{p}$ ▷ synchronize $\mathbf{q}$ by MPI communication
20: $\quad \rho \Leftarrow (\mathbf{z}, \mathbf{r})$ ▷ synchronize ρ by MPI communication
21: $\quad \gamma \Leftarrow (\mathbf{p}, \mathbf{q})$ ▷ synchronize γ by MPI communication
22: $\quad \alpha \Leftarrow \rho/\gamma$
23: $\quad \mathbf{r} \Leftarrow \mathbf{r} - \alpha\mathbf{q}$
24: $\quad \delta\mathbf{u} \Leftarrow \delta\mathbf{u} + \alpha\mathbf{p}$
25: $\quad i \Leftarrow i + 1$
26: **end while**

we prolongate the solution to the fine level system and use it as an initial guess to solve $\bar{\mathbf{K}}\bar{\mathbf{z}} = \bar{\mathbf{r}}$ by a conjugate gradient iteration. We refer to this iteration as inner loop level 0. The conjugate gradient solver used in inner loops is shown in Algorithm 3.

At lines 5 and 13 of Algorithm 2, we convert double-precision variables to single precision and vice versa so that computations within inner loops of multigrid preconditioner are performed using single-precision arithmetic. As a result, the computation cost is shifted from the outer loop to the multigrid inner loops, enabling double-precision results computed mostly with single-precision computation. This shift halves the memory footprint, memory transfer size, and communication size. If lines from 5 to 13 are skipped, the solver becomes a standard preconditioned conjugate gradient solver. We refer to this as standard solver (Ichimura et al., 2014).

In line 1 and line 19 of Algorithm 2, we use the element-by-element (EBE) method for computing sparse matrix-vector products. In the EBE method, matrix-vector products

are calculated by summing element-wise matrix-vector products as

$$\mathbf{q} = \sum_e \mathbf{Q}_e \mathbf{K}_e \mathbf{Q}_e^T \mathbf{p}$$

where $\mathbf{K}_e$ and $\mathbf{Q}_e$ indicate the element stiffness matrix and mapping between local and global node numbers. Instead of storing the element stiffness matrix in memory (e.g., Compressed Row Storage CRS format), the matrix-vector product is computed using nodal coordinates and material properties every time. This method is especially effective when targeting architectures with high arithmetic capability per memory bandwidth capability, such as GPU-based systems.

Note that in inner loops, we also compute sparse matrix-vector products (see line 10 of Algorithm 3). For efficient computation, we utilized arrays in CRS format for inner loops 1 and 2, while the EBE method was used for inner loop 0.

Algorithm 3 Conjugate gradient iteration with block Jacobi preconditioner used to solve $\bar{\mathbf{K}}_j \bar{\mathbf{z}}_j = \bar{\mathbf{r}}_j$ at inner loop level j in Algorithm 2. $\bar{\epsilon}$ and N^{max} are the tolerance for the relative error and the maximum loop iterations. $\bar{\mathbf{B}}_j$ is the block Jacobi matrix. Matrix-vector multiplication $\bar{\mathbf{K}}_j \bar{\mathbf{p}}$ is computed using the EBE method for inner loop $j = 0$, while for $j = 1, 2$ is computed using arrays in Compressed Row Storage format.

1: $\bar{\mathbf{e}} \Leftarrow \bar{\mathbf{r}}_j - \bar{\mathbf{K}}_j \bar{\mathbf{z}}_j$
2: $\bar{\beta} \Leftarrow 0, i \Leftarrow 1$
3: **while** $||\bar{\mathbf{e}}||_2 / ||\bar{\mathbf{r}}||_2 > \bar{\epsilon}$ and $i < N^{max}$ **do**
4: $\quad \bar{\mathbf{x}} \Leftarrow \bar{\mathbf{B}}_j^{-1} \bar{\mathbf{e}}$
5: $\quad \bar{\rho}_a \Leftarrow (\bar{\mathbf{x}}, \bar{\mathbf{e}})$ ▷ synchronize $\bar{\rho}_a$ by MPI communication
6: $\quad$ **if** $i > 1$ **then**
7: $\quad\quad \bar{\beta} \Leftarrow \bar{\rho}_a / \bar{\rho}_b$
8: $\quad$ **end if**
9: $\quad \bar{\mathbf{p}} \Leftarrow \bar{\mathbf{x}} + \bar{\beta} \bar{\mathbf{p}}$
10: $\quad \bar{\mathbf{q}} \Leftarrow \bar{\mathbf{K}}_j \bar{\mathbf{p}}$ ▷ synchronize $\bar{\mathbf{q}}$ by MPI communication
11: $\quad \bar{\gamma} \Leftarrow (\bar{\mathbf{p}}, \bar{\mathbf{q}})$ ▷ synchronize $\bar{\gamma}$ by MPI communication
12: $\quad \bar{\alpha} \Leftarrow \bar{\rho}_a / \bar{\gamma}$
13: $\quad \bar{\rho}_b \Leftarrow \bar{\rho}_a$
14: $\quad \bar{\mathbf{e}} \Leftarrow \bar{\mathbf{e}} - \bar{\alpha} \bar{\mathbf{q}}$
15: $\quad \bar{\mathbf{z}}_j \Leftarrow \bar{\mathbf{z}}_j + \bar{\alpha} \bar{\mathbf{p}}$
16: $\quad i \Leftarrow i + 1$
17: **end while**

3.3 Acceleration with Aggregation-based correction (ABC)

To improve the efficiency of the state-of-the-art solver, we analyzed the accuracy of the initial solution predicted by the DDP method, $\mathbf{e}_{DDP} = \delta \mathbf{u}^i_{DDP} - \delta \mathbf{u}^i_{sol}$. By identifying and addressing the sources of error, the prediction of the initial solution can be improved, resulting in enhanced speed and performance of the iterative solver.

Fig. 4 shows the error in data-driven prediction for displacement in the Z direction, grouped according to blocks. At first glance, there is evident a significant low-frequency error within the blocks used in data-driven prediction. This trend of low-frequency error

is also observed in the X and Y directions. Therefore, removing the significant low-frequency error will accelerate the solver. Additionally, the figure includes a zoomed-in view of the error within a specific block, allowing for better visibility of the high-frequency error components.

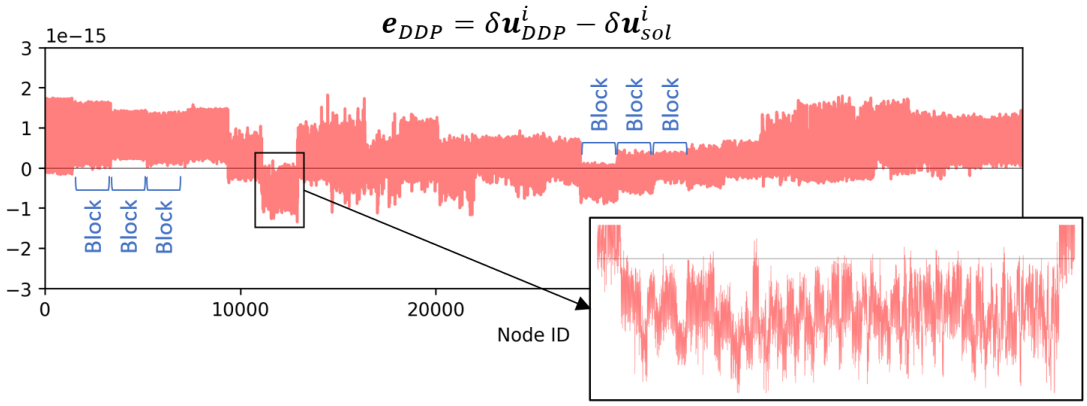


Fig. 4: Low-frequency and high-frequency error in data-driven prediction of initial guess.

Considering this observation, the error in data-driven prediction at i-th time step can be decomposed into two components: low-frequency (LF) error and high-frequency (HF) error:

$$\mathbf{e}_{DDP} = \mathbf{e}_{LF} + \mathbf{e}_{HF} = \delta\mathbf{u}^i_{DDP} - \delta\mathbf{u}^i_{sol} \tag{11}$$

We can employ a multigrid approach to address the significant low-frequency error present within the blocks used in the DDP method. This approach iteratively corrects the error by solving the problem on a coarse grid, effectively capturing the low-frequency components. For instance, applying the multigrid method solved low-frequency errors in large-scale problems effectively (Fujita et al., 2017). In implementing the multigrid method to eliminate low-frequency errors in blocks, we should consider the block structure when constructing the coarse grid. Specifically, each block in the fine grid should represent a new node in the coarse grid. In this way, we can improve the initial solution at low computational cost by aggressively coarsening the system from the number of DOF in the fine level to the number of DOF in the coarse block level (e.g., from 10^{9-10}DOF to 10^{4-5}DOF).

Among multigrid methods, we chose the aggregation-based multigrid method because it is suitable to remove low-frequency errors within aggregates or blocks. This correction is achieved by grouping a set of nodes into aggregates equivalent to constructing the coarse block level (Vanek et al., 1996). To prevent any confusion of terms, in the context of multigrid, we used the term "aggregates" to refer to the blocks used in the DDP method, while in domain partitioning, we used the term "small domain".

We named the aggregation-based multigrid method designed to correct the low-frequency error in the data-driven prediction as Aggregation-based Correction (ABC). The implementation of this method can be seen in Algorithm 4. Note that the ABC method, which removes low-frequency error, is applied after data-driven prediction and before solving $\mathbf{K}^v\delta\mathbf{u}^i = \delta\mathbf{f}^i$, where the high-frequency error is removed. In line 5 of Algorithm 4, we solve the system at the coarse block level and obtain $\mathbf{e}_b$. Then, we prolongate it to the fine level and remove the low-frequency error $\mathbf{e}_{LF}$.

Algorithm 4 Aggregate-based correction to remove low-frequency error at i-th time step. Here, $\mathbf{K}_b, \mathbf{R}_b$, and $\mathbf{P}_b$ represent the global stiffness matrix, restriction, and prolongation operator related to the coarse block level.

predict solution $\delta\mathbf{u}^i$ based on previous steps ▷ step from Algorithm 1
*** Correction starts ***
$\mathbf{r} \Leftarrow \mathbf{K}^v\delta\mathbf{u}^i - \delta\mathbf{f}^i$ ▷ compute residual
$\mathbf{r}_b \Leftarrow \mathbf{R}_b\mathbf{r}$ ▷ restrict to coarse block level
solve $\mathbf{K}_b\mathbf{e}_b = \mathbf{r}_b$ ▷ solve by Algorithm 3
$\mathbf{e}_{LF} \Leftarrow \mathbf{P}_b\mathbf{e}_b$ ▷ prolongate to fine level
$\delta\mathbf{u}^i \Leftarrow \delta\mathbf{u}^i - \mathbf{e}_{LF}$ ▷ remove low-frequency error
*** Correction ends ***
solve $\mathbf{K}^v\delta\mathbf{u}^i = \delta\mathbf{f}^i$ ▷ continue with Algorithm 1

3.4 Acceleration with Adaptive time-stepping (ATS)

Considering the governing equations (1),(2), and (3), the viscoelastic response decays exponentially with time as illustrated in Fig. 5.

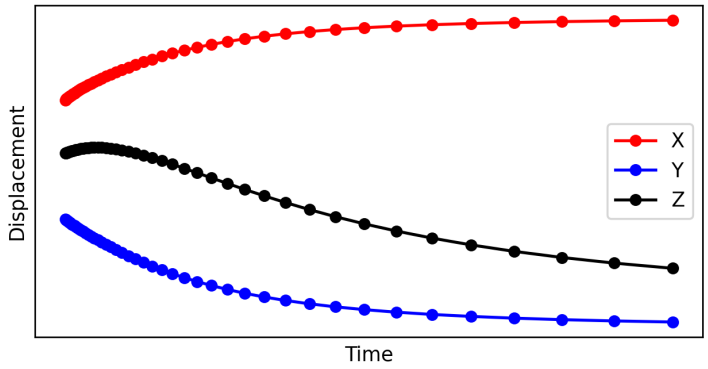


Fig. 5: Decay of viscoelastic response over time. Dots represent the solution in a logarithmic time spacing.

Considering this characteristic, previous studies considered logarithmic time spacing with shorter time steps after the earthquake. For instance, Johnson and Tebo (2018) analyzed the elastic/viscoelastic response of a simplified 2D model using this type of spacing. However, the application of this approach to a more complex 3D heterogeneous model has yet to be investigated. This study introduces an Adaptive Time-stepping method (ATS) to perform 3D viscoelastic simulations efficiently.

If we increase the time step dt during the simulation, the viscoelastic linear system $\mathbf{K}^v$ will change as described by Eq. (4) resulting in a challenging task of modifying a large linear system at each time step. However, we can exploit the linear viscoelastic behavior to separate the time-dependent systems conveniently, as described below:

$$\mathbf{K}^v = \sum_{k=1}^{N^{mat}} \mathbf{K}_k^0 + \sum_{k=1}^{N^{mat}} \frac{1}{1+\frac{\alpha dt\mu_k}{\eta_k}}\mathbf{K}_k^v \tag{12}$$

with

$$\mathbf{K}_k^0 = \sum_e \int_{\Omega_e} \mathbf{B}^T\mathbf{D}_k^0\mathbf{B}d\Omega_e, \quad \mathbf{K}_k^v = \sum_e \int_{\Omega_e} \mathbf{B}^T\mathbf{D}_k^v\mathbf{B}d\Omega_e$$

where N^{mat} is the number of layers with different materials. The linear viscoelasticity matrix for k-th material, is given by:

$$\mathbf{D}_k = \mathbf{D}_k^0 + \frac{1}{1 + \frac{\alpha dt \mu_k}{\eta_k}} \mathbf{D}_k^v \tag{13}$$

for elements in k-th elastic layer,

$$\mathbf{D}_k^0 = \begin{bmatrix} \lambda_k + 2\mu_k & \lambda_k & \lambda_k & 0 & 0 & 0 \\ \lambda_k & \lambda_k + 2\mu_k & \lambda_k & 0 & 0 & 0 \\ \lambda_k & \lambda_k & \lambda_k + 2\mu_k & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu_k & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu_k & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu_k \end{bmatrix}, \mathbf{D}_k^v = 0$$

and in the case of elements in k-th viscoelastic layer,

$$\mathbf{D}_k^0 = \frac{3\lambda_k + 2\mu_k}{3} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{D}_k^v = \frac{\mu_k}{3} \begin{bmatrix} 4 & -2 & -2 & 0 & 0 & 0 \\ -2 & 4 & -2 & 0 & 0 & 0 \\ -2 & -2 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

In this way, we can adapt dt depending on the viscoelastic response. Since the viscoelastic response decays exponentially with time, we propose the following expression:

$$dt = dt_0 e^{c \times i}, \quad c = \frac{dt_f - dt_0}{t_{total}}$$

where i is the i-th time step and c is the growth rate defined by the initial time step dt_0 and the final time step dt_f at the end of simulation, which occurs at time t_{total}. In the implementation of ATS, we update not only the system $\mathbf{K}^v$ according to the time step dt as described in Eq. (12), but also the auxiliary arrays required when solving the system $\mathbf{K}^v \delta\mathbf{u} = \mathbf{f}^v$ in Algorithm 2 and Algorithm 3.

For the computation in the fine level with double precision (e.g., $\mathbf{q} \Leftarrow \mathbf{Kp}$ and $\mathbf{z} \Leftarrow \mathbf{B^{-1}r}$), we update the linear viscoelasticity matrix according to the time step dt following Eq. (13) and compute using the EBE method. To update the three-level multigrid preconditioner arrays $(\bar{\mathbf{K}}, \bar{\mathbf{K}}_\mathbf{1}, \bar{\mathbf{K}}_\mathbf{2})$, we first precompute the non-time-dependent and time-dependent systems as shown in Eq.(12). Then, we updated the linear system $\mathbf{K}^v$ by only adding the precomputed systems in each time step. A similar process can be followed to update the block Jacobi preconditioner arrays $(\bar{\mathbf{B}}, \bar{\mathbf{B}}_\mathbf{1}, \bar{\mathbf{B}}_\mathbf{2})$.

Implementing the ATS method may seem straightforward initially, but it involves significant computation when working with large arrays in each time step. This implication can result in increased computation time. To address this challenge, we leverage the computational power of GPUs using OpenACC. By offloading the computation to GPUs, we can effectively accelerate the implementation and reduce its overall computation time.

4 Performance measurement

4.1 System environment for computation

We perform computations for viscoelastic crustal deformation and measure performance on a parallel GPU-based cluster at the Earthquake Research Institute of The University of Tokyo. The cluster comprises fifteen nodes, each node equipped with a 2.80 GHz 24-core AMD EPYC 7402P CPU and four NVIDIA Tesla A100 GPUs. This high-performance computing setup enables us to perform viscoelastic simulations efficiently. To evaluate the efficiency of our proposed methods, we measured their performance using a maximum number of 56 A100 GPUs.

4.2 3D high-fidelity crustal model

To demonstrate the utility of the proposed methods in viscoelastic analysis on a high-fidelity crustal structure model, we calculate post-seismic viscoelastic crustal deformation off the coast of the Nankai region of Japan due to a fault slip. The Nankai region is one of the plate subduction zones where large earthquakes occur.

A 3D finite-element model for computing Green's function is constructed based on the JIVSM (Koketsu et al., 2009, 2012). The crustal structure data are projected onto the cartesian coordinate system, originating at P (135°E, 33.5°N) and the ellipsoidal height 0 m. The target area domain is set between -1248 km $< x <$ 1248 km, -1248 km $< y <$ 1248 km, and -1100 km $< z$. Dirichlet boundary conditions were imposed on the bottom and sides of the finite-element model.

Fig. 6a provides an overview of the 3D FE model for Nankai Trough subduction zone. In Fig. 6b, a close-up view of the FE model illustrates the discretization of the target crustal structure model using second-order tetrahedral elements. Additionally, Fig. 6c shows the profile section A-B crossing at origin P, allowing a better appreciation of the layers within the model. The material properties of these layers were obtained from the JIVSM and are summarized in Table 1.

Table 1: Material properties of layers in Fig. 6a. V_p, V_s, ρ, and η are primary wave velocity, secondary wave velocity, density, and viscosity, respectively. ID indicates the layer number ID in JIVSM.

Layer name	**ID**	V_p (m/s)	V_s (m/s)	ρ (kg/m^3)	η (Pa·s)
Eurasian plate	1-16	2100 – 6400	700 – 3800	2050 – 2800	—
Mantle wedge (elastic)	17	7500	4500	3200	—
Mantle wedge (viscoelastic)	17	7500	4500	3200	2×10^{18}
Philippine sea plate	18-19	5000 – 6800	2900 – 4000	2400 – 2900	—
Oceanic mantle (elastic)	20	8000	4700	3200	—
Oceanic mantle (viscoelastic)	20	8000	4700	3200	10×10^{18}
Pacific plate (PAC)	21-22	5400 – 6500	2800 – 3500	2600 – 2800	—
Oceanic mantle (PAC)	23	8100	4600	3400	—

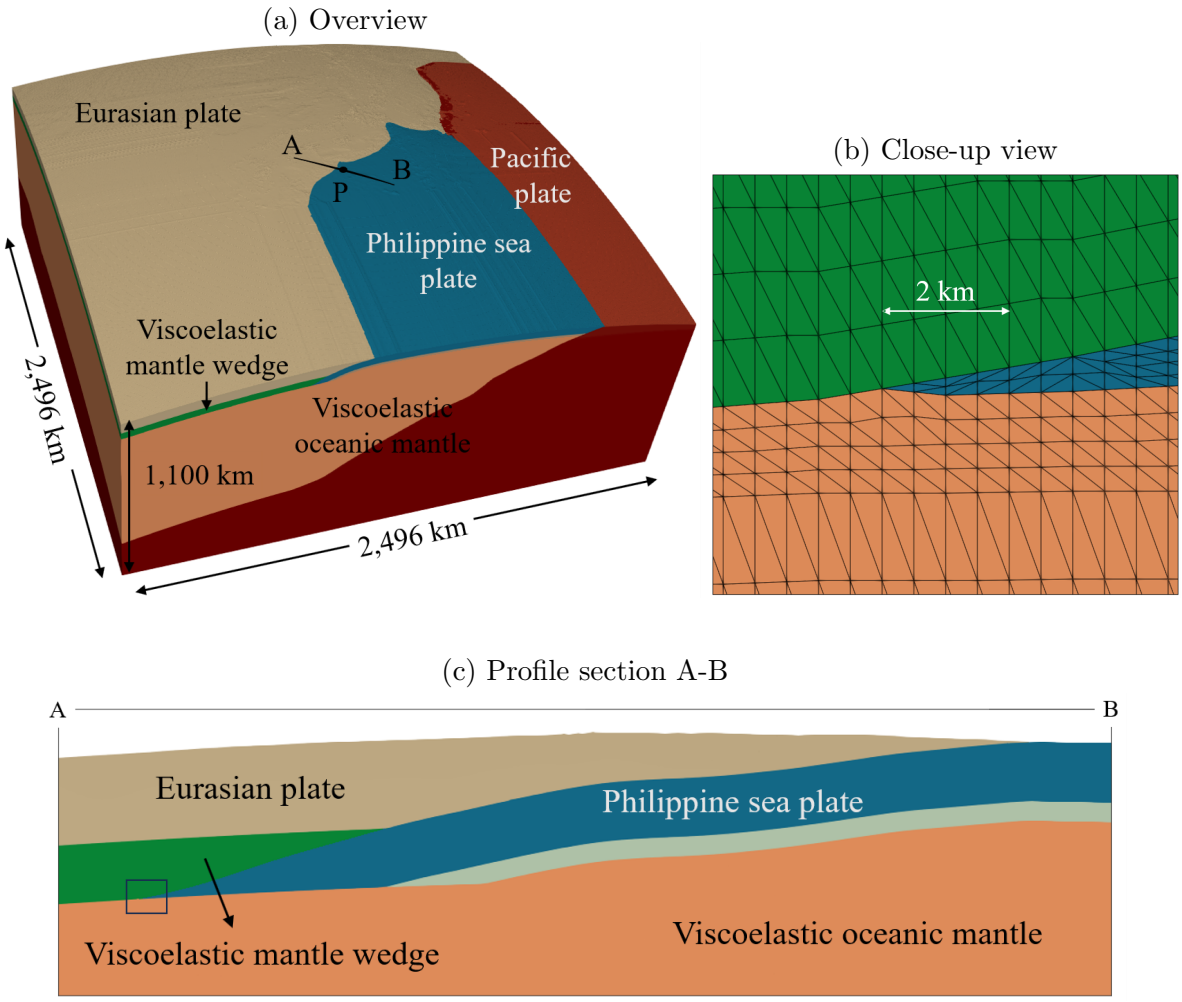


Fig. 6: 3D Finite-element model constructed for Nankai Trough subduction zone. (a) Overview, (b) close-up view, and (c) Profile section A-B. The black rectangle in profile A-B indicates the area shown in close-up view.

4.3 Performance measurement of ABC

To evaluate the performance of ABC, we keep the minimum element size to 2000 m and run simulations for one hundred time steps with $dt = 648,000s$. The resulting 3D finite-element model consisted of 2.8×10^8 DOF. We partitioned the model into 16 MPI domains to efficiently deal with the computational demand, enabling parallel computation across 16 GPUs.

The performance was measured from time steps 51 to 100 because the data-driven predictor needs information of previous steps to predict the initial guess of the current time step. By removing the low-frequency error in the initial guess of DDP, the number of iterations is reduced by 1.46-fold compared with the multigrid solver without ABC (see Table 2). As a result, the elapsed time per time step decreased to 1.18s, which is 1.22 times faster compared to using the state-of-the-art solver without ABC (see Table 3).

Table 2: Average elapsed time per step of each solver for time steps 51-100.

Solver name	Average elapsed time per step (s)				Speedup	
	Total	Solver	Predictor	Correction	Total	Solver
Multigrid + DDP	1.44	1.32	0.030	-	1.00	1.00
Multigrid+DDP+ABC	1.18	0.97	0.027	0.050	1.22	1.36

Table 3: Average required number of iterations per step for time steps 51-100.

Solver name	Average number of iterations per step			
	Outer	Inner 0	Inner 1	Inner 2
Multigrid + DDP	6.3	23.4	71.8	326.3
Multigrid+DDP+ABC	4.3	16.1	45.6	276.5

4.4 Performance measurement of ATS

To evaluate the performance of the ATS method, we conducted simulations with a minimum element size of 500 m over one hundred years. The resulting 3D finite-element model consisted of 4.2×10^9 DOF. In this case, we partitioned the model into 56 MPI domains, enabling parallel computation across 56 GPUs.

We considered 4800 time steps for fixed time-stepping simulations using a constant time step of $dt = 648,000$ s. In contrast, the number of steps for ATS simulation is only 537 with an initial time step $dt_0 = 648,000$ s and a final time step $dt_f = 32 \times dt_0$. Table 4 presents the simulation elapsed time and speedup for four different simulations with standard solver, multigrid solver, multigrid solver with data-driven predictor, and incorporating the ATS method. Elapsed times for simulations with standard and multigrid solvers are estimated by taking the average of the first 100 steps.

The ATS method demonstrated superior efficiency, reducing the number of time steps by 8.93-fold, which resulted in a significant 6.02-fold reduction in simulation elapsed time. Note that the reduction in time steps does not directly correspond to an equivalent reduction in elapsed time due to additional operations involved in updating simulation arrays (explained in Section 3.4).

In this study, we conducted simulations to estimate viscoelastic crustal deformation given a single fault slip vector per simulation. However, this computation can be enhanced by simultaneously solving for multiple vectors, which reduces the amount of memory access per vector. Recently, Murakami et al. (2023) developed a GPU solver designed explicitly for this purpose, which achieved an impressive 8.6-fold speedup compared to the multigrid solver. Therefore, we anticipate more significant speed improvements when incorporating this feature into our ATS implementation.

Table 4: Elapsed time for 100-year crustal deformation simulation.

Solver name	Elapsed time (min)			Speedup
	Total (steps)	Solver	Predictor	
Standard solver*	31152(4800)	31121	-	1.00
Multigrid solver*	1140.5(4800)	1130.5	-	27.31
Multigrid + DDP	371.1(4800)	318.4	4.01	86.21
Multigrid+DDP+ATS	61.60(537)	48.29	4.00	505.7

Notes: Elapsed times for simulations using standard and multigrid solver are estimated by taking the average of first 100 steps.

The varying time step dt can lead to error accumulation over time when using adaptive time-stepping. This error accumulation can result in deviations from the exact solution, compromising the accuracy of the simulation. To analyze this deviation, we examine a more rigorous scenario by extending the simulation time to 350 years and setting $dt_f = 96 \times dt_0$. Then, we compare the solution obtained from the ATS simulation with the solution of a reference simulation using a fixed time step. Fig. 7a shows the magnitude of horizontal displacement at the end of a 350-year simulation. Fig. 7b illustrates the difference of ATS displacement from the reference displacement on the surface at the end of the simulation. This difference is quantified as the error, denoted by $u_{ATS} - u_{ref}$, at the end of the simulation. Note that the maximum magnitude of the error is 1 mm, which is smaller than the typical precision range of surface deformation measurements (see Table 5).

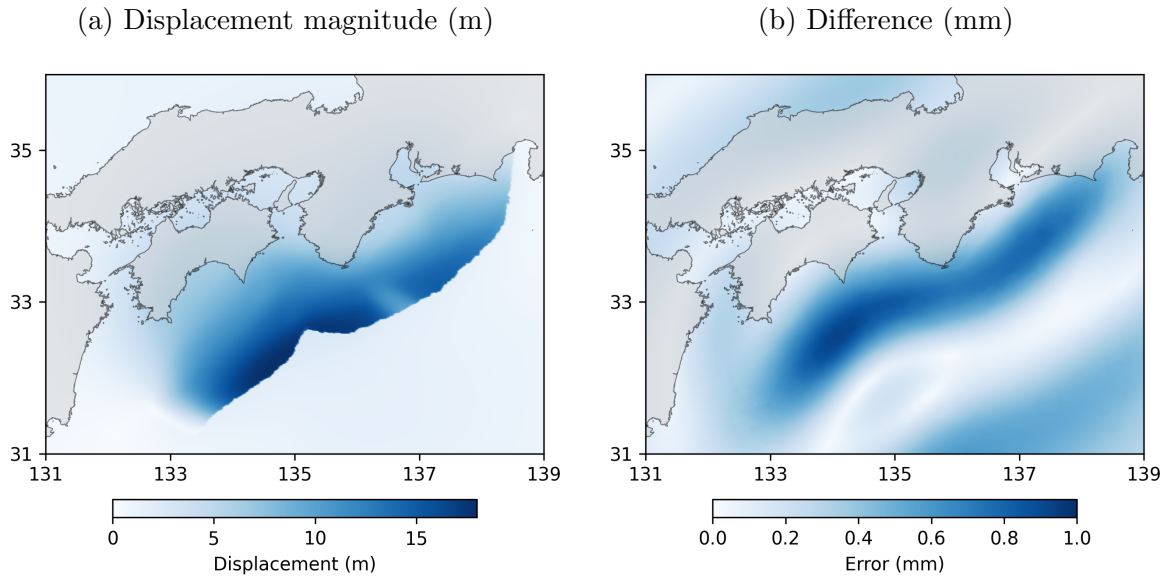


Fig. 7: (a) Horizontal displacement magnitude at the end of 350-year simulation and (b) Displacement difference between the solution of ATS method and the reference solution using a fixed time-stepping.

5 Application example

This section shows an application example to demonstrate the potential use of the developed long-term viscoelastic computation method. In previous studies, elastic or completely relaxed inversion methods have often been conducted to estimate slip-deficit distribution using data collected at specific times (Noda et al., 2018; Hori et al., 2021). However, we can improve the slip-deficit estimation accuracy by using the time-history viscoelastic response captured in geodetic observations. Furthermore, such time-history observation data might include information from historical earthquake slips. In this application example, we use the time-history viscoelastic Green's functions enabled by the developed method to do inversions of the slip-deficit as well as the coseismic slip of past earthquakes and compare the slip-deficit estimated by using Green's functions assuming the completely relaxed state.

In the first test, we use synthetic observation data collected 350 years after the occurrence of coseismic slip. With this data, we compare the estimation of slip-deficit distribution through viscoelastic inversion with the estimation obtained through completely relaxed inversion. The estimation is acceptable for both cases because a completely relaxed state is valid (after 350 years). In addition, coseismic slip is estimated by viscoelastic inversion.

In the second test, we simulated the coseismic slip during the 1944 and 1946 earthquakes in the Nankai region as a fused slip distribution in 1945 (Japan Cabinet Office, 2015). Synthetic observation data is recorded from 2006 to 2023 when onshore and seafloor geodetic observation is available. Using this data, we compare the estimation of slip-deficit distribution through viscoelastic inversion with the estimation through completely relaxed inversion. Additionally, viscoelastic inversion estimated coseismic slip with synthetic observation data that captured viscoelastic relaxation around 80 years after the earthquake.

We generated synthetic observation data for both tests by utilizing coseismic slip and inter-seismic slip-deficit distributions obtained from Japan Cabinet Office (2015) and Hori et al. (2021), respectively. We defined 80 unit slip parameters to represent each slip distribution and combined them as described in Eq. (6).

The combination of 80 parameters for coseismic slip is illustrated in Fig. 8a, while Fig. 8b depicts the combination of 80 parameters for inter-seismic slip-deficit. Therefore, we consider 160 input parameters to represent both slip distributions. The details of the 80 unit slips used in the combination are explained in Section 5.1.

(a) Reference coseismic slip.

(b) Reference inter-seismic slip-deficit.

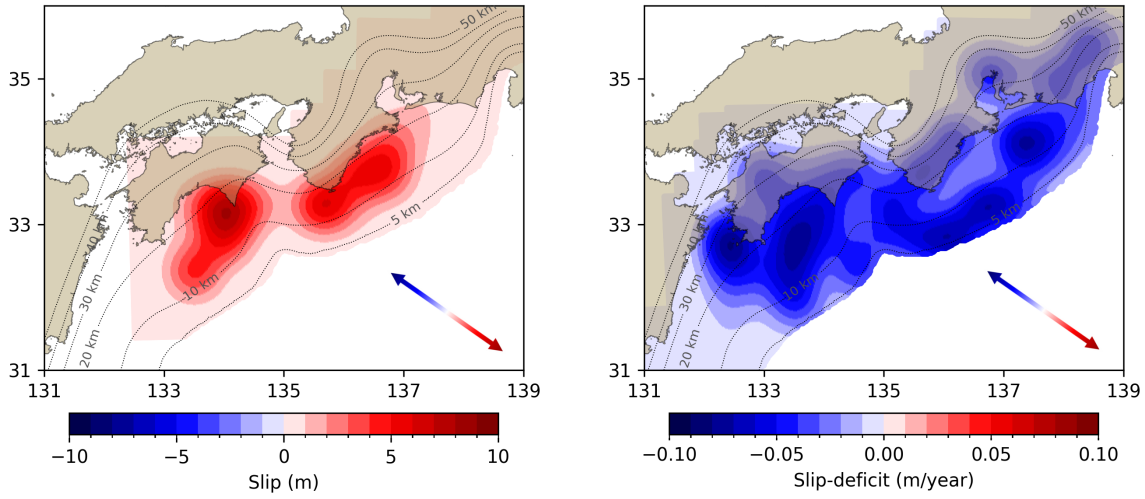


Fig. 8: Input reference slip distributions for synthetic tests.

To quantify the accuracy of the inversion methods, we use the misfit value r defined as:

$$r = \frac{||\mathbf{d}_{calc} - \mathbf{d}_{obs}||_2}{||\mathbf{d}_{obs}||_2}$$

Here, $||\cdot||_2$ denotes L2 norm, and $\mathbf{d}_{obs}$ is the synthetic observation displacement. The vector $\mathbf{d}_{calc} = \mathbf{G}\mathbf{a}_{calc}$ is the estimated displacement, where $\mathbf{a}_{calc}$ is the estimated model parameters from the inversion analysis.

To conduct the inversion analysis mentioned above, we compute long-term viscoelastic Green's functions for 350 years in the Nankai Trough subduction zone using the efficient ATS method proposed in this study.

5.1 Computation of Green's functions

From the examination result of Murakami et al. (2022), an element size of around 500 m is required for numerical convergence of the results; thus, elements are refined such that the element size in the horizontal direction is 500 m in the range of -448 km $< x <$ 448 km, -448 km $< y <$ 448 km, and $(h(x,y)-10)$ km $< z < (h(x,y)+10)$ km. Here, $h(x,y)$ is the z-coordinate of the plate boundary (for the region beyond the trench axis, $h(x,y)$ represents the z-coordinate of the ground surface). The generated FE model comprises 1.04×10^9 quadratic tetrahedral elements with 4.2×10^9 DOF. To efficiently deal with the computational demand, we partitioned the model into 56 MPI domains enabling parallel computation across 56 GPUs.

To compute Green's functions in the Nankai Trough subduction zone, we positioned 80 structured grid nodes along the plate boundary of our constructed model. These nodes are spaced in a 50 km grid, and we specifically selected nodes within a depth limit of 50 km. Then, we introduce unit slips centered at these nodes with third-order B-spline shape as basis function (Yabuki and Matsu'ura, 1992). Considering one direction of slip on each unit slip, we computed 80 Green's functions in total. Fig. 9 shows an example of a unit slip distribution and the location of 80 unit slips applied on the plate interface in the Nankai Trough subduction zone. The unit slip or part of its distribution located beyond the trench or Trough axis is set forcibly to zero.

The computation of 80 viscoelastic Green's functions using the FE model with 4.2×10^9 DOF for 350 years entails a high computational cost. However, the analysis time is reduced by employing the proposed adaptive time-stepping method and GPU implementation. This approach enabled the computation of all Green's functions in 2.96×10^5 s (3.42 days).

Finally, we output the crustal deformation response (Green's functions), due to unit slips, at the location of onshore and seafloor observation stations. We then transform coordinates from the crustal structure model to the global standard geodetic reference following Hori et al. (2021). The coordinates of the observation stations were obtained from Yokota et al. (2016) and are shown in Fig. 10.

5.2 Estimating slip-deficit using data from 350 years after past megaquake

In this application, we utilize long-term viscoelastic Green's functions to estimate interseismic slip-deficit distribution from synthetic observation data recorded 333-350 years after the last major earthquake. Here, we compare the conventional case using the completely relaxed inversion with the case using the time-history viscoelastic Green's functions for inversion.

The time-history data for 350 years was generated following Eq. (10), by multiplying the viscoelastic Green's functions with the input model parameters derived from the reference coseismic slip and slip-deficit. Fig. 11 shows the vertical displacement on the surface generated from the input model parameters over 350 years for two observation stations: Station 144 (onshore) and Station 200 (seafloor). Note that the surface vertical

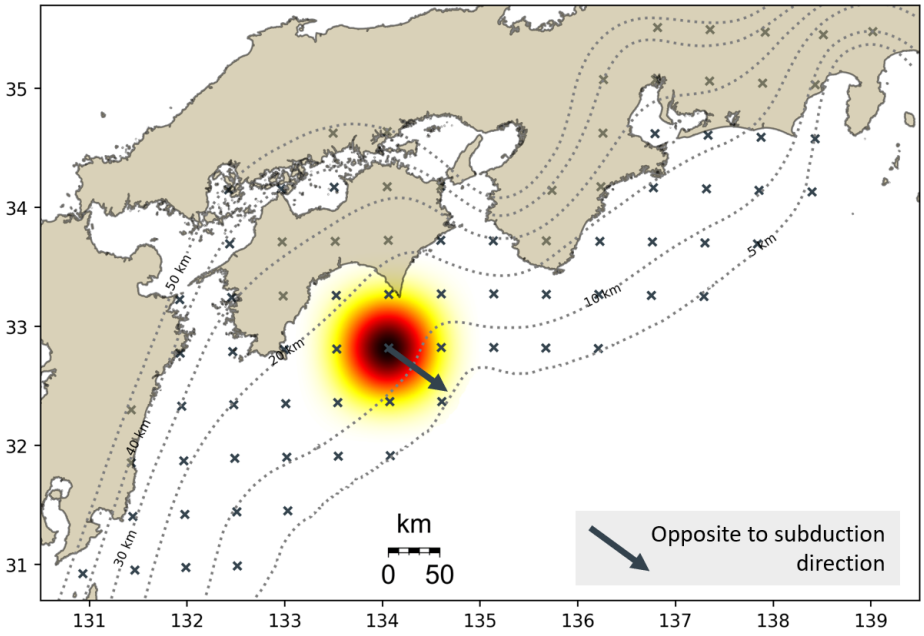


Fig. 9: Example of unit slip input in Nankai Trough subduction zone and location of 80 unit slips defined in direction opposite to subduction.

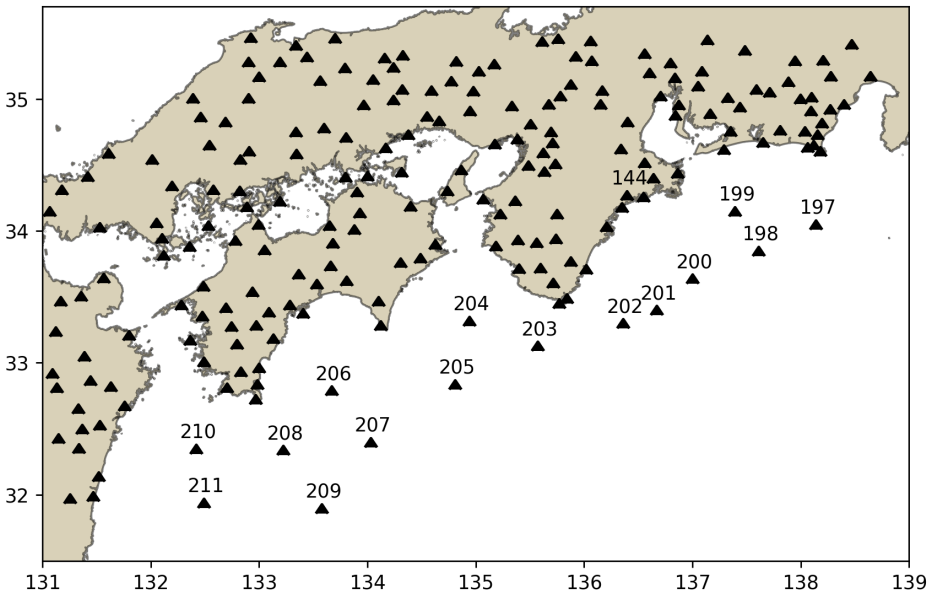


Fig. 10: Location of onshore and seafloor observation stations. Only the station IDs of seafloor observation and onshore station 144 are shown.

displacement in seafloor station 144 is more significant than in onshore station 200, but the viscoelastic relaxation decays over time in both stations. In addition, we introduce Gaussian noise attributed to the precision of geodetic observation instruments, which is shown in Table 5. The observation data is generated after a significant time has elapsed since the occurrence of the coseismic slip (within the range of 333-350 years), where the assumption of a completely relaxed state is valid.

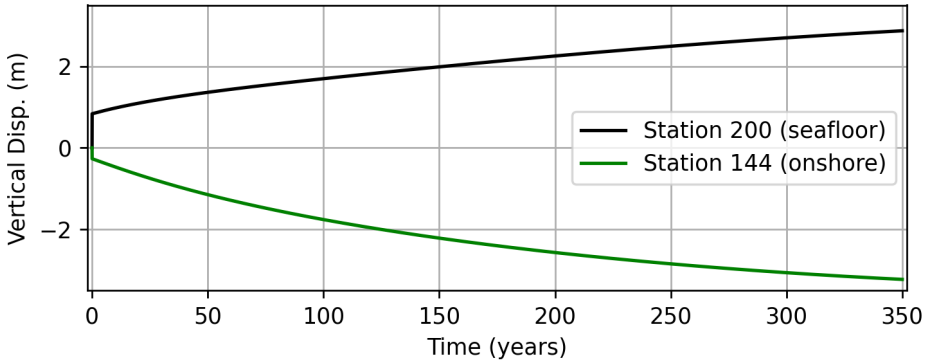


Fig. 11: Vertical displacement of onshore station 144 and seafloor station 200 for 350 years relative to year 0 (right before earthquake occurrence).

Table 5: Precision assumed for geodetic observation in inter-seismic periods.

	Onshore observation	**Seafloor observation**
Stations IDs	1-196	197-211
Precision	H: 4mm/year and V: 8mm/year	H: 5mm/year and V: 20mm/year

Notes: H and V indicate horizontal and vertical components, respectively.

The observation data generated in 333-350 years with the consideration of error due to the precision of geodetic instruments can be seen in Fig. 12. To consider a realistic case, we assume that observation data is available from year 333 and generate data relative to this year's displacement. Note that the magnitude of the vertical displacement rate is around 5 mm/year, comparable with the geodetic instrument's precision. Therefore, we also include horizontal components in the inversion analysis.

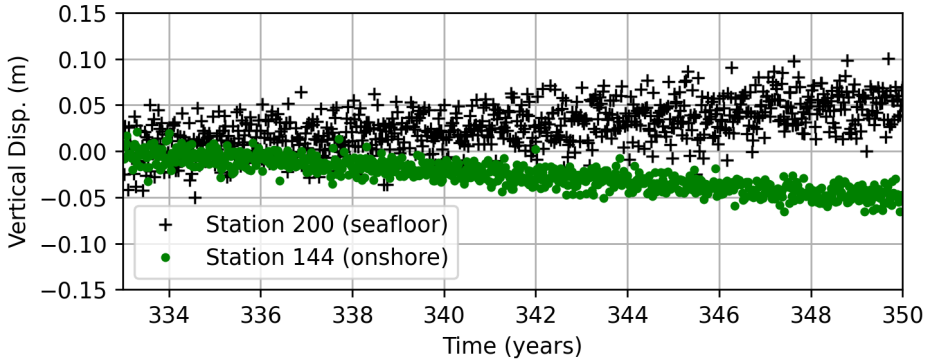


Fig. 12: Vertical displacement of onshore station 144 and seafloor station 200 for 333-350 years relative to year 333 (when observation data is available).

To estimate the slip-deficit distribution through the completely relaxed inversion, we calculate the difference between the displacement at year 350 and the displacement at year 345, which effectively neglects the viscous relaxation due to the coseismic slip during this period. This approach allows us to estimate only 80 model parameters associated with the slip-deficit. On the other hand, for the viscoelastic inversion, we consider all the time history data in the interval of 333-350 years and estimate 160 model parameters related to the coseismic slip and slip-deficit.

Fig. 13a shows the results of slip-deficit estimation. Both inversion methods yield reliable estimations, but the viscoelastic inversion outperforms the completely relaxed inversion (see the difference from reference slip-deficit Fig. 13b). In quantitative terms, the viscoelastic inversion achieves a misfit value of 5.05%, while the completely relaxed inversion exhibits a misfit value of 17.0%. These results were expected since the completely relaxed state is valid after a prolonged period following the coseismic slip.

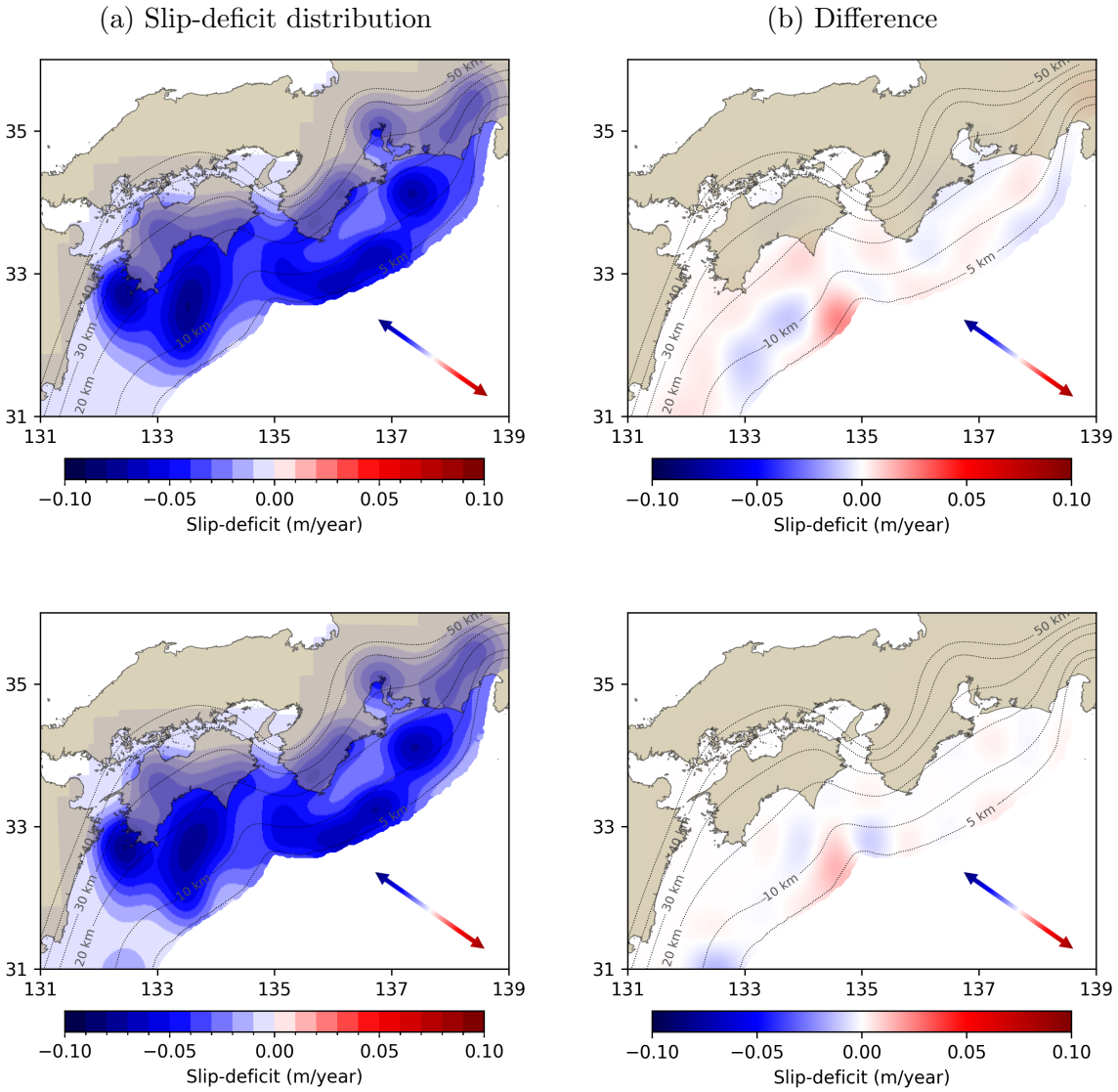
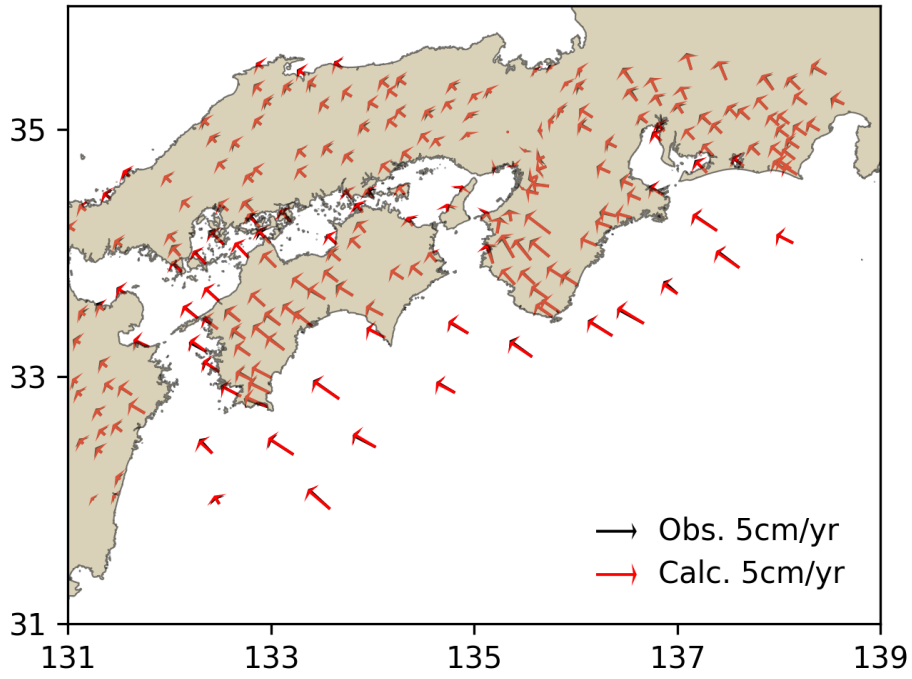


Fig. 13: Estimation results of slip-deficit distribution (350 years after earthquake). The slip-deficit and difference from reference distribution are shown from left to right for Completely Relaxed inversion (Above) and Viscoelastic inversion (Bottom).

From the estimated model parameters related to the slip-deficit, we plot the horizontal surface displacement rate detected with the data around 350 years after the earthquake (see Fig. 14). The red arrows in Fig. 14a and Fig. 14b represent the displacement rate calculated from completely relaxed and viscoelastic inversion, respectively. Compared to the vertical components, the horizontal displacement rate is larger in magnitude (around 4 cm/year).

(a) Completely relaxed inversion



(b) Viscoelastic inversion

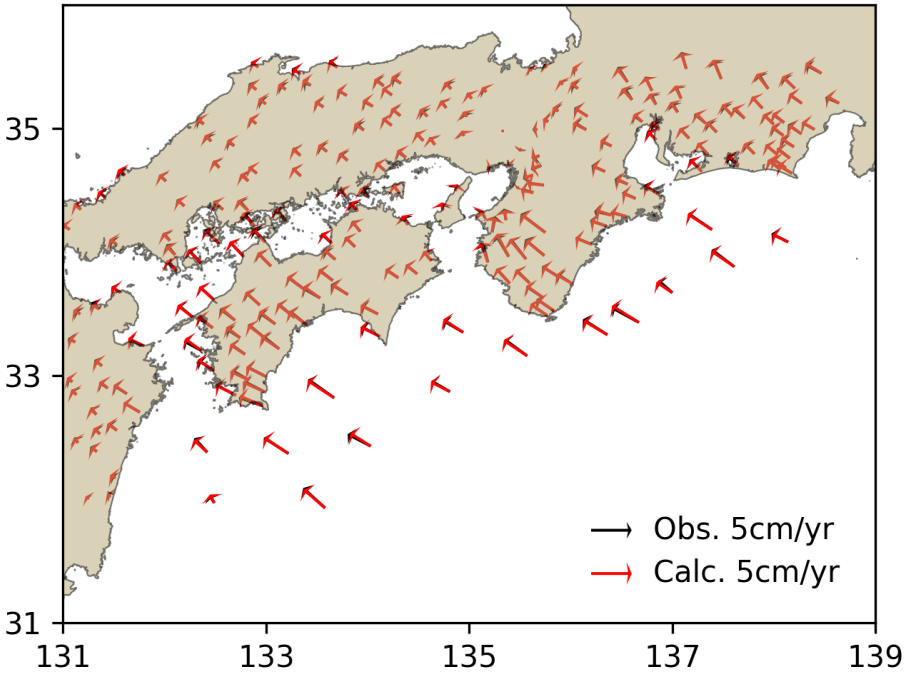


Fig. 14: Observed surface deformation rate at observation stations (black arrow) and calculated deformation rate (red arrow) at 80 years after earthquake.

In addition to estimating the slip-deficit, the viscoelastic inversion enables the estimation of coseismic slip. Despite the relaxation that occurred over time and the effect of coseismic slip after 350 years is small, we were able to obtain a rough slip estimation.

Fig. 15a shows the estimated distribution of the 350-year-ago coseismic slip. Note that the difference from the reference coseismic slip is significant (see Fig. 15b). The estimated moment magnitude (Mw) is 8.61, near the magnitude of reference slip (8.54), indicating that the earthquake's magnitude can be estimated. It is important to clarify that the Nankai region has experienced multiple megaquakes in the last 350 years (megathrust earthquakes occurred in 1707, 1854, 1944, and 1946). However, for this application, we only considered a single megathrust earthquake.

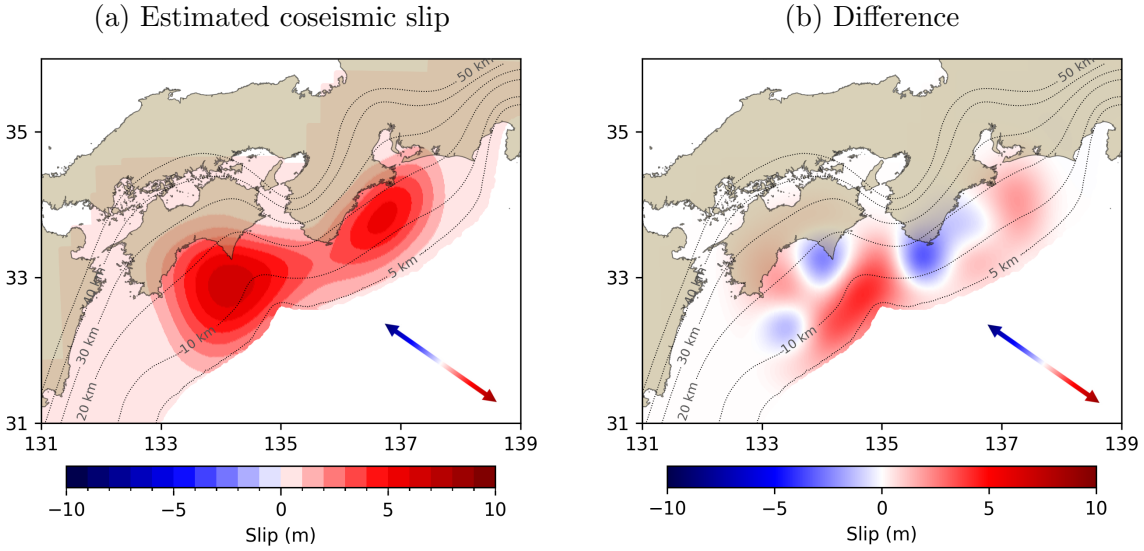


Fig. 15: Coseismic slip distribution estimated from long-term viscoelastic inversion considering synthetic data during 333-350 years at observation location.

5.3 Estimating slip-deficit using data from 80 years after past megaquake

In this application, we consider a realistic case of an earthquake in 1945, and only geodetic data is available from 2006 to 2023. Using synthetic observation data generated in this period, we utilize long-term viscoelastic Green's functions to estimate inter-seismic slip-deficit distribution.

Like the previous application, the time-history data from 1945 to 2023 was generated following Eq. (10). Fig. 16 shows the vertical displacement on the surface generated from the input model parameters over 78 years for Station 144 (onshore) and Station 200 (seafloor). The observation data generated from 2006 to 2023 with the consideration of error due to the precision of geodetic instruments can be seen in Fig. 17. Assuming that observation data is available from 2006, we generated data relative to this year's displacement.

To conduct a completely relaxed inversion, we calculate the difference between the displacement in 2018 and the displacement in 2023. Using this data, we estimated 80 model parameters associated with the slip-deficit. On the other hand, for the viscoelastic inversion, we consider all the time history data from 2006 to 2023 years and estimate 160 model parameters related to the coseismic slip and slip-deficit.

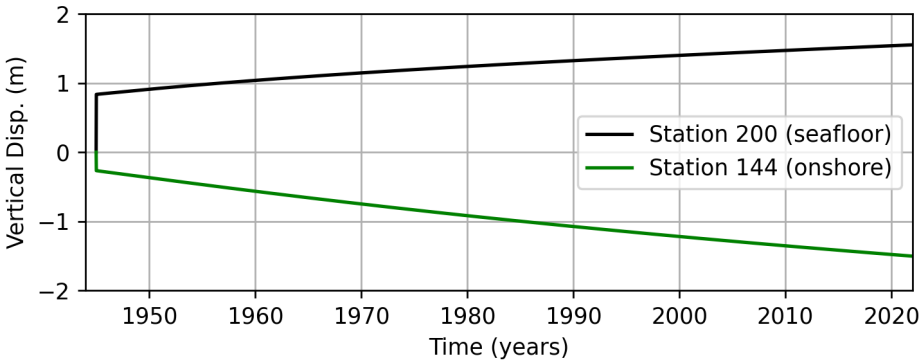


Fig. 16: Vertical displacement of onshore station 144 and seafloor station 200 for 1945-2023 years relative to year 1945 (right before earthquake occurrence).

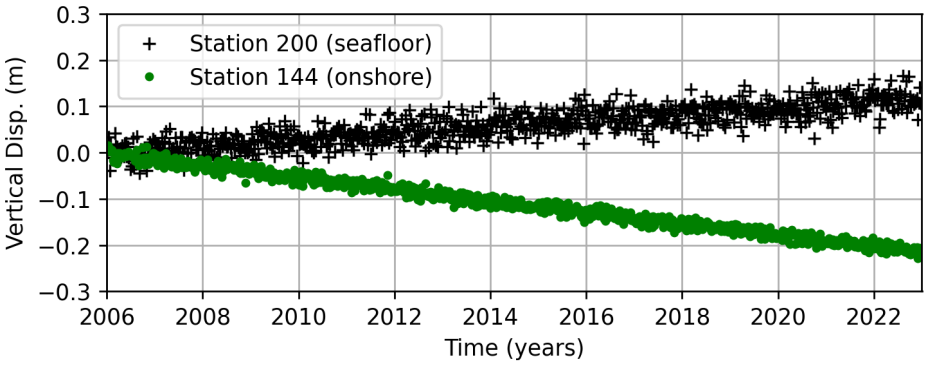


Fig. 17: Vertical displacement of onshore station 144 and seafloor station 200 for 2006-2023 years relative to year 2006 (when observation data is available)

Fig. 18a shows the results of slip-deficit estimation. Once again, the viscoelastic inversion outperforms the completely relaxed inversion (see Fig. 18b). In quantitative terms, the viscoelastic inversion achieves a misfit value of 4.45%, while the completely relaxed inversion exhibits a misfit value of 33.8%. The larger misfit value in the completely relaxed inversion is due to its omission of the viscous relaxation recorded in the observation data. Fig. 19 shows the horizontal surface displacement rate obtained from the estimated model parameters related to the slip-deficit. The red arrows in Fig. 19a and Fig. 19b represent the displacement rate calculated from completely relaxed and viscoelastic inversion, respectively.

Fig. 20a shows the estimated distribution of the 80-year-ago coseismic slip. Note that the difference from the reference coseismic slip is small (see Fig. 20b). The estimated moment magnitude (Mw) is 8.57, almost the same as that of the reference slip (8.54), indicating that the earthquake's magnitude is accurately estimated.

The main results of the application example are presented in Table 6. The viscoelastic inversion outperforms the completely relaxed inversion. That is because the viscoelastic inversion incorporates the time-history data rather than relying on a difference calculation at two specific times.

Furthermore, the coseismic slip can only be estimated through the viscoelastic inversion. It yields better results when utilizing observation data around 80 years after the earthquake than data around 350 years.

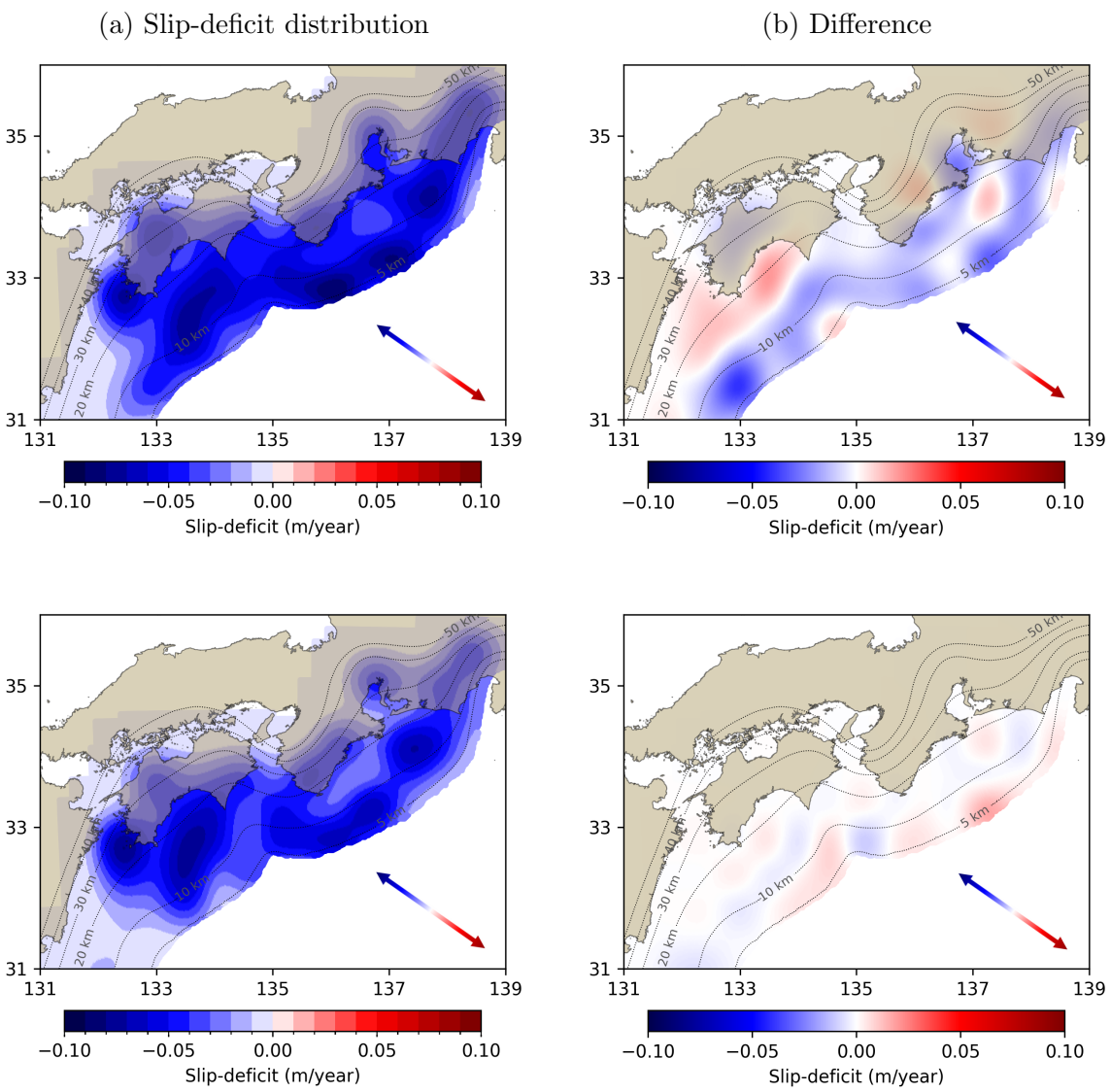
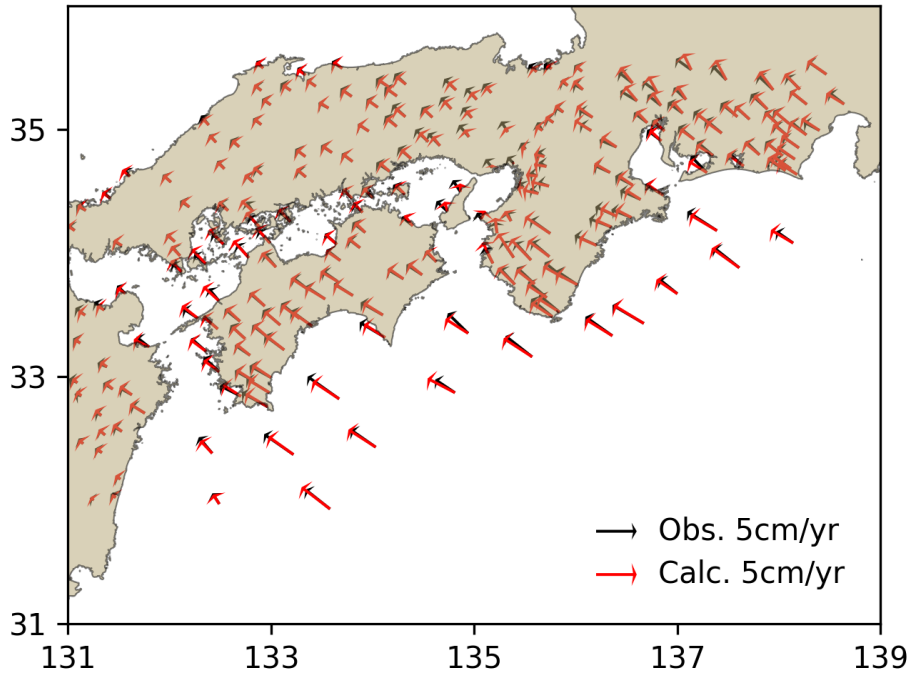


Fig. 18: Estimation results of slip-deficit distribution (78 years after earthquake). The slip-deficit and difference from reference distribution are shown from left to right for Completely Relaxed inversion (Above) and Viscoelastic inversion (Bottom).

Table 6: Comparison of completely relaxed inversion and viscoelastic inversion results.

		Completely relaxed inversion	**Viscoelastic inversion**
350-year-ago earthquake	Misfit	17.0%	5.05%
	Mw	—	8.61
80-year-ago earthquake	Misfit	33.8%	4.45%
	Mw	—	8.57

(a) Completely relaxed inversion



(b) Viscoelastic inversion

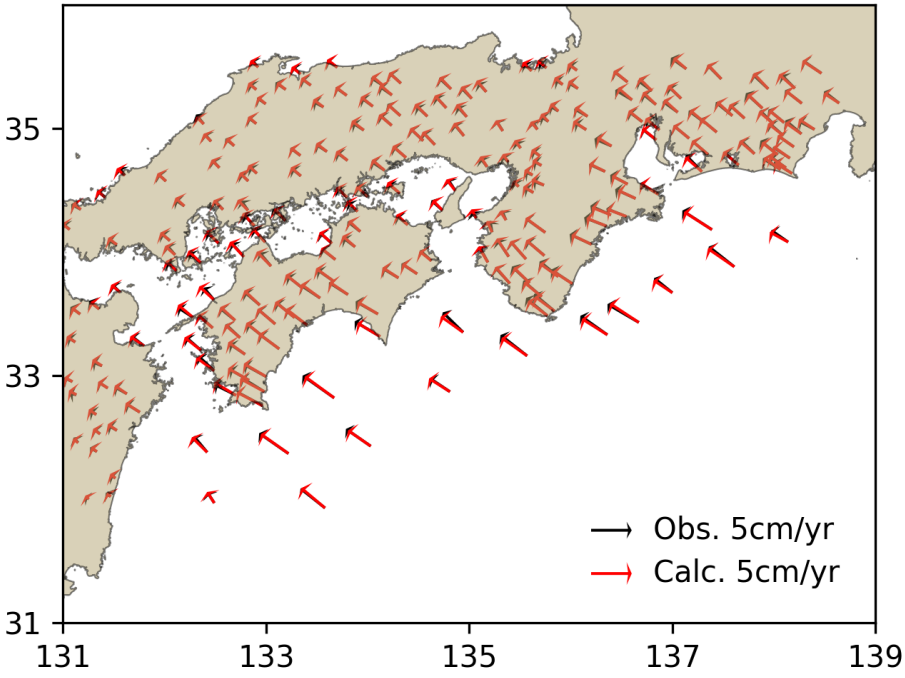


Fig. 19: Observed surface deformation rate at observation stations (black arrow) and calculated deformation rate (red arrow) at 80 years after earthquake.

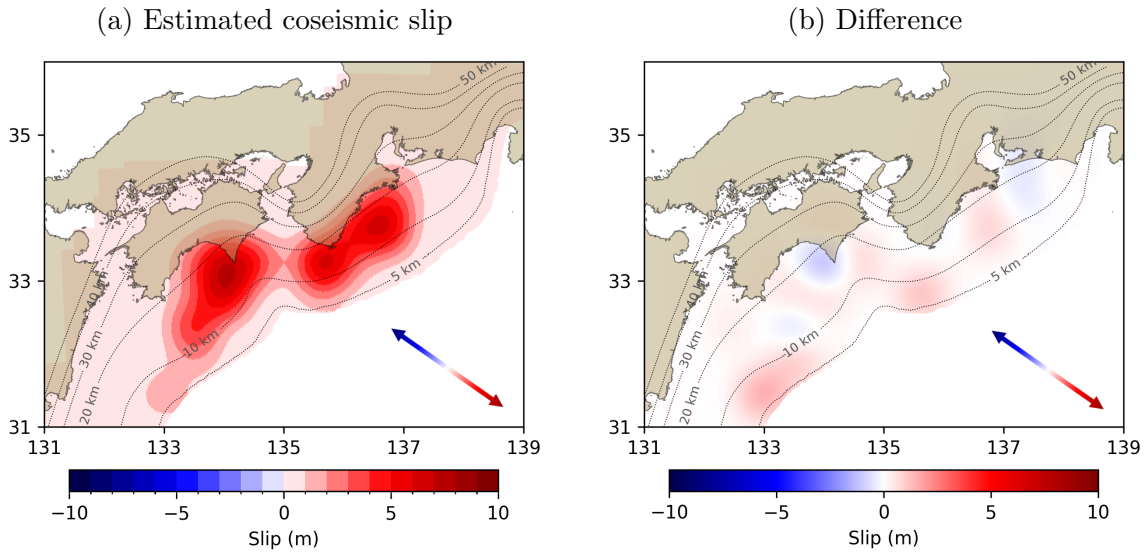


Fig. 20: Coseismic slip distribution estimated from long-term viscoelastic inversion considering synthetic data during 2006-2023 years at observation location.

6 Conclusions

In this study, we developed an Aggregate-based Correction (ABC) to improve the initial guess in solving the viscoelastic crustal deformation problem and an Adaptive Time-Stepping (ATS) method suitable for viscoelastic analysis. We constructed a large-scale 3D crustal model of 4.2×10^9 DOF to evaluate the proposed methods for the Nankai Trough subduction zone. The implementation of the ABC yielded a remarkable 1.47-fold reduction in the number of solver iterations compared to the state-of-the-art solver. With a small increase in memory footprint, the ATS method led to a 6.02-fold speedup from the highly tuned GPU-based multigrid solver with Data-Driven Predictor. The speedup obtained in this study is expected to improve earthquake source models on 3D crustal structures.

By enabling the computation of long-term viscoelastic deformation in practical time, we calculated 80 viscoelastic Green's functions for the 3D crustal model using 56 A100 GPUs. As an application example, we performed inversion analysis to estimate inter-seismic slip-deficit and historical earthquake slip distribution using a synthetic observation dataset. Inversion considering long-term viscoelastic response yielded more accurate predictions for inter-seismic slip-deficit. In our future work, we plan to explore integrating additional data sources, such as leveling survey data and tidal gauge data, to enhance the estimation of coseismic slip and slip-deficit. Furthermore, we intend to incorporate unknown parameters that account for crustal properties and additional slip distributions.

References

Central Disaster Management Council, . http://www.bousai.go.jp/jishin/nankai/index.html (in Japanese; accessed 4 July 2023).

Fujita, K., Ichimura, T., Koyama, K., Inoue, H., Hori, M., Maddegedara, L., 2017. Fast and scalable low-order implicit unstructured finite-element solver for earth's crust deformation problem, in: Proceedings of the Platform for Advanced Scientific Computing Conference, Association for Computing Machinery, New York, NY, USA. pp. 1–10. URL: https://doi.org/10.1145/3093172.3093236, doi:10.1145/3093172.3093236.

Fujita, K., Murakami, S., Ichimura, T., Hori, T., Hori, M., Lalith, M., Ueda, N., 2022. Scalable finite-element viscoelastic crustal deformation analysis accelerated with data-driven method, in: 2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH), pp. 18–25. doi:10.1109/ScalAH56622.2022.00008.

Fukahata, Y., Nishitani, A., Matsu'ura, M., 2004. Geodetic data inversion using ABIC to estimate slip history during one earthquake cycle with viscoelastic slip-response functions. Geophysical Journal International 156, 140–153. URL: https://doi.org/10.1111/j.1365-246X.2004.02122.x, doi:10.1111/j.1365-246X.2004.02122.x.

Fukushima, Y., Nishikawa, T., Kano, Y., 2023. High probability of successive occurrence of nankai megathrust earthquakes. Scientific reports 13, 63.

HDF5, 2017. HDF5 - Hierarchical Data Format 5. https://www.hdfgroup.org/solutions/hdf5.

Hori, T., Agata, R., Ichimura, T., Fujita, K., Yamaguchi, T., Iinuma, T., 2021. High-fidelity elastic Green's functions for subduction zone models consistent with the global standard geodetic reference system. Earth, Planets and Space 73, 41. URL: https://doi.org/10.1186/s40623-021-01370-y, doi:10.1186/s40623-021-01370-y.

Ichimura, T., Agata, R., Hori, T., Hirahara, K., Hashimoto, C., Hori, M., Fukahata, Y., 2016. An elastic/viscoelastic finite element analysis method for crustal deformation using a 3-D island-scale high-fidelity model. Geophysical Journal International 206, 114–129. URL: https://doi.org/10.1093/gji/ggw123, doi:10.1093/gji/ggw123.

Ichimura, T., Fujita, K., Tanaka, S., Hori, M., Lalith, M., Shizawa, Y., Kobayashi, H., 2014. Physics-based urban earthquake simulation enhanced by 10.7 blndof × 30 k time-step unstructured fe non-linear seismic wave simulation, in: SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 15–26. doi:10.1109/SC.2014.7.

Japan Cabinet Office, 2015. Strong motion fault model and tsunami fault model of past earthquakes along the nankai trough. https://www.bousai.go.jp/jishin/nankai/pdf/jishinnankai20151217_03.pdf (in Japanese; accessed 4 July 2023).

Johnson, K.M., Tebo, D., 2018. Capturing 50 years of postseismic mantle flow at nankai subduction zone. Journal of Geophysical Research: Solid Earth 123, 10,091–10,106. URL: https://agupubs.onlinelibrary.wiley.com/doi/

abs/10.1029/2018JB016345, doi:https://doi.org/10.1029/2018JB016345, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018JB016345.

Karypis, G., Kumar, V., 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing 20, 359–392. URL: https://doi.org/10.1137/S1064827595287997, doi:10.1137/S1064827595287997.

Koketsu, K., Miyake, H., Afnimar, Tanaka, Y., 2009. A proposal for a standard procedure of modeling 3-d velocity structures and its application to the tokyo metropolitan area, japan. Tectonophysics 472, 290–300. URL: https://www.sciencedirect.com/science/article/pii/S0040195108002539, doi:https://doi.org/10.1016/j.tecto.2008.05.037. deep seismic profiling of the continents and their margins.

Koketsu, K., Miyake, H., Suzuki, H., 2012. Japan integrated velocity structure model version 1, in: Proceedings of the 15th World Conference on Earthquake Engineering, Lisbon. pp. 1–4.

Li, S., Moreno, M., Bedford, J., Rosenau, M., Oncken, O., 2015. Revisiting viscoelastic effects on interseismic deformation and locking degree: A case study of the peru-north chile subduction zone. Journal of Geophysical Research: Solid Earth 120, 4522–4538. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015JB011903, doi:https://doi.org/10.1002/2015JB011903, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2015JB011903.

Melosh, H.J., Raefsky, A., 1981. A simple and efficient method for introducing faults into finite element computations. Bulletin of the Seismological Society of America 71, 1391–1400. URL: https://doi.org/10.1785/BSSA0710051391, doi:10.1785/BSSA0710051391.

Murakami, S., Fujita, K., Ichimura, T., Hori, T., Hori, M., Lalith, M., Ueda, N., 2023. Development of 3d viscoelastic crustal deformation analysis solver with data-driven method on gpu, in: Computational Science – ICCS 2023, Springer Nature Switzerland, Cham. pp. 423–437.

Murakami, S., Ichimura, T., Fujita, K., Hori, T., Ohta, Y., 2022. Impact of ambiguity of physical properties of three-dimensional crustal structure model on coseismic slip and interseismic slip deficit in the nankai trough region. GeoHazards 3, 162–177. URL: https://www.mdpi.com/2624-795X/3/2/9, doi:10.3390/geohazards3020009.

Noda, A., Saito, T., Fukuyama, E., 2018. Slip-deficit rate distribution along the nankai trough, southwest japan, with elastic lithosphere and viscoelastic asthenosphere. Journal of Geophysical Research: Solid Earth 123, 8125–8142. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JB015515, doi:https://doi.org/10.1029/2018JB015515.

Parker, J., Lyzenga, G., Norton, C., Zuffada, C., Glasscoe, M., Lou, J., Donnellan, A., 2008. Geophysical finite-element simulation tool (geofest): Algorithms and validation for quasistatic regional faulted crust problems. Pure and Applied Geophysics 165, 497–521. URL: https://doi.org/10.1007/s00024-008-0325-9, doi:10.1007/s00024-008-0325-9.

Sherrill, E.M., Johnson, K.M., 2021. New insights into the slip budget at nankai: An iterative approach to estimate coseismic slip and afterslip. Journal of Geophysical Research: Solid Earth 126, 2020JB020833. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020JB020833, doi:https://doi.org/10.1029/2020JB020833. 2020JB020833 2020JB020833.

Tomita, F., Iinuma, T., Ohta, Y., Hino, R., Kido, M., Uchida, N., 2020. Improvement on spatial resolution of a coseismic slip distribution using postseismic geodetic data through a viscoelastic inversion. Earth, Planets and Space 72, 84. URL: https://doi.org/10.1186/s40623-020-01207-0, doi:10.1186/s40623-020-01207-0.

Vanek, P., Mandel, J., Brezina, M., 1996. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Computing 56, 179–196.

Yabuki, T., Matsu'ura, M., 1992. Geodetic data inversion using a bayesian information criterion for spatial distribution of fault slip. Geophysical Journal International 109, 363–375. URL: https://doi.org/10.1111/j.1365-246X.1992.tb00102.x, doi:10.1111/j.1365-246X.1992.tb00102.x.

Yamaguchi, T., Ichimura, T., Yagi, Y., Agata, R., Hori, T., Hori, M., 2017. Fast crustal deformation computing method for multiple computations accelerated by a graphics processing unit cluster. Geophysical Journal International 210, 787–800. URL: https://doi.org/10.1093/gji/ggx203, doi:10.1093/gji/ggx203, arXiv:https://academic.oup.com/gji/article-pdf/210/2/787/17724974/ggx203.pdf.

Yokota, Y., Ishikawa, T., Watanabe, S.i., Tashiro, T., Asada, A., 2016. Seafloor geodetic constraints on interplate coupling of the nankai trough megathrust zone. Nature 534, 374–377.