Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 206 (2005) 162–181
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Multigrid iterative algorithm using pseudo-compressibility for three-dimensional mantle convection with strongly variable viscosity

Masanori Kameyama *, Akira Kageyama, Tetsuya Sato

Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), 3173-25 Showa-machi, Kanazawa, Yokohama 236-0001, Japan

Received 29 March 2004; received in revised form 24 November 2004; accepted 25 November 2004
Available online 21 January 2005

Abstract

A numerical algorithm for solving mantle convection problems with strongly variable viscosity is presented. Equations for conservation of mass and momentum for highly viscous and incompressible fluids are solved iteratively by a multigrid method in combination with pseudo-compressibility and local time stepping techniques. This algorithm is suitable for large-scale three-dimensional numerical simulations, because (i) memory storage for any additional matrix is not required and (ii) vectorization and parallelization are straightforward. The present algorithm has been incorporated into a mantle convection simulation program based on the finite-volume discretization in a three-dimensional rectangular domain. Benchmark comparisons with previous two- and three-dimensional calculations including the temperature- and/or depth-dependent viscosity revealed that accurate results are successfully reproduced even for the cases with viscosity variations of several orders of magnitude. The robustness of the numerical method against viscosity variation can be significantly improved by increasing the pre- and post-smoothing calculations during the multigrid operations, and the convergence can be achieved for the global viscosity variations up to 10^{10}.

Keywords: Mantle convection; Variable viscosity; Pseudo-compressibility; Multigrid method; Local time-stepping

1. Introduction

The Earth's mantle is the spherical shell composed of silicate rocks and it ranges from approximately 5–50 to 2900 km depth. Although the mantle behaves like an elastic solid on short time scales, it acts

* Corresponding author. Tel.: +81 45 778 5836; fax: +81 45 778 5493.
E-mail address: kameyama@jamstec.go.jp (M. Kameyama).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.11.030

like a highly viscous fluid on long time scales. The mantle also acts as a heat engine and it is in a convective motion in order to transport the heat from the hot interior to the cool surface [1,2]. The mantle convection is observed as the motion of tectonic plates at the Earth's surface. The motion of surface plates, in turn, drives seismicity, volcanism and mountain building at the plate margins. Thus, the mantle convection is the origin of the geological and geophysical phenomena observed at the Earth's surface. A major tool for understanding the mantle convection is numerical analysis. It has been playing an important role in the study of mantle convection, since a numerical simulation of mantle convection first arose [3,4].

Mantle convection requires different numerical techniques from those for ordinary fluids such as water because of its rheological properties. The viscosity of mantle materials is estimated as high as 10^{22} Pa s [1,5]. Since the mantle materials is highly viscous, both the nonlinear and time-derivative terms of velocity can be ignored in the equation of motion. This implies that the flow in the mantle is described by a steady-state Stokes flow balancing among the buoyancy force, pressure gradient and viscous resistance. Taken together with the assumption of incompressibility, one needs to solve elliptic differential equations for velocity and pressure at every timestep. In addition, the viscosity of mantle material varies by several orders of magnitude depending on temperature, pressure, and stress [6,7]. The strong variation in viscosity makes numerical techniques for ordinary isoviscous fluids, such as the spectral method [8,9], unfit for the numerical modeling of mantle convection. In order to get deep insights into the mantle convection, it is very important to develop efficient numerical techniques that can deal with the steady-state flow of highly viscous and incompressible fluids with a strongly variable viscosity.

The efficiency of numerical simulations of mantle convection strongly relies on numerical methods used for solving elliptic differential equations. One of the most efficient methods is the multigrid iteration [10]. The multigrid concept has been successfully applied to a wide range of problems, including calculations of incompressible fluid flow [11–13]. During the last two decades, various numerical models of mantle convection have been developed where the multigrid method is utilized. There are two strategies to apply the multigrid method to this problem, depending on how the steady-state Stokes equations are solved.

The first strategy solves the Stokes equations by splitting into the separate equations for velocities and pressure. The discretized equations for velocity components (or their proxy) are solved by the multigrid method, while the pressure is eliminated or solved separately. Parmentier et al. [14] developed convection models of isoviscous fluid in three-dimensional Cartesian geometry. By using a streamfunction formulation, the Stokes equations are reduced to a pair of Poisson equations which are solved by multigrid iterations. Baumgardner [15] developed a convection model of isoviscous fluid in a three-dimensional spherical geometry. He solved the elliptic equations for velocity components using a multigrid method, while the pressure fields are prescribed by the equation of state. Baumgardner and his colleagues [16–18] further developed convection models for fluids with variable viscosity in a three-dimensional spherical geometry. The Stokes equations are solved separately for velocity and pressure by so-called Uzawa iterative scheme [19]. The iteration for velocity is carried out by a multigrid method, while a conjugate gradient scheme is used for pressure iteration. This approach was also employed by Moresi and his colleagues [20,21] for convection problems with strongly variable viscosity in two- and three-dimensional Cartesian geometry.

The second strategy, on the other hand, solves the Stokes equations for velocity and pressure as a whole by the multigrid technique. The key issue of this strategy is a choice of the smoothing algorithm which reduces the errors of solution on a particular grid. Several methods for solving incompressible fluid flows have been utilized as a smoothing algorithm. Trompert and Hansen [22,23] and Albers [24] developed numerical methods for convection problems with variable viscosity in three-dimensional Cartesian geometry. The Stokes equations are solved by a multigrid method where the SIMPLER algorithm [25] is employed as a smoothing operation. Auth and Harder [26] used the symmetric coupled Gauss–Seidel

(SCGS) method [27] as a smoothing operation in the multigrid method for solving the Stokes equation. Tackley [28,29] employed a similar smoothing method where the velocity and pressure fields are updated in a somewhat coupled manner, and his method has been successfully applied to a variety of convection problems [29,30], including calculations with extremely variable viscosity [31–33].

In this paper, we present a solution algorithm for the three-dimensional mantle convection, following the second strategy where the Stokes equations are solved as a whole by the multigrid method. In Section 2 we introduce our solution algorithm based on the pseudo-compressibility method [34]. A notable feature of our algorithm is its simplicity and intuitiveness. This feature makes the algorithm easily fit to a smoothing operation of the multigrid iterations. In addition, taken together with the local time-stepping method, a strong variation in viscosity can be handled without a severe increase in computational costs. In Section 3, we develop a numerical model of mantle convection that can deal with a strongly variable viscosity based on the algorithm presented in Section 2 together with the multigrid technique. In order to demonstrate the validity and efficiency of our algorithm, we carry out several calculations for the mantle convection with a strongly variable viscosity. In this paper, an application of our algorithm is the thermal convection in a three-dimensional rectangular domain. It can be easily applied to the numerical models for more realistic problems of mantle convection, such as spherical shell geometry and thermo-chemical convection and so on.

2. Iterative algorithm for mantle convection

We consider thermal convection of a highly viscous and incompressible fluid with a strongly variable Newtonian viscosity in a three-dimensional Cartesian geometry ($x_1 = x$, $x_2 = y$, $x_3 = z$).

The nondimensional forms of the fundamental equations are (for example [35,36])

$$-\frac{\partial}{\partial x_j}\left[\eta\left(\frac{\partial v_j}{\partial x_i}+\frac{\partial v_i}{\partial x_j}\right)\right]+\frac{\partial p}{\partial x_i}=RaT\delta_{i3}\quad(i,j=1,2,3), \tag{1}$$

$$\frac{\partial v_k}{\partial x_k}=0\quad(k=1,2,3), \tag{2}$$

$$\frac{\partial T}{\partial t}+v_i\frac{\partial T}{\partial x_i}=\frac{\partial^2 T}{\partial x_j^2}+q, \tag{3}$$

where v_i ($i = 1,2,3$) are fluid velocities in ith direction, p pressure, η viscosity, T temperature, Ra the Rayleigh number, and q is the internal heating rate. We assumed that x_3-axis is the vertical axis pointing upward. Boussinesq approximation is employed in the energy equation (3) and, hence, the effects of adiabatic and viscous heating are ignored.

2.1. Overview of the iterative algorithm

In the following, we consider the solution algorithm for the conservation equations of momentum (1) and mass (2). In order to simplify the solution algorithm, the energy equation (3) is solved separately from Eqs. (1) and (2): the temperature T at new time is calculated using the velocity at old time in the advection term. This algorithm solves Eqs. (1) and (2) for the velocity and pressure at new time using the new temperature. In addition, the viscosity η at new time is already known by some means and kept unchanged while the velocity and pressure are computed. Because of these assumptions, Eqs. (1) and (2) are taken to be linear with respect to velocity and pressure.

The discretized equations of (1) and (2) can be written in a matrix form of

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{4}$$

where

$$\boldsymbol{A} \equiv \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} & \boldsymbol{C}_{13} & \boldsymbol{G}_1 \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} & \boldsymbol{C}_{23} & \boldsymbol{G}_2 \\ \boldsymbol{C}_{31} & \boldsymbol{C}_{32} & \boldsymbol{C}_{33} & \boldsymbol{G}_3 \\ \boldsymbol{D}_1 & \boldsymbol{D}_2 & \boldsymbol{D}_3 & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{p} \end{bmatrix}, \qquad \boldsymbol{b} \equiv \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ Ra\boldsymbol{\theta} \\ \boldsymbol{0} \end{bmatrix}. \tag{5}$$

In (5), $\boldsymbol{v}_i$ ($i = 1,2,3$) are the vectors of unknown fluid velocities in i-direction, $\boldsymbol{p}$ the vector of unknown pressures, $\boldsymbol{\theta}$ the vector of known temperatures, $\boldsymbol{C}_{ij}$ ($i,j = 1,2,3$) the matrices representing the spatial discretization of velocities for calculating viscous stress, $\boldsymbol{G}_i$ ($i = 1,2,3$) the matrices representing the discretization of the pressure gradient, $\boldsymbol{D}_i$ ($i = 1,2,3$) the matrices representing the discretization of the divergence of velocity, and $\boldsymbol{0}$ is the zero matrix or zero vector. One of the major difficulties in solving (4) is that a zero block appears in the main diagonal in $\boldsymbol{A}$. This implies that basic iterative methods for solving linear simultaneous equations (such as Jacobi and Gauss–Seidel methods) which make use of the inverse of the main diagonal fail to solve (4).

The algorithm for solving (4) that we propose in this paper is another kind of iterative method: let $\boldsymbol{x}^n$ be an approximate at nth iteration ($n = 0,1,2,\ldots$). We then calculate the approximate at the iteration ($n+1$) by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \boldsymbol{T}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^n). \tag{6}$$

Here, a regular matrix $\boldsymbol{T}$ is introduced so as to control the convergence rate of (6), and is chosen to be a diagonal matrix whose nonzero elements are positive. This iterative procedure is used together with the multigrid method, and is repeated until $\boldsymbol{x}^n$ converges to a steady solution. It is obvious that the steady solution of (6) satisfies (4), and that it does not depend on $\boldsymbol{T}$. In addition, the presence of zero block in the main diagonal of $\boldsymbol{A}$ does not spoil the convergence of this iterative procedure.

This iterative algorithm is suitable for large-scale numerical simulations on massively parallel computers and, in particular, massively vector-parallel supercomputers because (i) the procedure (6) consists of multiplication of matrices and vectors and summation of vectors and, hence, can be easily vectorized and parallelized, and (ii) large memory storage for the matrix $\boldsymbol{T}$ is not required, since $\boldsymbol{T}$ is a diagonal matrix.

Another benefit of the iterative procedure (6) is that it is easily implemented into the multigrid method. This is because the procedure (6) is quite similar to that representing the basic iterative methods which are commonly used as a smoothing operator of the multigrid method. Suppose an iterative procedure for (4) by Jacobi method

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \boldsymbol{D}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^n) \quad (n = 0,1,2,\ldots), \tag{7}$$

where a matrix $\boldsymbol{D}$ is the main diagonal of $\boldsymbol{A}$. The two procedures (6) and (7) are the same except that the diagonal matrix $\boldsymbol{T}$ is used in (6) in place of $\boldsymbol{D}^{-1}$ in (7). Therefore, the iterative procedure (6) can be used as a smoothing operator of the multigrid method in a similar manner to Jacobi method.

In the following subsections, we will introduce the ideas which led us to the iterative procedure (6). We will also discuss an appropriate choice of the matrix $\boldsymbol{T}$, which is a key parameter in this procedure.

2.2. Ingredient 1: pseudo-compressibility method for highly viscous, incompressible fluids

When the matrix $\boldsymbol{T}$ is diagonal, Eq. (6) can be explicitly written as ($i,j,k = 1,2,3$)

$$v_i^{n+1} = v_i^n + \tau_{v_i}\left\{-\frac{\partial p^n}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\eta\left(\frac{\partial v_j^n}{\partial x_i} + \frac{\partial v_i^n}{\partial x_j}\right)\right] + RaT\delta_{i3}\right\}, \tag{8}$$

$$p^{n+1} = p^n - \tau_{\mathrm{p}}\frac{\partial v_k^n}{\partial x_k}, \tag{9}$$

where τ_{v_i} $(i = 1, 2, 3)$ and τ_{p} are the diagonal elements appearing in $\boldsymbol{T}$.

In order to discuss the physical meaning of these equations more clearly, we introduce the following "auxiliary" set of equations:

$$M_{v_i}\frac{\partial v_i}{\partial \tilde{t}} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\eta\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right)\right] + RaT\delta_{i3}, \tag{10}$$

$$K\frac{\partial p}{\partial \tilde{t}} = -\frac{\partial v_k}{\partial x_k}. \tag{11}$$

Here $\tilde{t}$ is analogous to time, M_{v_i} $(i = 1, 2, 3)$ and K are positive constants analogous to density and compressibility, respectively. These equations come from the modification of the "auxiliary" set of equations used in the pseudo-compressibility method [34], which is used to solve steady-state flow of incompressible fluid with high Reynolds number. The difference between Eqs. (10) and (11) and those employed for high Reynolds-number flow is that the nonlinear term of velocity has been eliminated in (10), because the viscosity of mantle materials is significantly high and, in other words, the Reynolds number is significantly small.

By discretizing (10) and (11) in the direction of $\tilde{t}$ by a first-order explicit scheme, we get

$$M_{v_i}\frac{v_i^{n+1} - v_i^n}{\Delta\tilde{t}} = -\frac{\partial p^n}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\eta\left(\frac{\partial v_j^n}{\partial x_i} + \frac{\partial v_i^n}{\partial x_j}\right)\right] + RaT\delta_{i3}, \tag{12}$$

$$K\frac{p^{n+1} - p^n}{\Delta\tilde{t}} = -\frac{\partial v_k^n}{\partial x_k}. \tag{13}$$

We notice that Eqs. (12) and (13) and Eqs. (8) and (9) are identical if we choose τ_{v_i} and τ_p as

$$\tau_{v_i} = \Delta\tilde{t}/M_{v_i}, \quad \tau_{\mathrm{p}} = \Delta\tilde{t}/K. \tag{14}$$

In other words, τ_{v_i} and τ_{p} are "effective" timesteps for the evolution of v_i and p, respectively.

The iterative procedure given by (12) and (13) converges to an incompressible flow field regardless of the choice of M_{v_i} and K. To show this more clearly, we here assume that M_{v_i}, K and η are constants. From Eqs. (10) and (11) we obtain the pseudo-temporal evolution of $\nabla\cdot\boldsymbol{v}$ as

$$\frac{\partial^2}{\partial\tilde{t}^2}(\nabla\cdot\boldsymbol{v}) = \frac{1}{KM}\nabla^2(\nabla\cdot\boldsymbol{v}) + \frac{2\eta}{M}\frac{\partial}{\partial\tilde{t}}\left[\nabla^2(\nabla\cdot\boldsymbol{v})\right]. \tag{15}$$

(Here we denote $M_{v_i} \equiv M$.) The first term of the right-hand side represents the effect of propagation of "pseudo-sound wave" whose velocity is $1/\sqrt{KM}$, and the second term represents the effect of diffusion due to viscosity. This equation indicates that the pseudo-temporal evolution of $\nabla\cdot\boldsymbol{v}$ is characterized by a decaying oscillation. Therefore, $\nabla\cdot\boldsymbol{v}$ approaches to an asymptotic value (=0) as $\tilde{t}$ increases to infinity. Consequently, the procedure of (12) and (13) leads the steady velocity field with $\nabla\cdot\boldsymbol{v} = 0$, as long as the numerical integration scheme is stable.

We also note that the iterative procedure (8) and (9) should be used in combination with multigrid method [10]. Recall that we are trying to find a steady-state solution of evolution equations (10) and (11) through a repetition of temporal integration, which inevitably requires a large number of iterations.

Moreover, the elliptic nature of (10) for velocities v_i results in a slow reduction in their errors, particularly in their smooth components with large spatial wavelengths. By incorporating into the multigrid procedure, we are able to obtain a sufficiently fast convergence. In fact, the multigrid technique is adopted in most of recent numerical analysis using pseudo-compressibility method (for example [37,38]).

2.3. Ingredient 2: local time stepping method for strongly variable viscosity

In most of previous numerical analysis using pseudo-compressibility method (for example [37–40]), the viscosity of fluids has been assumed to be constant. In this subsection, we apply this method for the cases with a strong variation in viscosity.

The convergence rate of the original pseudo-compressibility method deteriorates in proportion to the viscosity variation in the entire domain, when the spatial variation in viscosity is introduced. This is because the viscosity η is a diffusion coefficient for v_i in Eq. (10). The spatial variation in η results in the spatial variation in the local convergence rate of v_i. The values of v_i in the region with smaller η reach to an appropriate solution only very slowly, whereas the values in the region with larger η converge quickly. This suggests that the convergence should be accelerated particularly in the region with small η.

Here we try to accelerate the convergence for the case with a spatial variation in η by controlling the stepping of the pseudo-time $\tilde{t}$ in accordance with η. This idea is known as the local time stepping method, which is used to obtain steady-state solution of evolutionary equations. This method allows updating each variable using a timestep which is based on the local numerical stability criterion. We thus spatially vary the effective timesteps τ_{v_i} and τ_{p} in (8) and (9) in accordance with η.

The spatial variation of τ_{v_i} and τ_{p} can be estimated from the local numerical stability criterion of (15). In the following derivation, we assume for simplicity that Eq. (15) holds even in the case of variable viscosity. From the criterion of the diffusion of $\nabla \cdot \boldsymbol{v}$ (the second term in the right-hand side of (15)) we require that the nondimensional parameter α defined by

$$\alpha \equiv \frac{2\eta/M}{\Delta^2/\Delta\tilde{t}} \tag{16}$$

is sufficiently small. Here Δ is the mesh size. Similarly, from the criterion of the pseudo-sound propagation we require that the nondimensional parameter β defined by

$$\beta \equiv \frac{1/\sqrt{KM}}{\Delta/\Delta\tilde{t}} \tag{17}$$

is sufficiently small. By substituting (16) and (17) into (14) we get

$$\tau_{v_i} = \frac{\Delta\tilde{t}}{M} = \frac{\alpha}{2}\frac{\Delta^2}{\eta}, \qquad \tau_{\mathrm{p}} = \frac{\Delta\tilde{t}}{K} = \frac{2\beta^2}{\alpha}\eta. \tag{18}$$

Here we assumed $M_{v_i} = M$.

When τ_{v_i} and τ_{p} are defined by (18), the spatial variations of the diffusion of velocity components and the pseudo-sound propagation become more modest than that of viscosity. Eq. (18) implies that M ($\equiv M_{v_i}$ in (10)) and K are taken to be proportional to η and η^{-1}, respectively. By assuming that η/M and $K\eta$ are constant in (10) and (11), we obtain the pseudo-temporal evolution of $\nabla \cdot \boldsymbol{v}$ in the presence of spatial variation in η as

$$\frac{\partial^2}{\partial\tilde{t}^2}(\nabla\cdot\boldsymbol{v}) = \frac{1}{KM}\nabla^2(\nabla\cdot\boldsymbol{v}) + \frac{2\eta}{M}\frac{\partial}{\partial\tilde{t}}[\nabla^2(\nabla\cdot\boldsymbol{v})] + \frac{1}{KM}\frac{\partial}{\partial x_i}\left[\frac{\partial(\ln\eta)}{\partial x_i}(\nabla\cdot\boldsymbol{v})\right] + \frac{\eta}{M}\frac{\partial}{\partial x_i}\left[\frac{\partial(\ln\eta)}{\partial x_j}\frac{\partial}{\partial\tilde{t}}\left(\frac{\partial v_j}{\partial x_i}+\frac{\partial v_i}{\partial x_j}\right)\right]. \tag{19}$$

The first and second terms in the right-hand side of (19) are equivalent with those in the right-hand side of (15). Note that the effects of spatial variation in η appear in the third and fourth terms in the form of $\ln\eta$. Thus we expect that $\nabla \cdot \boldsymbol{v}$ reduces to zero in the entire computational domain at a rate comparable to that for the case with constant η.

Another constraint for τ_{v_i} and τ_p can be obtained directly from (6). Let $\boldsymbol{e}^n \equiv \boldsymbol{x}^n - \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\boldsymbol{r}^n \equiv \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^n$ to be the error and the residual at nth iteration, respectively. From (6), we get the evolution equations for $\boldsymbol{e}^n$ and $\boldsymbol{r}^n$ as

$$\boldsymbol{e}^{n+1} = (\boldsymbol{I} - \boldsymbol{T}\boldsymbol{A})\boldsymbol{e}^n, \qquad \boldsymbol{r}^{n+1} = (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{T})\boldsymbol{r}^n, \tag{20}$$

where $\boldsymbol{I}$ is the identity matrix. These equations indicate that the convergence rate of (6) becomes optimal when $\boldsymbol{T} = \boldsymbol{T}_{\mathrm{opt}} \equiv \boldsymbol{A}^{-1}$. (In this sense, the matrix $\boldsymbol{T}$ can be regarded as a "preconditioner" for Eq. (4).) In the present algorithm using local time stepping, we approximate $\boldsymbol{T} \simeq \boldsymbol{T}_{\mathrm{opt}}$ by a diagonal matrix. In addition, Eq. (20) implies that the necessary condition for the convergence is that the spectral radius of the matrix $\boldsymbol{I} - \boldsymbol{T}\boldsymbol{A}$ (or, identically, $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{T}$) is smaller than one. Indeed, this condition is identical to the Courant–Friedrichs–Lewy (CFL) condition of Eqs. (12) and (13), i.e., $\alpha < \alpha_c$ and $\beta < \beta_c$ where α_c and β_c are threshold values.

3. Results

The algorithm described in the previous section has been incorporated into a mantle convection simulation program in a three-dimensional rectangular domain. We carried out several calculations for thermal convection with strongly variable viscosity, in order to demonstrate the validity and efficiency of the iterative algorithm.

3.1. Details of numerical method

The basic equations are discretized by the finite volume method. A staggered grid is used; temperature and pressure are located at the center of the grid cells, while the velocity components are at the center of the cell faces normal to the direction of the velocity components. A uniform mesh is employed. The basic equations are nondimensionalized with a length scale of h (height of the box), time scale of h^2/κ (where κ is thermal diffusivity), temperature scale of ΔT (temperature drop across the box), and pressure scale of $\eta_0\kappa/h^2$ (where η_0 is reference viscosity). In the following subsections, we present two kinds of calculations. First is to obtain the instantaneous flow fields for prescribed distributions of buoyancy and viscosity, and second is to obtain the temporal developments of thermal convection.

When the purpose is to obtain the flow fields, we only solve the conservation equations for momentum (1) and mass (2). These equations are solved by the multigrid method based on correction-storage algorithm, because of the linear nature of (4). The smoothing operation at each grid level is carried out by (6). The diagonal elements of $\boldsymbol{T}$ for unknown velocities (τ_{v_i}) were chosen to be 0.5 times the inverse of the corresponding diagonal elements of $\boldsymbol{A}$, while those for unknown pressures (τ_p) are to be 0.25 times the viscosity at the corresponding cell centers. The values of τ_{v_i} and τ_p defined thus are approximately equal to those with $\alpha = \beta = 1/8$ in (18). (We observed that the iteration by (6) diverged when τ_{v_i} or τ_p is larger than the above values.) A linear interpolation is used in both fine-to-coarse (restriction) and coarse-to-fine (prolongation) operations. The values of viscosity at cell centers on coarser grids is calculated by a linear interpolation from those on the finer grids by one grid level. On the other hand, the values of viscosity at the midpoints of cell edges on each grid level, which are necessary to calculate viscous shear stress, are calculated by an interpolation proposed by Ogawa et al. [36] from the values at cell centers on the same grid level. Since the aim of this paper is to demonstrate the efficiency and robustness of the smoothing algorithm

(6), only the V-cycle is used in the present calculations, although more complicated multigrid iteration cycles (such as W-, F-cycles) are expected to be more robust (for example [26,24]).

When the purpose is to obtain a time-dependent or steady-state convection, we solve the energy equation (3) in addition to Eqs. (1) and (2). In the calculations presented in this paper, the internal heating rate q is taken to be zero. The energy equation is discretized by a first-order Euler method in time. An upwind scheme, called power-law scheme [25], is used to evaluate the contributions of heat transport by advection and conduction. The discretized equation for T is solved by a fully implicit scheme when we seek for a steady-state convection, or by a fully explicit scheme when we seek for a time-dependent convection.

3.2. Benchmark comparison

In order to test the validity of our numerical code, we compare our results with those of earlier studies [41,42,36]. The temperature T is fixed at 0 and 1 at the top ($z = 1$) and bottom ($z = 0$) boundaries, respectively, while the vertical side walls are adiabatic. In most cases, the viscosity η is assumed to be dependent on temperature T and depth $(1 - z)$ as

$$\eta = \eta_t \exp[-E_T T + E_p(1 - z)], \tag{21}$$

where η_t is the viscosity at the top surface ($z = 1$, $T = 0$), and E_T and E_p are constants describing the temperature- and depth-dependence of viscosity, respectively.

3.2.1. Benchmark for two-dimensional convection

First we present the benchmark comparison with Blankenbach et al. [41] for the cases of steady-state convection in two-dimensional rectangular domains of aspect ratio (width/depth) λ. We carried out calculations of five cases listed in Table 1. Cases 1a–1c are the convection of isoviscous fluid in a square box ($\lambda = 1$), Case 2a is the convection of fluids with temperature-dependent viscosity in a square box, and Case 2b is the convection with temperature- and depth-dependent viscosity in a box of $\lambda = 2.5$. The impermeable and shear-stress-free conditions are adopted along the all boundaries. The initial conditions are chosen to make single convective cell with an ascending flow along the left-side wall ($x = 0$). We carried out these calculations by varying the number of mesh divisions in x- and z-directions, while the mesh division in y-direction was kept to 1, in order to obtain two-dimensional flow patterns in the x–z plane.

We summarized in Table 1 the parameter values and mesh divisions employed in the present calculations, and the obtained values of (i) Nusselt number Nu, (ii) root-mean-square velocity V_{rms}, and (iii) vertical temperature gradient q_1–q_4 at $(x, z) = (0, 1)$, $(\lambda, 1)$, $(\lambda, 0)$, and $(0, 0)$, respectively. We also show in Table 1 the results from the benchmark standard [41]. Table 1 shows that the agreement between the results of our calculations and benchmark standards is satisfactory for all the values. We conclude that the present numerical code accurately handles two-dimensional convection of fluids with both constant and variable viscosity.

3.2.2. Benchmark for three-dimensional convection with modest viscosity variation

Second, we compare our results with those of Busse et al. [42] for the cases of steady-state convection with both constant (Case 1a) and modestly temperature-dependent viscosity (Case 2) in a three-dimensional rectangular box of $a \times b \times 1$. Fig. 1 shows the convective flow patterns for both cases. Case 1a is a bimodal convection of isoviscous fluid with $Ra = 3 \times 10^4$ in a box of $a = 1.0079$ and $b = 0.6283$, while Case 2 is a square-cell convection in a cube ($a = b = 1$) with modestly temperature-dependent viscosity. In Case 2, the viscosity η depends on temperature T as

$$\eta(T) = \exp\left[\frac{Q}{T + G} - \frac{Q}{0.5 + G}\right], \qquad Q = \frac{225}{\ln(r_\eta)} - 0.25\ln(r_\eta), \quad G = \frac{15}{\ln(r_\eta)} - 0.5. \tag{22}$$

Table 1
Benchmark comparison with Blankenbach et al. [41]

Mesh	16×16	32×32	64×64	128×128	Benchmark standard
(a) *Case 1a (constant viscosity, $Ra = 10^4$, $\lambda = 1$)*					
Nu	4.726885	4.840006	4.872745	4.881429	4.884409 ± 0.000010
V_{rms}	41.755217	42.560118	42.785441	42.844667	42.864947 ± 0.000020
q_1 ($=q_3$)	7.695426	7.966605	8.036044	8.053536	8.059384 ± 0.000003
q_2 ($=q_4$)	0.887088	0.664927	0.608154	0.593695	0.588810 ± 0.000003
(b) *Case 1b (constant viscosity, $Ra = 10^5$, $\lambda = 1$)*					
Nu	9.235550	10.082608	10.394786	10.495490	10.534095 ± 0.000010
V_{rms}	181.18346	189.19219	191.98526	192.87381	193.21454 ± 0.00010
q_1 ($=q_3$)	14.424512	17.764322	18.731196	18.990358	19.079440 ± 0.000040
q_2 ($=q_4$)	3.441283	1.425553	0.898253	0.767465	0.722751 ± 0.000020
(c) *Case 1c (constant viscosity, $Ra = 10^6$, $\lambda = 1$)*					
Nu	13.505592	18.963003	20.884342	21.604074	21.972465 ± 0.000020
V_{rms}	675.69469	780.80943	814.68028	827.43096	833.98977 ± 0.00020
q_1 ($=q_3$)	16.63581	31.21463	41.44228	44.72368	45.96425 ± 0.00030
q_2 ($=q_4$)	9.836338	6.941879	2.477731	1.255997	0.877170 ± 0.000010
(d) *Case 2a ($Ra_t = 10^4$, $E_T = \ln(10^3)$, $E_p = 0$, $\lambda = 1$)*					
Nu	10.3634	10.4780	10.1666	10.0862	10.0660 ± 0.00020
V_{rms}	372.2759	457.4738	475.5292	478.9647	480.4334 ± 0.1000
q_1	17.42173	20.41169	18.33101	17.72910	17.53136 ± 0.00400
q_2	2.25932	1.16135	1.03954	1.01498	1.00851 ± 0.00020
q_3	14.1505	20.2221	25.3614	26.6611	26.8085 ± 0.0100
q_4	7.036967	4.248769	1.557456	0.743177	0.497380 ± 0.000100
Mesh	40×16	80×32	160×64	320×128	Benchmark standard
(e) *Case 2b ($Ra_t = 10^4$, $E_T = \ln(16384)$, $E_p = \ln(64)$, $\lambda = 2.5$)*					
Nu	7.4020	7.2761	7.0450	6.9605	6.9299 ± 0.0005
V_{rms}	201.478	171.085	175.343	172.806	171.755 ± 0.020
q_1	18.4789	30.6337	20.9644	19.0478	18.4842 ± 0.0100
q_2	0.37860	0.17729	0.18048	0.17873	0.17742 ± 0.00003
q_3	14.5474	14.7261	14.7092	14.3514	14.1682 ± 0.0050
q_4	3.51882	1.28403	0.79348	0.66355	0.61770 ± 0.00005

See the text as for the meaning of the symbols.

Here, the viscosity contrast r_η is set to be 20, and the Rayleigh number defined by the viscosity for $T = 0.5$ is 2×10^4. In these cases the top and bottom surfaces are assumed to be rigid ($v_1 = v_2 = v_3 = 0$), while the vertical side walls are planes of mirror symmetry. The initial conditions are chosen to make single ascending and descending flow at $(x, y) = (0, 0)$ and (a, b), respectively.

We summarize the result of benchmark comparison for three-dimensional convection in Table 2. We calculated the values of (i) Nu and V_{rms}, (ii) vertical velocity w and temperature T at specified points at mid-depth of the convecting vessel ($z = 0.5$), and (iii) vertical temperature gradient Q at specified points at the top surface ($z = 0$). We also show in Table 2 the results from the benchmark standard [42]. The agreement between the obtained results and the benchmark standard is satisfactory for all of the values. In particular, we obtained a good agreement for Case 2. We thus conclude that the present numerical code accurately handles three-dimensional convection of fluids with mildly variable viscosity.

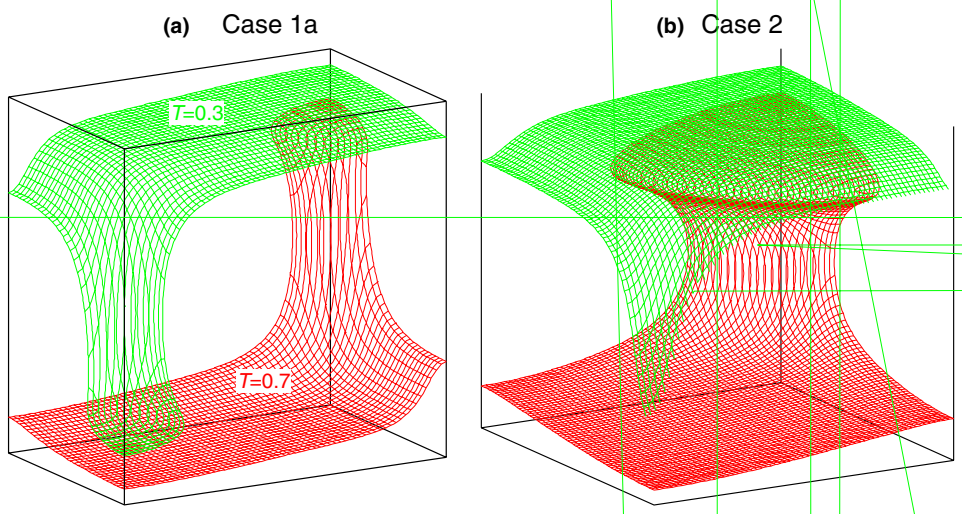

(a) Case 1a
T=0.3
T=0.7
(b) Case 2

Table 3
Comparison of the flow patterns, Nusselt numbers Nu and the minimum mesh sizes in z-direction dz_{min} between the present method ("KKS") and Ogawa et al. [36] ("OSZ")

Case	Ra_t	r_η	Pattern	Code	KKS			OSZ
				Mesh	$32\times16\times32$	$64\times32\times64$	$128\times64\times128$	$24\times14\times22^*$
				dz_{min}	0.03125	0.015625	0.0078125	3.2×10^{-2}
1	10^5	1	WL-3D	Nu	9.816	10.071	10.101	9.72
4	10^3	10^2	WL-2D	Nu	4.060	4.073	4.077	4.13
10	10^3	10^3	WL-3D	Nu	4.961	4.932	4.921	4.96
				Mesh	$32\times16\times32$	$64\times32\times64$	$128\times64\times128$	$44\times18\times30^*$
				dz_{min}	0.03125	0.015625	0.0078125	1.7×10^{-2}
16	10^3	3.2×10^3	WL-3D	Nu	5.284	5.271	5.271	5.37
				Mesh	$32\times16\times32$	$64\times32\times64$	$128\times64\times128$	$44\times18\times30^*$
				dz_{min}	0.03125	0.015625	0.0078125	1.3×10^{-2}
17	10^2	3.2×10^4	SL-3D	Nu	3.561	3.631	3.659	3.61
18	32	10^5	SL-3D	Nu	3.119	3.174	3.197	3.17

Note that in OSZ [36] the mesh spacing was taken to be non-uniform with finer resolution near the boundaries so as to improve the accuracy in Nu.

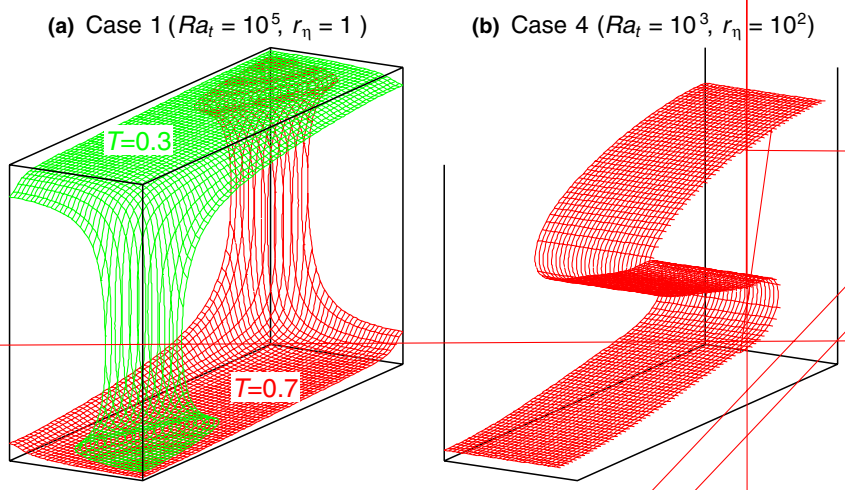
3.2.3. Benchmark for three-dimensional convection with strong viscosity variation

We also carried out the calculations similar to Ogawa et al. [36] in order to test our code for a much stronger temperature-dependence of viscosity. A steady-state convection in a box of $1.7\times0.5\times1$ is considered. The impermeable and shear-stress-free conditions are adopted along the all boundaries. We carried out calculations for several cases of Ogawa et al. [36] with several mesh divisions. In these calculations we varied Ra_t, the Rayleigh number defined with η_t, and the viscosity contrast $r_\eta = \exp(E_T)$ between the top and bottom boundaries, as listed in Table 3. Fig. 2 shows the convective flow patterns for several cases. We obtained the same flow patterns and the change in flow patterns depending on Ra_t and r_η as in Ogawa et al. [36]. The convective flow patterns are classified into three-dimensional whole-layer convection (WL-3D) for Case 1 ($Ra_t = 10^5$ and $r_\eta = 1$), two-dimensional roll of whole-layer convection (WL-2D) for Case 4 ($Ra_t = 10^3$ and $r_\eta = 10^2$), again in WL-3D for Case 16 ($Ra_t = 10^3$ and $r_\eta = 3.2\times10^3$), and in three-dimensional stagnant-lid convection (SL-3D) for Case 18 ($Ra_t = 32$ and $r_\eta = 10^5$).

We summarize the comparison of the flow patterns and Nusselt numbers Nu obtained by the present code ("KKS") and those by Ogawa et al. [36] ("OSZ") in Table 3. The values of Nu obtained by both codes agree within at most 4% deviation for all cases. The largest deviation for Case 1 may come from the differences in adopted mesh sizes. The present calculations are carried out with the minimum mesh size in z-direction of 1/128, which is more than four times finer than that employed in Ogawa et al. [36]. It is most likely that a finer spatial resolution is required for Case 1 than that employed in Ogawa et al. [36] in order to resolve the thermal boundary layers.

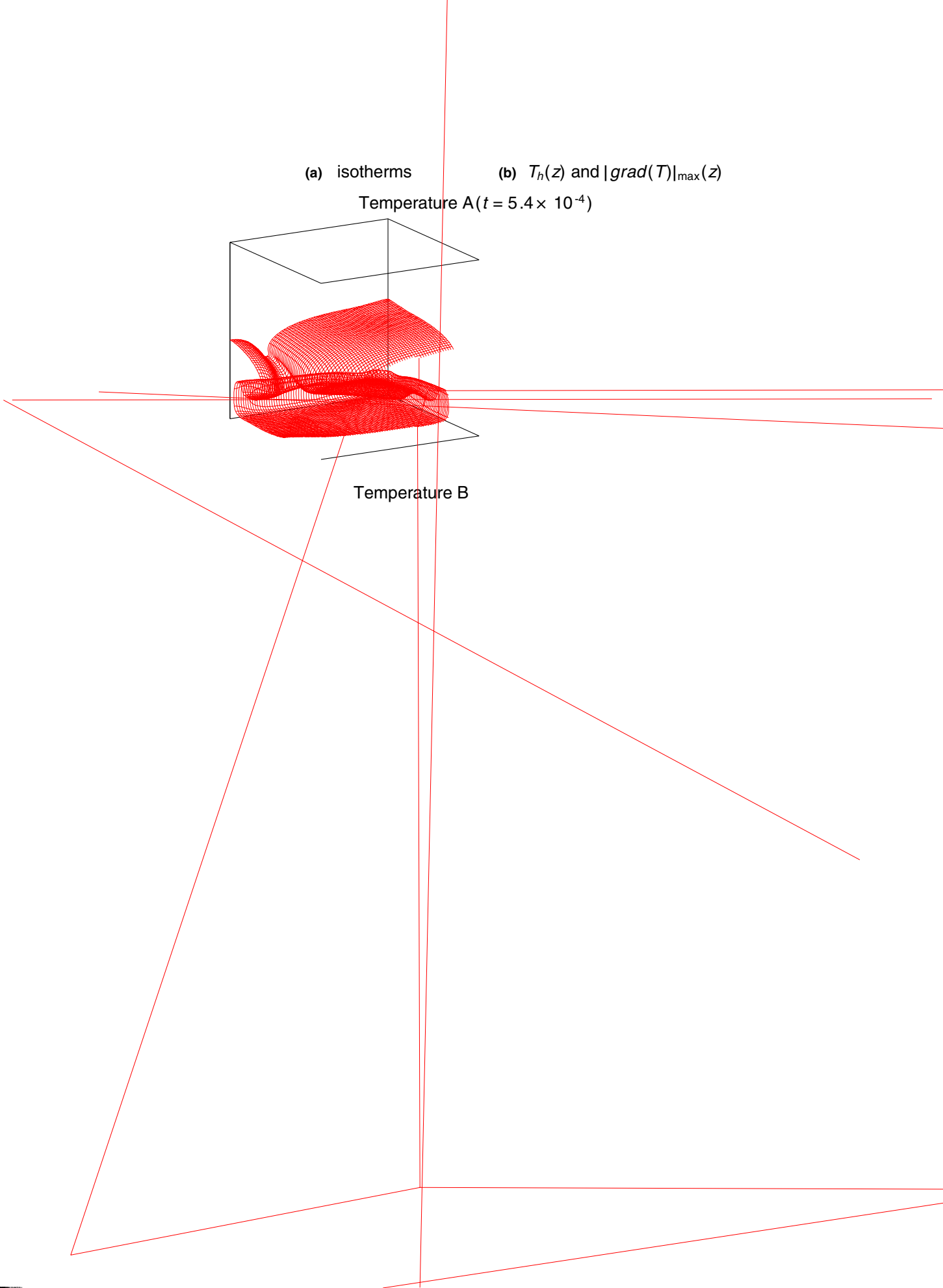
3.3. Robustness and efficiency of multigrid iteration against spatial variation in viscosity

To demonstrate the robustness and efficiency of the present algorithm against the spatial variation in viscosity, we performed convergence tests similar to Albers [24]. Here we solve Eqs. (1) and (2) only for prescribed distributions of buoyancy and viscosity. The distributions of buoyancy and viscosity are given by assuming the Rayleigh number Ra_t and the temperature-dependence of viscosity E_T for a prescribed distribution of temperature. (The distributions of temperature used for the tests will be introduced below.) We take into account the change of the viscosity variation in the convecting vessel by increasing the global



viscosity contrast $r_\eta = \exp(E_T)$ from 1 to 10^{10}, and estimate a threshold value of r_η (hereafter denoted by r_η^c) below which the multigrid iteration converges. The values of velocity and pressure are initially set to zero, and are iteratively updated until the L^2-norm of the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ in Eq. (4) becomes smaller than that of $\boldsymbol{b}$ by 8 orders of magnitude. All of the calculations were done with equally spaced $64 \times 64 \times 64$ mesh divisions and 6 grid levels, where the grid is coarsest for the grid level $\ell = 1$ and finest for $\ell = 6$. The mesh spacing is successively doubled as ℓ decreases, and the mesh spacing is 1/2 for grid level $\ell = 1$.

We used three distributions of temperature (hereafter denoted by Temperatures A, B, and C). In Fig. 3 we present (a), in the left column, several isothermal surfaces and (b), in the right column, the plots of the horizontally averaged temperature T_h and the maximum of the magnitude of local temperature gradient $|\mathrm{grad}(T)|_{max}$ at each height z, for these temperatures. The temperature fields are taken from snapshots of a three-dimensional time-dependent thermal convection of fluid with temperature-dependent viscosity in a cubic box. The Rayleigh number Ra_t defined by viscosity η_t is 10^3, and $E_T = \ln(10^5)$ is assumed. In Temperature A, which is taken from an early stage of the temporal evolution (nondimensional time


(a) isotherms
(b) $T_h(z)$ and $|grad(T)|_{max}(z)$
Temperature A ($t = 5.4 \times 10^{-4}$)
Temperature B

$t = 5.4 \times 10^{-4}$), the convective flow occurs only in the lowermost part of the box, and the local variation in the temperature is very small in the entire box (see the top panel of Fig. 3(b)). In Temperature B ($t = 1.52 \times 10^{-2}$), the cold fluid with $T < 0.5$ sinks into the hot interior, and a large local variation in temperature occurs around the descending flow near the bottom surface (see the middle panel of Fig. 3(b)). In Temperature C ($t = 3.6 \times 10^{-2}$), the interior of the box is nearly isothermal and, as can be also seen from the smaller $|\mathrm{grad}(T)|_{\max}$ than in Temperature B, the temperature contrast between the descending flow and the surroundings becomes smaller.

In addition, to find an appropriate implementation of the present algorithm for the multigrid operation, we employed four different types of smoothing procedures during the multigrid V-cycles. In these types, we varied the number of iterations N_s by (6) for the pre- and post-smoothing calculations at each grid level, as listed in Table 4. In Type 0, N_s is taken to be 8 at all grid levels, and is the smallest among the four types employed in these tests. In Type 1, N_s depends on the grid level ℓ. The value of N_s is taken to be equal to that in Type 0 at the finest grid ($\ell = 6$), and is successively doubled as the grid becomes coarser. In Types 2 and 3, in contrast, N_s is assumed to be dependent on both the grid level ℓ and the global viscosity contrast r_η. In both types, N_s at the finest grid is taken to be $[\ln(r_\eta) + 1]$ times larger than that in Type 1, and the values of N_s is successively multiplied by two and four as the grid becomes coarser in Types 2 and 3, respectively.

We compared the convergence behaviors of different types of smoothing procedures, for three distributions of T and various values of r_η. In Fig. 4 we show the plots against r_η of (a) the number of V-cycles N_V and (b) the function W_s given by

$$W_s = \frac{\sum_{\ell=1}^{6} n(\ell) \times m(\ell)}{m(\ell = 6)}, \tag{23}$$

where $n(\ell)$ is the total number of iterations by (6) in the smoothing procedures on grid level ℓ, and $m(\ell) \equiv 4 \times (64/2^{6-\ell})^3$ is the number of elements of the unknown vector $\boldsymbol{x}$ (i.e., the number of unknown variables; see (6)) on grid level ℓ. The numerator in (23) represents the total number of updating operations of unknown variables during the entire multigrid iterations, while the denominator represents the number of updating operation during one smoothing calculation at the finest grid ($\ell = 6$). Namely, W_s is the measure of the computational cost of the multigrid iterations, and it indicates that the total computational cost spent in the smoothing procedures during the entire multigrid iterations is equal to that virtually spent in W_s-times smoothing calculations on the finest grid. In the figure, the values are plotted only for $r_\eta \leqslant r_\eta^c$, below which the multigrid iteration converges.

The comparisons of r_η^c for different types of smoothing procedures in Fig. 4 show that the value of r_η^c becomes larger from Types 0 to 3 for all temperature distributions. For Type 0, r_η^c is 10^3 for Temperature A, and 10^2 for Temperatures B and C. By changing from Types 0 to 1, r_η^c increases to 10^7 for Temperature A, and to 10^3 for Temperatures B and C. The value of r_η^c significantly increases by further changing to

Table 4
Numbers of pre- and post-smoothing calculations N_s on grid level ℓ employed in the various types of the smoothing procedures in this study

Type	$N_s(\ell)$ on grid level ℓ
0	8
1	$8 \times 2^{N_{\mathrm{grid}}-\ell}$
2	$[\ln(r_\eta) + 1] \times 8 \times 2^{N_{\mathrm{grid}}-\ell}$
3	$[\ln(r_\eta) + 1] \times 8 \times 4^{N_{\mathrm{grid}}-\ell}$

Here, N_{grid} is the number of employed grids and r_η is the magnitude of global viscosity contrast. In the present calculations N_{grid} is taken to be 6.

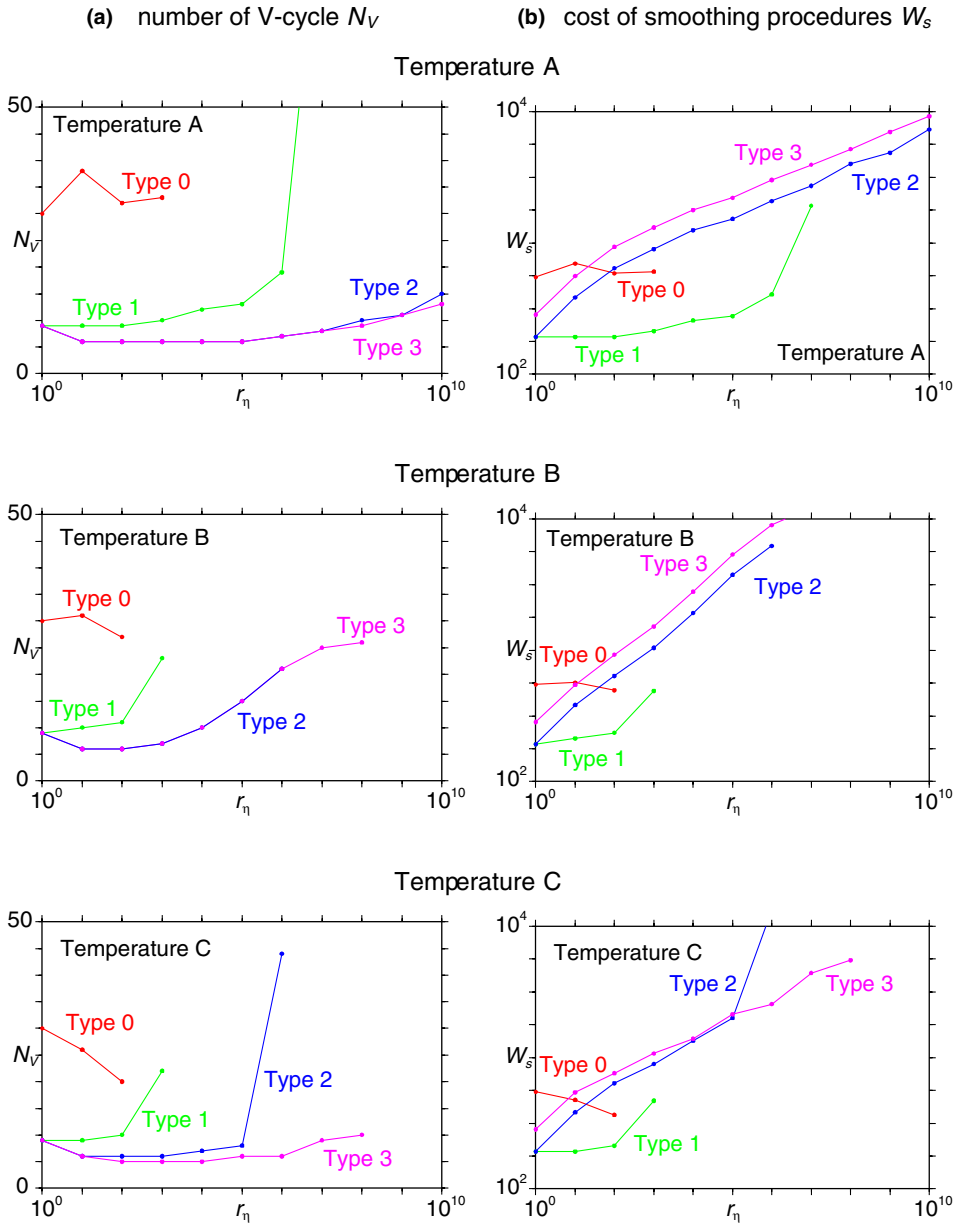


Fig. 4. Convergence behavior for various values of global viscosity contrast r_η and for temperature fields in Fig. 3 by using various types of smoothing procedures listed in Table 4. In (a) the number of V-cycles N_V is plotted against r_η. In (b) the measure of total computational cost in the smoothing procedures W_s is plotted against r_η. The values are plotted only for $r_\eta \leqslant r_\eta^c$, below which the multigrid iteration converges.

Types 2 and 3. For Type 2, r_η^c is larger than 10^{10} for Temperature A, and 10^6 for Temperatures B and C. For Type 3, r_η^c is larger than 10^{10} for Temperature A, and 10^8 for Temperatures B and C. From the comparison between these types, we conclude that the robustness of the multigrid iteration is improved by increasing

N_s. In particular, a significant improvement is obtained for large r_η by increasing N_s in proportion to $[\ln(r_\eta) + 1]$.

The multigrid iteration becomes more robust against r_η for larger N_s, since larger number of iterations of (6) can reduce the errors more effectively. To see this more clearly, we show in Fig. 5 the evolution of the L^2-norm of residual during the multigrid iteration on the finest grid ($\ell = 6$) for the cases (a) where the iteration converged with Type 1, and (b) where the iteration diverged with Type 0. For Type 1 where the multigrid iteration converges (Fig. 5(a)), the L^2-norm of residual on the finest grid is significantly reduced after coming back from the smoothing on coarser grids. Even if the residual is increased during the smoothing calculations on the finest grid (it can sometimes happen in this kind of decaying oscillation system; see Eq. (15) or (19)), it is significantly reduced by the coarse-grid correction. As a result, the residual is successively reduced by the multigrid iterations. For Type 0, where the multigrid iteration diverges (Fig. 5(b)), in contrast, the L^2-norm of residual on the finest grid is increased after coming back from the smoothing on coarser grids. The smoothing calculations on the finest grid reduces the residual, except for the first multigrid iteration. However, the decrease in the residual by the smoothing calculations is too small to overcome the increase by the coarse-grid correction. As a result, the residual is successively increased by the multigrid iterations. From the comparison between these cases, we conclude that sufficient number of smoothing calculations are necessary on coarser grids in order to ensure the convergence of multigrid iteration when a large spatial variation in viscosity is involved.

Fig. 4 also shows that the convergence rates of our method differ between the types of smoothing procedures. The comparison in the convergence rates between Types 0 and 1 for small r_η indicates that a successive increase in N_s to coarser grids significantly reduces the number of V-cycles N_V (Fig. 4(a)) as well as the computational cost W_s (Fig. 4(b)). This stabilizing effect of increasing the number of smoothing iterations on coarser grids is in agreement with the features of previous results using multigrid methods (for example [22,24]). The comparison between Types 1 to 3 in Fig. 4(a) indicate that N_V is significantly reduced by increasing N_s in proportion to $[\ln(r_\eta) + 1]$. This feature is also consistent with the earlier results (for example [22,24]). As can be seen in Fig. 4(b), however, the computational cost W_s of smoothing calculations increases as N_s becomes larger from Types 1 to 3. This is because the cost of smoothing calculations during one V-cycle becomes significantly larger.

We also note from Fig. 4 that the convergence behavior differs between the prescribed temperature fields. This difference may reflect the difference in the magnitude of the local temperature gradient |grad(T)| shown

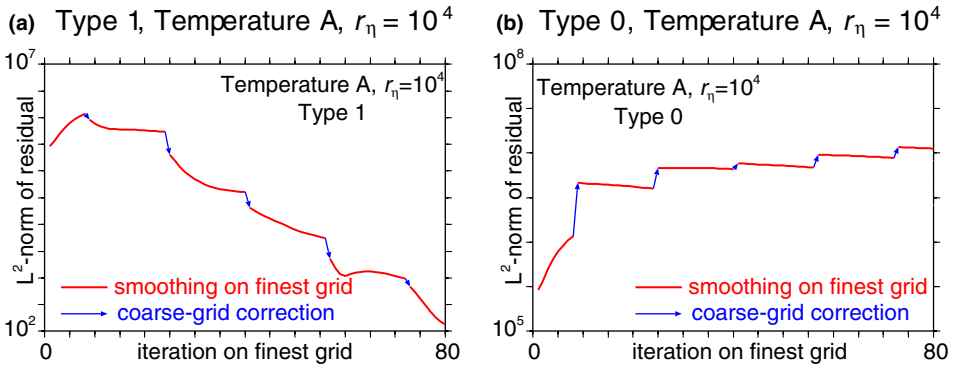


Fig. 5. Comparison in the evolution of the L^2-norm of residuals on the finest grid (grid level $\ell = 6$) during the initial five multigrid V-cycles between the case with the smoothing procedures of (a) Type 1 and (b) Type 0. Both calculations were performed with Temperature A and $r_\eta = 10^4$. The red lines indicate the evolutions obtained by the smoothing calculations on the grid level $\ell = 6$, while the blue arrows indicate the changes of residuals for $\ell = 6$ due to the coarse-grid correction.

in Fig. 3(b). The different temperature distributions provide the different distributions of viscosity. The distribution of viscosity determines the nature of the coefficient matrix $\boldsymbol{A}$ in (4) and, hence, determines the convergence behavior of (6). In particular, a larger |grad(T)| generates a larger local variation of viscosity for given r_η, and makes the convergence of (6) more difficult. The convergence behaviors presented in Fig. 4 are qualitatively consistent with the above conjecture. The multigrid iteration converges most easily for Temperature A with the smallest |grad(T)|, and least easily for Temperature B with the largest |grad(T)|. In Fig. 4, the value of r_η^c is the largest for Temperature A for any type of smoothing procedures. Although the values of r_η^c are the same for Temperatures B and C when the same type of smoothing procedures is employed, the number of V-cycles N_V is smaller for Temperature C than for Temperature B, except for the case with Type 2 for Temperature C and $r_\eta = 10^6$. These convergence behaviors are also consistent with those in the earlier studies (for example [23,24]).

4. Discussion and concluding remarks

We developed a numerical algorithm for solving mantle convection problems with strongly variable viscosity. Equations for conservation of mass and momentum for highly viscous and incompressible fluids are solved iteratively by a multigrid method in combination with pseudo-compressibility and local time stepping techniques. In order to demonstrate its efficiency, the present algorithm has been implemented into a mantle convection simulation program based on the finite-volume discretization in a three-dimensional rectangular domain. Benchmark comparison with previous two- and three-dimensional calculations including the temperature- and/or depth-dependent viscosity revealed that accurate results are obtained even for the cases with viscosity variations of several orders of magnitude. We could also significantly improve the robustness of the numerical method against a spatial variation in viscosity by increasing the pre- and post-smoothing calculations in the multigrid operations. We achieved convergence even for the viscosity contrasts up to 10^{10}, although the convergence rate deteriorates with increasing viscosity variations. The present algorithm can be further applied to the numerical models under more realistic conditions, such as spherical shell geometry and so on.

The results of convergence tests described in Section 3.3 suggest that the convergence of multigrid method in combination with the present algorithm is determined by the accuracy of the coarse-grid correction when a large spatial variation in viscosity is incorporated. In the present convergence tests, the accuracy of coarse-grid correction was improved by increasing the amount of smoothing calculations on coarser grids during one multigrid iteration. This strategy always works, since the present iterative procedure never diverges as long as the local time stepping satisfies the CFL condition (see Section 2.3). However, it turned out that total computational costs significantly increased as the magnitude of viscosity contrast becomes larger (see Fig. 4(b)). Therefore, one needs to further accelerate the rate convergence of (6) in order to improve the efficiency of smoothing calculations and, in turn, to improve the accuracy of coarse-grid correction.

The discussion in Section 2.3 also suggests that the rate of convergence of the present smoothing algorithm can be accelerated by choosing the matrix $\boldsymbol{T}$ in (6) more properly. As has been pointed out in Section 2.3, the matrix $\boldsymbol{T}$ acts as a "preconditioner" for (4). That is, the convergence of (6) becomes faster as $\boldsymbol{T}$ better approximates $\boldsymbol{A}^{-1}$. In the present application to mantle convection problems, we had chosen $\boldsymbol{T}$ as a diagonal matrix by the use of the local time stepping approach. In addition, we had determined the values of nonzero elements of $\boldsymbol{T}$ from the assumption that both the rate of "diffusion" in velocity components and the rate of propagation of "pseudo-sound wave" were kept almost uniform in space. This choice of $\boldsymbol{T}$ corresponds to the preconditioning similar to the diagonal scaling of (4). However, apart from the local time stepping approach, we can use any arbitrary matrix $\boldsymbol{T}$ rather than a diagonal matrix. In other words, the convergence of (6) is accelerated if $\boldsymbol{T}$ preconditions (4) more properly than in the present paper. We can

construct $\boldsymbol{T}$, for example, by an incomplete factorization of $\boldsymbol{A}$. Since the matrix $\boldsymbol{T}$ defined thus is most likely to be a better approximation of $\boldsymbol{A}^{-1}$ than that employed here, the convergence of (6) is expected to be faster than in the present cases. On the other hand, constructing $\boldsymbol{T}$ by an incomplete factorization has several disadvantages because (i) more memory is required to store all of the nonzero elements of $\boldsymbol{T}$ and (ii) the operation of an incomplete factorization is difficult to vectorize and parallelize. We should, therefore, take into account the specific computer architecture (such as scalar or vector processors) in choosing the most appropriate $\boldsymbol{T}$ in terms of preconditioning techniques.

However, the most essential improvement possible for the present numerical method is the multigrid operation itself, rather than the smoothing calculations during multigrid iteration. As has been suggested in Section 3.3, the reason why the multigrid iteration diverges for larger r_η is that the coarse-grid correction becomes less efficient as r_η increases (see Figs. 4 and 5). The inefficiency of coarse-grid correction may come from an inappropriate treatment of the influences of variable viscosity in the multigrid operation. In the present numerical method, as well as in most of earlier models based on finite-volume discretization [28,22,24], we followed a "standard" strategy of multigrid. Namely, a linear interpolation was used for transferring the residual and error between adjacent grid levels. In addition, the discretized equations on coarser grids were derived by directly discretizing the differential equations on the particular coarser grids. However, as has been already acknowledged in the literature of multigrid [12,13], the standard strategy does not efficiently work when the differential equations contain strongly varying coefficients. One of the potential remedies is to develop the discretized equations based on Galerkin coarse-grid approximation [12,13], where the coefficient matrix $\bar{\boldsymbol{A}}$ on a coarse grid is defined from the matrix on a fine grid $\boldsymbol{A}$ by

$$\bar{\boldsymbol{A}} = \boldsymbol{RAP}, \tag{24}$$

where $\boldsymbol{R}$ and $\boldsymbol{P}$ are the restriction and prolongation operators, respectively. The Galerkin coarse-grid approximation can automatically define the coefficient matrix $\bar{\boldsymbol{A}}$ on the coarse grid, so as to sufficiently approximate $\boldsymbol{A}$ defined on the fine grid. But it is not straightforward to use this approach in finite-volume models, since it makes the coefficient matrices on the coarse grids very complicated, with increased stencils, compared to those coming from the direct discretization. On the other hand, another potential remedy is to use the operator-dependent transfer operators [12,13]. These transfer operators utilize the structure of $\boldsymbol{A}$ with the variations in its coefficients and, hence, improve the accuracy of the restriction and prolongation operations. Indeed, Yang and Baumgardner [18] incorporated these transfer operators together with the Galerkin coarse-grid operators into their finite-element models, and demonstrated significant improvements in the robustness and efficiency of their multigrid procedures even for the cases with strongly viscosity variations. We expect that the robustness of our multigrid method presented in Section 3 can be improved by incorporating the operator-dependent transfer operators into the present model, although their efficiency is left uncertain when used without the Galerkin approximation [24].

It is also an important issue to improve the robustness of the multigrid iterations against a "sharp" or "discontinuous" variation in viscosity, in addition to a "smooth" variation considered here. One of the major unsolved problems in the numerical study of mantle dynamics is to reproduce the motion of surface plates in the framework of mantle convection. Earlier numerical studies (for a review [43]) had demonstrated that the generation of localized zones of low viscosity around the plate boundaries plays an important role in reconciling the fluid-like flow of mantle with the discrete motion of surface plates. However, it is very difficult in essence for the multigrid method to deal with a local variation in viscosity as long as uniform mesh spacing is considered, since such local features are hardly "visible" on coarser grids. Recently, in several fields of geophysical fluid dynamics, adaptive gridding techniques, such as the local mesh refinement [24], its combination with spectral-element method [44], and the wavelet-based method [45], are utilized. Since these methods allow taking the spatial resolution non-uniformly in space, a local variation of viscosity can be handled accurately by, for example, using finer resolution around the regions of local variation. We thus speculate that the combination of these techniques together with the

present method is one promising approach to reproduce the motion of surface plates in the numerical model of mantle convection.

Acknowledgments

We thank Tomoeki Nakakuki, David A. Yuen, Paul J. Tackley, and Masaki Ogawa for discussion and comments. We also thank two anonymous reviewers for valuable comments which greatly improved the manuscript. The calculations presented in this paper were in part done by Earth Simulator at Japan Agency for Marine-Earth Science and Technology.

References

[1] D.L. Turcotte, G. Schubert, Geodynamics: Applications of Continuum Physics to Geological Problems, Wiley, New York, 1982, p. 450.
[2] F.D. Stacey, Physics of the Earth, third ed., Brookfield Press, Brisbane, Australia, 1992.
[3] D.L. Turcotte, E.R. Oxburgh, Finite amplitude convection cells and continental drift, J. Fluid Mech. 28 (1967) 29–42.
[4] D.P. McKenzie, J. Roberts, N.O. Weiss, Convection in the Earth's mantle, Tectonophysics 19 (1973) 89–103.
[5] G. Schubert, D.L. Turcotte, P. Olson, Mantle convection in the Earth and planets, Cambridge University Press, Cambridge, 2001.
[6] J. Weertman, The creep strength of the Earth's mantle, Rev. Geophys. 8 (1970) 145–168.
[7] S.-I. Karato, P. Wu, Rheology of the upper mantle: a synthesis, Science 260 (1993) 771–778.
[8] B. Travis, P. Olson, G. Schubert, The transition from two-dimensional to three-dimensional planforms in infinite-Prandtl-number thermal convection, J. Fluid Mech. 216 (1990) 71–91.
[9] U. Christensen, H. Harder, 3D convection with variable viscosity, Geophys. J. Int. 104 (1991) 213–226.
[10] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comput. 31 (1977) 333–390.
[11] A. Brandt, Multilevel adaptive computations in fluid dynamics, AIAA J. 18 (1980) 1165–1172.
[12] P. Wesseling, An Introduction to Multigrid Methods, Wiley, New York, 1992.
[13] U. Trottenberg, C. Oosterlee, A. Schüller, Multigrid, Academic Press, New York, 2001.
[14] E.M. Parmentier, C. Sotin, B.J. Travis, Turbulent 3D thermal convection in an infinite Prandtl number, volumetrically heated fluid: implications for mantle dynamics, Geophys. J. Int. 116 (1994) 241–251.
[15] J.R. Baumgardner, Three-dimensional treatment of convective flow in the Earth's mantle, J. Statist. Phys. 39 (1985) 501–511.
[16] H.-P. Bunge, M.A. Richards, J.R. Baumgardner, Effect of depth-dependent viscosity on the planform of mantle convection, Nature 379 (1996) 436–438.
[17] H.-P. Bunge, M.A. Richards, C. Lithgow-Bertelloni, J.R. Baumgardner, S.P. Grand, B.A. Romanowicz, Time scales and heterogeneous structure in geodynamic Earth models, Science 280 (1998) 91–95.
[18] W.-S. Yang, J.R. Baumgardner, A matrix-dependent transfer multigrid method for strongly variable viscosity infinite Prandtl number thermal convection, Geophys. Astrophys. Fluid Dyn. 92 (2000) 151–195.
[19] A. Ramage, A.J. Wathen, Iterative solution techniques for the Stokes and Navier–Stokes equations, Int. J. Numer. Methods Fluids 19 (1994) 67–83.
[20] L.-N. Moresi, V. Solomatov, Numerical investigation of 2D convection with extremely large viscosity variations, Phys. Fluids 7 (1995) 2154–2162.
[21] L.-N. Moresi, M. Gurnis, Constraints on the lateral strength of slabs from three-dimensional dynamic flow models, Earth Planet. Sci. Lett. 138 (1996) 15–28.
[22] R.A. Trompert, U. Hansen, The application of a finite volume multigrid method to three-dimensional flow problems in a highly viscous fluid with a variable viscosity, Geophys. Astrophys. Fluid Dyn. 83 (1996) 261–291.
[23] R. Trompert, U. Hansen, On the Rayleigh number dependence of convection with a strongly temperature-dependent viscosity, Phys. Fluids 10 (1998) 351–364.
[24] M. Albers, A local mesh refinement multigrid method for 3D convection problems with strongly variable viscosity, J. Comput. Phys. 160 (2000) 126–150.
[25] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington, DC, 1980, p. 197.
[26] C. Auth, H. Harder, Multigrid solution of convection problems with strongly variable viscosity, Geophys. J. Int. 137 (1999) 793–804.
[27] S.P. Vanka, Block implicit multigrid solution of Navier–Stokes equations in primitive variables, J. Comput. Phys. 65 (1986) 138–158.

[28] P.J. Tackley, Three-dimensional models of mantle convection: influence of phase transitions and temperature-dependent viscosity, Ph.D. Thesis, California Institute of Technology, 1994.
[29] P.J. Tackley, Effects of strongly variable viscosity on three-dimensional compressible convection in planetary mantles, J. Geophys. Res. 101 (1996) 3311–3332.
[30] J.T. Ratcliff, P.J. Tackley, G. Schubert, A. Zebib, Transitions in thermal convection with strongly variable viscosity, Phys. Earth Planet. Inter. 102 (1997) 201–212.
[31] P.J. Tackley, Self-consistent generation of tectonic plates in three-dimensional mantle convection, Earth Planet. Sci. Lett. 157 (1998) 9–22.
[32] P.J. Tackley, Self-consistent generation of tectonic plates in time-dependent, three-dimensional mantle convection simulations. 1. Pseudoplastic yielding, Geochem. Geophys. Geosyst. 1 (2000).
[33] P.J. Tackley, Self-consistent generation of tectonic plates in time-dependent, three-dimensional mantle convection simulations. 2. Strain weakening and asthenosphere, Geochem. Geophys. Geosyst. 1 (2000).
[34] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[35] H. Schmeling, W.R. Jacoby, On modelling the lithosphere in mantle convection with nonlinear rheology, J. Geophys. 50 (1981) 89–100.
[36] M. Ogawa, G. Schubert, A. Zebib, Numerical simulations of three-dimensional thermal convection in a fluid with strongly temperature-dependent viscosity, J. Fluid Mech. 233 (1991) 299–328.
[37] F. Sotiropoulos, G. Constantinescu, Pressure-based residual smoothing operator for multistage pseudocompressibility algorithms, J. Comput. Phys. 133 (1997) 129–145.
[38] C.-H. Tai, Y. Zhao, Parallel unsteady incompressible viscous flow computations using an unstructured multigrid method, J. Comput. Phys. 192 (2003) 277–311.
[39] S.E. Rogers, D. Kwak, U. Kaul, On the accuracy of the pseudocompressibility method in solving the incompressible Navier–Stokes equation, Appl. Math. Modelling 11 (February) (1987) 35–44.
[40] J. Shen, On a new pseudocompressibility method for the incompressible Navier–Stokes equations, Appl. Numer. Math. 21 (1996) 71–90.
[41] B. Blankenbach, F. Busse, U. Christensen, L. Cserepes, D. Gunkel, U. Hansen, G. Jarvis, M. Koch, G. Marquatt, D. Moore, P. Olson, H. Schmeling, T. Schnaubelt, A benchmark comparison for mantle convection codes, Geophys. J. Int. 98 (1989) 23–38.
[42] F.H. Busse, U. Christensen, R. Clever, L. Cserepes, C. Gable, E. Giannandrea, L. Guillow, G. Houseman, H.-C. Nataf, M. Ogawa, M. Parmentier, C. Sotin, B. Travis, 3D convection at infinite Prandtl number in Cartesian geometry – a benchmark comparison, Geophys. Astrophys. Fluid Dyn. 75 (1993) 39–59.
[43] D. Bercovici, The generation of plate tectonics from mantle convection, Earth Planet. Sci. Lett. 205 (2003) 107–121.
[44] A. Fournier, M.A. Taylor, J.J. Tribbia, The spectral element atmosphere model (SEAM): high-resolution parallel computation and localized resolution of regional dynamics, Monthly Weather Review 132 (2004) 726–748.
[45] O.V. Vasilyev, D.A. Yuen, Y.Y. Podladchikov, Applicability of wavelet algorithm for geophysical viscoelastic flow, Geophys. Res. Lett. 24 (1997) 3097–3100.