Large-scale nonlinear viscoelastic simulation for crustal deformation accelerated by data-driven method and multi-grid solver

Sota Murakami[1], Kohei Fujita[1,2], Tsuyoshi Ichimura[1,2,4], Ryota Kaneko[1], Takane Hori[3], Muneo Hori[3], Maddegedara Lalith[1], and Naonori Ueda[4]

[1] Earthquake Research Institute and Department of Civil Engineering, The University of Tokyo, Tokyo, Japan
{souta,fujita,ichimura,ryota-24,lalith}@eri.u-tokyo.ac.jp
[2] Center for Computational Science, RIKEN, Kobe, Japan
[3] Japan Agency for Marine-Earth Science and Technology, Yokohama, Japan
{horit,horimune}@jamstec.go.jp
[4] Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan
naonori.ueda@riken.jp

Abstract. We developed a fast method for spatially highly detailed nonlinear viscoelastic crustal deformation analysis by a combination of a data-driven method and a multi-grid solver. Here, highly accurate estimations of the solution of the next time step are obtained using a data-driven predictor based on the history of past solutions, which reduces the number of iterative solver iterations and thus reduces the computation cost. Although this method has been shown to enable fast linear viscoelastic analysis, its validity has not been confirmed for nonlinear viscoelastic problems. Numerical experiments have shown that the data-driven method reduces the number of iterations by 3.35-fold. To achieve further acceleration, we introduced a multi-grid solver capable of efficiently solving large systems of equations. The proposed combination of the data-driven method and multi-grid solver is applied to nonlinear viscoelastic crustal deformation analysis of the Nankai Trough region, and it is shown that the proposed method achieved a 15.1-fold speedup, which enabled many large-scale crustal deformation simulations within reasonable computational costs. The fast nonlinear viscoelastic analysis of spatially highly detailed crustal structure models enabled by this study is expected to contribute to the advance of interplate state estimation.

Keywords: unstructured finite-element method · data-driven predictor · GPU · OpenACC · power-law rheology.

1 Introduction

Accurate estimation of interplate conditions such as plate locking and sliding is considered to play an important role in improving the understanding of large-scale earthquakes. Such an estimation is expected to be realized through viscoelastic response analysis that can analyze the response of crust which is believed to behave viscoelastically, based on two types of recently well-accumalated

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

data: (1) observation data directly above the seismogenic zone through the installation of seafloor observation networks and (2) the crustal structure data from subsurface structural explorations. Such viscoelastic response analysis must accurately reflect realistic rheological models and crustal structure data, and must be conducted many times. As one of the realistic rheological models for a part of the mantle, a nonlinear rheological model that assumes that the stress-strain relationship follows a power law has been proposed [5] and has been shown to successfully explain crustal deformation for past earthquakes (e.g., [3, 13]). Also, to accurately calculate the response in a detailed 3D crustal structure model, a region of 10^{7-8}m $\times$ 10^{7-8}m $\times$ 10^{7-8}m must be computed on a fine mesh with element sizes as small as 10^{3-4} m [7]. As a result, to achieve interplate state estimation, nonlinear viscoelastic analyses must be performed about 10^{3-6} times on a highly detailed model with $10^9 - 10^{10}$ degrees of freedom. This requires huge computational cost and thus a reduction in computational cost is required.

The governing equation for nonlinear viscoelastic crustal deformation becomes an elliptic partial differential equation which requires solving a large system of linear equations at each time step, requiring huge computational cost. Iterative solvers are often used to solve such large system of linear equations. In this case, the number of iterations required for the convergence of the solution is expected to decrease if a highly accurate initial solution is given; thus, data-driven prediction methods, which aim to further reduce the number of iterations from conventional methods by more aggressively predicting the solution for the next time step based on the history of past solutions, are being developed [10]. In addition, multi-grid solvers have been developed as efficient methods for solving large system of linear equations in crustal deformation problems such as linear viscoelastic analysis on high-performance computers [9], and a combination with the predictor method has been shown to enable fast linear viscoelastic analysis [6, 15]. On the other hand, although its validity has not been confirmed for the nonlinear viscoelastic problems targeted in this study, the deformation due to nonlinear viscoelasticity is quasi-static, similar to linear viscoelastic variation, and the above prediction based on changes in the spatial solution structure provides an initial solution accurate enough to speed up the solver. Also, the above solution prediction is performed by dividing the analysis target into small regions; however, the effect of this region division on the initial solution estimation performance for this type of viscoelastic problem has not yet been confirmed.

Based on the above, this study aims to reduce the computational cost of nonlinear viscoelastic analysis on highly detailed crustal structure models by developing a new numerical solution method with a data-driven method for highly accurate estimation of the initial solution and a multi-grid solver for reduction of solver iterations. The rest of this paper is organized as follows. Section 2 describes the nonlinear viscoelastic analysis method and its validation, the data-driven predictor method, and the multi-grid solver used in this study. In Section 3, the effectiveness of the developed method is demonstrated by comparison with conventional methods. Here, a detailed comparison that has not been done before, such as considering the effects of domain decomposition, is conducted. Section 4

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

describes an application example of a nonlinear viscoelastic crustal deformation analysis in the Nankai Trough earthquake region. Section 5 provides a summary of this study.

2 Method

2.1 Nonlinear Viscoelastic Problem

In this study, the crust is assumed to follow a power-law Maxwell rheology as follows

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

$$\dot{\sigma}_{ij} = \lambda \dot{\epsilon}_{kk} \delta_{ij} + 2\mu \dot{\epsilon}_{ij} - \frac{\mu}{\eta} |s|^{n-1} s_{ij} \tag{2}$$

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{3}$$

Here, u, f, σ, ϵ, λ, and μ indicate displacement, outer force, stress, strain, and Lame's constants, respectively. Einstein's summation convention is used, and δ, $\dot{(\cdot)}$, and $|\ |$ indicate Kronecker delta, time differentiation, and the Frobenius norm, respectively. s, n, and η indicate the deviatoric stress, stress exponent, and coefficient for viscous relaxation, respectively. Particularly, when $n = 1$ the material becomes linear Maxwell viscous with viscous coefficient η, and when $\eta = \infty$ the material becomes linear elastic.

In this study, the finite element method with second-order tetrahedral elements is used to model heterogeneous crustal structures with traction-free boundary conditions. We use the same algorithm used in the Geophysical Finite Element Simulation Tool [16], a finite element software package for geophysical and other continuum domain applications, for analysis (see Algorithm 1). The external forces equivalent to fault slip during an earthquake are provided as input to the analysis using the split-node technique [14]. Gravity is taken into account by applying force on the ground surface according to the displacement in the gravity direction [4]. In this study, we adopt control parameter $\alpha = 0$ in Algorithm 1, which results in solving a nonlinear viscoelastic problem using an explicit method. In this case, the calculation of crustal deformation in each time step is approximated as a linear elastic problem. The finite-element model is generated by the method in [9] which automatically and robustly generates a tetrahedral mesh of stratified ground models from Digital Elevation Map (DEM) data.

Numerical validation of the developed nonlinear viscoelastic analysis program is conducted as follows. We consider a horizontally stratified two-layer model where spatially uniform shear stress $\tau(t)$ is applied to the layer interface. One layer is a nonlinear viscoelastic layer and the other is an elastic layer, each with a thickness of L, with Dirichlet boundary conditions on the top and bottom surfaces and periodic conditions on the sides. In this setting, the displacement at the layer boundary becomes equivalent to the Maxwellian standard linear

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

Algorithm 1 Algorithm for solving nonlinear viscoelastic response of crust. Here, superscript $(\ )^i$ is the variable in the i-th time step. Δt is the time increment. $\mathbf{B}$ is the displacement-strain transformation matrix. $\mathbf{D}$ is the matrices indicating elastic material property. $\boldsymbol{\beta}$ represents the Maxwell viscoplastic strain rate, and $\boldsymbol{\beta'}$ its Jacobian matrix. α is the controlling parameter. Ω is the viscoelastic body.

$i \Leftarrow 0$
Compute $\mathbf{f}^0$ by split-node technique
$\mathbf{K} \Leftarrow \sum_k \int_{\Omega_e^k} \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{d\Omega_e}$
solve $\mathbf{K}\mathbf{u}^0 = \mathbf{f}^0$
$\sigma^0 \Leftarrow \mathbf{D}\mathbf{B}\mathbf{u}^0$
while $i < N_t$ **do**
  $\mathbf{K}^i \Leftarrow \sum_k \int_{\Omega_e^k} \mathbf{B}^T \left(\mathbf{D}^{-1} + \alpha \Delta t \beta'\right)^{-1} \mathbf{B} \mathbf{d\Omega_e}$
  $\mathbf{f}^i \Leftarrow \sum_e \int_{\Omega_e} \mathbf{B}^T \left\{ \left(\mathbf{D}^{-1} + \alpha \Delta t \beta'\right)^{-1} \Delta t \beta^i - \sigma^i \right\} d\Omega_e + \mathbf{f}^0 + \mathbf{g}^i$
  if $i \geq 20$ **then**
    guess δu by data-driven predictor
  else if $i \geq 3$ **then**
    guess δu by 2nd order Adams-Bashforth method
  end if
  solve $\mathbf{K}^i \delta \mathbf{u}^i = \mathbf{f}^i$ with initial solution $\delta \hat{u}$
  $\mathbf{u}^i \Leftarrow \mathbf{u}^{i-1} + \delta \mathbf{u}^i$
  $\sigma^i \Leftarrow \sigma^{i-1} + \left(\mathbf{D}^{-1} + \alpha \Delta t \beta'\right)^{-1} \left(\mathbf{B} \delta \mathbf{u^i} - \mathbf{\Delta t} \beta^{\mathbf{i}}\right)$
  $i \Leftarrow i + 1$
end while

solid model with Young's modulus of the spring μ and a dashpot following the power-law. In particular, for $n > 1$ and given a shear stress $\tau(t) = \tau^0 H(t)$, the displacement $u(t)$ at the layer boundary can be expressed as

$$u(t) = \frac{L}{\mu} \times \left[\tau^0 - \left\{ \left(\frac{\tau^0}{2} \right)^{-(n-1)} + \frac{(n-1)\mu t}{2\eta} \right\}^{-\frac{1}{n-1}} \right], \tag{4}$$

where $H(t)$ is the Heaviside step function. Since nonlinear viscoelasticity can be regarded as linear viscoelasticity with a certain local viscosity at a local time and place, we define this effective viscosity as

$$\eta_{eff} = \frac{\eta}{2|s|^{n-1}}. \tag{5}$$

The parameters used in the numerical validation are shown in Fig. 1a, and Fig. 1b,c shows the time history of displacement at the layer boundary and the effective viscosity in the nonlinear viscoelastic layer. The solution follows the analytical solution expressed in Eq. (4) and thus we can see that a correct solution is obtained.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

a) problem setting

Variable	Value
η	30GPa
μ	3×10^{32} Pa^{3}s
n	3
τ^0	1.0×10^{15} Pa
L	60,000m
dt	86,400 s
ds	1,000 m

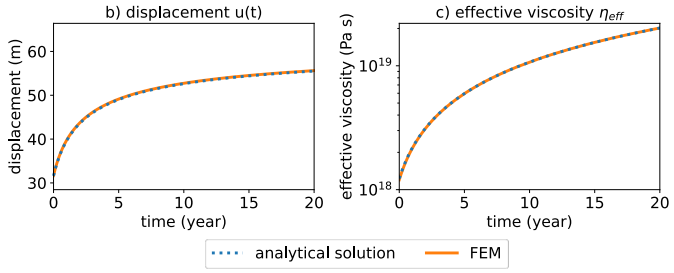

Fig. 1. Problem settings and time history response for the numerical verification problem. a) Problem settings, b) displacement response, c) effective viscosity.

2.2 Data-driven Predictor

In nonlinear viscoelastic analysis, it is necessary to efficiently solve a large system of linear equations obtained using the finite element method (line 14 of Algorithm 1). When using an iterative method for solving the system of linear equations, the number of iterations required for the convergence of the solution is expected to decrease if an initial solution close to the target solution can be predicted.

One method for predicting the initial solution is the Adams-Bashforth method. In the second-order Adams-Bashforth method, the solution is predicted pointwise as

$$\delta\hat{\mathbf{u}}^i_{adam} \Leftarrow \mathbf{u}^{i-3} - 3\mathbf{u}^{i-2} + 2\mathbf{u}^{i-1}. \tag{6}$$

In general, pointwise prediction methods such as the Adams-Bashforth method are limited to prediction within lower-order polynomial approximation, such as in Eq. (6), because the prediction results become unstable when using higher-order polynomial approximation.

On the other hand, the idea of dynamic mode decomposition (DMD) has been introduced to utilize historical data of a larger number of modes to predict the initial solution with higher accuracy [10]. Here, based on the idea of exact DMD, we consider m pairs of input data and output data $(\mathbf{x}^j, \mathbf{y}^j)$ with data length l, and define two $l \times m$ matrices (in the primitive approach, l corresponds to the total degrees of freedom in space and m to the time step) as follows

$$\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^m), \tag{7}$$

$$\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^m). \tag{8}$$

The goal in exact DMD is to extract the spatiotemporal behavior of the dataset of interest, for example by computing the eigenvectors and eigenvalues of operator $\mathbf{C}$ defined as

$$\mathbf{C} = \mathbf{Y}\mathbf{X}^\dagger. \tag{9}$$

Based on the datasets $\mathbf{X}, \mathbf{Y}$ and the input data $\mathbf{x}$, the output data $\mathbf{y}$ used for the prediction of the initial solution can be obtained as

$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{10}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

As a linear operator $\mathbf{C}$ is used to predict the output data $\mathbf{y}$, there is no guarantee that the prediction for nonlinear problems are effective. However, since deformation due to nonlinear viscoelasticity is a weakly nonlinear phenomenon in which the effective viscosity changes slowly with time, it behaves within a quasi-static range with a similar response to a linear viscoelastic problem (see Fig. 1,b). Furthermore, by constructing local prediction operators for the solution of the next step from recent historical data, and by using local prediction to capture nonlinear patterns of spatial variation, we can expect to predict initial solutions accurate enough to speed up the solver even when using linear operators.

In this study, when predicting the solution of the i-th time step, we define $\Delta\mathbf{u}^i = \mathbf{u}^i_{adam} - \delta\mathbf{u}^i$ as the correction between the predicted and actual values in the Adams-Bashforth method, and we use the input and output data pairs $(\mathbf{x}^j, \mathbf{y}^j) = (\Delta\mathbf{u}^{i-j-1}, \Delta\mathbf{u}^{i-j})$ to predict the solution. In this case, the operator $\mathbf{C}$ is a one-step-ahead prediction operator for the correction value, and by using the value $\Delta\mathbf{u^{i-1}}$ from the previous step as the input data, the prediction value $\Delta\hat{\mathbf{u}}^{\mathbf{i}}$ for the i-th correction value is obtained. By predicting the dominant deformation mode using the Adams-Bashforth method and applying a data-driven prediction method to the correction values, more deformation modes can be considered. Furthermore, the relaxation of the ill-condition of $\mathbf{X}$ allows for a highly accurate estimation of $\mathbf{X}^\dagger$ regardless of the computational method.

To improve tractability to spatially local modes, the entire domain is divided into subregions, and a data-driven predictor is applied to each of these subregions to predict the solution. On the other hand, if the size of the subregion is too small, the global deformation may not be adequately captured. Therefore, it is possible that the suitable subdomain size may vary according to the characteristics of the target problem to achieve high estimation performance, and verification is necessary.

In this study, the output data $\mathbf{y}$ is obtained by first computing $\mathbf{a} = \mathbf{X}^\dagger\mathbf{x}$ and then computing $\mathbf{y} = \mathbf{X}\mathbf{a}$, without explicitly computing the operator $\mathbf{C}$. For some datasets, the ill-condition of $\mathbf{X}$ may make the computation of $\mathbf{X}^\dagger\mathbf{x}$ unstable and degrade the accuracy of the estimated value $\mathbf{y}$. However, considering the computational cost and the effectiveness in linear viscoelasticity problems, we use QR decomposition to compute $\mathbf{X}^\dagger\mathbf{x}$ to obtain $\mathbf{y}$ (here, the ill-condition is also devised by changing the data to be learned as shown above, i.e., learning the correction from the prediction from the Adam-Bashforth method). Specifically, first, based on Gram-Schmidt's orthogonalization method, we decompose $\mathbf{X} = \mathbf{QR}$ and then solve the system of linear equations $\mathbf{Ra} = \mathbf{Q}^T\mathbf{x}$ to obtain $\mathbf{a}$.

As the computational cost and memory footprint of QR decomposition increases with the data length l, we reduce the dimensionality of the matrix. Specifically, using a $l' \times l$ matrix $\mathbf{A}$, we transform the dataset $\mathbf{X}$ and input data $\mathbf{x}$ as $\mathbf{X}' \Leftarrow \mathbf{AX}$ and $\mathbf{x}' \Leftarrow \mathbf{Ax}$. While we can use $\mathbf{A} = \mathbf{X}^T$ for the transformation matrix, the computational accuracy deteriorates as the condition number of $\mathbf{X}^T\mathbf{X}$ becomes the square of that of $\mathbf{X}$. Therefore, we use an $l' \times l$ matrix with random values (here, $l' \ll l$) for the transformation matrix $\mathbf{A}$. In the application example, a matrix $\mathbf{X}$ of 8000×16 is transformed into a smaller matrix $\mathbf{X}'$ of

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

96×16 for each of the 5×10^5 subregions. Since the history of the input data $\mathbf{x}'$ corresponds to $\mathbf{X}'$, only the transformation of $\mathbf{x}' \Leftarrow \mathbf{A}\mathbf{x}$ is required at each time step. Furthermore, this can be computed efficiently as a matrix-matrix product by computing multiple subregions together (i.e., for the data $\mathbf{x}_k$ in each subregion k, data can be transformed as $[\mathbf{x}'_1, \cdots, \mathbf{x}'_b] \Leftarrow \mathbf{A}[\mathbf{x}_1, \cdots, \mathbf{x}_b]$ using the same transformation matrix $\mathbf{A}$). This results in low cost and relatively low memory footprint in the data-driven predictor when compared to that of the solver.

2.3 Multi-grid solver

An iterative solver is used to effectively utilize the highly accurate initial solution predicted by the data-driven predictor in the previous section. In an iterative solver, the use of highly accurate initial solutions is expected to reduce the number of iterations required for the convergence of the solution. Although this study is concerned with nonlinear viscoelastic crustal deformation, we solve a linear set of equations with the stiffness matrix identical to that of the linear viscoelastic problem. Since the multi-grid solver has been shown effective for linear viscoelastic crustal deformation problems, it is expected to be effective for nonlinear viscoelastic problems as well. The multi-grid solver has also been shown to perform well on massively parallel CPU computer systems and multi-GPU computers systems due to its high parallelism and scalability.

The multi-grid solver (Algorithm 2) is an adaptive conjugate gradient method that comprises a multi-grid model as a preconditioner. Here we use a two-level geometric multi-grid method where a second-order tetrahedral element model is used as a fine grid and a first-order tetrahedral element model is used as the coarse grid (here, the first-order tetrahedral element model is generated by extracting the vertex nodes of the second-order tetrahedral element model). We refer to the original conjugate gradient iteration as the outer loop and the conjugate gradient iteration for the preconditioner as the inner loop. In each iteration of the outer loop, the solution is first estimated using the 3×3 block Jacobi method. This block Jacobi solution is used as an initial solution to the inner loop with the first-order tetrahedral element model (Algorithm 2 a line 9; inner coarse loop). After refinement by the inner coarse loop, the solution is used as an initial solution to the inner loop with the second-order tetrahedral element model (Algorithm 2 a line 11; inner fine loop). In these inner loops, the conjugate gradient method with a block Jacobi preconditioner is used (Algorithm 2 b). By introducing a coarse model, the computational cost per iteration is reduced and the solver convergence rate is improved, resulting in the efficient computation of the preconditioner.

In the outer loop, FP64 computation is used to guarantee numerical accuracy, while FP32 computation is used in the inner loops. This reduces computational costs through reduction in memory access and MPI communication, utilization of fast FP32 arithmetic hardware, and improved cache utilization. The Element-by-Element (EBE) method [17] is used in the matrix-vector product. In the EBE method, the matrix-vector product is computed element-wise; i.e., the local matrix vector product is computed element wise and the local results are added

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

Algorithm 2 The multi-grid solver to solve $\mathbf{Kx} = \mathbf{b}$. The input variables are $\mathbf{x}, \mathbf{K}, \mathbf{b}, \mathbf{M}, \mathbf{P}, \epsilon_c^{in}, \epsilon^{in}, N^{max}, N_c^{max}$. Here, $\mathbf{K}$ and $\mathbf{P}$ represent global stiffness matrix and the mapping matrices between grids. $\mathbf{M}$ is 3×3 block Jacobi matrix of K. The other variables are temporal. Here, $(\bar{})$ represents FP32 variables, while the others are FP64 variables.

(a) outer loop

$\mathbf{r} \Leftarrow \mathbf{Kx}$ (computed using EBE)
$\mathbf{r} \Leftarrow \mathbf{b} - \mathbf{r}$
$\beta \Leftarrow 0$
$i \Leftarrow 1$
while $|\mathbf{r}|/|\mathbf{b}| > \epsilon$ **do**
  $\bar{\mathbf{x}} \Leftarrow \bar{\mathbf{M}}^{-1}\bar{\mathbf{r}}$
  $\bar{\mathbf{r}}_c \Leftarrow \bar{\mathbf{P}}^T\bar{\mathbf{r}}$
  $\bar{\mathbf{x}}_c \Leftarrow \bar{\mathbf{P}}^T\bar{\mathbf{x}}$
  solve $\bar{\mathbf{K}}_c\bar{\mathbf{x}}_c = \bar{\mathbf{r}}_c$ using (b) with ϵ_c^{in} and N_c^{max} (*inner coarse loop*)
  $\bar{\mathbf{x}} \Leftarrow \bar{\mathbf{P}}\bar{\mathbf{x}}_c$
  solve $\bar{\mathbf{K}}\bar{\mathbf{x}} = \bar{\mathbf{r}}$ using (b) with ϵ^{in} and N^{max} (*inner fine loop*)
  $\mathbf{z} \Leftarrow \bar{\mathbf{x}}$
  if i>1 **then**
    $\gamma \Leftarrow (\mathbf{z}, \mathbf{q})$
    $\beta \Leftarrow \gamma/\rho$
  end if
  $\mathbf{p} \Leftarrow \mathbf{z} + \beta\mathbf{p}$
  $\mathbf{q} \Leftarrow \mathbf{Kp}$ (computed using EBE)
  $\rho \Leftarrow (\mathbf{z}, \mathbf{r})$
  $\gamma \Leftarrow (\mathbf{p}, \mathbf{q})$
  $\alpha \Leftarrow \rho/\gamma$
  $\mathbf{r} \Leftarrow \mathbf{r} - \alpha\mathbf{q}$
  $\mathbf{x} \Leftarrow \mathbf{x} + \alpha\mathbf{p}$
end while

(b) inner loop

$\bar{\mathbf{r}} \Leftarrow \bar{\mathbf{K}}\bar{\mathbf{x}}$ (computed using EBE)
$\bar{\mathbf{r}} \Leftarrow \bar{\mathbf{b}} - \bar{\mathbf{r}}$
$\bar{\beta} \Leftarrow 0$
$i \Leftarrow 1$
while $|\bar{\mathbf{r}}|/|\bar{\mathbf{b}}| > \epsilon^{in}$ & $i \leq N^{max}$ **do**
  $\bar{\mathbf{z}} \Leftarrow \bar{\mathbf{M}}^{-1}\bar{\mathbf{r}}$
  $\bar{\rho}_a \Leftarrow (\bar{\mathbf{z}}, \bar{\mathbf{r}})$
  if $i > 1$ **then**
    $\bar{\gamma} \Leftarrow \bar{\rho}_a/\bar{\rho}_b$
    $\bar{\beta} \Leftarrow \bar{\gamma}/\bar{\rho}$
  end if
  $\bar{\mathbf{p}} \Leftarrow \bar{\mathbf{z}} + \bar{\beta}\bar{\mathbf{p}}$
  $\bar{\mathbf{q}} \Leftarrow \bar{\mathbf{K}}\bar{\mathbf{p}}$ (computed using EBE)
  $\bar{\gamma} \Leftarrow (\bar{\mathbf{p}}, \bar{\mathbf{q}})$
  $\bar{\alpha} \Leftarrow \bar{\rho}_a/\bar{\gamma}$
  $\bar{\rho}_b \Leftarrow \bar{\rho}_a$
  $\bar{\mathbf{r}} \Leftarrow \bar{\mathbf{r}} - \bar{\alpha}\bar{\mathbf{q}}$
  $\bar{\mathbf{x}} \Leftarrow \bar{\mathbf{x}} + \bar{\alpha}\bar{\mathbf{p}}$
end while

to obtain the global product. By using EBE, the memory footprint and memory transfers can be reduced when compared with methods storing the global stiffness matrix on memory and reading it during matrix-vector product computations.

3 Numerical experiment

While the data-driven predictor and multi-grid solver described in Section 2 has been demonstrated effective for linear viscoelastic crustal deformation problems, it has not been applied to nonlinear viscoelastic crustal deformation problems. Since their performance depends on the characteristics of the problem, we verify their effectiveness for nonlinear viscoelastic problems through a numerical experiment on a model following an actual crustal deformation problem.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

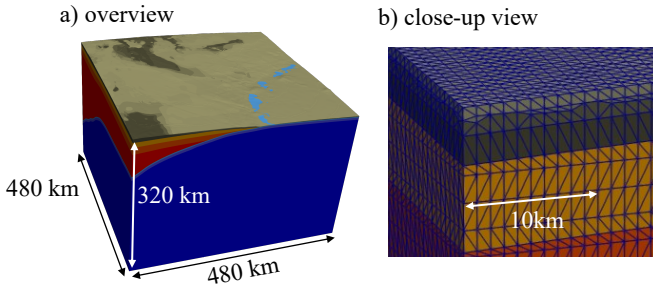

Fig. 2. Finite element model used for the numerical experiment problem.

3.1 Problem settings

We generated a finite-element model with element size $ds = 1000$ m for a region of 480 km $\times$ 480 km $\times$ 320 km for the numerical experiment (see Fig. 2). The degrees of freedom and number of tetrahedral elements are 1.95×10^8 and 4.83×10^7, respectively. The parameters for the nonlinear viscoelastic layer are set to $n = 3$, $\eta = 3\times10^{32}$ Pa^3 s. The sides and bottom of the finite element model are fixed to zero by Dirichlet boundary conditions. $N_t = 50$ time steps were computed with time step increment width of $dt = 86,400$ s. We evaluate the effectiveness of the data-driven predictor by comparing the required number of solver iterations with those obtained by using the second-order Adams-Bashforth method as the initial solution predictor. Two types of iterative solvers were employed: the standard 3×3 block Jacobi preconditioned conjugate gradient solver (PCGE) and the multi-grid solver. The PCGE solver corresponds to skipping lines 6 to 11 in Algorithm 2. All equations are solved up to relative error $\epsilon = 10^{-8}$. The inner loop threshold and maximum number of iterations in the multi-grid solver are set to $(\epsilon_c^{in}, \epsilon^{in}) = (0.05, 0.1)$ and $(N_c^{max}, N^{max}) = (300, 20)$, respectively. The entire domain is divided into 24,576 subdomains (approximately 7930 degrees-of-freedom per subdomain), and data of the past $m = 16$ time steps were used to predict the solution in each subdomain. Here, we used METIS [1] for partitioning the finite-element model into subregions. We evaluate the average number of solver iterations between time steps 21–50 for evaluation of the predictor performance.

The proposed method is designed to work on both CPU and GPU. For measurement, we use a computer system comprising 48 NVIDIA A100 40 GB PCIe GPUs. For GPU computation, we use a programming model called OpenACC [2], which has high portability but can achieve similar performance as an implementation with a low-level programming model such as CUDA. Almost all of the computation in Algorithms 1 and 2, except for pre-post processing, MPI communication, and conditional branch, is accelerated by GPU computation, which also minimizes CPU-GPU data transfer. As the computational performance is similar to that of linear viscoelastic problems, we refer to [15] for the computational performance of the developed method.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

Table 1. Number of solver iterations per time step for the numerical experiment problem.

case	solver	predictor	number of iterations		
			outer	inner fine	inner coarse
a	PCGE solver	Adams-Bashforth	3340	-	-
b	PCGE solver	data-driven predictor	659	-	-
c	Multi-grid solver	Adams-Bashforth	6.4	106	1763
d	Multi-grid solver	data-driven predictor	3.6	41	495

3.2 Performance results

We verify the effectiveness of the proposed method by confirming that the number of solver iterations has reduced. When using the data-driven predictor instead of the Adams-Bashforth method for the initial solution for PCGE solver, the number of iterations has decreased significantly from 3340 to 659 iterations (see cases a and b in Table 1). This is due to the reduction of the residual of the initial solution from 1.11×10^{-3} to 3.39×10^{-5}, together with the improvement in convergence rate at the beginning of the iterative refinement process (see Fig. 3.a).

When using the multi-grid solver, most of the iterations are replaced by inner coarse loops, which are computationally less expensive. As a model with fewer degrees of freedom improves convergence, the total number of iterations was reduced from 3340 to 1880 by changing from the PCGE solver to the multi-grid solver. Furthermore, when the data-driven predictor is applied to the multi-grid solver, the residuals of the initial solution range from 1.11×10^{-3} to 1.91×10^{-5}, which is similar to the case using the PCGE solver. However, unlike the PCGE solver, the convergence rate of the outer loop does not improve (Fig. 3.b). The amount of improvement in relative error in the outer loop is driven by the error threshold ϵ^{in} when computing the approximate solution in the inner loop. In the multi-grid solver, the inner coarse loop accounts for most of the computational cost, so it is important to reduce the number of iterations in the inner coarse loop. Convergence was improved in the inner coarse loop, and the number of iterations was significantly reduced from 1763 to 495 (Table 1.c,d).

Next, we evaluated the performance of the data-driven predictor when the number of subregions is varied. As shown in Table 2, the number of iterations was almost constant regardless of the number of subregions, indicating that the data-driven predictor can robustly reduce the computation time independent of the subregion size within a range finer than a certain degree.

Finally, we compare the effectiveness of the data-driven predictor with a linear viscoelastic problem. Table 3 shows the number of iterations required for solving a linear viscoelastic problem on the same mesh. We can see that the reduction in the number of iterations by the data-driven predictor was similar for the nonlinear and linear cases, confirming that data-driven predictor can predict initial solutions of nonlinear viscoelastic problems with high accuracy.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

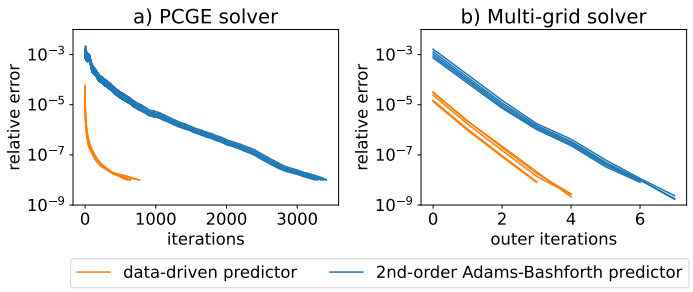

Fig. 3. Convergence history of a) PCGE solver and b) Multi-grid solver for the numerical experiment problem. Convergence history for every 2 time steps among 21-30 time steps are plotted.

Table 2. Number of multi-grid solver iterations per time step with varying subdomain sizes of the data-driven predictor in the numerical experiment problem.

			number of iterations		
case	# of subdomains	DOF per subdomain	outer	inner fine	inner coarse
a	98304	2000	3.73	41.3	460
b	24576	8000	3.60	41	495
c	6144	16000	3.63	40.8	513
d	1536	64000	3.40	35.7	485
e	384	256000	3.40	34.8	496

4 Application example

As an application example, post-seismic crustal deformation was analyzed for a hypothetical Nankai Trough earthquake [8]. Here we show the effect of nonlinear viscoelasticity in crustal deformation, and verify the effectiveness of the developed solver on an actual problem setting. Here we used the crustal structure data of Japan integrated velocity structure model version 1 [12, 11] to model a 2496 km $\times$ 2496 km $\times$ 1100 km area around the Nankai Trough earthquake region with element size of $ds = 1000$ m (see Fig. 4). The degrees of freedom and number of tetrahedral elements are 4.50×10^9 and 1.11×10^9, respectively. We assume that the thickness of the subducting plate is 30 km and that the upper 10 km of the ocean mantle beneath it is nonlinear viscoelastic. Assuming the effective viscosity pattern would be similar to that of the 2011 Tohoku-oki earthquake [3], which have comparable Magnitude levels of the Nankai Trough earthquake, we set the parameters for the nonlinear viscosity layer as Fig. 4.f). The sides and bottom of the finite element model are fixed to zero by Dirichlet boundary conditions. We computed the response after the earthquake for a period of 3 years using a simulation of $N_t = 147$ time steps with time step increment of $dt = 86400$ s. To compare the differences between the results of nonlinear viscoelastic analysis and those of linear viscoelastic analysis, we computed four

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

Table 3. Number of solver iterations per time step for linear viscoelastic model in the numerical experiment problem.

			number of iterations		
case	solver	predictor	outer	inner fine	inner coarse
a	PCGE solver	Adams-Bashforth	3430	-	-
b	PCGE solver	data-driven predictor	664	-	-
c	Multi-grid solver	Adams-Bashforth	6.00	110	1800
d	Multi-grid solver	data-driven predictor	3.77	43.5	537

cases of linear viscoelastic models with viscosity close to the effective viscosity in the nonlinear viscoelastic analysis (see Fig. 4.f-g). We used the same computer system as used in Section 3. Figure 5 shows the elapsed time and the number of solver iterations for the nonlinear viscoelastic case. We can see that the number of iterations was significantly reduced and the elapsed time was accelerated by 2.56-fold by applying the data-driven predictor. Since solving the system of linear equations accounts for 98% The data-driven predictor can be computed very fast (0.03 s). Thus, the combination of the multi-grid solver and the data-driven predictor resulted in a 15.1-fold speedup compared to the standard iterative solver (PCGE solver combined with Adams-Bashforth predictor). The overall computation time for the entire analysis was also improved by a factor of 12.5, from 45600 s to 3650 s, demonstrating the effectiveness of the data-driven predictor and the multi-grid solver in nonlinear viscoelastic analysis. Introducing multi-grid model with more levels by algebraic multigrid methods may reduce computation time and is expected to benefit from data-driven predictors as well ([6, 15] employed a three-level multigrid solver for linear viscoelastic problems and obtained similar results).

Figure 4.f shows the difference in surface displacement response for nonlinear viscoelastic and linear viscoelastic models, where the response varies significantly due to the difference in viscosity. Compared to the linear viscoelastic model, a rapid fluctuation immediately after the earthquake followed by a rapid decay was observed for the nonlinear viscoelastic model. This is because the nonlinear viscoelastic layer exhibits low effective viscosity immediately after the earthquake and undergoes rapid viscous relaxation, while the effective viscosity increases after a few years and transitions to gradual viscous relaxation, as shown in Fig. 4. The difference between linear and nonlinear viscoelastic response was significantly larger than the observation error, indicating the need for nonlinear viscoelastic analysis. In this application example, only the coseismic slip was used as an input; however, in actual crustal deformation analysis, crustal deformation due to continuous afterslip should also be considered. Whether the data-driven predictor is effective in problem settings where external force terms act continuously has not been tested in this study, and is a subject for future work.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

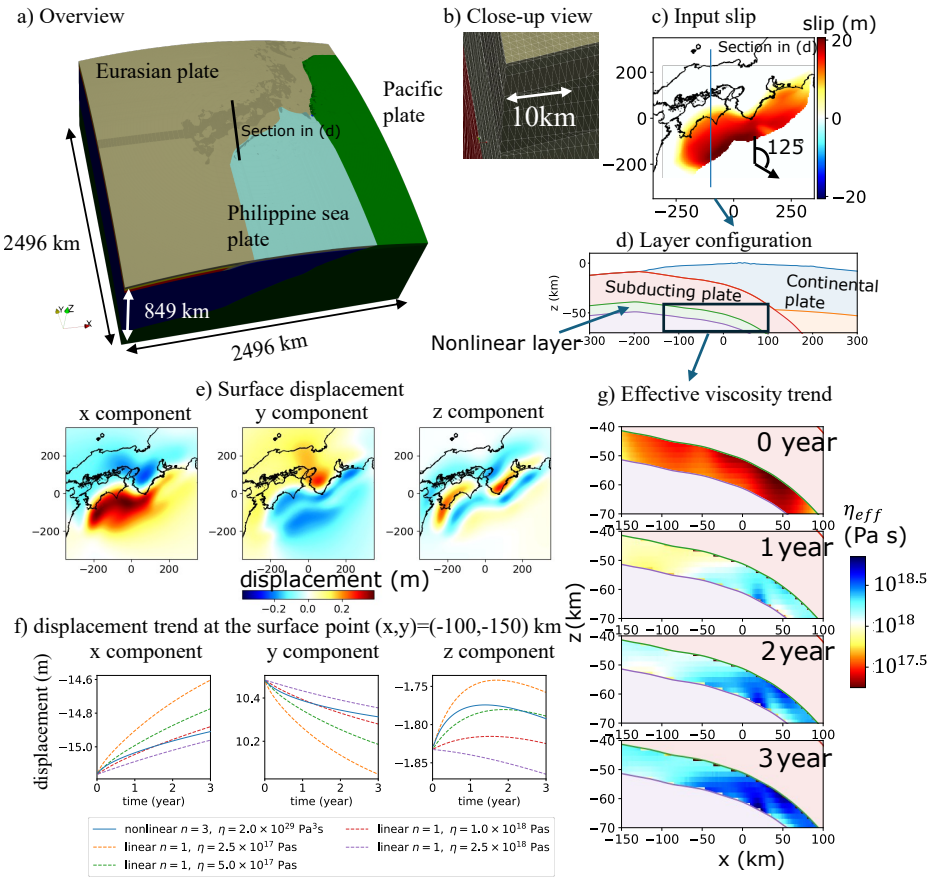

Fig. 4. Finite-element model and analysis results for application examples.

5 Conclusion

Aiming to reduce the analysis cost of nonlinear viscoelastic analysis of spatially highly detailed crustal structure models, we developed a new analysis method combining a data-driven method for highly accurate estimation of the initial solution and a multi-grid solver to reduce the number of solver iterations. First, the developed nonlinear viscoelastic analysis method was verified via a numerical verification problem. Next, we confirmed that the introduction of the data-driven predictor enabled predicting the solution of the next time step with higher accuracy than the conventional method, which led to a reduction in the number of solver iterations by about 3.35-fold. Furthermore, the combination with a multi-grid solver enabled a 15.1-fold speedup of the analysis. It was also confirmed that the data-driven predictor can predict the initial solution with high accuracy, in-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

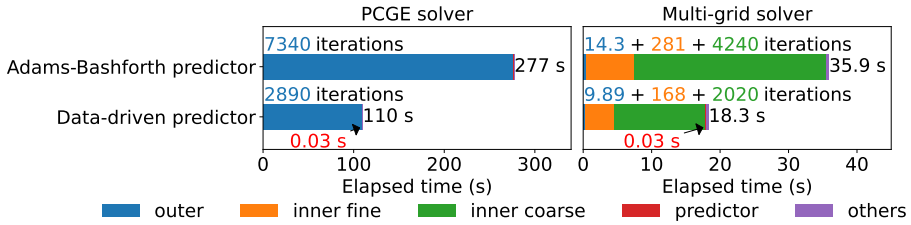

Fig. 5. Number of solver iterations and computation time per time step for the application problem with nonlinear viscoelastic case.

dependent of the number of subregions used for the data-driven predictor. As an application example, the nonlinear viscoelastic response to a hypothetical Nankai Trough earthquake was computed using a highly detailed crustal structure model with 4.5×10^9 degrees of freedom. The fast nonlinear viscoelastic analysis of spatially highly detailed crustal structure models enabled by this study is expected to contribute to the advance of interplate state estimation.

Since the data-driven method developed in this study is very fast compared to the solver with a relatively simple configuration, there is still room for development using more advanced techniques. We can expect further improvement of the method for application to a wider range of problems, improvement in robustness, and further acceleration.

Acknowledgements

This work was supported by JST FOREST Program, Grant Number JPMJFR215Q. This work was supported by JSPS KAKENHI Grant Numbers 23H00213, 22K12057. This work was supported by Research Project for Disaster Prevention on the great Earthquakes along the Nankai trough, Japan Grant Number JPJ008967, funded by MEXT. Part of the results were obtained using computational resource of Fugaku at Center for Computational Science, RIKEN. This work was supported by MEXT as "Program for Promoting Researches on the Supercomputer Fugaku" (Large-scale numerical simulation of earthquakes by Fugaku supercomputer, JPMXP1020230213).

References

1. Metis. https://github.com/KarypisLab/METIS, accessed 24 Feb 2025
2. OpenACC. https://www.openacc.org/, accessed 24 Feb 2025
3. Agata, R., Barbot, S.D., Fujita, K., Hyodo, M., Iinuma, T., Nakata, R., Ichimura, T., Hori, T.: Rapid mantle flow with power-law creep explains deformation after the 2011 Tohoku mega-quake. Nature Communications **10**(1), 1385 (2019)
4. Barbot, S., Fialko, Y.: A unified continuum representation of post-seismic relaxation mechanisms: semi-analytic models of afterslip, poroelastic rebound and viscoelastic flow. Geophysical Journal International **182**(3), 1124–1140 (2010)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1

5. Bürgmann, R., Dresen, G.: Rheology of the lower crust and upper mantle: Evidence from rock mechanics, geodesy, and field observations. Annu. Rev. Earth Planet. Sci. **36**(1), 531–567 (2008)
6. Fujita, K., Murakami, S., Ichimura, T., Hori, T., Hori, M., Lalith, M., Ueda, N.: 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. IEEE (2022)
7. Hori, T., Agata, R., Ichimura, T., Fujita, K., Yamaguchi, T., Iinuma, T.: High-fidelity elastic Green's functions for subduction zone models consistent with the global standard geodetic reference system. Tech. rep., Springer (2021)
8. Hyodo, M., Hori, T.: Re-examination of possible great interplate earthquake scenarios in the Nankai Trough, southwest Japan, based on recent findings and numerical simulations. Tectonophysics **600**, 175–186 (2013)
9. Ichimura, T., Agata, R., Hori, T., Hirahara, K., Hashimoto, C., Hori, M., Fukahata, Y.: An elastic/viscoelastic finite element analysis method for crustal deformation using a 3-D island-scale high-fidelity model. Geophysical Journal International **206**(1), 114–129 (2016)
10. Ichimura, T., Fujita, K., Koyama, K., Kusakabe, R., Kikuchi, Y., Hori, T., Hori, M., Maddegedara, L., Ohi, N., Nishiki, T., et al.: 152K-computer-node parallel scalable implicit solver for dynamic nonlinear earthquake simulation. In: International Conference on High Performance Computing in Asia-Pacific Region. pp. 18–29 (2022)
11. Koketsu, K., Miyake, H., Suzuki, H.: Japan integrated velocity structure model version 1. In: Proceedings of the 15th world conference on earthquake engineering. vol. 1, p. 4. Lisbon (2012)
12. Koketsu, K., Miyake, H., Tanaka, Y., et al.: A proposal for a standard procedure of modeling 3-D velocity structures and its application to the Tokyo metropolitan area, Japan. Tectonophysics **472**(1-4), 290–300 (2009)
13. Masuti, S., Barbot, S.D., Karato, S.i., Feng, L., Banerjee, P.: Upper-mantle water stratification inferred from observations of the 2012 Indian Ocean earthquake. Nature **538**(7625), 373–377 (2016)
14. Melosh, H., Raefsky, A.: A simple and efficient method for introducing faults into finite element computations. Bulletin of the Seismological Society of America **71**(5), 1391–1400 (1981)
15. Murakami, S., Fujita, K., Ichimura, T., Hori, T., Hori, M., Lalith, M., Ueda, N.: Development of 3D viscoelastic crustal deformation analysis solver with data-driven method on GPU. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M. (eds.) Computational Science – ICCS 2023. pp. 423–437. Springer Nature Switzerland, Cham (2023)
16. Parker, J., Lyzenga, G., Norton, C., Zuffada, C., Glasscoe, M., Lou, J., Donnellan, A.: Geophysical finite-element simulation tool (GeoFEST): algorithms and validation for quasistatic regional faulted crust problems. Pure and Applied Geophysics **165**, 497–521 (2008)
17. Winget, J.M., Hughes, T.J.: Solution algorithms for nonlinear transient heat conduction analysis employing element-by-element iterative strategies. Computer Methods in Applied Mechanics and Engineering **52**(1-3), 711–815 (1985)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_1