https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

A GPU-accelerated geodynamic modelling framework: JustRelax.jl v0.8

Albert de Montserrat[1,★], Pascal S. Aellig[2,3,★], Ludovic Räss[4], Christian Schuler[4], Thibault Duretz[5], and Boris J.P. Kaus[2,3]

[1]Geophysical Fluid Dynamics, Institute of Geophysics, ETH Zürich, Zürich, Switzerland
[2]Institute of Geosciences, Johannes Gutenberg University Mainz, Mainz, Germany
[3]Mainz Institute of Multiscale Modelling, Johannes Gutenberg University, Mainz, Germany
[4]Institute of Earth Sciences, University of Lausanne, Lausanne, Switzerland
[5]Institut für Geowissenschaften, Goethe-Universität Frankfurt, Frankfurt, Germany
★These authors contributed equally to this work.

Correspondence: Albert de Montserrat (albertdemontserratnavarro@eaps.ethz.ch)

Abstract. The rapid evolution of high-performance computing and the widespread adoption of graphics processing units (GPUs) require corresponding advances in solver design and strategies in computational geodynamics. Pseudo-transient, or dynamic-relaxation, approaches are particularly promising for multi-physics problems on structured staggered grids and are well suited to efficient parallel execution. Here, we introduce JustRelax.jl, a geodynamic code written in the Julia programming language that runs seamlessly on GPUs, from a single device to distributed multi-device systems. JustRelax.jl separates material-property calculations, particle-in-cell advection, and (distributed) parallelism into dedicated packages, facilitating the integration of new physical processes, numerical methods, and solver strategies while limiting modifications to the solver core. We demonstrate the capabilities of JustRelax.jl using a suite of geodynamic applications, including plastic shear-band localization, thermal convection, subduction, and magmatic systems. We assess its parallel scalability on modern GPU-based supercomputers, showing efficient scaling to hundreds of GPUs and providing a foundation for future extensions in computational geodynamics.

1 Introduction

In recent years, the high-performance computing (HPC) landscape has changed substantially owing to the increasing availability of graphics processing units (GPUs), with contemporary supercomputer centres deploying thousands of devices from multiple vendors. The transition from CPU-centric to GPU-centric architectures poses challenges that extend beyond conventional optimization and straightforward porting. As a result, many monolithic community codes require extensive modifications that are both time-consuming and technically demanding. This transition also motivates changes in solver strategy, because matrix-based approaches can be inefficient on GPUs owing to memory-access patterns, storage requirements, and the cost of global communication. Iterative, matrix-free methods are therefore particularly attractive for multi-physics problems on structured staggered grids.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Räss et al. (2022) demonstrated the feasibility of solving Stokes flow on thousands of GPUs with near-optimal algorithmic weak scalability while effectively utilizing the available memory bandwidth. Their approach uses dynamic relaxation, or the accelerated pseudo-transient (APT) method, in which the governing equations are augmented with a pseudo-time derivative and damped acceleration term, and iterated until a prescribed numerical tolerance is reached (e.g., Frankel, 1950; Papadrakakis, 1981). The resulting matrix-free formulation circumvents constructing a Jacobian matrix and the associated cost of evaluating matrix–vector products. Pseudo-transient methods have also been applied successfully to a wide range of geodynamic problems (e.g. Duretz et al., 2019; Kiss et al., 2023; Alkhimenkov et al., 2024; Spang et al., 2025; Aellig et al., 2026).

Although hundreds to thousands of programming languages exist, a small subset —notably C/C++, Fortran, Python, and MATLAB— has historically dominated computational geodynamics and computational science more broadly. Interpreted and dynamic languages such as MATLAB, e.g. MILAMIN (Dabrowski et al., 2008), LaCoDe (de Montserrat et al., 2019), and Python, e.g. Underworld (Moresi et al., 2002), are widely used for rapid prototyping and method development. In contrast, when computational performance is paramount, statically compiled languages such as C/C++, e.g. ASPECT (Bangerth et al., 2023), LaMEM (Kaus et al., 2016), and Fortran, e.g. StagYY (Tackley, 2008), remain common choices. These established codes comprise large and complex code bases and often rely on external libraries such as PETSc (Balay et al., 2026) and deal.II (Arndt et al., 2021), among many others.

In this study, we present JustRelax.jl, a high-performance computational geodynamics package written in the Julia programming language (Bezanson et al., 2017). To the best of our knowledge, this is the first production-ready, high-performance solid-Earth geodynamics library written in a language other than those traditionally used in the field. Julia offers a lower barrier to entry for prototyping while still enabling high performance through just-in-time (JIT) compilation, addressing the so-called "two-language barrier" and enabling both prototyping and production development within a single language. Trade-offs relative to established community codes and limitations of the approach are discussed in Sect. 7.

In the remainder of this paper, we describe the software design, underlying methods, and discretization, demonstrate the capabilities of JustRelax.jl using representative geodynamic applications spanning single-device to multi-device simulations, and evaluate scalability through strong- and weak-scaling experiments.

2 Software design

We present JustRelax.jl (de Montserrat et al., 2026c), a geoscientific modelling package written in the Julia programming language (Bezanson et al., 2017). We developed it as a high-performance tool for solid-Earth geodynamic modelling that is portable across hardware (CPUs and GPUs) and parallel architectures (multithreading, MPI, and GPU-aware MPI). Julia's multiple dispatch and metaprogramming capabilities make it easier to target this range of backends than many other languages do, while retaining the performance required for large-scale simulations.

JustRelax.jl provides a high-level API for heat diffusion and for the compressible Stokes equations, both solved with an iterative accelerated pseudo-transient (APT) relaxation method, as well as a set of underlying building blocks packed as independent self-consistent packages. The same solver core runs on CPUs and on NVIDIA and AMD GPUs without modifica-

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

tion. This portability is enabled by ParallelStencil.jl (Omlin and Räss, 2024), which uses metaprogramming to generate device-agnostic parallel kernels at parse time.

JustRelax.jl supports both 2D and 3D simulations via dedicated submodules. To run on a GPU, users load the vendor-specific package CUDA.jl (Besard et al., 2018) or AMDGPU.jl (Samaroo et al., 2026) and switch the ParallelStencil.jl initialization from Threads to the corresponding GPU backend (Algorithm 1). For example, to set up a 3D environment on an NVIDIA device, users need only add the relevant imports at the top level of their script:

```
const architecture = :cuda # :cpu, :cuda, or :amdgpu
if architecture == :cuda
    using CUDA
elseif architecture == :amdgpu
    using AMDGPU
end

using JustRelax, JustRelax.JustRelax3D
using ParallelStencil

if architecture == :cuda
    @init_parallel_stencil(CUDA, Float64, 3)
elseif architecture == :amdgpu
    @init_parallel_stencil(AMDGPU, Float64, 3)
else
    @init_parallel_stencil(Threads, Float64, 3)
end
```

Listing 1. Example script to load JustRelax.jl and select the target backend.

We designed JustRelax.jl to be modular and to delegate key functionality to a set of core dependencies. These dependencies are intentionally independent of the higher-level API provided by JustRelax.jl, so they can be reused by other software and libraries. The key dependencies are: (**i**) GeoParams.jl for local, solver-agnostic material-property calculations (e.g. density, viscosity, plasticity); (**ii**) JustPIC.jl for device-agnostic particle advection; (**iii**) ParallelStencil.jl as an abstraction layer for generating device-agnostic kernels; and (**iv**) ImplicitGlobalGrid.jl for distributed parallelism across multiple CPUs and GPUs. Additional minor dependencies are listed in the `Project.toml` files of the respective repositories.

2.1 Main dependencies

We describe the main dependencies and their roles below, and we summarize them in Fig. 1.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

2.1.1 Material physics: GeoParams.jl

GeoParams.jl provides a framework for local material-physics calculations, including equations of state, thermal parameters, and partial-melt parameterizations. Separating these calculations from the solver keeps GeoParams.jl independent of the discretization method (e.g. finite differences, finite elements, finite volumes, or discrete elements) and reusable across modelling frameworks. It also supports automatic conversion between dimensional and non-dimensional parameters, includes an extensive database of nonlinear creep laws, and is compatible with the Julia ecosystem for automatic differentiation.

2.1.2 Advection: JustPIC.jl

JustPIC.jl (de Montserrat et al., 2026a) provides particle-in-cell advection and bidirectional grid–particle interpolation of physical fields, specialized for staggered-grid discretizations. Further details are given in Sect. 4.

2.1.3 Shared memory parallelization: ParallelStencil.jl

Shared-memory parallelization typically requires backend-specific kernel implementations, for example, multithreading on CPUs or GPU kernels. ParallelStencil.jl (Omlin and Räss, 2024) provides an abstraction layer that generates shared-memory parallel kernels for the active hardware backend, reducing implementation-specific boilerplate while maintaining a backend-agnostic code base. Currently, CPUs and GPUs from NVIDIA and AMD are supported. Although ParallelStencil.jl also supports Apple GPUs (Metal), we do not currently support them in JustRelax.jl because they are limited to single-precision floating-point arithmetic.

2.1.4 Distributed parallelization: ImplicitGlobalGrid.jl

We provide distributed-memory parallelism through ImplicitGlobalGrid.jl (Omlin et al., 2024), which uses the Message Passing Interface (MPI) for domain decomposition and inter-process communication across multiple CPUs or GPUs. ImplicitGlobalGrid.jl builds on MPI.jl (Byrne et al., 2021), which provides Julia bindings to standard MPI libraries.

2.1.5 I/O and visualization

For data output and visualization, JustRelax.jl relies on established formats and libraries, including VTK, HDF5, and the Julia-specific, HDF5-compatible format JLD2. We provide checkpointing routines for the required fields and particles via JustPIC.jl. Visualization is supported either through ParaView in combination with WriteVTK.jl, or through Makie.jl (Danisch and Krumbiegel, 2021) for in situ visualization.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

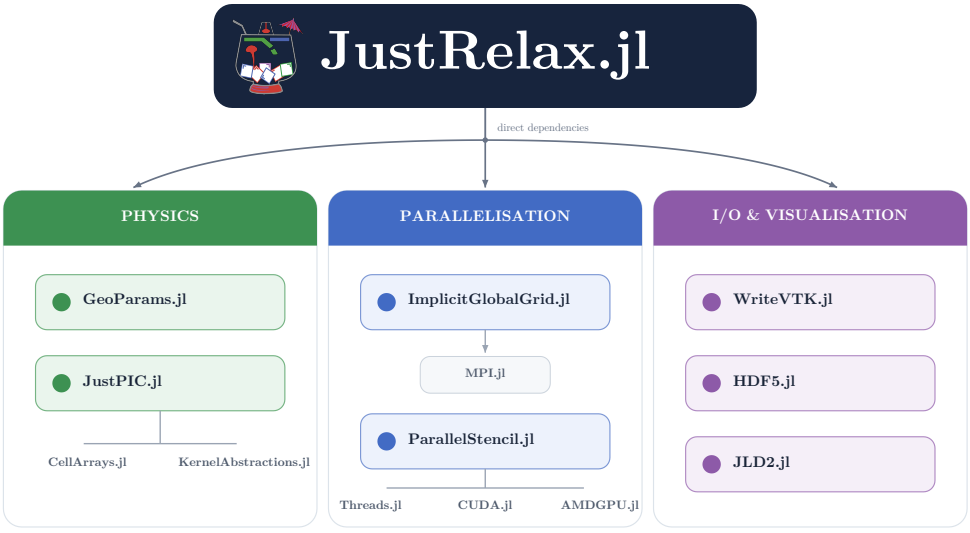


Figure 1. Dependency tree of JUSTRELAX.JL. Arrows indicate package dependencies; lighter nodes denote supporting packages and execution backends.

3 Methods

This section summarizes the mathematical and numerical framework used in this study. We first introduce the governing conservation laws and constitutive assumptions, and then outline the spatial discretization and iterative solution strategy used to compute the coupled velocity–pressure and temperature fields.

3.1 Governing equations

Geological processes in the solid Earth typically evolve over long timescales and involve slow deformation. Inertial terms can therefore be neglected, and the equations describing its deformation reduce to the compressible Stokes equations, expressed through the conservation of momentum and mass, respectively,

$$\nabla \cdot \boldsymbol{\tau} - \nabla p = \boldsymbol{f} \tag{1}$$

$$\nabla \cdot \boldsymbol{v} = \alpha \frac{\mathrm{D}T}{\mathrm{D}t} - \beta \frac{\mathrm{D}p}{\mathrm{D}t}, \tag{2}$$

where $\boldsymbol{v}$ is the velocity, p is the pressure, $\boldsymbol{\tau}$ is the deviatoric stress tensor, and $\boldsymbol{f} = \rho \boldsymbol{g}$ is the body-force term (density ρ and gravity vector $\boldsymbol{g}$). The operator $\mathrm{D}(\cdot)/\mathrm{D}t$ denotes the material derivative ($\mathrm{D}(\cdot)/\mathrm{D}t \equiv \partial(\cdot)/\partial t + \boldsymbol{v} \cdot \nabla(\cdot)$), and α and β are the thermal expansivity and compressibility, respectively.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

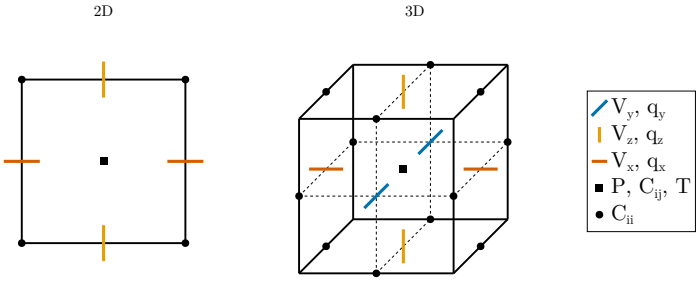


Figure 2. Staggered-grid finite-difference discretization of the Stokes and energy equations in 2D and 3D.

The system is closed with the constitutive law. In its simplest isotropic viscous form, this relationship is given by $\boldsymbol{\tau} = 2\eta\dot{\boldsymbol{\varepsilon}}$, where η is the (effective) viscosity and the deviatoric strain-rate tensor is $\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathrm{T}}\right) - \frac{1}{3}(\nabla\cdot\boldsymbol{v})\,\mathbf{I}$, with $\mathbf{I}$ the identity matrix. More complex constitutive laws are available in JustRelax.jl and are described in Sect. 3.3.

JustRelax.jl supports various density parameterizations through its coupling with GeoParams.jl. By default, density depends on pressure and temperature according to

$$\rho = \rho_0\left(1 - \alpha(T - T_0) + \beta(p - p_0)\right), \tag{3}$$

where ρ_0, T_0, and p_0 are the reference density, temperature, and pressure, respectively. Furthermore, thermal evolution is modelled through the conservation of energy,

$$\rho C_p \frac{\mathrm{D}T}{\mathrm{D}t} = \nabla\cdot(k\nabla T) + \boldsymbol{\tau} : \dot{\boldsymbol{\varepsilon}} + \alpha T\frac{\mathrm{D}p}{\mathrm{D}t} + Q, \tag{4}$$

where T is temperature, C_p is the specific heat capacity, k is the thermal conductivity, and Q represents additional volumetric heat sources or sinks. The terms $\boldsymbol{\tau} : \dot{\boldsymbol{\varepsilon}}$ and $\alpha T\mathrm{D}p/\mathrm{D}t$ correspond to shear heating and adiabatic heating, respectively. In JustRelax.jl, we discretize Eqs. (1)–(4) using finite differences on a staggered grid (Fig. 2; e.g. Gerya and Yuen (2003)).

3.2 Solution method: accelerated pseudo-transient method

JustRelax.jl provides solver strategies based on the accelerated pseudo-transient (APT) method for Eqs. (1)–(4). With APT, the steady governing equations are augmented with pseudo-time derivatives and iterated in pseudo-time ψ until the residuals satisfy a prescribed tolerance. In residual form, the pseudo-transient system reads:

$$\theta\frac{\partial\boldsymbol{v}}{\partial\psi} = \nabla\cdot\boldsymbol{\tau} - \nabla p - \boldsymbol{f} \tag{5}$$

$$\zeta\frac{\partial p}{\partial\psi} = -\nabla\cdot\boldsymbol{v} + \alpha\frac{\mathrm{D}T}{\mathrm{D}t} - \beta\frac{\mathrm{D}p}{\mathrm{D}t} \tag{6}$$

$$\xi\frac{\partial T}{\partial\psi} = -\rho C_p\frac{\mathrm{D}T}{\mathrm{D}t} + \nabla\cdot(k\nabla T) + \boldsymbol{\tau} : \dot{\boldsymbol{\varepsilon}} + \alpha T\frac{\mathrm{D}p}{\mathrm{D}t} + Q, \tag{7}$$

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Here, θ, ζ, and ξ are damping coefficients that control the relaxation of velocity, pressure, and temperature, respectively. Their values depend on the spectral properties of the discrete operators. In JUSTRELAX.JL, we typically use damping coefficients derived from a dispersion analysis of the pseudo-transient equations, following Räss et al. (2022). The resulting expressions are robust for the pseudo-transient energy equation; however, for the pseudo-transient Stokes equations (5) and (6), convergence can degrade in the presence of strong material nonlinearities and sharp viscosity contrasts. In these regimes, suitable pseudo-transient parameters can be difficult to choose and may change during a simulation as the spectral properties of the discrete operator evolve. To improve robustness, JUSTRELAX.JL also includes the auto-tuned APT strategy of Duretz et al. (2026), in which the damping coefficients are updated automatically during iterations. This approach decouples velocity and pressure using a Powell–Hestenes (PH) formulation and updates the velocity using dynamic relaxation in an inner iteration loop. Duretz et al. (2026) reported that this strategy keeps the number of pseudo-transient iterations approximately constant throughout a simulation and can substantially reduce the time-to-solution, particularly at high spatial resolution.

3.3 Rheology

Material behaviour is represented using a Maxwell-type rheology, in which the total strain-rate tensor $\dot{\boldsymbol{\varepsilon}}$ is decomposed into the sum of N deformation mechanisms, $\dot{\boldsymbol{\varepsilon}} = \sum_{i=1}^{N} \dot{\boldsymbol{\varepsilon}}_i$. In the visco-elasto-viscoplastic formulation employed in JUSTRELAX.JL, this decomposition comprises viscous, elastic, and plastic contributions. The constitutive relation couples Eqs. (1) and (2) through the pressure field and closes the governing system:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\text{viscous}} + \dot{\boldsymbol{\varepsilon}}^{\text{elastic}} + \dot{\boldsymbol{\varepsilon}}^{\text{plastic}} = \frac{1}{2\eta^{\text{eff}}}\boldsymbol{\tau} + \frac{1}{2G}\frac{\mathrm{D}\boldsymbol{\tau}}{\mathrm{D}t} + \dot{\lambda}\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\tau}}, \tag{8}$$

where the superscripts denote the viscous, elastic, and plastic components, respectively, η^{eff} is the effective viscosity, G is the shear modulus, $\mathrm{D}\boldsymbol{\tau}/\mathrm{D}t$ is the objective stress rate, $\dot{\lambda}$ is the plastic multiplier, and $\mathcal{Q}$ is the plastic flow potential (see Sect. 3.3.3). The effective viscosity in Eq. (8) is calculated by combining multiple viscous creep mechanisms through a harmonic sum,

$$\eta^{\text{eff}} = \left(\sum_{m=1}^{N^{\text{v}}} \frac{1}{\eta_m^{\text{viscous}}} \right)^{-1}, \tag{9}$$

where N^v is the number of active viscous rheology elements. For numerical stability, the effective viscosity is capped so that $\eta^{\text{min}} \leq \eta^{\text{eff}} \leq \eta^{\text{max}}$, where the viscosity contrast between the maximum and minimum viscosity caps is typically $\eta^{\text{max}}/\eta^{\text{min}} \approx 10^5$.

3.3.1 Creep laws

A wide range of creep laws are supported natively by JUSTRELAX.JL, including the non-Newtonian flow laws commonly used in geodynamic modelling. The viscous strain-rate tensor for mechanism m is given by

$$\dot{\varepsilon}_{\text{II},m} = A_m \tau_{\text{II}}^n d^{-p_m} f_{\text{H}_2\text{O}}^{r_m} \exp\left(-\frac{E_m + pV_m}{RT} \right), \tag{10}$$

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

where A is the pre-exponential factor, n is the stress exponent, d is the grain size, p is the grain size exponent, $f_{\mathrm{H_2O}}$ is the water fugacity, E is the activation energy, V is the activation volume, and R is the universal gas constant; m denotes the active creep mechanism.

In addition, JUSTRELAX.JL natively supports any other creep law implemented in GEOPARAMS.JL (Kaus et al., 2026) (e.g. the Herschel–Bulkley flow law (Herschel and Bulkley, 1926; Bird et al., 1987), or melt viscosity (Giordano et al., 2008)).

3.3.2 Elastic stress

The elastic stress contribution in the Maxwell-type rheology is computed using an objective stress rate to account for material rotations during deformation. The Jaumann derivative of the deviatoric stress tensor is,

$$\frac{\mathrm{D}\boldsymbol{\tau}}{\mathrm{D}t} = \frac{\partial \boldsymbol{\tau}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{\tau} + \boldsymbol{\tau}\boldsymbol{\Omega} - \boldsymbol{\Omega}\boldsymbol{\tau}, \tag{11}$$

where $\boldsymbol{\Omega} = \frac{1}{2}\left(\nabla\boldsymbol{v} - (\nabla\boldsymbol{v})^{\mathrm{T}}\right)$ is the spin tensor. The rotational correction can be approximated by a truncated Taylor expansion $\boldsymbol{\tau}^{\star} = \mathbf{R}\boldsymbol{\tau}^{t}\mathbf{R}^{\mathrm{T}}$ (Beuchert and Podladchikov, 2010). In two dimensions, the rotation angle is $\theta = \Omega_{xy}\Delta t$ and the rotation matrix $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{12}$$

In three dimensions, the rotation matrix is computed following the Euler–Rodrigues formula:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -n_3\sin\theta & n_2\sin\theta \\ n_3\sin\theta & \cos\theta & -n_1\sin\theta \\ -n_2\sin\theta & n_1\sin\theta & \cos\theta \end{bmatrix} + (1-\cos\theta)\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}, \tag{13}$$

where in this case $\theta = \gamma\Delta t/2$ is the incremental rotation angle, $\gamma = \sqrt{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}}$, with $\boldsymbol{\omega} = (\Omega_{yz}, \Omega_{xz}, \Omega_{xy})$, and $\boldsymbol{n} = \gamma^{-1}\boldsymbol{\omega}$ is the unit rotation axis associated with the axial vector of the spin tensor. The objective stress rate is then computed from the rotated stress tensor as

$$\frac{\mathrm{D}\boldsymbol{\tau}}{\mathrm{D}t} = \frac{\boldsymbol{\tau} - \boldsymbol{\tau}^{\star}}{\Delta t}. \tag{14}$$

This formulation provides an objective update of the elastic stress during each time step.

3.3.3 Plastic failure

Brittle deformation is represented using the regularized Drucker–Prager yield criterion following Duretz et al. (2021). The corresponding yield function $\mathcal{F}$ is given by

$$\mathcal{F}^{\mathrm{trial}} = \tau_{\mathrm{II}}^{\mathrm{trial}} - p^{\mathrm{trial}}\sin\phi - C\cos\phi - \dot{\lambda}\eta^{\mathrm{reg}}, \tag{15}$$

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

where ϕ is the friction angle, C is the cohesion, η^{reg} is a regularization term with units of viscosity, $\dot{\lambda}$ is the plastic multiplier, and η^{ve} is the effective visco-elastic viscosity,

$$\eta^{\text{ve}} = \left(\frac{1}{G\Delta t} + \frac{1}{\eta^{\text{eff}}} \right)^{-1}. \tag{16}$$

with G being the shear modulus. In the trial state, $\dot{\lambda}$ is zero. A non-associative plastic potential $\mathcal{Q} = \tau_{II} - p\sin\psi$ is used, where ψ is the dilation angle. The plastic multiplier $\dot{\lambda}$ is then

$$\dot{\lambda} = \begin{cases} \mathcal{F}^{\text{trial}} \left(\eta^{\text{ve}} + \eta^{\text{reg}} + K_b \Delta t \frac{\partial \mathcal{F}}{\partial p} \frac{\partial \mathcal{Q}}{\partial p} \right)^{-1}, & \text{if } \mathcal{F} > 0 \\ 0, \quad \text{otherwise} \end{cases}, \tag{17}$$

where K_b is the bulk modulus. For non-zero dilation angles, the volumetric plastic strain contribution is incorporated by augmenting the pressure term,

$$p \leftarrow p + K_b \dot{\lambda} \Delta t \sin\psi. \tag{18}$$

Finally, upon convergence, the deviatoric stress tensor is updated as:

$$\boldsymbol{\tau} = 2\eta^{\text{ve}} \left(\dot{\boldsymbol{\varepsilon}} - \dot{\lambda} \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\tau}} \right). \tag{19}$$

The description above represents one of the most common formulations for brittle failure; however, we must note that any other brittle failure model available in GeoParams.jl (e.g., Drucker–Prager plasticity with a tensile cap (Popov et al., 2025)) is also natively supported.

3.4 Resolving nonlinearities

A key advantage of the APT method is that it can accommodate the nonlinearities present in the governing equations, such as non-Newtonian fluids and plastic failure, without modifying the solver formulation (e.g. Räss et al., 2022; Kiss et al., 2023; Alkhimenkov and Podladchikov, 2025). These nonlinearities are solved explicitly during the pseudo-transient iterations, thereby avoiding more complex nonlinear schemes that require the computation of Jacobians, as in Newton-type methods. For strongly nonlinear, non-Newtonian rheologies, continuation on the effective viscosity is applied, which substantially improves convergence of the iterative solver:

$$\eta^{n+1} \leftarrow \Lambda \eta^{n+1} + (1 - \Lambda)\eta^{n}, \tag{20}$$

where $0 \leq \Lambda \leq 1$ is a damping coefficient. In practice, values in the range $10^{-3} \leq \Lambda \leq 10^{-2}$ typically perform well, depending on the degree of nonlinearity and the complexity of the model configuration.

4 Advection

Advection of material properties, temperature, and associated phase ratios is a key component of geodynamic simulations. JustRelax.jl provides two complementary advection schemes: (i) a particle-in-cell (PIC) method implemented in JustPIC.jl

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

(de Montserrat et al., 2026a), and (ii) an array-based fifth-order weighted essentially non-oscillatory (WENO-5) scheme (Jiang and Shu, 1996b).

4.1 Particle-in-cell

The advection of Lagrangian markers on an Eulerian grid is widely used in computational geodynamics (e.g., van Keken et al., 1997; Gerya and Yuen, 2003; Gerya, 2019; Pusok et al., 2017; Dominguez et al., 2024; Schuler et al., 2025) and dates back to Harlow and Welch (1965).

4.1.1 Grid-to-particle interpolation

Field values defined at grid nodes are mapped to particle positions using n-linear interpolation. Let $d \in \{2,3\}$ denote the spatial dimension, $\boldsymbol{t} = (t_1, \ldots, t_d) \in [0,1]^d$ be the particle position normalized by the lower and upper bounds of its parent cell, and $v_{\boldsymbol{b}}$ be the field value at the cell corner indexed by $\boldsymbol{b} = (b_1, \ldots, b_d) \in \{0,1\}^d$. Here, $b_k = 0$ and $b_k = 1$ refer to the lower and upper corner, respectively, in coordinate direction k. The n-linear interpolant is therefore

$$I(\boldsymbol{t}) = \sum_{\boldsymbol{b}\in\{0,1\}^d} v_{\boldsymbol{b}} \prod_{k=1}^{d} \left[b_k t_k + (1-b_k)(1-t_k) \right]. \tag{21}$$

For $d = 2$, Eq. (21) is the usual bilinear interpolant over the four cell corners; for $d = 3$, it is the trilinear interpolant over the eight corners. In practical terms, $I(\boldsymbol{t})$ is implemented by recursively applying the one-dimensional *lerp* kernel

$$\mathrm{lerp}(t, v_0, v_1) = (1-t)v_0 + tv_1 = \mathrm{fma}(t, v_1, \mathrm{fma}(-t, v_0, v_0)), \tag{22}$$

along the coordinate directions. The recursion is fully inlined by the compiler, so it introduces no dimension-dependent branching in the resulting kernel. Here, $\mathrm{fma}(a,b,c) = ab + c$ denotes the hardware *fused multiply-add* instruction, which combines the multiplication and addition into a single operation before rounding and can improve both performance and numerical accuracy on modern processors.

4.1.2 Velocity interpolation schemes

Several velocity interpolation schemes are implemented in JustPIC.jl. The default option uses bilinear interpolation in 2D and trilinear interpolation in 3D. More advanced schemes are also available, including linear-plus-pressure (LinP) (Pusok et al., 2017; Gerya, 2019) and the modified quadratic spline (MQS) method (Gerya et al., 2021). LinP and MQS have been shown to improve marker distributions in regimes dominated by pure shear. These schemes can be combined with second- and fourth-order Runge–Kutta time integration, and a particle-free semi-Lagrangian (SL) backtracking advection approach is also provided.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

4.1.3 Particle-to-grid interpolation

After advection, information stored on the particles must be transferred back to the grid to update the fields. An inverse-distance-weighted interpolant is used to distribute particle information to the grid. For a given grid location $\boldsymbol{x}$, the interpolated field value $\bar{a}$ is computed as

$$\bar{a} = \frac{\sum_i^n \omega_i a_i}{\sum_i^n \omega_i}, \quad \text{with} \quad \omega_i = \frac{1}{|\boldsymbol{x} - \boldsymbol{x}_i|^q}, \tag{23}$$

where a_i is the value carried by the i-th particle, $\boldsymbol{x}_i$ is its position, and $\boldsymbol{x}$ is the target grid location. The weighting factor ω_i is determined by the distance between the particle and the grid location, with exponent q controlling the spatial decay. The exponent q is chosen as a positive integer, where larger values increase the relative contribution of nearby particles compared with more distant particles; by default, we take $q = 2$.

4.1.4 Memory layout and implementation

Several approaches exist for organizing particle data into suitable structures. A common approach is to store particle data linearly in vectors or column arrays (concatenated or not), depending on the programming language storage convention. However, this approach introduces several potential issues:

1. **Suboptimal spatial locality**. Upon initialization, particles are sorted according to their parent cells. Strong advection progressively redistributes particles throughout the domain, leading to non-consecutive memory access, reduced spatial locality, and increased cache misses.

2. **Thread safety**. The interpolant in Eq. (23) involves two reduction operations, both of which introduce potential race conditions during parallel execution. Therefore, $2n_\mathrm{p}$ atomic operations, where n_p is the total number of particles, are required to ensure thread safety. In addition, two intermediate arrays storing local reductions may be required when iterating over particles, because the number of particles contributing to each grid cell is not known *a priori*.

To address these limitations, particles are stored in *CellArrays* objects provided by CellArrays.jl. To maintain spatial locality throughout a simulation, each cell is checked for incoming particles after the advection step. If a particle moves into a new cell, all associated metadata are transferred to the first available memory location within the new parent cell (Fig. 3). Consequently, particles are preferentially arranged from top to bottom within each cell and stored as contiguously as possible in memory.

4.2 Grid-based advection

Although particle-in-cell methods provide a flexible approach for tracking material interfaces, they are not optimal for all geodynamic applications. Grid-based advection schemes are advantageous for problems involving transport of scalar fields

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

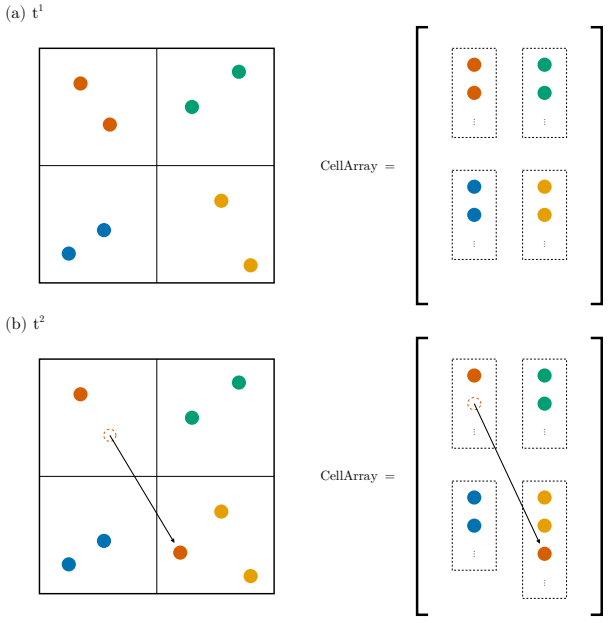


Figure 3. (a) Particle distribution at $t = t^1$ and (b) at $t = t^2$ after advection. The red particle moves into a different grid cell and is therefore transferred from CellArray_{11} to CellArray_{22}.

(e.g. temperature or chemical composition), where high-order conservative operators are required. For example, WENO-based schemes have been applied successfully to chemical advection during magma ascent (Dominguez et al., 2024) and multi-phase flow in porous to suspension flow regimes in magmatic systems (Wong and Keller, 2023). JUSTRELAX.JL provides a WENO-5 scheme to advect grid quantities, including temperature, phase ratios, and other fields (Jiang and Shu, 1996a).

While the advection of scalar fields is generally straightforward using an array-based approach, the phase-ratio representation requires additional treatment because phase information is stored at all staggered-grid locations. To ensure consistency of the phase ratios after advection, each phase is represented by an individual array containing its phase fraction, $\phi_i \in [0,1]$. These phase arrays are advected independently and subsequently combined to update the phase-ratio structure.

5 GPU acceleration and scaling

5.1 GPU acceleration

To assess the benefit of GPU acceleration for matrix-free pseudo-transient Stokes solvers, we compare CPU and GPU performance for 2D and 3D models. We use the shear-band localization example described in Sect. 6.1; the 3D case is identical to the 2D setup, but with a cubic domain $\Omega \in [0,1] \times [0,1] \times [0,1]$ and a spherical inclusion of radius $r = 0.1$ at its centre. Both models were run for 20 time steps at resolutions of 256×256 (2D) and $64 \times 64 \times 64$ (3D). The same high-level Julia code was

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

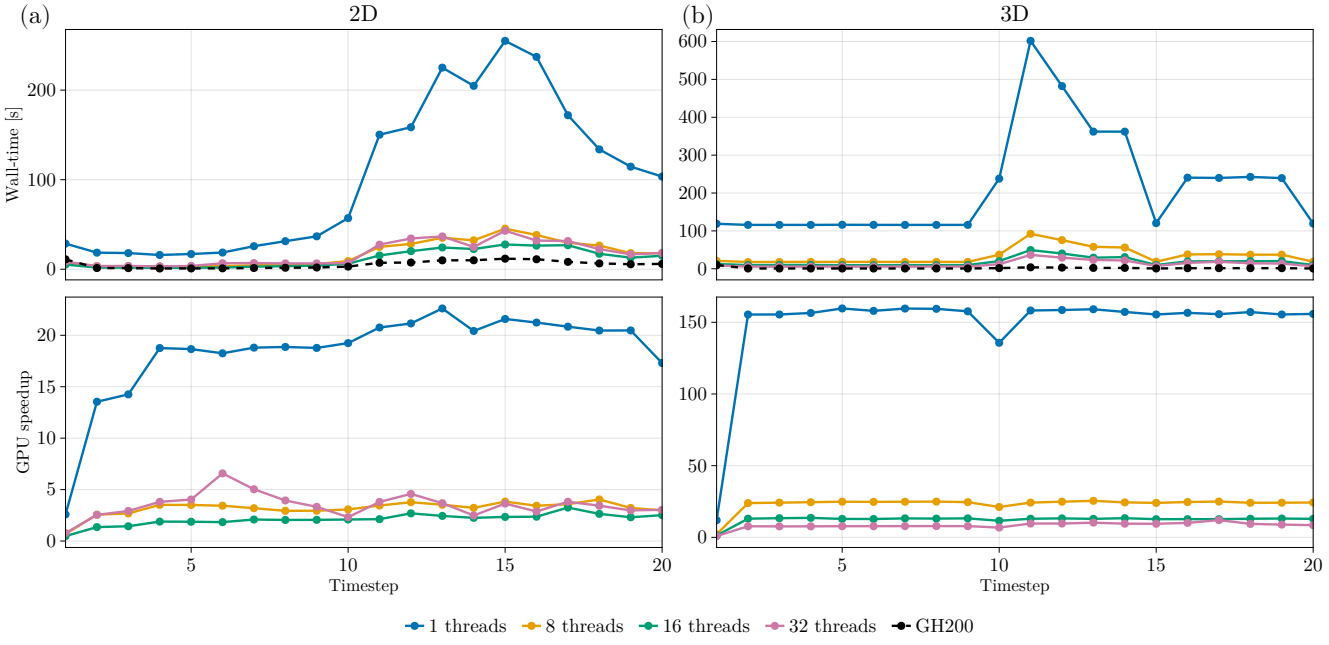


Figure 4. Comparison of CPU and GPU wall times for different thread counts. Panel (a) corresponds to the 2D shear-band localization example, and panel (b) shows timings for the same setup in 3D. The top-row plots show wall time per time step; the bottom-row plots show GPU speedup relative to runs executed with different numbers of CPU threads.

executed on both backends, using CPU multithreading or a GPU, respectively. The wall times reported in Fig. 4 correspond to the per-time-step Stokes solve after warm-up, excluding compilation and initialization overheads.

Figure 4 also shows the speedup provided by GPU execution. Peak speedups are $\times 22$ (2D) and $\times 158$ (3D) relative to single-thread CPU runs, and $\times 3$ (2D) and $\times 9$ (3D) relative to 32-thread CPU runs. These numbers may vary depending on the hardware being employed. Here, we used NVIDIA GH200 Grace Hopper superchips with theoretical CPU and GPU peak bandwidths of up to 500 GB/s and 4.0 TB/s, respectively, as reported by the vendor's specifications.

5.2 Scalability

Scalability is assessed through strong and weak scaling tests in 2D and 3D. In weak scaling tests, the local problem size per GPU is kept constant while the global problem size increases with the number of GPUs; in strong scaling tests, the global problem size is fixed while the local problem size per GPU decreases as additional GPUs are used. Note that the APT time-to-solution depends on model resolution. To isolate the scaling of the kernels that comprise the Stokes solver, we therefore fix the number of pseudo-transient iterations to 100, independent of resolution. For these tests, we used the APT method of Räss et al. (2022). The 2D models consist of a visco-elastic domain under pure-shear velocity boundary conditions with a weak

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

visco-elastic circular inclusion at its centre. For the 3D models, the domain contains denser visco-elastic spherical inclusions; free-slip velocity boundary conditions are prescribed on all boundaries, such that the flow is purely buoyancy driven.

The benchmarks were performed on the Alps supercomputer at the Swiss National Supercomputing Centre (CSCS) in Lugano, Switzerland. Each computational node is equipped with four NVIDIA GH200 Grace Hopper superchips. Performance is reported as the time per pseudo-transient iteration of the Stokes solver, t_{iter} (s). Wall-clock times are normalized by the corresponding single-GPU performance and do not include compilation, the warm-up phase, or I/O operations.

5.2.1 Strong scaling

On modern GPUs, strong scaling efficiency typically decreases as the number of GPUs increases because the local workload per GPU becomes smaller, communication costs become relatively more important, and small kernels tend to become launch-bound. Strong scaling tests were performed using fixed global domain sizes of 8192^2 cells in 2D and 400^3 cells in 3D. These resolutions were selected based on available GPU memory, as the simulations additionally require the allocation of billions of particles and associated auxiliary arrays. With each grid-coarsening step, the local resolution is reduced by 50 %, reaching minimum local resolutions of 256^2 and 50^3 cells in 2D and 3D, respectively.

The results show near-linear speedup up to two levels of grid coarsening, corresponding to 64 GPUs and local resolutions of 1024^2 and 100^3 cells in 2D and 3D, respectively. Beyond this point, scaling efficiency decreases as the local problem size becomes too small to adequately utilize GPU resources (Fig. 5).

5.2.2 Weak scaling

Weak scaling tests were performed with fixed local resolutions of 8192^2 cells in 2D and 256^3 cells in 3D. These resolutions were chosen to ensure sufficient workload per GPU and to avoid under-utilization of accelerator hardware, which would otherwise lead to artificially reduced weak-scaling efficiency. Although these resolutions exceed those commonly required for many geodynamic applications, they provide a meaningful assessment of the maximum computational capability of the solver.

The weak scaling results (Fig. 5) show high parallel efficiency for both the 2D and 3D Stokes solvers. Parallel efficiency remains above 82 % for up to 256 GPUs in 2D and above 94 % for up to 512 GPUs in 3D. The small variation in efficiency between 64 and 256 GPUs may reflect differences in communication overhead, node selection, and likely garbage collection.

6 Examples of geodynamic applications

This section presents examples of geodynamic problems that demonstrate the applicability of JustRelax.jl. The examples vary in physical complexity and span 2D and 3D, ranging from mantle convection and plume ascent to visco-elasto-viscoplastic shear bands and nonlinear rheology in magmatic systems. All models presented below were run on one or multiple NVIDIA GH200 Grace Hopper superchips at the CSCS supercomputing centre.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

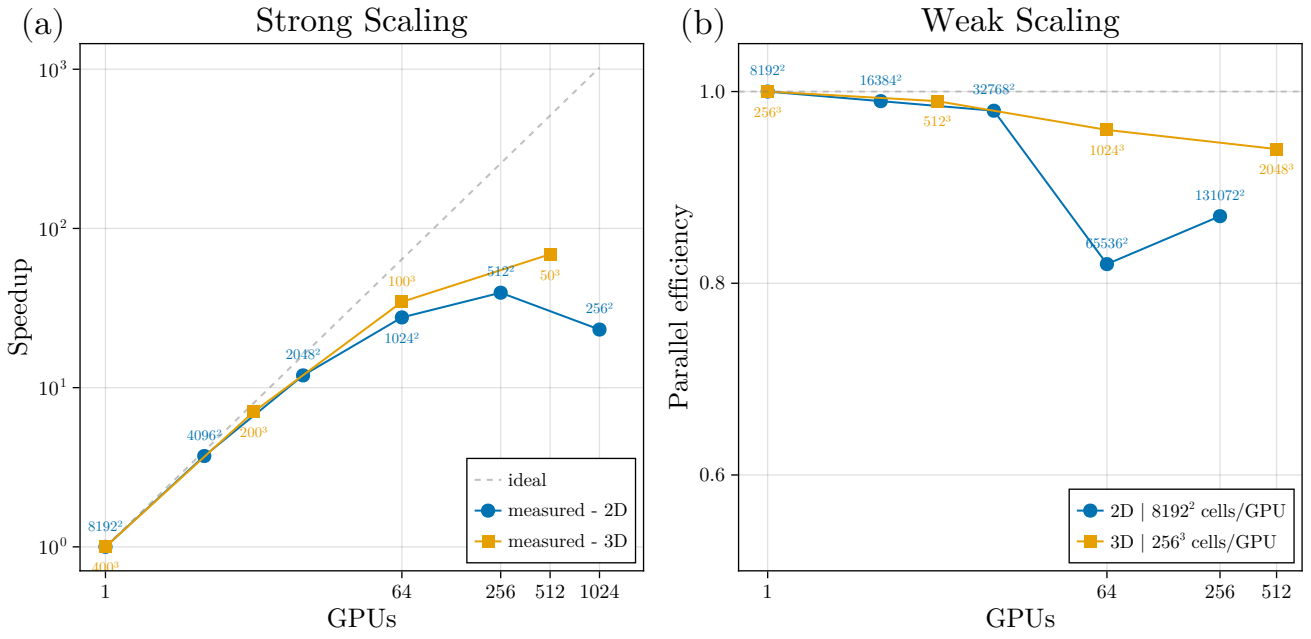


Figure 5. Scaling performance of the Stokes solver in 2D and 3D. (a) Strong scaling speedup for fixed global grid resolutions of 8192^2 (2D) and 400^3 (3D). The local resolution decreases by 50 % with each grid-coarsening step, reaching minimum resolutions of 256^2 (2D) and 50^3 (3D). (b) Weak scaling parallel efficiency for fixed local grid resolutions of 8192^2 (2D) and 256^3 (3D).

6.1 Shear band localization

Accurate representation of brittle deformation is essential for simulating geodynamic processes, particularly those involving lithospheric deformation. We assess the plastic-failure implementation described in Sect. 3.3.3 by reproducing shear-band localization benchmarks from previous studies (e.g., Duretz et al., 2018, 2021).

The model consists of a circular inclusion of radius $r = 0.1$ embedded at the centre of a stronger matrix within the domain $\Omega \in [0,1] \times [0,1]$. Both materials are described by a visco-elasto-viscoplastic rheology and share the same mechanical parameters: viscosity $\eta = 1$, cohesion $C = 1.6$, friction angle $\phi = 30°$, and regularized plastic viscosity $\eta^{\mathrm{reg}} = 8 \times 10^{-3}$. The only difference between the two materials is the shear modulus, with $G^{\mathrm{m}} = 1$ for the matrix and $G^{\mathrm{inc}} = G^{\mathrm{m}}/2$ for the inclusion. Pure-shear velocity boundary conditions are applied on the domain boundaries with a background strain rate $\dot{\varepsilon}_{\mathrm{bg}} = 1$.

The resulting accumulated plastic strain and shear-band localization are shown in Fig. 6. The simulation was performed on 64 GPUs, achieving a global grid resolution of 40946×40946 cells (more than five billion degrees of freedom when considering the primary unknown fields $\boldsymbol{v}_x$, $\boldsymbol{v}_y$, and P). The grid is partitioned equally across all GPUs, with halo widths of two cells. If the non-dimensional domain is interpreted as a $1\,\mathrm{m} \times 1\,\mathrm{m}$ box, this corresponds to a spatial resolution of approximately $24\,\mu\mathrm{m}$;

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

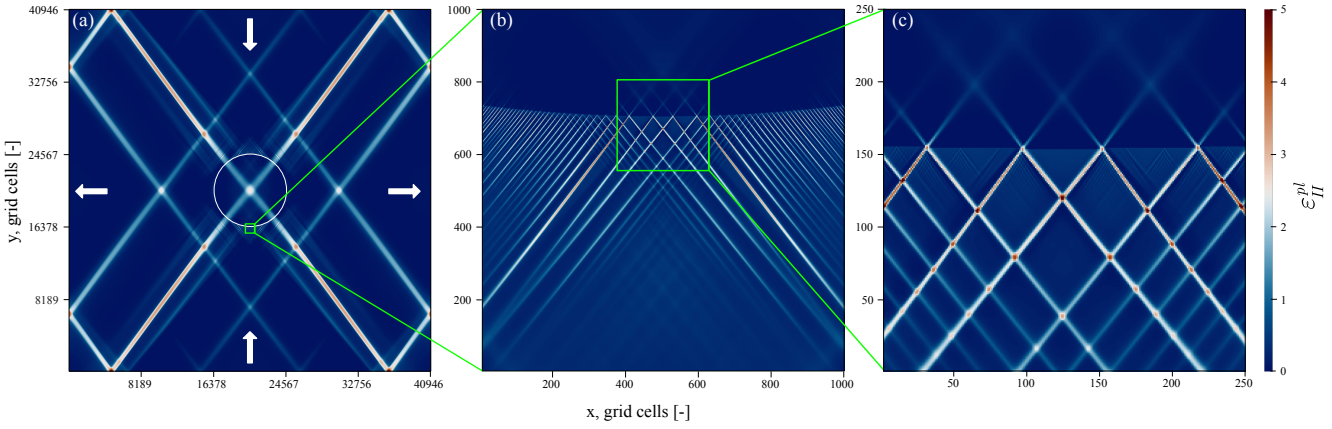


Figure 6. Accumulated plastic strain ε_{II}^{pl} during shear-band development in a strong matrix containing a weaker inclusion (white circle). The non-dimensional model domain is $\Omega \in [0,1] \times [0,1]$ with a grid resolution of 40946×40946 cells. Panels show progressively increasing levels of detail, highlighting localization near the matrix–inclusion interface.

interpreting the domain as a 40 km-thick crustal section yields metre-scale resolution. The wall-clock time was 20 h for 20 time steps. This timing includes compilation, initialization, the Stokes solve, and I/O.

6.2 Subduction of oceanic lithosphere

Subduction processes are fundamental to understanding lithospheric-scale deformation and global plate tectonics (e.g., Gerya, 2011; Bessat et al., 2020; Schuler et al., 2025). Here, we investigate subduction dynamics using a 2D visco-elasto-plastic model of a convergent subduction zone. The model domain is $\Omega \in [0, 4000] \times [-660, 30]$ km, where the upper 30 km is a sticky-air layer that approximates a nearly stress-free surface (Crameri et al., 2012). The numerical resolution is 1024×192, yielding an effective grid spacing of approximately 5 km in both dimensions; this model was run on a single GPU.

Figure 7a shows the initial temperature distribution, which was created with the Julia package GEOPHYSICALMODELGENERATOR.JL (Kaus et al., 2024). The oceanic lithosphere is located on the left and gradually thins toward the left boundary, while the continental lithosphere occupies the right side of the model. An initial weak zone separates the oceanic and continental lithosphere and facilitates subduction initiation. Material parameters are summarized in Table C1. Free-slip boundary conditions are prescribed along all model boundaries.

The oceanic lithosphere subducts beneath the continental lithosphere (Fig. 7). The subduction velocity increases until about 4 Myr, reaches a maximum of approximately $5\,\mathrm{cm\,yr^{-1}}$, and subsequently decreases as the slab approaches the lower model boundary. The stress distribution (Fig. 7c,f,i) shows a pronounced stress concentration within the slab. The viscosity shown in Fig. 7d,g,j corresponds to the diagnostic visco-elasto-plastic viscosity,

$$\eta_{\mathrm{vep}} = \frac{\tau_{\mathrm{II}}}{2\dot{\varepsilon}_{\mathrm{II}}}. \tag{24}$$

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

which exceeds $10^{24}\,\mathrm{Pa\,s}$ in the lithosphere but decreases to around $10^{19}\,\mathrm{Pa\,s}$ in the subjacent mantle beneath the oceanic lithosphere.

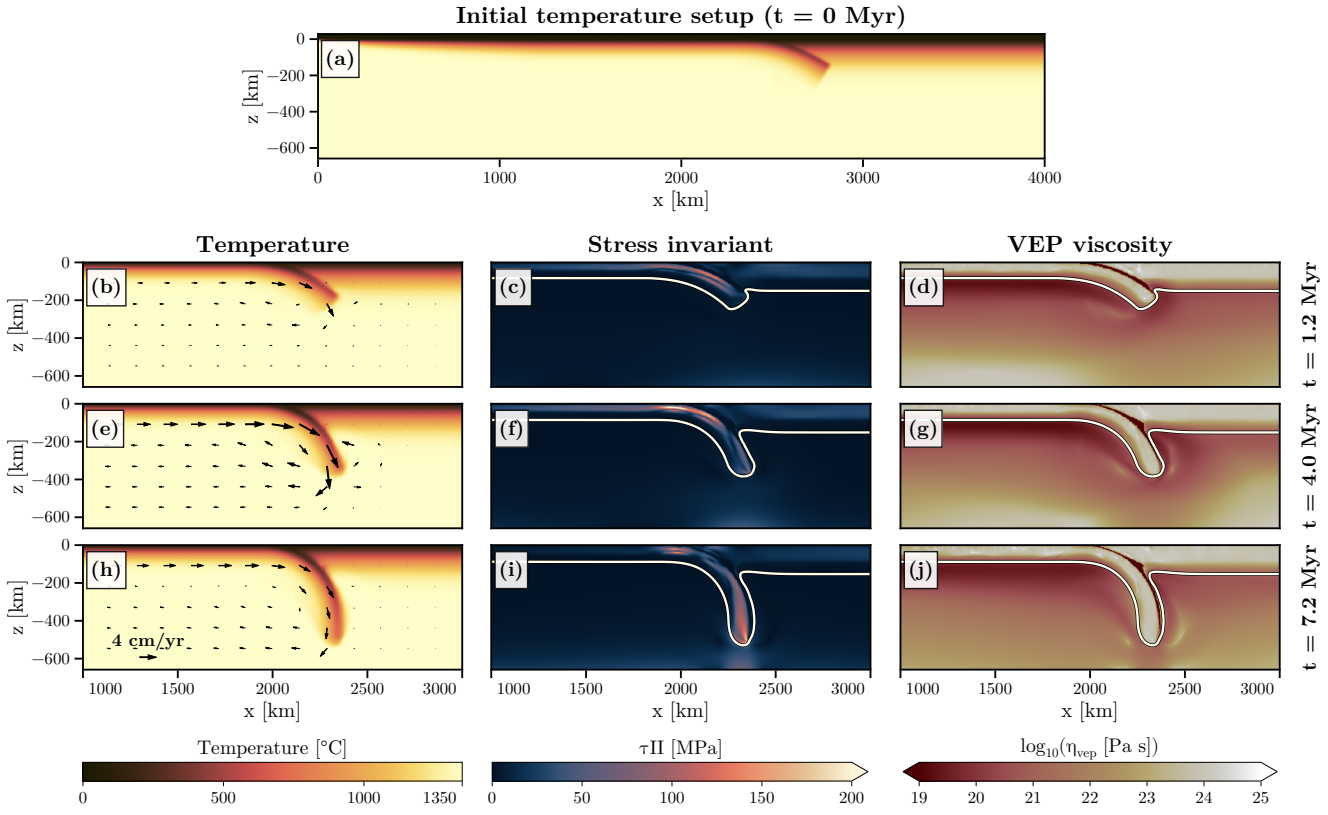


Figure 7. Temporal evolution of the 2D subduction model. Panel (a) shows the initial temperature distribution across the complete model domain. The subsequent rows show the model at 1.2, 4, and 7.2 Myr over the region from 1000 to 3000 km in the x-direction. The columns display temperature and velocity vectors (b, e, h), the second invariant of deviatoric stress, τ_{II} (c, f, i), and visco-elasto-plastic viscosity, η_{vep} (d, g, j). The outlined curve marks the 1200 °C isotherm and highlights the evolving slab geometry. Color scales are shared between timesteps, and the reference arrow in panel (h) represents a velocity of $4\,\mathrm{cm\,yr^{-1}}$.

6.3 Magmatic sill convection

Lithospheric-scale magmatic systems exhibit complex rheology and thermodynamics. Their multi-phase nature—here, solid (crystals), fluid (melt), and gas (volatiles)—depends on fractionation history, magma source, and pressure–temperature conditions in the lithosphere.

Although JustRelax.jl solves the mass and momentum conservation equations for a single phase, the multi-phase character of magma reservoirs can be incorporated through melt-dependent viscosity (Eqs. B1–B4; Giordano et al., 2008; Costa et al., 2009), density (Iacovino and Till, 2019; Huber et al., 2010) (Eqs. 25 and B10), and specific heat capacity C_p (including

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

latent heat of crystallization), together with parameterizations of C_p and thermal conductivity k from Whittington et al. (2009) available in GeoParams.jl. This enables magma convection and related processes to be modelled without changes to the code design.

In this example, sill convection is simulated for a high-silica rhyolitic magma ($75.75\,\mathrm{wt}\,\%$ SiO_2) emplaced within a low-SiO_2 host rock ($70.78\,\mathrm{wt}\,\%$) (see Table B1). These compositions are used for viscosity and density calculations and correspond to the Pre-Kilgore and Kilgore Tuff at the Heise volcanic field (USA), which hosted multiple caldera-forming eruptions (Watts et al., 2011). The model domain contains a 10 m sill ($\approx 950\,^\circ\mathrm{C}$) embedded within 1.25 m of host rock above and below. Melt fraction is obtained from precomputed MAGEMin (Riel et al., 2022) phase diagrams for each composition and captures the steep decrease in melt fraction for felsic magmas as temperature approaches the eutectic. For numerical stability, a smoothed polynomial is used to reduce the step-like decrease towards the eutectic.

Magma density follows the three-phase mixture of Degruyter and Huber (2014):

$$\rho(P,T,\boldsymbol{X}_{\mathrm{H_2O}}) = \phi_m\rho_m + \phi_g\rho_g + \phi_x\rho_x, \tag{25}$$

where the melt density ρ_m depends on composition, dissolved H_2O, P, and T (Iacovino and Till, 2019); the gas density ρ_g follows a modified Redlich–Kwong equation of state for H_2O (Huber et al., 2010); and the crystal density ρ_x uses the P–T parameterization in Eq. (3). Gas and crystal volume fractions follow from H_2O solubility (Liu et al., 2005) and melt fraction ϕ while enforcing the saturation constraint $\phi_m + \phi_g + \phi_x = 1$.

Finally, temperature, composition, and dissolved and exsolved H_2O are advected using WENO-5 to minimize numerical diffusion that would otherwise smear the thermal and compositional structure of rising plumes during convection. The model represents a pre-eruptive sill, capturing the progressive increase in viscosity and the evolution of magmatic volatiles during crystallization until the melt fraction decreases to 40 %, at which point the magma is considered to have entered the mush regime.

6.4 Ascent of a 3D thermal and compositional anomaly

The model domain is a 3D box with $\Omega \in [0,700]\times[0,700]\times[-700,0]\,\mathrm{km}$. Free-slip velocity boundary conditions are prescribed on all boundaries. Temperature is fixed at the top and bottom boundaries, $T(z=0\,\mathrm{km}) = 0\,^\circ\mathrm{C}$ and $T(z=-700\,\mathrm{km}) = 1563.85\,^\circ\mathrm{C}$, and zero lateral heat flux is imposed on the side boundaries. The domain is rheologically layered into upper crust, lower crust, lithospheric mantle, and mantle, with thicknesses of 21 km, 14 km, 55 km, and 610 km, respectively.

The initial temperature field is prescribed as a piecewise-linear geotherm. Temperature increases from $0\,^\circ\mathrm{C}$ at the surface to $650\,^\circ\mathrm{C}$ at 35 km depth ($\partial T/\partial z = 18.6\,^\circ\mathrm{C\,km^{-1}}$), then to $1219\,^\circ\mathrm{C}$ at 110 km ($\partial T/\partial z = 7.59\,^\circ\mathrm{C\,km^{-1}}$), and finally to $1564\,^\circ\mathrm{C}$ at the base of the domain ($\partial T/\partial z = 0.585\,^\circ\mathrm{C\,km^{-1}}$). The steep upper gradients represent the conductive crustal and lithospheric-mantle geotherms, whereas the weak gradient below 110 km approximates a near-adiabatic sublithospheric mantle.

To trigger buoyant ascent, we superimpose a hot, compositionally buoyant anomaly within the cube $\Omega_{\mathrm{anomaly}} \in [300,400]\times[300,400]\times[-660,-560]$ km. Within this cube, temperature increases linearly from $1729\,^\circ\mathrm{C}$ (top) to $1789\,^\circ\mathrm{C}$ (bottom), cor-

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

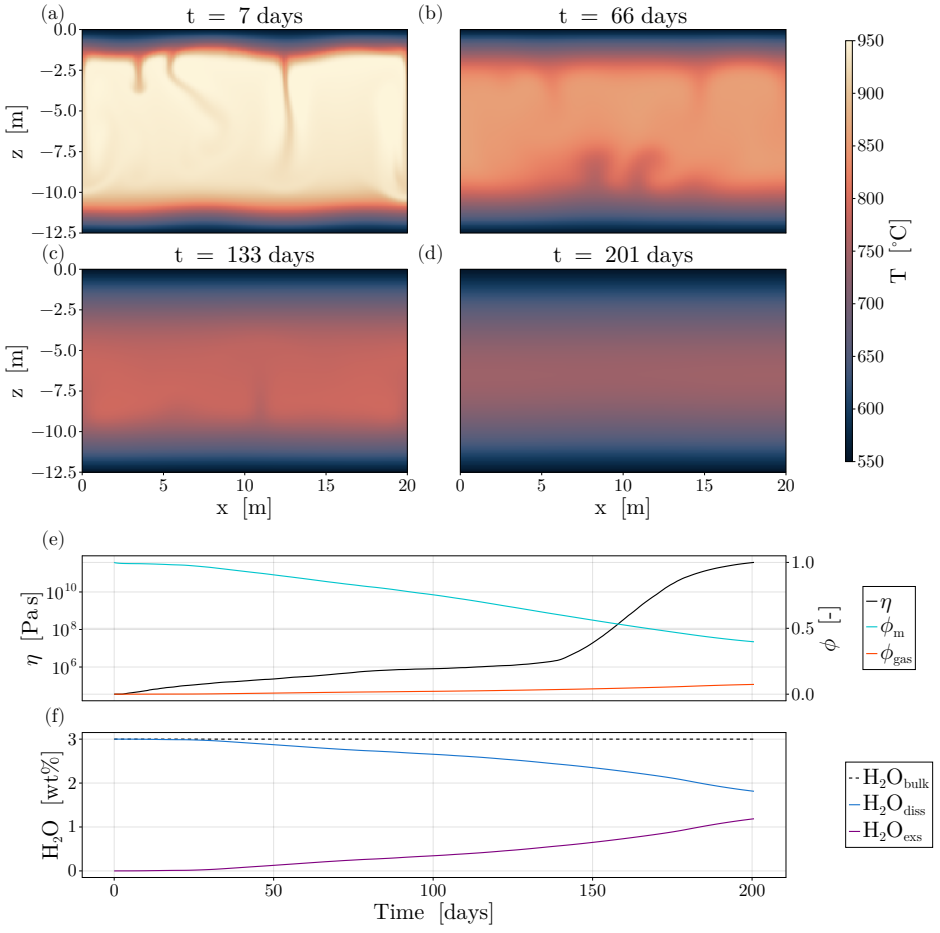


Figure 8. Magmatic sill convection of a 10 m high-silica magma (75.75 wt % SiO_2) emplaced within a low-SiO_2 host rock (70.78 wt %). (a)–(d) Thermal evolution, with high-density, high-crystallinity downwellings and plumes forming. (e) Evolution of magma viscosity and the melt and gas fractions ($\phi_{m,gas}$). (f) Temporal evolution of H_2O in the magma, showing the increase in exsolved H_2O with the corresponding decrease of dissolved H_2O over the course of the simulation. The increase in exsolved H_2O is linked to cooling at the sill boundaries, where the exsolution threshold is reached.

responding to a uniform excess of $\Delta T \approx 247\,^{\circ}\mathrm{C}$ relative to the background geotherm at the same depth. We additionally prescribe a density contrast of $\Delta\rho = -50\,\mathrm{kg\,m^{-3}}$ relative to the ambient mantle, so that the driving buoyancy is both thermal and compositional. Thermal conductivity follows the temperature- and pressure-dependent parameterization of Gerya (2019) $k = (a_k + b_k\,(T + c_k))\,(1 + d_k P)$, where a_k, b_k, c_k, and d_k are material parameters. All layers are modelled as visco-elasto-viscoplastic with a non-Newtonian viscosity; material parameters are listed in Table C2.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Fig. 9 summarizes the model setup and illustrates the evolution of this model. The goal of this example is not to reproduce a specific geodynamic scenario, but to demonstrate the applicability of JUSTRELAX.JL to 3D thermo-mechanical problems with a complex, nonlinear rheology.

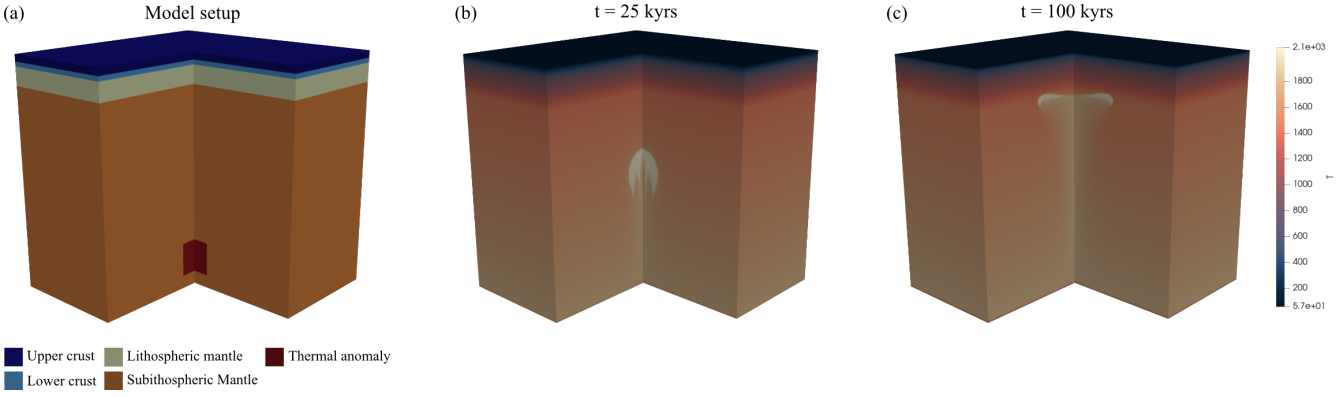


Figure 9. (a) Model configuration. Snapshots of the model evolution at (b) 25 kyr and (c) 100 kyr. The model is discretized on a regular grid of 256^3 cells. Details of the geometry and material properties are provided in Sect. 6.4.

7 Discussion

JUSTRELAX.JL is intended to complement well-established geodynamics codes, e.g. ASPECT (Bangerth et al., 2023) and LaMEM (Kaus et al., 2016), that have benefited from decades of development and community contributions. Many widely used solvers were designed primarily for CPU architectures and matrix-based linear algebra, e.g. I3ELVIS (Gerya and Yuen, 2007) and CITCOM-S (Zhong et al., 2000), and porting such mature code bases efficiently to heterogeneous multi-xPU — where by *xPU* we refer to code that executes on one CPU or GPU, and *multi-xPU* refers to distributed execution across many such devices via MPI communication—systems can be challenging and often requires substantial redevelopment. Even where matrix-free components exist, most production codes remain predominantly CPU-oriented in their current state.

JUSTRELAX.JL targets this emerging hardware landscape by providing a portable toolkit across hardware (CPU/GPU) and parallel paradigms (multithreading, MPI, and GPU-aware MPI) implemented in native Julia. Its core is a matrix-free Accelerated Pseudo-Transient (APT) solver designed to scale on multi-xPU architectures. Julia's code-generation capabilities are used to produce backend-agnostic kernels, reducing code redundancy while supporting common devices in both workstation and HPC environments. JUSTRELAX.JL is comparatively young and currently lags behind long-standing community codes in terms of user base and number of available features (e.g., adaptive mesh refinement, petrology, spherical coordinates), which will be addressed in future releases.

Another key difference from many existing geodynamic codes is that JUSTRELAX.JL and its core dependencies are designed to be compatible with automatic differentiation (AD) tools in the Julia ecosystem. This enables differentiation through the Stokes solve to compute adjoint sensitivities for gradient-based inversions, e.g. using reverse-mode AD with ENZYME.JL

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

(Schuler, 2025). It also enables the local use of AD to construct consistent tangent operators for nonlinear rheologies, among other possibilities, without requiring handwritten derivatives or Jacobians.

From a numerical perspective, the matrix-free APT formulation is xPU-friendly, but it inherits the convergence characteristics of iterative relaxation methods. The number of pseudo-transient iterations typically increases with spatial resolution; for representative benchmark problems, the iteration count has been shown to grow approximately linearly with resolution (Räss et al., 2022), thereby increasing the time to solution at very high resolutions.

Convergence can be further hindered by sharp contrasts in material properties (particularly viscosity), nearly incompressible materials, and strongly nonlinear rheologies. The auto-tuning APT strategy of Duretz et al. (2026) substantially mitigates iteration-scaling limitations by dynamically adjusting pseudo-transient parameters; nevertheless, 3D simulations can still require thousands of pseudo-transient iterations. This reflects a trade-off between the scalability and hardware portability of matrix-free methods and the faster convergence achievable with more advanced preconditioned solvers.

Many geodynamic applications require an evolving free surface to capture the coupling between deep-seated deformation and surface processes. Benchmark studies have shown that including a free surface can substantially influence model evolution and is therefore important for reproducing a wide range of tectonic processes (Schmeling et al., 2008). In finite-difference staggered-grid models, a common workaround is the *sticky-air* approach, in which a low-viscosity, low-density layer is placed above the lithosphere (Crameri et al., 2012; Kaus et al., 2010; Duretz et al., 2011). While straightforward to implement, sticky air introduces a sharp viscosity contrast at the surface that can significantly degrade the convergence of iterative Stokes solvers. This degradation is particularly evident for the APT Stokes solver based in Räss et al. (2022), to the point of potentially stalling the Stokes solver. Fortunately, the improved choice of damping coefficients from Duretz et al. (2021) significantly helps to improve convergence rates for the pseudo-transient Stokes solver. More sophisticated approaches to tackle the free-surface in FD staggered grids have been proposed based on eliminating degrees of freedom (Duretz et al., 2016; Larionov et al., 2017), which could help improve the convergence rates of APT solvers by avoiding an artificial low-viscosity layer.

Future efforts should focus on assessing the viability and efficiency of alternative matrix-free iterative solvers on cutting-edge GPU architectures. One promising avenue is to investigate coupling geometric multigrid (GMG) techniques with the APT. While multigrid approaches lower iteration counts by effectively damping errors over a range of spatial scales, APT is based on single-level relaxation. Developing and evaluating a multilevel APT–GMG approach is thus a sensible direction for future research, but it lies outside the scope of the present work.

From a software-design perspective, these choices come with trade-offs. The modular architecture and composable API expose numerical building blocks directly, making it straightforward to implement and combine new physical parameterizations, constitutive models, and solver strategies, either within JustRelax.jl itself or in external packages that follow established interfaces. For example, the complex equation of state and magma-dependent viscosity used in Sect. 6.3, as well as alternative plastic formulations, are implemented in GeoParams.jl and typically require few (or no) modifications to the internals of JustRelax.jl. In contrast, adding comparable capabilities to many community codes often requires detailed knowledge of their internal structure.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

This differs from many established codes, where simulations are primarily configured through parameter files and predefined interfaces, while the underlying numerical algorithms are less directly exposed, e.g. LaMEM (Kaus et al., 2016) and ASPECT (Bangerth et al., 2023). As a result, JUSTRELAX.JL shifts some responsibility to the user: simulations are assembled by composing components or adapting examples (Sect. 6), which is flexible but requires basic programming skills and careful verification.

A further consequence of modularity is reliance on external Julia packages that evolve independently. While this encourages code reuse, upstream updates can introduce incompatibilities or require patch updates to JUSTRELAX.JL. Continuous integration (CI) and comprehensive test suites help identify such issues, but maintaining compatibility remains an ongoing effort.

8 Conclusions

The increasing availability of high-performance GPUs, from workstations to large-scale HPC systems, motivates computational strategies for geodynamic modelling that are both hardware portable and scalable.

This study introduced JUSTRELAX.JL, a modular Julia framework for thermo-mechanical geodynamic simulations built around a matrix-free accelerated pseudo-transient (APT) solver. Our results show that separating material physics, particle-in-cell advection, and (distributed) parallelism into specialized components can preserve performance while enabling simulations that scale from a single device to several hundred GPUs.

At the same time, JUSTRELAX.JL introduces a practical workflow where the user can prototype and debug on CPUs — which are available on any laptop and workstation— and execute the same code on GPUs at later stages, typically requiring only minor code changes in the worst case scenario.

We demonstrated applicability across representative geodynamic problems spanning a wide range of spatial and computational scales, including visco-elasto-viscoplastic shear-band localization, subduction, mantle-plume dynamics, and magmatic sill convection. Together, these examples illustrate how a common software infrastructure can support diverse thermo-mechanical scenarios while remaining extensible.

The current implementation provides a portable and scalable foundation, but further numerical and physical developments are needed to catch up with long-lived community codes. Priorities include reducing time to solution (especially for strongly nonlinear rheologies and large viscosity contrasts), supporting full differentiability through Julia's AD ecosystem to enable, for example, sensitivity analysis, and extending the framework towards multi-phase processes. Overall, JUSTRELAX.JL is intended as an open and extensible platform for developing and sharing GPU-ready geodynamic models on modern multi-xPU architectures.

Code availability. JUSTRELAX.JL is open source under the MIT licence. Version v0.8.0, the version described in this paper, is archived on Zenodo—https://doi.org/10.5281/zenodo.22123264— (de Montserrat et al., 2026d). The complete source and input-script bundle prepared

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

for this GMD submission is archived separately in (de Montserrat et al., 2026b). The project webpage, source repository, documentation, and installation instructions are available at https://github.com/PTsolvers/JustRelax.jl. The software requires at least Julia 1.9, and runs on multicore CPUs. GPU execution additionally requires a supported NVIDIA or AMD GPU and the corresponding CUDA.JL or AMDGPU.JL backend; distributed runs require an MPI-capable system.

Data availability. No third-party or observational data were used in this study; all results presented here are model output generated with JUSTRELAX.JL. The input scripts defining the model setups, boundary conditions and material parameters for the benchmarks and application examples are distributed in **??**, in the `GMD_files` directory of the archived release repository, and are covered by the same MIT licence; the figures can be reproduced by running them. Reference solutions for the benchmarks in Appendix A are taken from the publications cited there.

Author contributions. AdM and PSA contributed equally to this work. They jointly conceptualized the study, developed the methodology, implemented the software, and performed validation. AdM and PSA prepared the original draft of the manuscript, and both authors contributed to the review and editing of the manuscript. TD contributed to the methodology and reviewed and edited the manuscript. CS contributed to software development and methodology and reviewed and edited the manuscript. LR contributed to the methodology, software development, manuscript review and editing, and funding acquisition. BJPK contributed to manuscript review and editing, funding acquisition, and supervision of PSA. All authors discussed the results and contributed to the final manuscript.

Competing interests. Two of the (co-)authors are members of the editorial board of *Geoscientific Model Development*.

Acknowledgements. The authors acknowledge the Swiss National Supercomputing Centre (CSCS) for the computing resources provided under project IDs c23 and c44, awarded via the PASC projects GPU4GEO and ∂GPU4GEO. Generative AI tools were used to improve the language and readability of the manuscript. The authors reviewed and edited all resulting text and take full responsibility for the content of the publication.

Financial support. AdM, CS, and LR were supported by the Swiss University Conference and the Swiss Council of Federal Institutes of Technology through the Platform for Advanced Scientific Computing (PASC) program. PSA and BJPK gratefully acknowledge funding from the European Research Council through the MAGMA project (ERC Consolidator Grant #771143).

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

References

Aellig, P. S., de Montserrat, A., and Kaus, B. J. P.: Thermo-mechanical controls on central-vent caldera collapse, Journal of Geophysical Research: Solid Earth, https://doi.org/10.1029/2026JB034434, accepted, 2026.

Alkhimenkov, Y. and Podladchikov, Y. Y.: Accelerated pseudo-transient method for elastic, viscoelastic, and coupled hydromechanical problems with applications, Geoscientific Model Development, 18, 563–583, https://doi.org/10.5194/gmd-18-563-2025, 2025.

Alkhimenkov, Y., Khakimova, L., Utkin, I., and Podladchikov, Y.: Resolving strain localization in frictional and time-dependent plasticity: Two-and three-dimensional numerical modeling study using graphical processing units (GPUs), Journal of Geophysical Research: Solid Earth, 129, e2023JB028 566, https://doi.org/10.1029/2023JB028566, 2024.

Arndt, D., Bangerth, W., Davydov, D., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Pelteret, J.-P., Turcksin, B., and Wells, D.: The deal.II finite element library: Design, features, and insights, Computers & Mathematics with Applications, 81, 407–422, https://doi.org/10.1016/j.camwa.2020.02.022, 2021.

Balay, S., Abhyankar, S., Adams, M., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W., Hapla, V., Isaac, T., Jolivet, P., Karpeyev, D., Kaushik, D., Knepley, M., Kong, F., Kruger, S., May, D., McInnes, L. C., Mills, R., Mitchell, L., Munson, T., Roman, J., Rupp, K., Sanan, P., Sarich, J., Smith, B., Suh, H., Zampini, S., Zhang, H., Zhang, H., and Zhang, J.: PETSc/TAO Users Manual, Tech. Rep. ANL-21/39, Argonne National Laboratory, https://doi.org/10.2172/2998643, revision 3.25, 2026.

Bangerth, W., Dannberg, J., Fraters, M., Gassmoeller, R., Glerum, A., Heister, T., Myhill, R., and Naliboff, J.: ASPECT v2.5.0, https://doi.org/10.5281/zenodo.8200213, [code], 2023.

Besard, T., Foket, C., and De Sutter, B.: Effective Extensible Programming: Unleashing Julia on GPUs, IEEE Transactions on Parallel and Distributed Systems, https://doi.org/10.1109/TPDS.2018.2872064, 2018.

Bessat, A., Duretz, T., Hetényi, G., Pilet, S., and Schmalholz, S. M.: Stress and deformation mechanisms at a subduction zone: insights from 2-D thermomechanical numerical modelling, Geophysical Journal International, 221, 1605–1625, https://doi.org/10.1093/gji/ggaa092, 2020.

Beuchert, M. J. and Podladchikov, Y. Y.: Viscoelastic mantle convection and lithospheric stresses, geophysical Journal international, 183, 35–63, https://doi.org/10.1111/j.1365-246X.2010.04708.x, 2010.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B.: Julia: A fresh approach to numerical computing, SIAM Review, 59, 65–98, https://doi.org/10.1137/141000671, 2017.

Bird, R. B., Curtiss, C. F., Armstrong, R. C., and Hassager, O.: Dynamics of Polymeric Liquids, Volume 2: Kinetic Theory, John Wiley & Sons, ISBN 978-0471802440, https://www.wiley.com/en-us/Dynamics+of+Polymeric+Liquids%2C+Volume+2%3A+Kinetic+Theory%2C+2nd+Edition-p-9780471802440, 1987.

Blankenbach, B., Busse, F., Christensen, U., Cserepes, L., Gunkel, D., Hansen, U., Harder, H., Jarvis, G., Koch, M., Marquart, G., Moore, D., Olson, P., Schmeling, H., and Schnaubelt, T.: A benchmark comparison for mantle convection codes, Geophysical Journal International, 98, 23–38, https://doi.org/10.1111/j.1365-246X.1989.tb05511.x, 1989.

Byrne, S., Wilcox, L. C., and Churavy, V.: MPI.jl: Julia bindings for the Message Passing Interface, Proceedings of the JuliaCon Conferences, 1, 68, https://doi.org/10.21105/jcon.00068, 2021.

Costa, A., Caricchi, L., and Bagdassarov, N.: A model for the rheology of particle-bearing suspensions and partially molten rocks, Geochemistry, Geophysics, Geosystems, 10, https://doi.org/10.1029/2008GC002138, 2009.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.



Crameri, F., Schmeling, H., Golabek, G. J., Duretz, T., Orendt, R., Buiter, S. J. H., May, D. A., Kaus, B. J. P., Gerya, T. V., and Tackley, P. J.: A comparison of numerical surface topography calculations in geodynamic modelling: an evaluation of the 'sticky air' method, Geophysical Journal International, 189, 38–54, https://doi.org/10.1111/j.1365-246X.2012.05388.x, 2012.

Dabrowski, M., Krotkiewski, M., and Schmid, D. W.: MILAMIN: MATLAB-based finite element method solver for large problems, Geochemistry, Geophysics, Geosystems, 9, https://doi.org/10.1029/2007GC001719, 2008.

Danisch, S. and Krumbiegel, J.: Makie.jl: Flexible high-performance data visualization for Julia, Journal of Open Source Software, 6, 3349, https://doi.org/10.21105/joss.03349, 2021.

de Montserrat, A., Morgan, J. P., and Hasenclever, J.: LaCoDe: A Lagrangian two-dimensional thermo-mechanical code for large-strain compressible visco-elastic geodynamical modeling, Tectonophysics, 767, 228 173, https://doi.org/10.1016/j.tecto.2019.228173, 2019.

de Montserrat, A., Aellig, P. S., Räss, L., and Duretz, T.: JustPIC.jl: Particle-in-Cell advection ready to rock the GPU, https://doi.org/10.5281/zenodo.21104534, [code], 2026a.

de Montserrat, A., Aellig, P. S., Räss, L. G., Schuler, C., Duretz, T., and Kaus, B. J.: JustRelax_GMD_v0.8.0, https://doi.org/10.5281/zenodo.22300683, 2026b.

de Montserrat, A., Aellig, P. S., Schuler, C., Navarrete, I., Räss, L., Fuchs, L., Kaus, B. J. P., and Dominguez, H.: JustRelax.jl: A Julia package for geodynamic modeling with matrix-free solvers, Journal of Open Source Software, 11, 9365, https://doi.org/10.21105/JOSS.09365, 2026c.

de Montserrat, A., Aellig, P. S., Schuler, C., Navarrete Jara, I., Räss, L., Fuchs, L., Kaus, B. J., and Dominguez, H.: JustRelax.jl: A Julia package for geodynamic modeling with matrix-free solvers, https://doi.org/10.5281/zenodo.22123264, [code], 2026d.

Degruyter, W. and Huber, C.: A model for eruption frequency of upper crustal silicic magma chambers, Earth and Planetary Science Letters, 403, 117–130, https://doi.org/10.1016/j.epsl.2014.06.047, 2014.

Dominguez, H., Riel, N., and Lanari, P.: Modelling chemical advection during magma ascent, Geoscientific Model Development, 17, 6105–6122, https://doi.org/10.5194/gmd-17-6105-2024, 2024.

Duretz, T., May, D. A., Gerya, T., and Tackley, P.: Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: A numerical study, Geochemistry, Geophysics, Geosystems, 12, https://doi.org/10.1029/2011GC003567, 2011.

Duretz, T., May, D., and Yamato, P.: A free surface capturing discretization for the staggered grid finite difference scheme, Geophysical Journal International, 204, 1518–1530, https://doi.org/10.1093/gji/ggv526, 2016.

Duretz, T., Souche, A., de Borst, R., and Le Pourhiet, L.: The Benefits of Using a Consistent Tangent Operator for Viscoelastoplastic Computations in Geodynamics, Geochemistry, Geophysics, Geosystems, 19, 4904–4924, https://doi.org/10.1029/2018GC007877, 2018.

Duretz, T., Räss, L., Podladchikov, Y., and Schmalholz, S.: Resolving thermomechanical coupling in two and three dimensions: spontaneous strain localization owing to shear heating, Geophysical Journal International, 216, 365–379, https://doi.org/10.1093/gji/ggy434, 2019.

Duretz, T., de Borst, R., and Yamato, P.: Modeling lithospheric deformation using a compressible visco-elasto-viscoplastic rheology and the effective viscosity approach, Geochemistry, Geophysics, Geosystems, 22, e2021GC009 675, https://doi.org/10.1029/2021GC009675, 2021.

Duretz, T., de Montserrat, A., Sevilla, R., Räss, L., Utkin, I., and Spang, A.: Automatic tuning of iterative pseudo-transient solvers for modeling the deformation of heterogeneous media, Geoscientific Model Development, 19, 5343–5362, https://doi.org/10.5194/gmd-19-5343-2026, 2026.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.



Frankel, S. P.: Convergence Rates of Iterative Treatments of Partial Differential Equations, Mathematical Tables and Other Aids to Computation, 4, 65, https://doi.org/10.2307/2002770, 1950.

Gerya, T.: Future directions in subduction modeling, Journal of Geodynamics, 52, 344–378, https://doi.org/10.1016/j.jog.2011.06.005, 2011.

Gerya, T.: Introduction to Numerical Geodynamic Modelling, Cambridge University Press, ISBN 9781316534243, https://doi.org/10.1017/9781316534243, 2019.

Gerya, T., Duretz, T., and Ludovic, R.: New continuity-based velocity interpolation scheme for staggered grids, EGU21, https://doi.org/10.5194/EGUSPHERE-EGU21-15308, 2021.

Gerya, T. V. and Yuen, D. A.: Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties, Physics of the Earth and Planetary Interiors, 140, 293–318, https://doi.org/10.1016/j.pepi.2003.09.006, 2003.

Gerya, T. V. and Yuen, D. A.: Robust characteristics method for modelling multiphase visco-elasto-plastic thermo-mechanical problems, Physics of the Earth and Planetary Interiors, 163, 83–105, https://doi.org/10.1016/j.pepi.2007.04.015, 2007.

Giordano, D., Russell, J. K., and Dingwell, D. B.: Viscosity of magmatic liquids: A model, Earth and Planetary Science Letters, 271, 123–134, https://doi.org/10.1016/J.EPSL.2008.03.038, 2008.

Harlow, F. H. and Welch, J. E.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, The Physics of Fluids, 8, 2182–2189, https://doi.org/10.1063/1.1761178, 1965.

Herschel, W. H. and Bulkley, R.: Konsistenzmessungen von gummi-benzollösungen, Kolloid-Zeitschrift, 39, 291–300, https://doi.org/10.1007/bf01432034, 1926.

Huber, C., Bachmann, O., and Manga, M.: Two competing effects of volatiles on heat transfer in crystal-rich magmas: Thermal insulation vs defrosting, Journal of Petrology, 51, 847–867, https://doi.org/10.1093/petrology/egq003, 2010.

Iacovino, K. and Till, C. B.: DensityX: A program for calculating the densities of magmatic liquids up to 1,627 °C and 30 kbar, Volcanica, 2, 1–10, https://doi.org/10.30909/VOL.02.01.0110, 2019.

Jiang, G.-S. and Shu, C.-W.: Efficient Implementation of Weighted ENO Schemes, Journal of Computational Physics, 126, 202–228, https://doi.org/10.1006/jcph.1996.0130, 1996a.

Jiang, G.-S. and Shu, C.-W.: Efficient Implementation of Weighted ENO Schemes, Journal of Computational Physics, 126, 202–228, https://doi.org/10.1006/jcph.1996.0130, 1996b.

Kaus, B. J. P., Mühlhaus, H., and May, D. A.: A stabilization algorithm for geodynamic numerical simulations with a free surface, Physics of the Earth and Planetary Interiors, 181, 12–20, https://doi.org/10.1016/J.PEPI.2010.04.007, 2010.

Kaus, B. J. P., Popov, A. A., Baumann, T., Pusok, A., Bauville, A., Fernandez, N., and Collignon, M.: Forward and inverse modelling of lithospheric deformation on geological timescales, in: Proceedings of NIC Symposium, ISBN 978-3-95806-109-5, 2016.

Kaus, B. J. P., Thielmann, M., Aellig, P., de Montserrat, A., de Siena, L., Frasunkiewicz, J., Fuchs, L., Piccolo, A., Ranocha, H., Riel, N., et al.: GeophysicalModelGenerator.jl: A Julia package to visualise geoscientific data and create numerical model setups, Journal of Open Source Software, 9, 6763, https://doi.org/10.21105/joss.06763, 2024.

Kaus, B. J. P., de Montserrat, A., Aellig, P. S., Dominguez, H., Medinger, N., Riel, N., Cosarinsky, M., Spang, A., Berlie, N., Kiss, D., Ranocha, H., Räss, L., Fuchs, L., Frasunkiewicz, J., and Luder, M.: GeoParams.jl: A Julia package for managing material parameters in geodynamic simulations, https://doi.org/10.5281/zenodo.20312540, [code], 2026.

Kiss, D., Moulas, E., Kaus, B. J. P., and Spang, A.: Decompression and Fracturing Caused by Magmatically Induced Thermal Stresses, Journal of Geophysical Research: Solid Earth, 128, e2022JB025 341, https://doi.org/10.1029/2022JB025341, 2023.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Larionov, E., Batty, C., and Bridson, R.: Variational Stokes: A Unified Pressure-Viscosity Solver for Accurate Viscous Liquids, ACM Trans. Graph., 36, https://doi.org/10.1145/3072959.3073628, 2017.

Liu, Y., Zhang, Y., and Behrens, H.: Solubility of H2O in rhyolitic melts at low pressures and a new empirical model for mixed H2O–CO2 solubility in rhyolitic melts, Journal of Volcanology and Geothermal Research, 143, 219–235, https://doi.org/10.1016/J.JVOLGEORES.2004.09.019, 2005.

Moresi, L., Dufour, F., and Mühlhaus, H.-B.: Mantle Convection Modeling with Viscoelastic/Brittle Lithosphere: Numerical Methodology and Plate Tectonic Modeling, pure and applied geophysics, 159, 2335–2356, https://doi.org/10.1007/s00024-002-8738-3, 2002.

Omlin, S. and Räss, L.: High-performance xPU Stencil Computations in Julia, Proceedings of the JuliaCon Conferences, 6, 138, https://doi.org/10.21105/jcon.00138, 2024.

Omlin, S., Räss, L., and Utkin, I.: Distributed Parallelization of xPU Stencil Computations in Julia, Proceedings of the JuliaCon Conferences, 6, 137, https://doi.org/10.21105/jcon.00137, 2024.

Papadrakakis, M.: A method for the automatic evaluation of the dynamic relaxation parameters, Computer Methods in Applied Mechanics and Engineering, 25, 35–48, https://doi.org/10.1016/0045-7825(81)90066-9, 1981.

Popov, A. A., Berlie, N., and Kaus, B. J.: A dilatant visco-elasto-viscoplasticity model with globally continuous tensile cap: stable two-field mixed formulation, Geoscientific Model Development, 18, 7035–7058, https://doi.org/10.5194/gmd-18-7035-2025, 2025.

Pusok, A. E., Kaus, B. J. P., and Popov, A. A.: On the Quality of Velocity Interpolation Schemes for Marker-in-Cell Method and Staggered Grids, Pure and Applied Geophysics, 174, 1071–1089, https://doi.org/10.1007/s00024-016-1431-8, 2017.

Räss, L., Utkin, I., Duretz, T., Omlin, S., and Podladchikov, Y. Y.: Assessing the robustness and scalability of the accelerated pseudo-transient method towards exascale computing, Geoscientific Model Development Discussions, 2022, 1–46, https://doi.org/10.5194/gmd-2021-411, 2022.

Riel, N., Kaus, B. J. P., Green, E. C. R., and Berlie, N.: MAGEMin, an Efficient Gibbs Energy Minimizer: Application to Igneous Systems, Geochemistry, Geophysics, Geosystems, 23, e2022GC010 427, https://doi.org/10.1029/2022GC010427, 2022.

Samaroo, J., Smirnov, A., Räss, L., Churavy, V., Montoison, A., Schaub, S. D., Hyatt, K., Baraldi, G., ffrancesco94, Osama, M., Evelyne, Hodgson, T., (Tee), A. F., Phillips, W., Besard, T., Ramadhan, A., TagBot, J., Schanen, M., Giordano, M., Bernoudy, W., Antholzer, S., Polanco, J. I., Moses, W., Arakaki, T., Guinard, C., Bauer, C., and jariji: JuliaGPU/AMDGPU.jl: v2.7.0, https://doi.org/10.5281/zenodo.4677700, [code], 2026.

Schmeling, H., Babeyko, A. Y., Enns, A., Faccenna, C., Funiciello, F., Gerya, T., Golabek, G. J., Grigull, S., Kaus, B. J., Morra, G., Schmalholz, S. M., and van Hunen, J.: A benchmark comparison of spontaneous subduction models-Towards a free surface, Physics of the Earth and Planetary Interiors, 171, 198–223, https://doi.org/10.1016/j.pepi.2008.06.028, 2008.

Schmid, D. W. and Podladchikov, Y. Y.: Analytical solutions for deformable elliptical inclusions in general shear, Geophysical Journal International, 155, 269–288, https://doi.org/10.1046/j.1365-246X.2003.02063.x, 2003.

Schuler, C.: Thermomechanical modeling of mantle dynamics and plate motion in the Mediterranean, Dissertation, Johannes Gutenberg University Mainz, https://doi.org/10.25358/openscience-13957, 2025.

Schuler, C., Kaus, B. J., Le Breton, E., Riel, N., and Popov, A. A.: Mantle Dynamics in the Mediterranean and Plate Motion of the Adriatic Microplate: Insights From 3D Thermomechanical Modeling, Geochemistry, Geophysics, Geosystems, 26, https://doi.org/10.1029/2024GC011996, 2025.

Spang, A., Thielmann, M., de Montserrat, A., and Duretz, T.: Transient propagation of ductile ruptures by thermal runaway, Journal of Geophysical Research: Solid Earth, 130, e2025JB031 240, https://doi.org/10.1029/2025JB031240, 2025.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.



Tackley, P. J.: Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid, Physics of the Earth and Planetary Interiors, 171, 7–18, https://doi.org/10.1016/j.pepi.2008.08.005, 2008.

van Keken, P. E., King, S. D., Schmeling, H., Christensen, U. R., Neumeister, D., and Doin, M.: A comparison of methods for the modeling of thermochemical convection, Journal of Geophysical Research: Solid Earth, 102, 22 477–22 495, https://doi.org/10.1029/97JB01353, 1997.

Watts, K. E., Bindeman, I. N., and Schmitt, A. K.: Large-volume rhyolite genesis in caldera complexes of the Snake River Plain: Insights from the Kilgore Tuff of the Heise volcanic field, Idaho, with comparison to Yellowstone and Bruneau-Jarbidge rhyolites, Journal of Petrology, 52, 857–890, https://doi.org/10.1093/petrology/egr005, 2011.

Whittington, A. G., Hofmeister, A. M., and Nabelek, P. I.: Temperature-dependent thermal diffusivity of the Earth's crust and implications for magmatism, Nature, 458, 319–321, https://doi.org/10.1038/nature07818, 2009.

Wong, Y.-Q. and Keller, T.: A unified numerical model for two-phase porous, mush and suspension flow dynamics in magmatic systems, Geophysical Journal International, 233, 769–795, https://doi.org/10.1093/gji/ggac481, 2023.

Zhong, S.: Analytic solutions for Stokes' flow with lateral variations in viscosity, Geophysical Journal International, 124, 18–28, https://doi.org/10.1111/j.1365-246X.1996.tb06349.x, 1996.

Zhong, S., Zuber, M. T., Moresi, L., and Gurnis, M.: Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection, Journal of Geophysical Research: Solid Earth, 105, 11 063–11 082, https://doi.org/10.1029/2000JB900003, 2000.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Appendix A: Benchmarks

To assess the accuracy of the accelerated pseudo-transient method for geodynamic applications, we reproduce three classic 2D viscous flow benchmarks (e.g. Duretz et al., 2011): (i) buoyancy-driven flow with smooth variations in density and viscosity (SolKz), (ii) buoyancy-driven flow with smooth density variations and a sharp viscosity contrast (SolCx), and (iii) far-field velocity-driven flow around a rigid circular inclusion embedded in a weak matrix (SolVi).

We also test JUSTRELAX.JL for its ability to model thermal and thermochemical convection at high Rayleigh numbers without numerical diffusion (Blankenbach et al., 1989) and to reproduce buoyancy-driven flow accurately (van Keken et al., 1997). The analytical solutions for SolKz and SolCx are implemented in EXACTFIELDSOLUTIONS.JL.

A1 Spatially varying material properties

A1.1 SolKz

The purpose of this benchmark is to evaluate the solver's ability to handle large, smooth viscosity variations. The non-dimensional model domain is defined as $\Omega \in [0,1] \times [0,1]$. Flow is driven purely by lateral buoyancy variations, with spatially smooth density and viscosity fields defined as:

$$\rho(x,y) = -\sin(2\pi y)\cos(3\pi x) \tag{A1}$$

$$\eta(y) = \exp(By), \tag{A2}$$

where $B = \ln(10^6)$, giving a viscosity contrast of six orders of magnitude. Free-slip boundary conditions are applied on all sides of the domain, implying zero tangential stress. The grid resolution is successively doubled, increasing from 16×16 to 1024×1024 cells. The convergence rates correspond to those reported by Duretz et al. (2011).

A1.2 SolCx

The SolCx benchmark tests the solver's performance with large, discontinuous viscosity contrasts. The non-dimensional domain is $\Omega \in [0,1] \times [0,1]$ and has free-slip boundary conditions. The setup includes both strong viscosity contrasts and spatial density variations, respectively defined as:

$$\eta(x,y) = \begin{cases} 1, & \text{if } 0 \leq x \leq 0.5 \\ 10^6, & \text{if } 0.5 < x \leq 1 \end{cases} \tag{A3}$$

$$\rho(x,y) = -\sin(\pi y)\cos(\pi x). \tag{A4}$$

As the discontinuous viscosity field is closer to typical geodynamic applications, the benchmark is ideally suited to assess the code's stability. The APT Stokes solver returns a good fit between the analytical solution described in Zhong (1996) and Duretz et al. (2011), with errors below 10^{-2} and 10^{-3} for pressure and velocities, respectively (see Fig. A1).

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

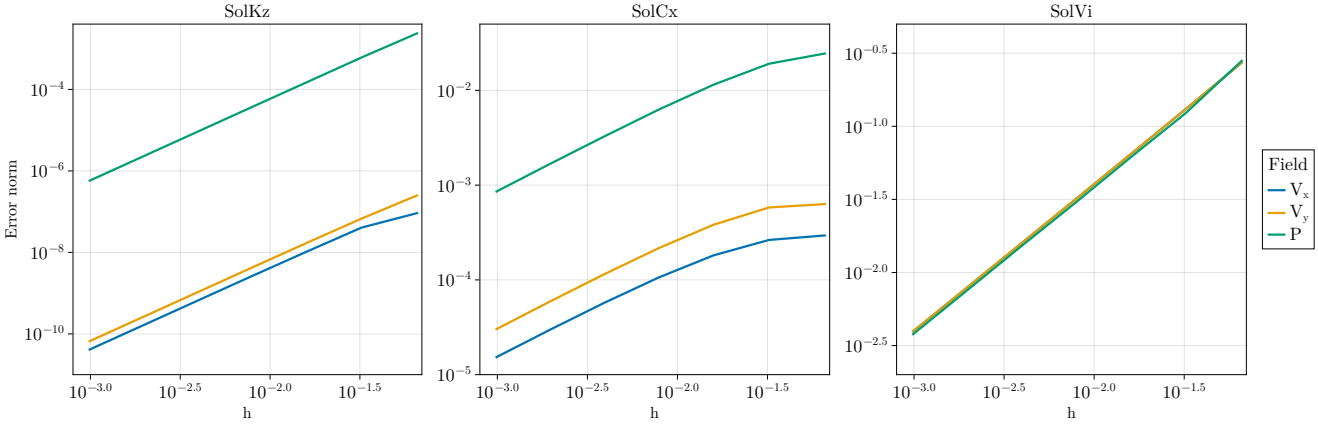


Figure A1. L1 norm for the SolKz and SolCx benchmarks, and the L2 norm for SolVi. Grid resolution ranges from 16×16 to 1024×1024 cells, with h being the grid spacing $(1/nx)$.

A1.3 SolVi

The setup consists of a circular inclusion of radius $r = 0.2$ embedded in the centre of a uniform matrix in the domain $\Omega \in [0,2] \times [0,2]$. The non-dimensional viscosity contrast between the weak matrix and the strong inclusion is $\Delta\eta = 10^3$. On the boundaries of the domain, we prescribe the velocity corresponding to the analytical solution with far-field pure shear boundary conditions described in (Schmid and Podladchikov, 2003) and a background strain rate of $\dot{\varepsilon} = 1$. We use the root-mean-squared (rms) error of the velocity and pressure fields as the metric to assess the accuracy of the numerical solution.

A1.4 Blankenbach thermal convection benchmark

Another commonly used benchmark in geodynamic codes is that of Blankenbach et al. (1989), which tests the time-dependent thermal convection in an isoviscous square box with various Rayleigh numbers and free-slip boundary conditions on all sides. This benchmark is especially useful for emulating mantle convection and for benchmarking the capability of a code to model higher Rayleigh numbers without numerical diffusion at the sub-grid level in both dimensional and non-dimensional form.

We achieve similar results to those in the benchmark study of Blankenbach et al. (1989) for their case 1a (isoviscous, $\mathrm{Ra} = 10^4$, square box). Additionally, we have performed a resolution test ranging from 32×32 to 512×512 on a single GPU (see Fig. A2). Although the number of time steps increases with higher resolutions, the root-mean-squared velocity and the Nusselt number at the top are consistent with the other resolutions and the reported numbers in the original study.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

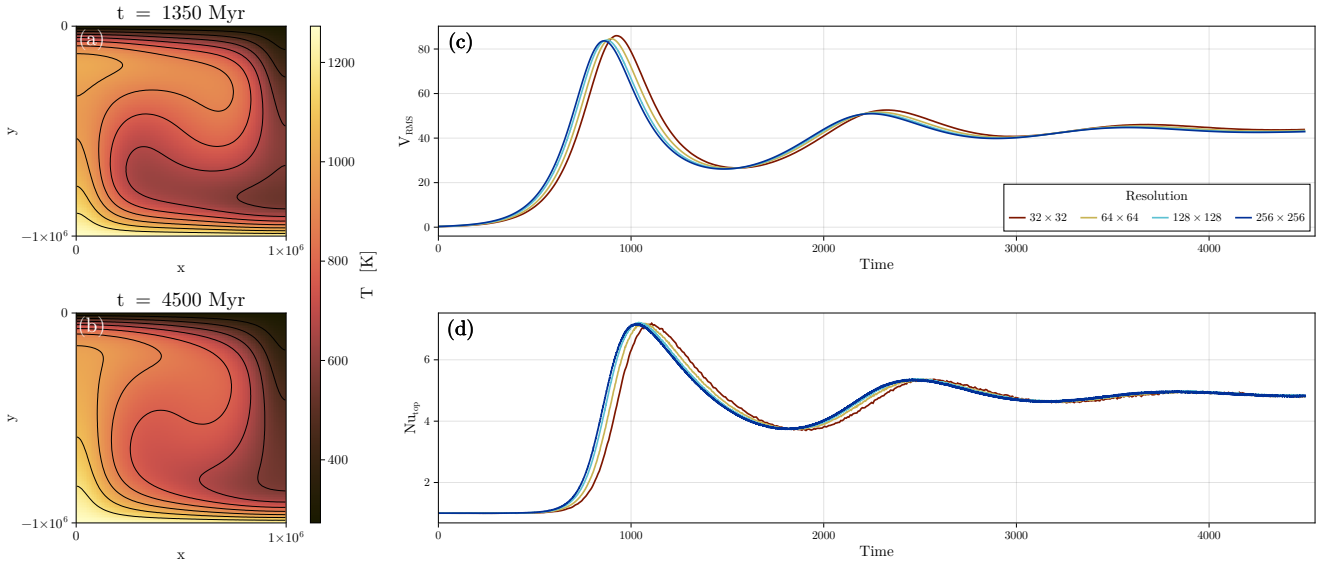


Figure A2. Thermal convection with $\mathrm{Ra} = 10^4$ after the benchmark paper of Blankenbach et al. (1989). The model corresponds to case 1a from the paper.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

A1.5 Van Keken thermochemical composition benchmark

We reproduce the thermochemical convection benchmark of van Keken et al. (1997). It consists of two incompressible, isoviscous liquids within a box domain $\Omega \in [0, \lambda] \times [0, 1]$, with $\lambda = 0.9142$. Convection is driven by differential buoyancy along the liquid interface, where the density contrast between the upper and lower liquids is $\Delta\rho = 1$. The interface between the fluids is $h = 0.2 + 0.02\cos\left(\frac{\pi x}{\lambda}\right)$. We also conducted a resolution test using 2^5 to 2^{10} grid points in each direction. The results agree with the values reported by van Keken et al. (1997) (see Fig. A3).

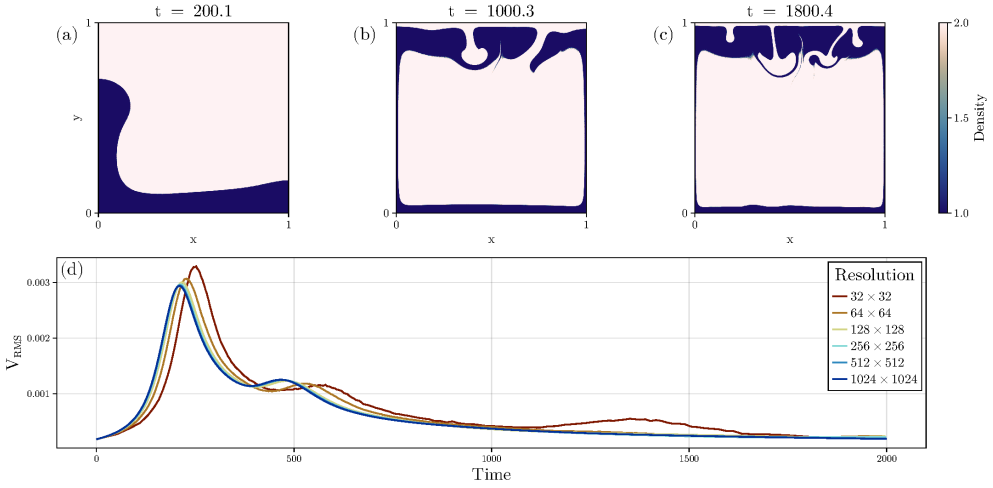


Figure A3. Thermochemical convection after van Keken et al. (1997) with a resolution test for a grid size of 2^n, with $n \in [5 - -10]$, in the x- and y-directions

Appendix B: Magma rheology and density parameterizations

For the magmatic sill convection setup (Sect. 6.3), we use the compositions of Watts et al. (2011) for the (Pre-)Kilgore Tuffs, which represent low- and high-silica rhyolites associated with the caldera-forming Kilgore eruption at the Heise volcanic field (USA).

To track the exsolution and evolution of dissolved H_2O, we added $3\,\mathrm{wt}\,\%$ water to the reported compositions (Table B1). The magma rheology consists of the temperature- and composition-dependent viscosity of Giordano et al. (2008) and the crystal-fraction correction term of Costa et al. (2009). We provide a detailed description of each parameter used in this rheology law. The magma viscosity is given by:

$$\eta_{\mathrm{magma}} = \eta_r \eta_{\mathrm{Giordano}} \tag{B1}$$

$$\eta_{\mathrm{Giordano}} = \eta_0 10^{\left(A_T + \frac{B_T}{T - C_T}\right)}, \tag{B2}$$

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

where $\eta_0 = 1\,\mathrm{Pa\,s}$ is a scaling viscosity, $A_T = -4.55$ is a constant, and B_T and C_T are computed from the oxide composition given. These initial compositions return $B_{T_{\mathrm{sill}}} = 10569.5442\,\mathrm{K}$, $C_{T_{\mathrm{sill}}} = 66.9511\,\mathrm{K}$, $B_{T_{\mathrm{host}}} = 9458.8842\,\mathrm{K}$, and $C_{T_{\mathrm{host}}} = 164.0830\,\mathrm{K}$. As the availability of a magmatic volatile phase (MVP) influences both the melt viscosity and melt fraction, B_T and C_T are re-evaluated as the exsolution of the MVP progresses. For the viscosity correction term η_r, we have:

$$\eta_r = \frac{1+\varphi^{\delta}}{\left[1 - F_m\left(\varphi, \dot{\varepsilon}, \gamma\right)\right]^{2.5\phi_*}} \tag{B3}$$

$$F_m = \alpha\,\mathrm{erf}\left(\frac{\sqrt{\pi}}{2\alpha}\varphi(1+\varphi^{\gamma})\right), \tag{B4}$$

where ϕ_c is the crystal volume fraction used in the Costa et al. (2009) correction, ϕ_* is the critical solid fraction, and $\varphi = \phi_c/\phi_*$. In the three-phase mixture used here, we set $\phi_c = \phi_x$; the gas fraction is not counted as crystals in the rheological correction. The remaining terms are given by

$$\theta = \log_{10}\left(\frac{\dot{\varepsilon}_{\mathrm{II}}}{\dot{\varepsilon}_0}\right) \tag{B5}$$

$$\phi_* = 0.066499\tanh(0.913424\theta + 3.850623) + 0.591806 \tag{B6}$$

$$\delta = -6.301095\tanh(0.818496\theta + 2.86) + 7.462405 \tag{B7}$$

$$\alpha = 1 - \xi = -0.000378\tanh(1.148101\theta + 3.92) + 0.999572 \tag{B8}$$

$$\gamma = 3.987815\tanh(0.8908\theta + 3.24) + 5.099645, \tag{B9}$$

where $\phi_*, \delta, \alpha, \gamma$ are dimensionless functions of θ, and the characteristic strain rate is $\dot{\varepsilon}_0 = 1\,\mathrm{s}^{-1}$. The total magma viscosity η_{magma} is constrained by an upper and lower bound for numerical stability, $\eta_{\min} \le \eta_{\mathrm{magma}} \le \eta_{\max}$, with $\eta_{\min} = 10^3\,\mathrm{Pa\,s}$ and $\eta_{\max} = 10^{12}\,\mathrm{Pa\,s}$. As stated in Sect. 6.3, the density is a three-phase mixture where each local density is given by:

$$\rho_m(P,T,\boldsymbol{X}_{\mathrm{H_2O}}) = \frac{\sum_{i=1}^{N_c} X_i\,MW_i}{\sum_{i=1}^{N_c} X_i V_i} \tag{B10}$$

$$V_i(P,T) = V_{\mathrm{ref},i} + \frac{dV_i}{dT}\left(T - T_{\mathrm{ref},i}\right) + \frac{dV_i}{dP}\left(P - P_0\right) \tag{B11}$$

where X_i is the mass fraction of the oxide component i, MW_i is the molecular weight, $N_c = 9$ for the composition in Table B1, and V_i is the partial molar volume. The quantities $V_{\mathrm{ref},i}$, dV_i/dT, dV_i/dP, and $T_{\mathrm{ref},i}$ are component-specific coefficients of the melt-density model.

The exsolved-water density is represented by a modified Redlich–Kwong fit from (Huber et al., 2010):

$$\rho_g(P,T) = \rho_{\mathrm{ref}}\left(a_1\tau^{-0.381} + a_2\omega^{-1.135} + a_3\tau^{-0.411\omega^{0.0333}}\right) \tag{B12}$$

where $\tau = T/T_c$ and $\omega = P/P_c$, where T_c and P_c are the critical temperature and pressure of water, ρ_{ref} is the reference density, and a_1, a_2, and a_3 are the dimensionless fit coefficients reported with the equation of state.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Table B1. Compositions of the (Pre-)Kilgore Tuff as reported by Watts et al. (2011) with added H_2O to track the evolution of the dissolved and exsolved volatiles. The Kilgore Tuff composition is used for the sill, the Pre-Kilgore Tuff composition for the host rock.

[wt %]	Host rock (Pre-Kilgore Tuff)	Sill (Kilgore Tuff)
SiO_2	70.78	75.75
TiO_2	0.55	0.28
Al_2O_3	15.86	12.48
FeO	3.93	2.14
MgO	1.11	0.09
CaO	1.20	0.48
Na_2O	2.54	3.53
K_2O	3.84	5.19
H_2O	3	3

The crystal density is approximated by a linearized P–T relation,

$$\rho_x(P,T) = \rho_{0,x}\left(1 - \alpha_x\left(T - T_0\right) + \beta_x\left(P - P_0\right)\right), \tag{B13}$$

where ρ_x is the crystal density at $T = T_0$ and $P = P_0$, and α_x and β_x are the crystal thermal expansion and compressibility coefficients, respectively. The mixture is then

$$\rho(P,T) = \phi_m \rho_m(P,T,\boldsymbol{X}_{\mathrm{H_2O}}) + \phi_g \rho_g(P,T) + \phi_x \rho_x(P,T) \tag{B14}$$

under the constraint $\phi_m + \phi_g + \phi_x = 1$, where these phase fractions are obtained from the melt fraction and water-solubility model. The melt fraction is computed from a polynomial fit to phase diagrams obtained with MAGEMin v1.9.10 (Riel et al., 2022).

Appendix C: Parameters for geodynamic applications

This section summarizes the material, rheological, and numerical parameters used in the geodynamic application models presented in the main text.

C1 Subduction of oceanic lithosphere

The following table lists the material properties and the visco-plastic rheology parameters used for the 2D subduction setup in Sect. 6.2.

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

Table C1. Material and rheological parameters of the active phases in the 2D subduction model. Mantle phases comprise the asthenosphere and both lithospheric mantles (phases 1–3). The subducting oceanic crust uses strong diabase dislocation creep, whereas the overriding continental crust uses wet quartzite dislocation creep; both crustal phases use dry-anorthite diffusion creep. The nonlinear solver bounds the creep viscosity between 10^{19} and 10^{24} Pa s; the listed 10^{18} Pa s weak-zone viscosity is its prescribed value before this global cut-off is applied.

Parameter	Units	Mantle phases	Oceanic crust	Continental crust	Weak zone	Sticky air
ρ_0	kg m^{-3}	3300	3300	2900	3300	100
α	K^{-1}	3×10^{-5}	3×10^{-5}	3×10^{-5}	3.0×10^{-5}	—
β	Pa^{-1}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	—
T_0	K	293	293	293	293	—
C_p	J kg^{-1} K^{-1}	10^3	10^3	10^3	10^3	10^6
k	W m^{-1} K^{-1}	3	34	3	3	1
G	Pa	6×10^{10}	6×10^{10}	6×10^{10}	6×10^{10}	—
ν	—	0.45	0.45	0.45	0.45	—
A_{disl}	MPa^{-n} s^{-1}	1.1×10^5	8	$10^{-11.2}$	—	—
n_{disl}	—	3.5	4.7	4	—	—
r_{disl}	—	0	0	1	—	—
E_{disl}	J mol^{-1}	5.30×10^5	4.85×10^5	1.35×10^5	—	—
V_{disl}	m^3 mol^{-1}	1.4×10^{-5}	0	0	—	—
A_{diff}	MPa^{-1} μm^3 s^{-1}	1.5×10^9	$10^{12.1}$	$10^{12.1}$	—	—
n_{diff}	—	1	1	1	—	—
p_{diff}	—	-3	-3	-3	—	—
E_{diff}	J mol^{-1}	3.75×10^5	4.60×10^5	4.60×10^5	—	—
V_{diff}	m^3 mol^{-1}	1.4×10^{-5}	2.4×10^{-5}	2.4×10^{-5}	—	—
d	m	10^{-3}	5×10^{-3}	5×10^{-3}	—	—
η	Pa s	—	—	—	10^{19}	10^{19}
C	MPa	10	10	10	10	—
ϕ	°	30	5	30	30	—
η_{reg}	Pa s	10^{22}	10^{22}	10^{22}	10^{22}	—

https://doi.org/10.5194/egusphere-2026-5366
Preprint. Discussion started: 2 November 2026
© Author(s) 2026. CC BY 4.0 License.

C2 Ascent of a 3D thermal and compositional anomaly

The table below summarizes the phase-wise rheology and thermal parameters adopted in the 3D plume (thermal and compositional anomaly) example in Sect. 6.4.

Table C2. Rheology parameters of the 3D plume example.

Parameter	Units	Upper crust	Lower crust	Lithospheric mantle	Sublithospheric mantle	Anomaly
ρ_0	$\mathrm{kg\,m^{-3}}$	2750	3000	3300	3300	3250
α	$\mathrm{K^{-1}}$	3.5×10^{-5}	3.5×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}
C_p	$\mathrm{J\,kg^{-1}\,K^{-1}}$	7.5×10^{2}	7.5×10^{2}	1.25×10^{3}	1.25×10^{3}	1.25×10^{3}
a_k	$\mathrm{W\,m^{-1}\,K^{-1}}$	0.64	0.64	0.73	0.73	0.73
b_k	$\mathrm{W\,m^{-1}}$	807	807	1293	1293	1293
c_k	K	0.77	0.77	0.77	0.77	0.77
d_k	MPa	4×10^{-11}	4×10^{-11}	4×10^{-11}	4×10^{-11}	4×10^{-11}
G	Pa	2.5×10^{10}	2.5×10^{10}	6.7×10^{10}	6.7×10^{10}	6.7×10^{10}
A_{disl}	$\mathrm{Pa^{-n}\,s^{-1}}$	5.07×10^{-18}	2.08×10^{-23}	2.51×10^{-17}	2.51×10^{-17}	2.51×10^{-17}
n_{disl}	—	2.3	3.2	3.5	3.5	3.5
E_{disl}	$\mathrm{J\,mol^{-1}}$	1.54×10^{5}	2.38×10^{5}	5.30×10^{5}	5.30×10^{5}	5.30×10^{5}
C	MPa	3	3	3	—	—
μ	—	0.2	0.2	0.2	0.3	—
η_{reg}	Pa s	10^{16}	10^{16}	10^{16}	—	—