Implicit solution of the material transport in Stokes flow simulation: Toward thermal convection simulation surrounded by free surface

Journal Article

Author(s):
Furuichi, Mikito; May, Dave A.

Publication date:
2015-07

Permanent link:
https://doi.org/10.3929/ethz-b-000100505

Originally published in:
Computer Physics Communications 192, https://doi.org/10.1016/j.cpc.2015.02.011

Computer Physics Communications 192 (2015) 1–11

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Implicit solution of the material transport in Stokes flow simulation: Toward thermal convection simulation surrounded by free surface

Mikito Furuichi [a,*], Dave A. May [b]

[a] Department of Mathematical Science and Advanced Technology, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), 3173-25 Showa-machi, Kanazawa, Yokohama, Japan

[b] Institute of Geophysics, Department of Earth Science, ETH Zürich, Sonneggstrasse 5, Zürich, Switzerland

ARTICLE INFO

Article history:
Received 11 July 2014
Received in revised form
9 February 2015
Accepted 17 February 2015
Available online 25 February 2015

Keywords:
Stokes flow
Free surface
Implicit time integration
JFNK
Geodynamics
TR-BDF2

ABSTRACT

We present implicit time integration schemes suitable for modeling free surface Stokes flow dynamics with marker in cell (MIC) based spatial discretization. Our target is for example thermal convection surrounded by deformable surface boundaries to simulate the long term planetary formation process. The numerical system becomes stiff when the dynamical balancing time scale for the increasing/decreasing load by surface deformation is very short compared with the time scale associated with thermal convection. Any explicit time integration scheme will require very small time steps; otherwise, serious numerical oscillation (spurious solutions) will occur. The implicit time integration scheme possesses a wider stability region than the explicit method; therefore, it is suitable for stiff problems. To investigate an efficient solution method for the stiff Stokes flow system, we apply first (backward Euler (BE)) and second order (trapezoidal method (TR) and trapezoidal rule–backward difference formula (TR-BDF2)) accurate implicit methods for the MIC solution scheme. The introduction of implicit time integration schemes results in nonlinear systems of equations. We utilize a Jacobian free Newton Krylov (JFNK) based Newton framework to solve the resulting nonlinear equations. In this work we also investigate two efficient implicit solution strategies to reduce the computational cost when solving stiff nonlinear systems. The two methods differ in how the advective term in the material transport evolution equation is treated. We refer to the method that employs Lagrangian update as "fully implicit" (Imp), whilst the method that employs Eulerian update is referred to as "semi-implicit" (SImp). Using a finite difference (FD) method, we have performed a series of numerical experiments which clarify the accuracy of solutions and trade-off between the computational cost associated with the nonlinear solver and time step size. In comparison with the general explicit Euler method, the second order accurate Imp methods reduce total computational cost successfully through the utilization of a large time step without sacrificing accuracy and stability. Moreover, the proposed SImp method is effective in reducing the computational cost associated with evaluating the nonlinear residual while obtaining a solution similar to the Imp method.

1. Introduction

Developing numerical schemes to model the dynamics of the systems described by Stokes flow coupled with material transport and a free surface boundary condition is a significant challenge in computational geodynamics (i.e. [1–5]). Simulating such systems over million-year time scales enables numerical investigation of the interaction between the surface geometry (topography) and interior dynamics of planets, which is of fundamental importance for understanding processes such as mountain building, subduction initiation and planetary core formation [2,6,7].

Although numerous free surface implementations have been proposed previously [4,8–11], almost all employ explicit time integration schemes. Such schemes are conditionally stable, and thus an inappropriate choice of the computation time step will result in spurious behavior, which typically manifests itself as an oscillation at the free surface interface [3,12,13]. The state of pressure within the Earth is dominated by a large background hydrostatic pressure. Assuming an average rock density to be 3300 kg/m^3, this results in a pressure gradient on the order of 32 MPa/km. Small perturbations from this background hydrostatic pressure can occur, particularly in regions close to the Earth's surface. For example, at depth,

[*] Corresponding author.
E-mail address: m-furuic@jamstec.go.jp (M. Furuichi).

http://dx.doi.org/10.1016/j.cpc.2015.02.011

the dynamic pressure associated with the presence of a negatively buoyant mantle plume will be insignificant compared to the hydrostatic pressure. However, as the plume rises, the dynamic pressure associated with the upward traveling plume will become comparable in magnitude to the hydrostatic pressure. Near the surface of the Earth, the negatively buoyant material will generate significant dynamic pressure and result in uplift of the crust. Physically, pressure changes and isostatic relaxation of the surface recover this "out of balanced" situation and return the pressure field to a predominately hydrostatic state. However the time scale required to resolve these relaxation processes is much smaller than the time scale associated with the physical processes of interest (e.g. thermal convection). By utilizing an explicit time integrator, one requires a small time step to capture the relaxation of the surface. As a result, the computational cost for stable time integration when free surface dynamics are incorporated is far higher than that when a free non-deformable surface is adopted.

Several approaches for stabilization, or implicit treatment, have been recently proposed to avoid free surface oscillations in the context of finite element and finite difference methods [3,12,13]. In this work, we propose to treat advection as a coordinate nonlinearity coupled to the momentum equation, thereby defining a fully implicit time integration scheme. In the geodynamics community, nonlinear solvers have been applied to processes involving material non linearities (i.e. power law creep or plastic rheologies). However, the application of a nonlinear solver for implicit material advection to define implicit time integrators for stable free surface evolution has not been examined.

In this paper, we apply several implicit time integration schemes based on the ordinary differential equation (ODE) stability theory [14] for solving the sticky-air free surface problem of Stokes flow [2,15]. These implicit methods are implemented within the FD framework combined with the MIC method with nonlinear residuals derived from the formulation of the material transport. Resulting nonlinear equations are solved iteratively by the Newton-based nonlinear solver. In addition, we propose two types of the advection of the material properties for the nonlinear solver. One is a full implicit method that uses the MIC throughout the solution process. The second is a semi-implicit method that uses an Eulerian advection scheme for the nonlinear residual evaluation. We examine the solution quality and efficiency of these methods by performing numerical experiments that simulate (i) viscous relaxation of a sinusoidal topographic high and (ii) thermal evolution in a radial gravity field coupled with a free surface. Based on our numerical experiments, we discuss the different stability characteristics and accuracies and analyze the trade-offs between time step size and computational efficiency in an attempt to determine optimal strategies to the solution of time-dependent viscous flow calculations.

2. Basic Stokes flow system

We begin with a purely mechanical problem of Stokes equations to explain our solution procedures. The equations for an incompressible variable viscosity Stokes fluid in a domain Ω are given as

$$\nabla p - \nabla \cdot \left(\eta(\phi)\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right) + \rho(\phi)\boldsymbol{g} = \boldsymbol{0}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where ρ and η denote the density and viscosity as function of the material composition ϕ. The pressure, velocity vector and gravity vector are denoted via p, $\boldsymbol{u}$ and $\boldsymbol{g}$ respectively. Along the boundary of Ω, denoted via $\partial\Omega$, we impose the "free-slip" boundary condition given by

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0; \qquad \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) \cdot \boldsymbol{t} = 0, \tag{3}$$

where $\boldsymbol{n}$, $\boldsymbol{t}$ denote the outward pointing unit normal vector and unit tangent vector to $\partial\Omega$.

The evolution of the material composition ϕ, expressed in Lagrangian frame of reference is given by

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}, \tag{4}$$

$$\frac{d\phi}{dt} = 0. \tag{5}$$

Eqs. (4) and (5) represent the fluid element defined at the position $\boldsymbol{x}$ advect phase ϕ (i.e. $\frac{D\phi}{Dt} = 0$ in the Eulerian frame).

3. Time integration

In this work, we used several explicit and implicit time integration schemes for an efficient solution of the numerically stiff system following the general ODE theory [14]. Assuming that the velocity $\boldsymbol{u}$ is uniquely obtained from the ϕ and $\boldsymbol{x}$, we can regard Eqs. (4) and (5) as a system of ODEs. The complete set of ODEs given by Eqs. (4) and (5) represents a stiff system with free surface deformation. The applied time integration method to improve the stability and accuracy for such problems are described below.

3.1. Explicit method (Exp)

It is common practice within geodynamics simulations of the Stokes flow, to perform a splitting between the equations describing the motion of the fluid (Eqs. (1), (2)), and those defining the transport of material phase (Eqs. (4), (5)) Such a splitting gives rise to the explicit Euler time-stepping method (Exp). The procedure to update the position $\boldsymbol{x}$ is given by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t\, \boldsymbol{u}^n, \tag{6}$$

where $(\cdot)^n$ is the value at nth time step. The Exp method is the most standard method used to solve the Stokes flow system because it is easy to implement and computationally inexpensive. However the Euler method is first order accurate in time and the explicit time step may require small Δt to avoid the spurious oscillations. We note that for the system of equations we consider in this work, there is no formal stability criterion defining a Δt which will yield a stable time-integration scheme.

3.2. One-step implicit methods (BE, TR)

For a stable time step integration that permits a large time step size to be used, an implicit time integration scheme is required [16]. We first introduce one-step methods, by which $\boldsymbol{x}^{n+1}$ is obtained directory by $\boldsymbol{x}^n$ without any intermediate steps. One of the classical and useful methods is the backward Euler method (BE), which can be expressed by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t\, \boldsymbol{u}^{n+1}. \tag{7}$$

This method is first order accurate and L-stable, which is unconditionally stable and suppresses unphysical numerical oscillation against the large time step integration.

Another frequently used one-step implicit method is the trapezoidal method (TR) [13,16], which can be described by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \frac{1}{2}\Delta t\, (\boldsymbol{u}^n + \boldsymbol{u}^{n+1}). \tag{8}$$

This averaging scheme is second order accurate; however it is not L-stable. Thus, the stability region without significant spurious oscillation is limited although better than the Exp method.

3.3. Two-step implicit method (TR-BDF2)

As a high order accurate and L-stable method, we implement a two-step method based on the trapezoidal and backward differentiation formula method (TR-BDF2) [14] with the two-step procedures outlined below:

$$\text{1st step: } \boldsymbol{x}^{n+1/2} = \boldsymbol{x}^n + \frac{1}{4}\Delta t\,(\boldsymbol{u}^n + \boldsymbol{u}^{n+1/2}), \tag{9}$$

$$\text{2nd step: } \boldsymbol{x}^{n+1} = \frac{1}{3}(4\boldsymbol{x}^{n+1/2} - \boldsymbol{x}^n) + \frac{1}{3}\Delta t\,\boldsymbol{u}^{n+1}. \tag{10}$$

The first stage is the normal TR method applied over $\Delta t/2$ for obtaining the intermediate state at $n + \frac{1}{2}$. Then the backward differentiation method is applied at the second stage of Eq. (10) for the solution at $n+1$. The TR-BDF2 method is second order accurate and L-stable; however, double calculation cost against the one-step methods is required due to the two-step procedures.

4. Spatial discretization

Eqs. (1) and (2) are discretized in space using a standard staggered grid finite difference (FD) scheme defined on a rectangular mesh consisting of $N \times M$ control volumes. A linear Stokes flow problem with density and viscosity variations discretized by the FD method (e.g. [15,17,18]) can be obtained in matrix form as

$$\mathbf{A}[\boldsymbol{\eta}_c]\mathbf{X} = \begin{pmatrix} \mathbf{K}[\boldsymbol{\eta}_c] & \mathbf{G} \\ \mathbf{G}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_g \\ \mathbf{p}_c \end{pmatrix} = \begin{pmatrix} \boldsymbol{\rho}_g g \\ 0 \end{pmatrix} = \mathbf{b}[\boldsymbol{\rho}_c], \tag{11}$$

where $\mathbf{X} = (\mathbf{u}_g, \mathbf{p}_c)^T$ is the discrete velocity and pressure vector. The notation $(\cdot)_g$ and $(\cdot)_c$ denote vector components on the faces of the cell, and scalars on the cell center, respectively. **K** and **G** are the discretized matrix form for the gradient of deviatoric stress and gradient of pressure.

Following [15,19], we discretize the composition field ϕ using Lagrangian markers, thereby allowing us to track the evolution of material properties (e.g. η and ρ). We will denote a value defined on the marker via $(\cdot)_m$. Accordingly, Eqs. (4) and (5) become

$$\frac{d\mathbf{x}_m}{dt} = \mathbf{u}_m, \tag{12}$$

$$\frac{d\boldsymbol{\phi}_m}{dt} = 0, \tag{13}$$

where $\mathbf{u}_m$ is the velocity at position $\mathbf{x}_m$. The velocity $\mathbf{u}_m$ is obtained by linearly interpolating values $\mathbf{u}_g$ from the FD grid. This interpolation from the FD grid to the marker positions is denoted via

$$\mathbf{u}_m = \Pi[\mathbf{u}_g]. \tag{14}$$

Material properties defined on the markers (e.g. $\boldsymbol{\eta}_m$, $\boldsymbol{\rho}_m$) are projected onto the centroid of each control volume using the following weighted-averaging scheme

$$(\cdot)_c^k = \left(\sum_{\Omega_p \subseteq \Omega_k} (\cdot)_m^p \, w^k{}_p(\mathbf{x}_m{}^p) \right) \Bigg/ \left(\sum_{\Omega_p \subseteq \Omega_k} w^k{}_p(\mathbf{x}_m{}^p) \right), \tag{15}$$

where the notations $(\cdot)_c^k$ and $(\cdot)_m^p$ are for the kth cell and pth marker from the complete discrete set $(\cdot)_c$ and $(\cdot)_m$, respectively, Ω_p is the quadrilateral (2D) or hexahedral (3D) region with size Δx centered at marker position $\mathbf{x}_m{}^p$, Ω_k is the kth cell region of the staggered grid and $w^k{}_p(\mathbf{x}_m{}^p)$ is the fractional area (or volume) of $\Omega_p \cap \Omega_k$. The variables associated with the remapping procedure are shown in Fig. 1. Note that $\boldsymbol{\rho}_g$ in Eq. (11) is obtained from $\boldsymbol{\rho}_c$ by linear interpolation.

This type of MIC scheme is commonly used in computational geodynamics (e.g. [15,19,20]), because it allows numerically non diffusive advection without numerical oscillation arising in general high order Eulerian advection scheme. In addition, with this type of MIC scheme, it is easy to implement a physical evolution model coupled to the Stokes flow dynamics. However, the computational cost is expensive compared with alternative full Eulerian methods.

In this work, we are interested in processes which involve a free surface boundary condition, e.g.

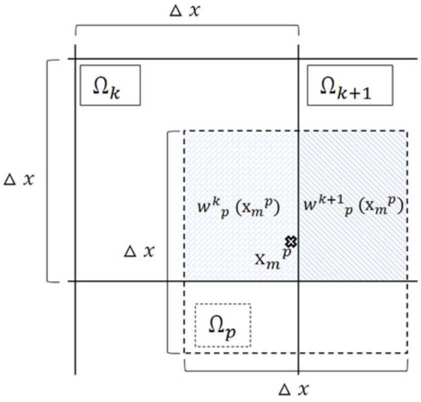

Fig. 1. Illustration of the weighted-averaging scheme that converts variables defined on the markers to the grid cell.

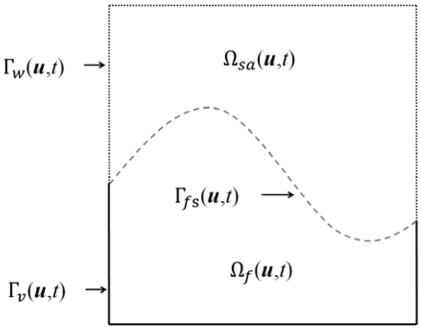

Fig. 2. Illustration of the domain for the sticky-air approach.

$$\left(\eta\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right) - p\boldsymbol{I}\right)\cdot\boldsymbol{n} = 0 \quad \text{on } \Gamma_{fs}(\boldsymbol{u}, t), \tag{16}$$

where $\Gamma_{fs}(\boldsymbol{u}, t)$ denotes the location of the time-dependent free surface of the fluid domain ($\Omega_f(\boldsymbol{u}, t)$). For a free-surface geometry which is not orthogonal to the coordinate system, the standard staggered grid discretization cannot be used to discrete the fluid domain $\Omega_f(\boldsymbol{u}, t)$. This short coming is alleviated via approximating the free-surface boundary condition using an approach routinely used in the geodynamics community known as the "sticky-air" technique [2]. The sticky-air approach is defined in the following way:

1. Given a fluid domain $\Omega_f(\boldsymbol{u}, t)$, we define free-surface boundaries $\Gamma_{fs}(\boldsymbol{u}, t)$ and rest of the boundaries as $\Gamma_v(\boldsymbol{u}, t) = \partial\Omega_f(\boldsymbol{u}, t) \setminus \Gamma_{fs}(\boldsymbol{u}, t)$ (see Fig. 2).
2. Define a computational domain Ω with faces orthogonal to the 2D or 3D coordinate system such that $\Omega_f(\boldsymbol{u}, t) \subset \Omega$, and which possesses a boundary defined by $\partial\Omega = \Gamma_v(\boldsymbol{u}, t) \cup \Gamma_w(\boldsymbol{u}, t)$.
3. The "sticky-air" region ($\Omega_{sa}(\boldsymbol{u}, t)$) is defined as the subdomain enclosed by $\Gamma_w(\boldsymbol{u}, t) \cup \Gamma_{fs}(\boldsymbol{u}, t)$. The material inside $\Omega_{sa}(\boldsymbol{u}, t)$ defines a new composition which is represented using the phase ϕ of Lagrangian markers. The sticky-air material is defined as a fluid with low viscosity and zero density which is evolved according to the governing equations given by Eqs. (1), (2), (4), (5).

Provided sticky-air viscosity is small relative to the viscosity inside the fluid domain, the viscous shear stress across the free-surface interface is small and the boundary condition in Eq. (16) is closely approximated. The validity and accuracy of this approach has been carefully examined in [2].

5. Implicit solution for nonlinear problem

An implicit time integration method to solve the Stokes flow system coupled with the material transport (Eqs. (1), (2), (4), (5)) requires the solution of a nonlinear problem.

5.1. Definition of the nonlinear residual

Here, we define our nonlinear residuals for the implicit time integration schemes. To evaluate the nonlinear residual, we first update the material properties $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$ for given trial velocity and pressure solution $\mathbf{X}' = (\mathbf{u}'_g, \mathbf{p}'_c)$. The nonlinear Stokes residual for a converged solution with MIC method is then evaluated for each time integration schemes.

5.1.1. Updating material property (Imp and SImp)

The nonlinearity for the implicit material transport comes from the updating material properties with $\mathbf{X}'$. We have examined two types of calculation method for $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$. The difference between them in the residual evaluation gives fully implicit (Imp) and semi-implicit (SImp) method.

The Imp method evaluates the nonlinearity of $\boldsymbol{\eta}_m$ and $\boldsymbol{\rho}_m$ directly from the marker coordinates. Therefore, it is a fully general implicit method of the MIC scheme. The trial marker position $\mathbf{x}'_m$ for a given velocity $\mathbf{u}'_g$ is calculated by the each time integration schemes of Section 3 via Eq. (14). Then, we evaluate the material properties $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$from the markers at $\mathbf{x}'_m$ via Eq. (15). The advection and remapping operations of markers should be performed every time nonlinearity is updated. Note that these operations are computationally expensive. Thus, the costs to evaluate the nonlinear residual with the MIC scheme is likely to dominate the total cost of the nonlinear solver when using the Imp method.

In the SImp method, the marker positions do not change when evaluating the nonlinear residual. Instead, an advection scheme on the Eulerian mesh is used to transport the material implicitly and thus update $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$. The position of each marker is only updated after obtaining the converged solution of the nonlinear problem. With the Eulerian advection method, we examined classical first and fifth order upwind FD methods (Appendix A). In this work, to avoid the time-step size limits by Courant–Friedrichs–Lewy (CFL) condition, the FD advection scheme is applied with the sub time steps, by which quantity is iteratively advected with the substep size Δt_s to reach the $\Delta t = N_s \Delta t_s$. The number of substeps N_s is chosen for Δt_s so as to not exceed $0.8\Delta x/|\mathbf{u}_{\max}|$.

The accuracy and stability obtained by this method can differ from the Imp method, because the profiles transported by the grid-based method are inconsistent with those defined by the markers. Therefore we consider this method as a semi-implicit time integration method. When using the Simp, we can expect reduced computational cost compared with Imp because the Eulerian advection does not require the expensive MIC remapping procedure at each nonlinear residual evaluation.

5.1.2. Stokes residual for implicit time-stepping method

The updated $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$ using $\mathbf{X}'$ generate the nonlinear grid-based Stokes residuals, which are different for the implicit time integration schemes. For the BE method of Eq. (7), each marker is updated with $\mathbf{X}'$ via $\mathbf{x}'_m = \mathbf{x}^n_m + \Delta t\, \mathbf{u}'_m$. Thus, the discrete nonlinear residual can be expressed as follows:

$$\mathbf{F}_{BE}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] = \mathbf{A}[\boldsymbol{\eta}_c]\mathbf{X}' - \mathbf{b}[\boldsymbol{\rho}_c]. \tag{17}$$

In the same manner, the TR method of Eq. (8) updates marker position using $\mathbf{x}'_m = \mathbf{x}^n_m + \frac{\Delta t}{2}\left(\mathbf{u}'_m + \mathbf{u}^n_m\right)$. The nonlinear residual for the TR method is given by

$$\mathbf{F}_{TR}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] = \mathbf{A}[\boldsymbol{\eta}_c]\left(\frac{1}{2}\mathbf{X}' + \frac{1}{2}\mathbf{X}^n\right) - \mathbf{b}[\boldsymbol{\rho}_c]. \tag{18}$$

For the two-step TR-BDF2 method of Eqs. (9) and (10), the residual F_{TR} of Eq. (18) is used in the first step to obtain $\mathbf{X}^{n+\frac{1}{2}}$ and $\mathbf{x}^{n+\frac{1}{2}}_m$. At the second step, since the markers are updated using $\mathbf{x}'_m = \mathbf{x}^{n+\frac{1}{2}}_m + \frac{\Delta t}{2}\left(\frac{2}{3}\mathbf{u}'_m + \frac{1}{6}\left(\mathbf{u}^{n+\frac{1}{2}}_m + \mathbf{u}^n_m\right)\right)$ from the intermediate state, the residual can be derived as

$$\mathbf{F}_{TR\text{-}BDF2}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] = \mathbf{A}[\boldsymbol{\eta}_c]\left(\frac{2}{3}\mathbf{X}' + \frac{1}{6}\left(\mathbf{X}^{n+\frac{1}{2}} + \mathbf{X}^n\right)\right) - \mathbf{b}[\boldsymbol{\rho}_c]. \tag{19}$$

The algorithm for the nonlinear residual of each time stepping scheme (i.e. BE, TR and TR-BDF2) with updating the material property by Imp or SImp method is shown in Algorithm 1.

```
Algorithm 1
Line 1.  If Imp is used then:
Line 2.    Update marker coordinate $\mathbf{x}'_m$ with
           $\mathbf{u}'_m = \Pi[\mathbf{u}'_g]$
Line 3.    Project $\boldsymbol{\eta}_m$, $\boldsymbol{\rho}_m$ at $\mathbf{x}'_m$ onto the mesh $\boldsymbol{\eta}_c$, $\boldsymbol{\rho}_c$
Line 4.  Elseif SImp is used then:
Line 5.    Update $\boldsymbol{\eta}_c$ and $\boldsymbol{\rho}_c$ by upwind Eulerian
           advection schemes with $\mathbf{u}'_g$
Line 6.  End if:
Line 7.  Compute $\mathbf{F}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c]$ where
         $\mathbf{F} = \mathbf{F}_{BE}$, $\mathbf{F}_{TR}$ or $\mathbf{F}_{TR\text{-}BDF2}$
```

5.2. Nonlinear solver

The nonlinear Stokes problem are solved by the Newton-based solver in a globalized JFNK framework [21]. To define the nonlinear solver, for clarity we will refer to the nonlinear residual as $\mathbf{F}$. Noting that depending on the choice of time integration scheme used, $\mathbf{F}$ will be defined via Eq. (17) ($\mathbf{F}_{BE}$: backward Euler), Eq. (18) ($\mathbf{F}_{TR}$: trapezoidal rule), or Eq. (19) ($\mathbf{F}_{TR\text{-}BDF2}$: trapezoidal rule—backward difference formula).

The Newton correction without forming the Jacobian can be expressed as

$$\hat{\mathbf{J}}\delta\mathbf{X}' = -\mathbf{F}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c], \tag{20}$$

with an inexact Jacobian $\hat{\mathrm{J}}_{ij} \approx \mathrm{J}_{ij}[\mathbf{X}'] = \partial\mathbf{F}_i[\mathbf{X}']/\partial\mathbf{X}'_j$. We solve Eq. (20) for the Newton direction $\delta\mathbf{X}'$ to update $\mathbf{X}'$. The simple line search approach [22] is applied for the globalization to calculate the step size s which is used in

$$\mathbf{X}' = \mathbf{X}' + s\delta\mathbf{X}'. \tag{21}$$

The Newton iteration is deemed to be converged when the nonlinear residual satisfies the following:

$$\|\mathbf{F}\|_2/\|\mathbf{F}_0\|_2 < eitr, \tag{22}$$

where $\|\mathbf{F}_0\|_2$ is the 2-norm of the initial residual (i.e. the problem dependent). The algorithm for the globalized Newton-based scheme of MIC method is shown in Algorithm 2.

```
Algorithm 2
Line 1.   $n = 0$: Given $\mathbf{X}^0$ and $\mathbf{x}^0_m$, $\boldsymbol{\phi}^0_m$ and $\Delta t$
Line 2.   Compute $\mathbf{F}_0$
Line 3.   While not converged:
Line 4.     Solve Linearized problem $\hat{\mathbf{J}}\,\delta\mathbf{X}' = -\mathbf{F}$:
          (Evaluate $\mathbf{F}$ using Algorithm 1)
Line 5.     Compute step size $s$
Line 6.     Update: $\mathbf{X}' = \mathbf{X}' + s\,\delta\mathbf{X}'$
Line 7.   End While:
Line 8.   If TR-BDF2 method is used then:
Line 9.     Update marker coordinate $\mathbf{x}^{n+\frac{1}{2}}_m$ and goto
          Line 3 with $\mathbf{F} = \mathbf{F}_{TR\text{-}BDF2}$
Line 10.  End if:
Line 11.  Update marker coordinate $\mathbf{x}^{n+1}_m$
Line 12.  $\mathbf{X}^{n+1} = \mathbf{X}'$
Line 13.  $n = n + 1$: goto Line 2
```

In the JFNK approach, the matrix free product of operator $\hat{\mathbf{J}}$ is given by

$$\hat{\mathbf{J}}\delta\mathbf{X}' = (\mathbf{F}[\mathbf{X}' + \epsilon\,\delta\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] - \mathbf{F}[\mathbf{X}', \boldsymbol{\eta}_c, \boldsymbol{\rho}_c])/\epsilon, \tag{23}$$

where the parameter $\epsilon = \sqrt{(1 + \|\mathbf{X}'\|)\epsilon_m}/\|\delta\mathbf{X}'\|$ with $\epsilon_m = 10^{-6}$ [23]. The linear solve of Eq. (20) and Line 4 of the Algorithm 2 is performed using FGMRES [24], which is restarted every five iterations. Krylov iterations are terminated when the 2-norm of the initial residual is reduced by a factor of 10^2. Although the choices of parameters in the matrix free product and Krylov subspace method affect the performance of our nonlinear solver, such side effects are limited and do not change our performance analysis drastically. We use the operator $\mathbf{A}[\boldsymbol{\eta}_c]$ defined in Eq. (11) as the preconditioner for the true Jacobian. Note that by replacing the operator $\hat{\mathbf{J}}$ defined by the JFNK framework with $\mathbf{A}[\boldsymbol{\eta}_c]$, we can convert the Newton algorithm into a Picard nonlinear solver, written in defect correction form [21].

The convergence of a globalized Newton-based scheme is highly dependent on the starting vector used. From numerical experimentation, we found that the robustness of time-dependent simulations can be improved by switching between $\hat{\mathbf{J}}$ and $\mathbf{A}[\boldsymbol{\eta}_c]$ alternately during the nonlinear iterations. Generally, efficient convergence is observed when Picard/JFNK is applied to nonlinear systems with initially large/small residuals, respectively. However the best choice of switching point is highly problem dependent. Thus we have switched between $\hat{\mathbf{J}}$ and $\mathbf{A}[\boldsymbol{\eta}_c]$ every five nonlinear iterations. CPU time was used as heuristic to determine the order to be applied between Newton–Picard–Newton $\cdots$ (Newton first) and Picard–Newton–Picard $\cdots$ (Picard first). Every 20 time steps, we changed the order to the alternate order, i.e. Newton first or Picard first, and compared CPU time required with that required by the previous time step. Here, we assume that the problems of neighboring time steps have similar numerical difficulty. Then, the better ordered method relative to CPU time is utilized for the next 20 time steps.

The linear problem defined in Eq. (11) (used by Exp) and application of the preconditioner used with our JFNK framework are both solved via the sparse direct solver PARDISO from the Intel MKL library [25].

6. Coupling with energy equation

As an extension to thermal convection of Boussinesq fluids operating in an infinite Prandtl number regime, we consider the coupling of the following simple energy equation in Lagrangian frame to the mechanical system discussed in Section 2:

$$\frac{d\mathbf{T}_m}{dt} = \kappa\nabla^2\mathbf{T}_m, \tag{24}$$

where, for simplicity, κ is the thermal diffusivity taken as a constant for the entire domain. Variations in the density associated with temperature perturbations (Boussinesq approximation) are introduced via

$$\boldsymbol{\rho}_m = \boldsymbol{\rho}_0[\boldsymbol{\phi}_m]\,(1 + \alpha\mathbf{T}_m)\,. \tag{25}$$

The change in the temperature field of Eq. (24) denoted $\Delta\mathbf{T}_c$ is evaluated on the grid cell with $\mathbf{T}_c$ given by

$$\Delta\mathbf{T}_c = \kappa\,\mathbf{L}\mathbf{T}_c, \tag{26}$$

where $\mathbf{L}$ is the discretized matrix of the Laplacian by the FD method. To update the temperature on the markers, we calculate

$$\mathbf{T}_m^{n+1} = \mathbf{T}_m^n + \Delta t\,\Delta\mathbf{T}_m^n, \tag{27}$$

where the linear interpolation is used for $\Delta\mathbf{T}_m = \Pi[\Delta\mathbf{T}_c]$, for the one-step methods discussed in Section 3.2. Although the subgrid diffusion procedure on markers for $\Delta\mathbf{T}_m$ is proposed by [15] in computational geodynamics field, we do not apply that for simplicity of error analysis in this work. For the two-step methods of Section 3.3, we calculate

$$\mathbf{T}_m^{n+\frac{1}{2}} = \mathbf{T}_m^n + \frac{1}{2}\Delta t\,\Delta\mathbf{T}_m^n, \tag{28}$$

$$\mathbf{T}_m^{n+1} = \mathbf{T}_m^n + \frac{1}{2}\Delta t\left(\Delta\mathbf{T}_m^n + \Delta\mathbf{T}_m^{n+\frac{1}{2}}\right). \tag{29}$$

These are explicit Runge–Kutta methods which are first and second order accurate for Eq. (27) and Eqs. (28), (29), respectively. Different from the solution of the mechanical equations of Eqs. (1), (2), (4), (5), we can assume that this explicit time-stepping scheme will not cause spurious solutions because the diffusion time scale of our target problem is much larger than the dynamical time scale of the Stokes flow.

7. Numerical experiments

We conducted numerical experiments with two different buoyancy-driven processes to assess the performance of the different time integration techniques introduced. The first experiment focused on the relaxation of large wavelength surface deformation under gravity ("bump relaxation"). This test was used to evaluate the performance of the proposed techniques for regional scale simulations of subduction and mountain building. The second example we studied considers thermal evolution under the influence of a radial gravity field. Here the fluid domain is time-dependent as we assume a free surface boundary condition on the perimeter of the fluid domain. These experiments investigated the coupling between various wavelength topographic anomalies coupled with an energy equation. This problem scenario is of relevance for planetary core formation (e.g. [7,8,26]).

In both experiments, the domain is two dimensional and defined by $\Omega = 1 \times 1$. The FD mesh employed to solve the Stokes problem utilized $N \times N$ control volumes in the x_1 and x_2 directions respectively. On the boundary of the FD mesh, we applied free slip boundary conditions. The free surface boundary condition approximated by surrounding low viscosity fluid with zero mass density, so-called "sticky-air" technique (see Section 4). Each control volume in the FD mesh was populated with markers. Markers were regularly spaced within each control volume with a displacement $\Delta x/6$, where $\Delta x = 1/N$ is the size of each control volume. The time step size Δt is taken as constant.

The presented quantities are reported in their non-dimensional form. We employed the following scales $x_i = x_i'/l$, $t = t'\Delta\eta/(4\pi G\Delta\rho^2 l^2)$, $\rho = \rho'/\Delta\rho T = T'/\Delta T, \eta = \eta'/\Delta\eta$ and gravity acceleration $g = 9.8/(4\pi G\Delta\rho l)$ where G is the gravitational constant [27].

Computational experiments were performed using 16 cores of a Xeon(R) E5-2650 v2 CPU.

7.1. Bump relaxation test

Fig. 3(a) shows the initial setting of the bump relaxation test consisting of two layers labeled $\phi = 1$ and $\phi = 2$. In this test, we assumed that $\alpha = 0$ for each material (Eq. (25)), thereby decoupling the energy transport from the momentum equations. Markers located below the cosine curve given by

$$x_2 < h_0 - \delta h\cos(k_0x_1), \tag{30}$$

where $h_0 = 0.5$, $\delta h = 0.025$ and $k_0 = 2\pi$ are the initial thickness, perturbed amplitude and wavenumber, represent the bottom dense material ($\phi = 2$). Otherwise markers are assumed to represent the sticky-air layer ($\phi = 1$). In the simulation, we observed gradual relaxation of the bump to a flat shape under the gravity $(0, -g)$. We monitored the marker position $(0.5, h_p(t))$ at

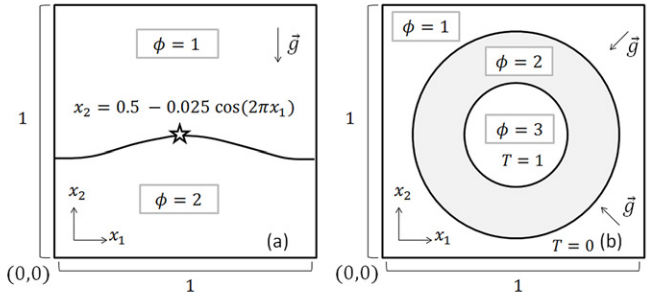

Fig. 3. Schematic of (a) bump relaxation test (Section 7.1) and (b) free surface thermal evolution test (Section 7.2). The star shown in (a) is used to monitor the peak topography in Figs. 4–7.

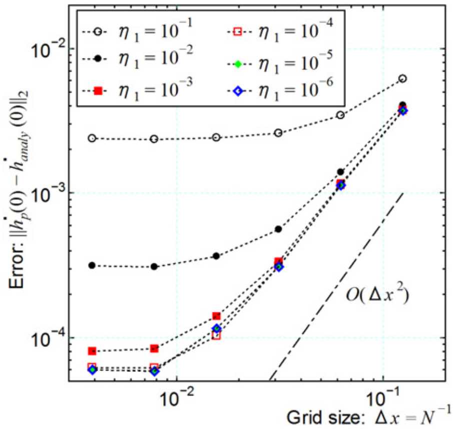

Fig. 4. Deviation of the velocity from analytic decay at peak topography position for the convergence test (Section 7.1). Errors are plotted against grid size N and sticky-air viscosity η_1.

Table 1
Non-dimensional material parameters used in numerical experiments (bump relaxation, spherical thermal evolution).

ϕ	Bump test Section 7.1		Thermal test Section 7.2	
	η	ρ_0	η	ρ_0
1	10^{-3}	0	10^{-2}	0
2	1	0.8	1	0.8
3	–	–	10^{-2}	1.6

the top of the bump (denoted by the star in Fig. 3(a)) to evaluate the topographic change.

The solution of this bump relaxation problem with small sticky-air viscosity approaches the exponential decay of topographic anomaly [13], using

$$\tau = \left(\frac{h_0k_0 + \sinh(h_0k_0)\,\cosh(h_0k_0)}{\sinh^2(h_0k_0)}\right)\left(\frac{2k_0\eta_2}{\rho_2 g}\right) = 64.43. \qquad (31)$$

With this decay time, the analytic top topography is given by $h_{analy}(t) = h_0 + \delta h\, e^{-t/\tau}$.

To check the convergence property, we first compared the initial velocity at the top of the bump $\dot{h}_p(t = 0)$ for various grid size N and sticky-air viscosity η_1. Fig. 4 shows the deviations of the numerical solution from the analytical velocity $\dot{h}_{analy}(t = 0)$ with $\rho_1 = 0$, $\rho_2 = 1$ and $\eta_2 = 1$. From the result, we used the model with $\eta_1 = 10^{-3}$ and $N = 128$ to analyze each time integration method with numerically converged solution to the analytic decay in the bump test. The material parameters used for each phase are also provided in Table 1.

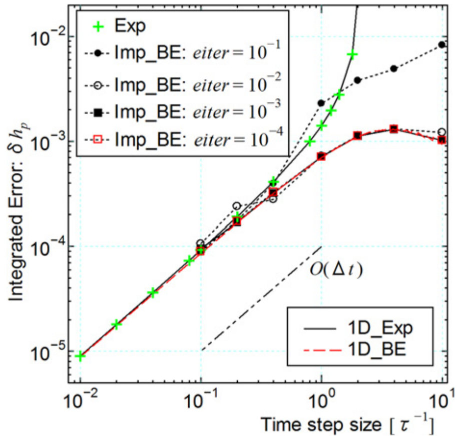

Fig. 5. Peak topography error from the analytical solution (δh_p of Eq. (32)) with first order accurate time-stepping (Exp and BE) methods (Section 7.1). The solutions for 1D IVP (i.e. 1D_Exp, 1D_BE) are also plotted for comparison.

To validate the accuracy of the bump test, we introduced the error integrated in time defined by

$$\delta h_p = \frac{1}{t^{n_{\max}}}\left(\sum_{n=1}^{n_{\max}} \left\| (h_p(t^n) - h_{analy}(t^n)) \right\|_2\right), \qquad (32)$$

where $t^n = n\Delta t$ and $n_{\max}$ is the number of time steps required to reach the target time with $t_{end} = 50\tau$ (i.e. $t^{n_{\max}-1} < t_{end} \le t^{n_{\max}}$). We evaluated the error not only for our 2D calculation result but also for the solution of the 1D initial value problem (IVP) of the exponential decay (Appendix B) to observe the consistency with ODE theory.

We denote the solution schemes using the Imp, SImp and 1D IVP, as Imp_*tstep*, SImp_*tstep* and 1D_*tstep*, respectively, for each time-stepping method *tstep* = BE, TR or TR-BDF2. The explicit method for 1D IVP is also referred to as the 1D_Exp.

The stopping condition for the nonlinear solver of Eq. (22) is given by

$$\mathbf{F}_0 = \mathbf{F}[\mathbf{X}_0, \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] \quad \text{where } \mathbf{X}_0^T = (\mathbf{0}, \mathbf{0}). \qquad (33)$$

7.1.1. Explicit method (Exp): bump test

The errors by the Exp method for several different Δt are plotted in Fig. 5. Note that our solution agrees with that of 1D_Exp. Following ODE theory, the solution obtained by the Exp demonstrates the first order accuracy for small step size $\Delta t/\tau < 1$, deviates from the first order for $1 \le \Delta t/\tau < 2$, and diverges at $\Delta t/\tau \ge 2$.

Fig. 6 shows the marker position against several different Δt. The solution by the Exp with small time step ($\Delta t/\tau = 0.1$) shows that the peak topographic height decays via viscous relaxation and then maintains a constant value for the remainder of the experiment. With $\Delta t/\tau = 2$, the peak topography initially collapses far faster than the analytic solution. It then oscillates strongly to achieve balance but ultimately fails to reach the steady state.

7.1.2. Implicit method (Imp): bump test

Next, we examined the solution using the Imp methods. Our implicit solvers based on a nonlinear iterative method have two control parameters, i.e. nonlinear relative residual tolerance (denoted *eitr* in Eq. (22)) and time step Δt. The smaller *eitr* or Δt are the higher the accuracy of the solution; however, the calculation is more expensive. We first examined the quality of solutions as a function of *eitr* and different Δt values.

Fig. 5 shows the error using the Imp_BE with different *eiter* values. For all examined Δt, *eiter* $= 10^{-3}$ was sufficiently small to

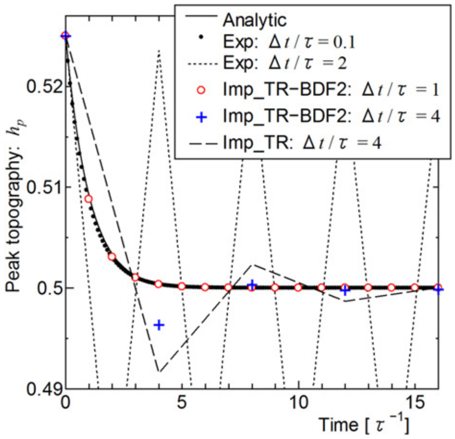

Fig. 6. Vertical coordinate history of the star marker (Fig. 3) by different time integration methods in the bump relaxation test (Section 7.1).

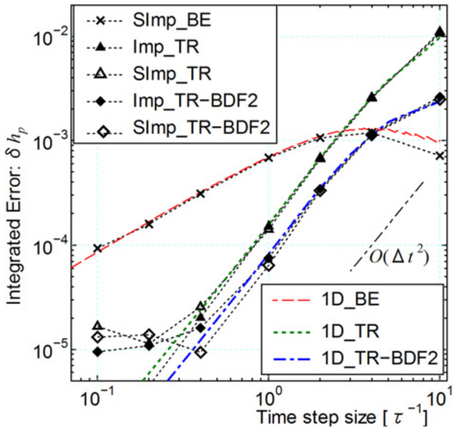

Fig. 7. Peak topography error from the analytical solution (δh_p of Eq. (32)) with second order accurate Imp and SImp methods (i.e. TR and TR-BDF2) (Section 7.1). The solutions for 1D IVP (i.e. 1D_BE, 1D_TR, 1D_TR-BDF2) and SImp_BE are also plotted for comparison.

agree with the 1D_BE solution and converged to the solution with *eiter* $= 10^{-4}$. Therefore, we use *eiter* $= 10^{-3}$ as the reference tolerance of the nonlinear solver for the implicit methods. Compared with the Exp method with $\Delta t/\tau > 1$, the BE formulation allows the use of larger time steps without compromising accuracy due to the oscillatory behavior. On the other hand, the accuracy of the Imp_BE is the first order as same as the Exp method.

For a higher order convergence method, we examined second order TR and TR-BDF2 schemes. The errors obtained by the second order Imp methods are plotted in Fig. 7. We can confirm numerically that the error curves for both integration methods are second order accurate and fitted to the 1D IVP solutions. The second order methods have a clear advantage in that they can achieve a certain solution accuracy at a larger time step size than that required by the first order Exp or BE methods.

The Imp_TR-BDF2 shows better convergence than the Imp_TR at the same step size. However note that the two step Imp_TR-BDF2 requires almost twice the calculation process than the Imp_TR. Thus, the advantage of using Imp_TR-BDF2 instead of Imp_TR is the oscillatory behavior of the solution at a large time step rather than the better accuracy for $\Delta t/\tau < 2$. With large Δt, the solution obtained using the TR scheme deviates significantly from the analytic solution with numerical oscillation. On the other hand, the TR-BDF2 is L-stable, same as BE, thus such deviation is saturated for larger time step size over $\Delta t/\tau \geq 4$ (note that the BE works much better at these large time step). The difference between Imp_TR and Imp_TR-BDF2 is also evident in the marker position shown in Fig. 6. The solution obtained using Imp_TR-BDF2 shows that the oscillation equilibrates rapidly to steady state.

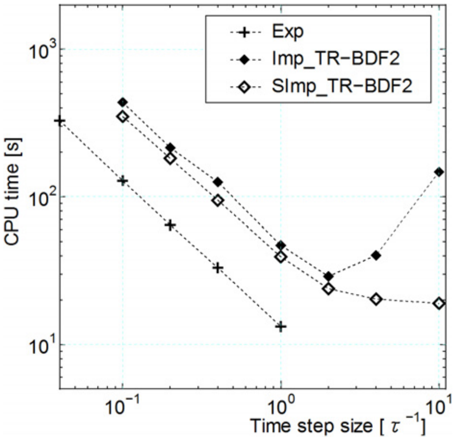

Fig. 8. CPU time required to evolve the bump relaxation test for $t_{end} = 50\tau$ (Section 7.1.4).

From these observations, we selected the Imp_TR-BDF2 as the standard implicit time integration scheme to obtain efficient second order convergence and avoid drastic degradation of solution accuracy at a large time step. In the reminder of this paper, we focus on the performance of the implicit solution methods with the TR-BDF2 time-stepping scheme.

7.1.3. Semi-implicit method (SImp): bump test

We examine the SImp method solved by the TR-BDF2 scheme (SImp_TR-BDF2) with nonlinear tolerance *eiter* $= 10^{-3}$. In the bump test, we show the result for the first order upwind FD method because we could not find the significant difference between first and fifth order FD methods. The observed errors using SImp_TR-BDF2 are shown in Fig. 7 (error using SImp_BE and SImp_TR are also plotted for reference). The results obtained using SImp_TR-BDF2 agree well with the Imp_TR-BDF2 solutions, indicating that FD advection captures the nonlinearity by the marker update successfully.

7.1.4. Computational performance comparison: bump test

Here, we compare the computational efficiency of the Exp and implicit TR-BDF2 methods. The computational cost is a function of the number of time step, and the CPU time required to perform each time step. CPU time per time step can be further decomposed as follows: (number of iterations of nonlinear solver) × (cost per nonlinear iteration).

In Fig. 8, we compare the CPU time required by different solution methods to reach $t_{end} = 50\tau$. Using time step $\Delta t/\tau = 0.1$, the ranking of CPU time is as follows: time(Exp) < time(SImp_TR-BDF2) < time(Imp_TR-BDF2). The high computational cost for the nonlinear residual evaluation of the Imp method results in the greatest CPU time. The SImp_TR-BDF2 is the second fastest owing to a less expensive nonlinear residual evaluation.

Per time step, the Exp method is the least expensive but use of a large time step with the Exp method is not possible due to the bounded stability region. On the other hand, the implicit solver is more expensive than the explicit method per time step. However, this additional cost comes with an advantage, i.e. the implicit method's time integrator can restrict the selection of the time step only on the basis of an acceptable temporal error. As for the error, the second order accuracy of the TR-BDF2 method achieves the same solution quality with larger time step size than that using the first order Exp method. In the bump test, the solutions at $\Delta t/\tau = 0.1$ by the Exp method and that at $\Delta t/\tau = 1$ by the implicit TR-BDF2 methods are similar in terms of $\delta h_p \sim 10^{-4}$. Although total elapsed CPU time depends on the cost of the nonlinear iteration, the use of Imp_TR-BDF2 with $\Delta t/\tau = 1$ results in a reduction of approximately 70% CPU time compared with the Exp method at $\Delta t/\tau = 0.1$, as shown in Fig. 8. Further reduction of the cost for the implicit method is obtained by the SImp_TR-BDF2 using a cheaper nonlinear solver than that associated with the Imp, as the nonlinear residual is evaluated using an Eulerian FD method, as opposed to the more computationally expensive marker-mesh remapping procedure by Eq. (15).

For $\Delta t/\tau > 2$, the Imp_TR-BDF2 fails to reduce CPU time with increasing step size and the difference between the cost of Imp_TR-BDF2 and SImp_TR-BDF2 increases. This observation is derived from the fact that the convergence of the nonlinear problem becomes increasingly more difficult as Δt increases; therefore, the number of nonlinear iterations which indicates the cost difference for Imp_TR-BDF2 and SImp_TR-BDF2, increases. This additional trade-off also indicates that the use of a large Δt does not always guarantee reduction in CPU time, even though time integration is stable for large Δt.

7.2. *Free surface thermal evolution test*

Here we discuss the numerical experiments of a thermally driven convecting fluid, surrounded by a low viscosity fluid. The initial material geometry with three layers is illustrated in Fig. 3(b). We assume that a radial gravity acceleration is given by $g_i = -g\hat{r}_{cen}$ where $\hat{r}_{cen}$ is the unit vector of $r_{cen} = (x_1 - 0.5, x_2 - 0.5)$. Although a self-gravitating force is commonly employed for g_i in the planetary problems [7,8,26], we simply use constant acceleration to focus on solution quality and solver performance.

The inner most layer ($\phi = 3$) consists of a high density and low viscosity fluid. The temperature of $\phi = 3$ is fixed at $T = 1$. This region mimics the planet's core, which allows us to approximate a stress free and hot thermal boundary condition of the middle layer. The middle layer ($\phi = 2$) represents a simplified mantle layer. We are primarily interested in studying the dynamics of the middle layer. The initial temperature of the middle layer is given by

$$T_{init} = \frac{\log(|r_{cen}|/0.4)}{\log(0.5)} - 0.01\cos\left(\frac{k_0\theta}{2\pi}\right)\sin\left(\frac{2\pi r_{cen}}{0.4}\right) \tag{34}$$

where the second term on the right hand side is the initial perturbation with $\cos(\theta) = x/r_{cen}$ and wavenumber $k_0 = 6\pi$ [28]. The outer most layer ($\phi = 1$) is the sticky-air layer, which has zero density, zero temperature $T = 0$ and a very low viscosity in order to numerically approximate the free surface condition of the middle layer.

In this series of experiments, we solved Stokes equations coupled with the energy equation of Eq. (24) using the grid size $N = 128$, a normalized thermal diffusivity of $\kappa = 8.01 \times 10^{-11}$ and thermal expansion coefficient $\alpha = 5 \times 10^{-2}$. Other properties for each layer are listed in Table 1. When we assume $l = 14,000$ km, $\Delta\eta = 4.3 \times 10^{22}$ Pa s, $\Delta\rho = 5000$ kg/m^3 and $\Delta T = 5000$ K, parameters result in the Rayleigh number $Re = \rho'_{\phi=2} g(\alpha/\Delta T)(0.2l)^3 \Delta T/(\eta'_{\phi=2}(\kappa\Delta\eta/(4\pi G\Delta\rho^2))) = 10^6$. To evaluate the deviation from the hydrostatic balance, the stopping condition for the nonlinear solver of Eq. (22) at the nth time step is defined with

$$\mathbf{F}_0 = \mathbf{F}[\mathbf{X}_0, \boldsymbol{\eta}_c, \boldsymbol{\rho}_c] \quad \text{where } \mathbf{X}_0^T = (\mathbf{u}_g = \mathbf{0}, \mathbf{p}_c^{n-1}). \tag{35}$$

Once we began the simulation, upwelling plumes rose from the inner layer to the outer layer as shown in Fig. 9(a). At the early stage of calculation, the growth of the initial perturbation dominated the upwelling plumes, but various wavelength instabilities evolved to create a complicated convection mode under the employed *Re* with the current of times. Thus the symmetry of the temperature profile was found to be slightly broken at $t = 4 \times 10^5$ as shown in Fig. 9(b–d).

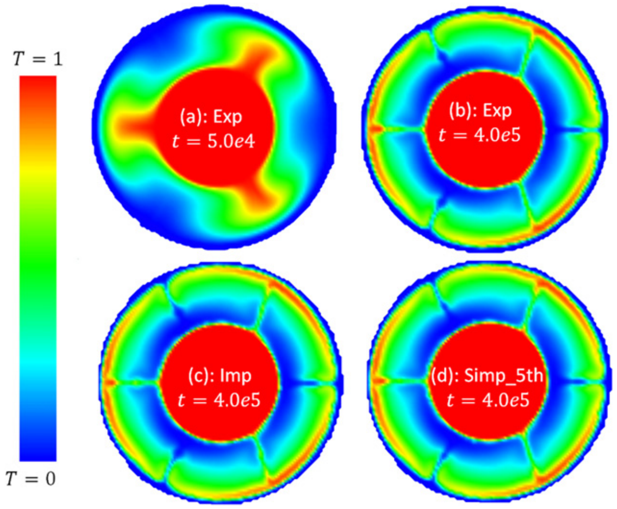

Fig. 9. Time development of temperature profile and free surface within the thermal evolution test Section 7.2. (a) Upwelling plumes at $t = 5 \times 10^4$ by Exp with $\Delta t/\tau = 0.2$. (b) Thermal state at $t = 4 \times 10^5$ by Exp with $\Delta t/\tau = 0.2$. (c) Thermal state at $t = 4 \times 10^5$ by Imp with $\Delta t/\tau = 4$. (d) Thermal state at $t = 4 \times 10^5$ by SImp_5th with $\Delta t/\tau = 4$.

To analyze each time-stepping method, we evaluated the error in the following form:

$$\delta u_{ave} = \frac{1}{n_{\max}}\left(\sum_{n=1}^{n_{\max}} \frac{\left\| (u_{ave,\Delta t}^{method}(t^n) - u^{REF}(t^n)) \right\|_2}{u^{REF}(t^n)}\right), \tag{36}$$

where $u_{ave,\Delta t}^{method}(t)$ is the averaged velocity on the FD grid over the computational domain as a representative value of the convective motion of the whole layer. We regard the solution obtained by TR-BDF2 at time step size $\Delta t/\tau = 0.001$ as the reference solution of $u^{REF} = u_{ave,\Delta t/\tau=0.001}^{Imp}$ in Eq. (36) because it is the most accurate numerical solution examined in this study. Here we use τ discussed in Section 7.1 as the characteristic decay time of surface deformation to represent the time step size Δt. In the thermal test, we employed the TR-BDF2 method as the implicit time integration scheme. Accordingly, we denote Imp_TR-BDF2 and SImp_TR-BDF2 as Imp and SImp, respectively, in the following subsections.

7.2.1. *Explicit method (Exp): thermal test*

First, we solve the problem using the explicit method. Fig. 10 shows the error deviation δu_{ave} of the averaged velocity for different time step sizes until $t = 2.0 \times 10^4$. We observed that the planetary body exhibits spurious vibrations when using a time step of $\Delta t/\tau > 0.2$ for the Exp. Since the computational time step size is apparently smaller than the time scale of the thermal diffusion (i.e. $\Delta t_{diff} = \Delta x^2/(2\kappa) = 3.8 \times 10^5 \gg \tau = 64.43$), this oscillation is not an artifact of the explicit time-stepping of the energy equation but from the out of balanced problem of the free surface. On the other hand, unphysical behavior is no longer observed for the solutions at $\Delta t/\tau \le 0.2$ and first order temporal accuracy is attained. This upper limit of Δt for the non-oscillatory behavior is smaller than that for the bump test $\Delta t/\tau \le 1.0$ because this problem deals with shorter wavelength than the bump test.

7.2.2. *Implicit method (Imp): thermal test*

First we determined an appropriate implicit solution parameter from δu_{ave} at $t = 2.0 \times 10^4$ calculated by the Imp for several different values of *eitr* and Δt as shown in Fig. 10. The deviation of the solutions converges with small *eiter*. The

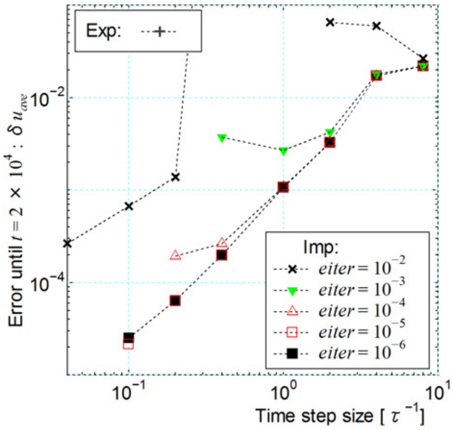

Fig. 10. Error in the average velocity δu_{ave} at $t = 2.0 \times 10^4$ by the Exp and Imp methods with various tolerance *eiter* in free surface thermal evolution test (Section 7.2.2). Results by Imp show convergence properties with implicit solution parameters.

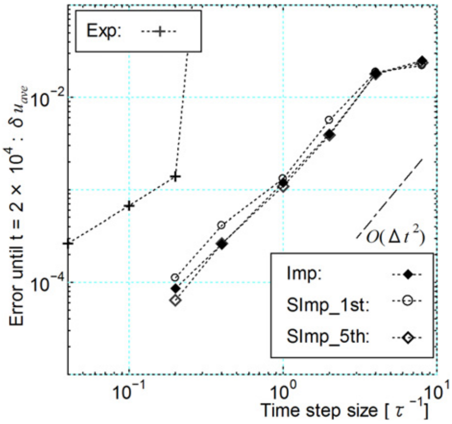

Fig. 11. Error in the average velocity δu_{ave} at $t = 2.0 \times 10^4$ with Exp, Imp, SImp_1st and SImp_5th methods in free surface thermal evolution test (Section 7.2). Implicit solutions are obtained with *eitr* $= 10^{-2}$ for $\Delta t/\tau = 8$, *eitr* $= 10^{-3}$ for $\Delta t/\tau = 4$, *eitr* $= 10^{-3}$ for $\Delta t/\tau = 2$, *eitr* $= 10^{-4}$ for $\Delta t/\tau = 1$ and *eitr* $= 10^{-5}$ for $\Delta t/\tau = 0.4$ and 0.2.

implicit solution with smaller Δt is found to require smaller tolerance for convergence. This means that the suitable tolerance *eitr* for efficient computational performance should be distinct for different Δt. The observed relationship suggests that to avoid the use of *eitr* that is too small at large Δt, the proper solution parameters in this test are (*eitr* $= 10^{-2}$ for $\Delta t/\tau = 8$), (*eitr* $= 10^{-3}$ for $\Delta t/\tau = 4$), (*eitr* $= 10^{-3}$ for $\Delta t/\tau = 2$), (*eitr* $= 10^{-4}$ for $\Delta t/\tau = 1$) and (*eitr* $= 10^{-5}$ for $\Delta t/\tau = 0.4$ and 0.2). In the following thermal tests, these tolerances were used for the nonlinear solver of the implicit solution methods.

The converged solutions by the Imp with these *eiter* values are replotted in Fig. 11, which shows close to the second order accuracy. At a similar level of accuracy, the Imp results in a larger time step than the Exp method. In addition, unlike the Exp method, the Imp scheme does not largely degrade the quality of solution for $\Delta t/\tau > 0.2$.

To see the impact of the error contribution of δu_{ave} in global scale phenomena, the temperature field obtained using the Imp with $\Delta t/\tau = 4$ at $t = 4.0 \times 10^5$ is illustrated in Fig. 9(c). In Fig. 11, the δu_{ave} obtained by the Imp with $\Delta t/\tau = 4$ is greater than that obtained by Exp with $\Delta t/\tau = 0.2$ at $t = 2.0 \times 10^4$. However, their thermal fields are visually consistent (see Fig. 9(b) for reference) because the transient surface relaxation process of small scale bumps does not play an important role for global scale phenomena. This suggests that the acceptable temporal error to capture thermal evolution surrounded by a free surface can be greater than the upper limit handled by the Exp method with $\Delta t/\tau = 0.2$.

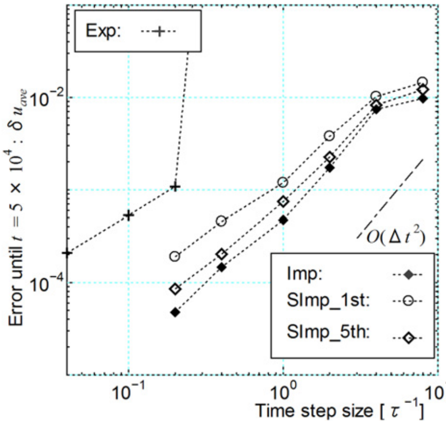

Fig. 12. Plot of the error δu_{ave}. Same as Fig. 11 but for integration time of $t = 5.0 \times 10^4$ (Section 7.2).

7.2.3. Semi-implicit method (SImp): thermal test

In this test, two types of Euler advection methods were examined to evaluate the nonlinear residual (see Section 5.1.1 and Appendix A). Here, we denote the SImp with the first and fifth order upwind methods as SImp_1st and SImp_5th, respectively.

Fig. 11 shows the numerical convergence errors at $t = 2.0 \times 10^4$ obtained by the Simp methods. The SImp_1st solution is found to deviate from that obtained by the Imp because thermal instabilities growing at the high wavenumber range are difficult to resolve using the first order FD advection scheme. The FD advection schemes inherently show dispersion and dissipation errors to transport Fourier components at a high wavenumber range (e.g. [29,30]), although, essentially such errors do not appear with the marker advection of the Imp. The deviation between the Imp and SImp_1st is not found in the bump test because the bump test only exhibits the growth of initial low wavelength mode. Since higher order FD methods are less diffusive and thus can capture a wider wavenumber range, the fifth order upwind method is expected to reduce such dispersion errors. In Fig. 11, the solution of the SImp_5th agrees well with that of the Imp at $t = 2.0 \times 10^4$. The improvement achieved using the fifth order FD method, compared with the first order method, is clearly evident. From these results, resolving higher wave modes by a lower diffusive advection scheme is found to improve the Simp solution quality.

However, when time is increased to $t = 5.0 \times 10^4$, the SImp_5th solution begins to deviate from the Imp solution as shown in Fig. 12, although SImp_5th shows better agreement with the Imp than SImp_1st. This is due to the growth of instability at much higher wavenumbers, which cannot well-resolved by the fifth order method. We can expect further improvement by using a more sophisticated lower diffusive advection method (e.g. [30–32]).

Fig. 9(d) shows the temperature profile at $t = 4 \times 10^5$ for the SImp_5th with $\Delta t/\tau = 4$. This structure and the structure using the Imp as shown in Fig. 9(c), are approximately the same. The fine scale structures differ slightly due to the numerical diffusivity in the temporal Eulerian advection of the SImp_5th. However the difference for the whole convective structures is negligible, and the numerical convergence against Δt and the higher order advection scheme are observed in the numerical error analysis. These results justify the use of the SImp_5th to obtain the similar solution qualities of the Imp.

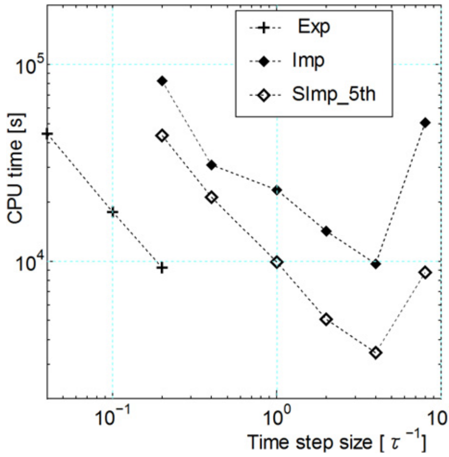

Fig. 13. CPU time required to evolve the free surface thermal evolution test forward in time until $t = 4 \times 10^5$ (Section 7.2.4).

7.2.4. Computational performance comparison: thermal test

We compared the CPU times required to reach time $t = 4 \times 10^5$ for different time integration schemes, i.e. Exp, Imp and SImp_5th in Fig. 13. The Exp shows the best cost performance without numerical solution oscillations at time step $\Delta t/\tau = 0.2$. With this time step size, the Exp is the fastest among the examined time integration methods.

The quality of solutions for the implicit methods with $\Delta t/\tau = 1$ were similar to that using the Exp with $\Delta t/\tau = 0.1$ in Fig. 12. When we compare the CPU time at $\Delta t/\tau = 1$, the SImp_5th shows the best performance due to reduction in CPU time associated with the calculation cost of the nonlinear solver. The SImp_5th spends approximately half of the CPU time required for the Exp at $\Delta t/\tau = 0.1$, although the Imp requires the approximately the same CPU time due to the cost increase incurred by the nonlinear solver.

Moreover, given the consistency between Fig. 9(b) and Fig. 9(c), (d), the errors for $\Delta t/\tau = 0.2$ for the explicit and $\Delta t/\tau = 4$ for the implicit methods can be an acceptable (i.e. $u_{ave} < 10^{-2}$ in Fig. 12). With these solution parameters, the SImp_5th is found to reduce approximately 65% CPU time compared with the Exp. On the other hand, it is still difficult for the Imp to outperform the Exp method.

Fig. 14 shows the detailed cost balance of each simulation for the Exp method with $\Delta t/\tau = 0.2$, the Imp with $\Delta t/\tau = 4$ and the SImp with $\Delta t/\tau = 4$. Since the Exp method requires many more time steps, overall most CPU time is spent in the linear solver. In contrast, the Imp shows smaller cost for the linear solver; however, the nonlinear residual evaluation employing the expensive remapping procedure dominates the execution time. We argue that the trade-off between implicit and explicit solver will become more apparent as the mesh spacing decreases. The SImp shows the reduced cost not only from using large Δt, but also from the inexpensive nonlinear residual evaluation.

8. Conclusions

We have examined several types of implicit schemes of the MIC method, which were applied to Stokes flow problems employing time-dependent material properties and an approximate free surface. From numerical experiments, we observed the following.

1. It is difficult to solve stiff problems with the sticky-air free surface deformation using the explicit time integration scheme because the available time step is bound by the decay time of the transient free surface relaxation. Our implicit material transport techniques with wide stability region can handle large Δt over such short decay time without numerical oscillation of the solution.
2. In the numerical experiments, the observed accuracy and stability of the implicit time integration methods are consistent with those obtained by ODE theory (i.e. first or second order, with or without L-stability). The TR-BDF2 method is attractive in terms of second order accuracy and is L-stable for solving the stiff free surface problems. For a target solution accuracy, the second order implicit methods can use larger Δt than the first order explicit method.
3. There is a complicated, problem dependent trade-off between the increasing cost of the nonlinear solver and the reduced number of time steps required by using large Δt values. In both applications examined, the implicit time integration succeeds in reducing total calculation cost using large Δt to obtain similar solutions of fully explicit methods with small Δt.
4. The Imp method which uses markers throughout the nonlinear solver gives stable and oscillation-less behavior for large Δt using L-stable time integration scheme. The cost of evaluating the nonlinear residual dominates total CPU time.
5. The SImp method advects quantities in the Eulerian frame to evaluate the nonlinearity associated with material transport. The SImp method produced solutions similar to those obtained by the Imp method in our experiments. Although the Euler advection method may generate a numerically diffusive error at a high wavenumber mode, the use of a higher order FD scheme could improve such degradation of solution accuracy. Compared with the Imp, the SImp can reduce the cost of the nonlinear solver significantly because the residual evaluation is less expensive.

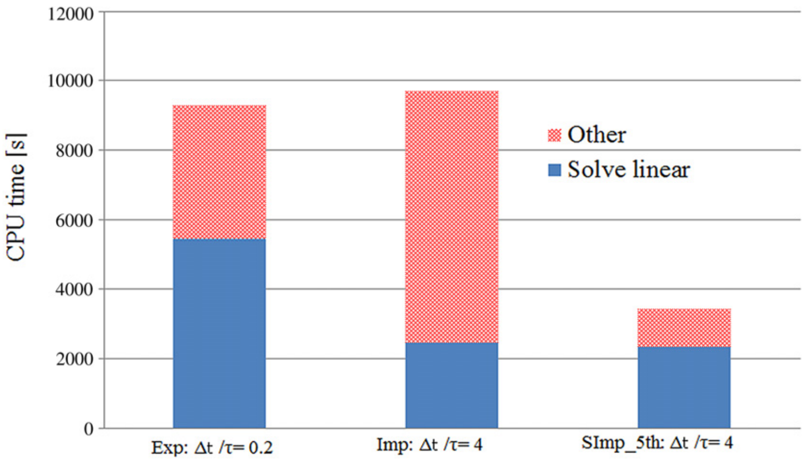

Fig. 14. Break down of the execution time for different methods when applied to the thermal evolution experiment (Section 7.2.4).

In summary, the implicit material transport implemented by the nonlinear iterative solver can reduce the CPU time required to solve buoyancy driven processes that possess a free surface. An enlarged stability domain, second-order accuracy, and the resulting larger time steps permitted were found to be beneficial, even though the implicit algorithms require the solution of a nonlinear problem. The proposed semi-implicit strategy reduced computational cost of the fully implicit method significantly.

In the future, we will apply the developed implicit advection strategy to three-dimensional planetary scale simulations. We are especially interested in simulating convective flow patterns to study the thermal anomaly associated with sinking iron diapirs, which is a process that plays a critical role during planetary core formation [7].

Acknowledgments

The authors would like to thank Takashi Nakagawa and Hide Sakaguchi for helpful discussion and support. The authors also thank the anonymous review of Marc Spiegelman whose comments helped improve the clarity of the paper. This study was supported by the Earth Simulator Project of Japan Agency for Marine-Earth Science and Technology, and a Grant-in-Aid for Young Scientists (24740310) from the Japan Society for the Promotion of Science (JSPS).

Appendix A. Upwind FD advection schemes

The advection equation of quantity q in the Eulerian frame is given as follows:

$$\frac{\partial q}{\partial t} = -u_i \frac{\partial q}{\partial x_i}. \tag{A.1}$$

In the treatment of SImp, we numerically solve Eq. (A.1) using the Euler method. In this work, we examined first and fifth order upwind FD discretization schemes for the spatial derivatives of q, which are given as follows:

$$\text{1st order } (\partial q/\partial x_i)_I (u_I > 0)\text{: } (q_I - q_{I-1})/\Delta x, \tag{A.2}$$

$$\text{1st order } (\partial q/\partial x_i)_I (u_I < 0)\text{: } -(q_I - q_{I+1})/\Delta x, \tag{A.3}$$

$$\begin{aligned}\text{5th order } (\partial q/\partial x_i)_I (u_I > 0)\text{: } &(-3q_{I+2} + 30q_{I+1} + 20q_I \\ &- 60q_{I-1} + 15q_{I-2} - 2q_{I-3})/(60\Delta x),\end{aligned} \tag{A.4}$$

$$\begin{aligned}\text{5th order } (\partial q/\partial x_i)_I (u_I < 0)\text{: } &-(-2q_{I+3} + 15q_{I+2} - 60q_{I+1} \\ &+ 20q_I + 30q_{I-1} - 3q_{I-2})/(60\Delta x),\end{aligned} \tag{A.5}$$

where I is the grid point in the ith direction.

Appendix B. One dimensional ODE solutions

To validate our Stokes flow solutions for different time-stepping methods in the bump test (Section 7.1), we calculate the numerical solution of the ODE of the viscous bump relaxation given by

$$\frac{dh_p}{dt} = v(t), \tag{B.1}$$

where $v(t)$ is the velocity at the top topography in the vertical coordinate. The velocity of the approximate solution of the bump problem is given by the exponential relaxation given by

$$v(t) = -\frac{(h_p(t) - h_0)}{\tau}. \tag{B.2}$$

Thus the ODE's solutions for each explicit and implicit time-stepping with constant Δt can be written as follows:

$$\text{1D\_Exp} : h_p(t + \Delta t) = (1 - \xi)h_p(t) + \xi h_0, \tag{B.3}$$

$$\text{1D\_BE} : h_p(t + \Delta t) = \frac{h_p(t) + \xi h_0}{(1 + \xi)}, \tag{B.4}$$

$$\text{1D\_TR} : h_p(t + \Delta t) = \frac{h_p(t)(1 - \xi/2) + \xi h_0}{(1 + \xi/2)}, \tag{B.5}$$

$$\begin{aligned}&\text{1st step of 1D\_TR-BDF2} : h_p(t + \Delta t/2) \\ &= \frac{h_p(t)(1 - \xi/4) + \xi h_0}{(1 + \xi/4)},\end{aligned} \tag{B.6}$$

$$\begin{aligned}&\text{2nd step of 1D\_TR-BDF2} : h_p(t + \Delta t) \\ &= \frac{4h_p(t + \Delta t/2) - h_p(t) + \xi h_0}{3(1 + \xi/3)},\end{aligned} \tag{B.7}$$

where $\xi = \Delta t/\tau$. The initial value is $h_p(0) = h_0 + \delta h$.

References

[1] S. Zhong, M. Gurnis, L. Moresi, Geophys. J. Int. 127 (3) (1996) 708–718.
[2] F. Crameri, H. Schmeling, G.J. Golabek, T. Duretz, R. Orendt, S.J.H. Buiter, D.A. May, B.J.P. Kaus, T.V. Gerya, P.J. Tackley, Geophys. J. Int. 189 (1) (2012) 38–54.
[3] B.J.P. Kaus, H. Mühlhaus, D.A. May, Phys. Earth Planet. Inter. 181 (1–2) (2010) 12–20.
[4] M. Furuichi, M. Kameyama, A. Kageyama, Phys. Earth Planet. Inter. 176 (1) (2009) 44–53.
[5] M. Furuchi, Procedia Comput. Sci. 4 (2011) 1506–1515.
[6] F. Crameri, P.J. Tackley, I. Meilick, T.V. Gerya, B.J.P. Kaus, Geophys. Res. Lett. 39 (3) (2012).
[7] G.J. Golabek, T.V. Gerya, B.J.P. Kaus, R. Ziethe, P.J. Tackley, Geochem. Geophys. Geosyst. 10 (11) (2009).
[8] T.V. Gerya, D.A. Yuen, Phys. Earth Planet. Inter. 163 (2007) 83–105.
[9] H. Schmeling, A.Y. Babeyko, A. Enns, C. Faccenna, F. Funiciello, T.V. Gerya, G.J. Golabek, S. Grigull, B.J.P. Kaus, G. Morra, S.M. Schmalholz, J. van Hunen, Phys. Earth Planet. Inter. 171 (1–4) (2008) 198–223.
[10] M. Dabrowski, M. Krotkiewski, D.W. Schmid, Geochem. Geophys. Geosyst. 9 (4) (2008).
[11] L. Moresi, S. Quenette, V. Lemiale, C. Meriaux, B. Appelbe, H.B. Mühlhaus, Phys. Earth Planet. Inter. 163 (2007) 69–82.
[12] T. Duretz, D.A. May, T.V. Gerya, P.J. Tackley, Geochem. Geophys. Geosyst. 12 (7) (2011).
[13] S.C. Kramer, C.R. Wilson, D.R. Davies, Phys. Earth Planet. Inter. 194 (2012) 25–37.
[14] R.J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, Vol. 98, SIAM, 2007.
[15] T.V. Gerya, D.A. Yuen, Phys. Earth Planet. Inter. 140 (4) (2003) 293–318.
[16] M. Spiegelman, R.F. Katz, Geochem. Geophys. Geosyst. 7 (4) (2006).
[17] P.J. Tackley, Ph.D. thesis, California Institute of Technology, 1994.
[18] M. Kameyama, A. Kageyama, T. Sato, J. Comput. Phys. 206 (1) (2005) 162–181.
[19] L. Moresi, F. Dufour, H.-B. Mühlhaus, J. Comput. Phys. 184 (2003) 476–497.
[20] P.J. Tackley, Phys. Earth Planet. Inter. 171 (1) (2008) 7–18.
[21] D.A. Knoll, D.E. Keyes, J. Comput. Phys. 193 (2) (2004) 357–397.
[22] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[23] M. Pernice, H.F. Walker, SIAM J. Sci. Comput. 19 (1) (1998) 302–318.
[24] Y. Saad, M.H. Schultz, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[25] O. Schenk, K. Gärtner, Parallel Comput. 28 (2) (2002) 187–197.
[26] J.-R. Lin, T.V. Gerya, P.J. Tackley, D.A. Yuen, G.J. Golabek, Icarus 204 (2) (2009) 732–748.
[27] R. Honda, H. Mizutani, T. Yamamoto, J. Geophys. Res. Solid Earth (1978–2012) 98 (B2) (1993) 2075–2089.
[28] S. Zhong, M. Gurnis, J. Geophys. Res. Solid Earth (1978–2012) 98 (B7) (1993) 12219–12232.
[29] P.C. Chu, C. Fan, J. Comput. Phys. 140 (2) (1998) 370–399.
[30] Y. Imai, T. Aoki, J. Comput. Phys. 217 (2) (2006) 453–472.
[31] C. Hu, C.-W. Shu, J. Comput. Phys. 150 (1) (1999) 97–127.
[32] M. Furuichi, M. Kameyama, A. Kageyama, J. Comput. Phys. 227 (10) (2008) 4977–4997.