Computers and Fluids 190 (2019) 217–253

Contents lists available at ScienceDirect

Computers and Fluids

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

Implementation of a Volume-of-Fluid method in a finite element code with applications to thermochemical convection in a density stratified fluid in the Earth's mantle

Jonathan M. Robey, Elbridge Gerry Puckett*

Department of Mathematics, University of California, Davis, CA 95616, USA

ARTICLE INFO

Article history:
Received 17 January 2018
Revised 16 May 2019
Accepted 20 May 2019
Available online 21 May 2019

Keywords:
Volume-of-Fluid method
Adaptive mesh refinement
Rayleigh–Bénard problem
Thermochemical convection
Rayleigh Taylor Instability
Compositionally stratified fluid
Large low shear wave velocity provinces

ABSTRACT

We describe the implementation of a second-order accurate Volume-of-Fluid interface tracking algorithm in the open source finite element code ASPECT that is designed to model convection and other processes in the Earth's mantle. This involves the solution of the incompressible Stokes equations coupled to an advection diffusion equation for the temperature, a Boussinesq approximation that governs the dependence of the density on the temperature, and an advection equation for a marker indicating two initial (constant) density states, that are passively advected in the underlying flow field. The Volume-of-Fluid method in ASPECT is fully parallelized and fully integrated with ASPECT's adaptive mesh refinement algorithm. We present the results of several interface tracking benchmarks in order to demonstrate the accuracy of the method, as well as the results of several benchmarks commonly used in the computational mantle convection community. Finally, we present the results of computations with and without adaptive mesh refinement of a model problem involving thermochemical convection in a computationally stratified fluid designed to provide insight into how thermal plumes, that eventually reach the Earth's surface as ocean island basalts, originate at structures near the core-mantle boundary known as Large Low Shear wave Velocity Provinces or "LLSVPs". LLSVPs are structures in parts of the lowermost portion of the Earth's mantle, characterized by slow shear wave velocities and higher density than the surrounding mantle, which were discovered by seismic tomography of the deep Earth.

1. Introduction

Over more than the past four decades there have been many numerical methods developed to study convection and other processes in the Earth's mantle. In particular, there have been a sequence of codes developed over this period of time that are now freely available to any individual who wishes to study mantle dynamics. They include HC [25,26,71], ConMan [38], CitCom S [48,75,85], Citcom CU [51,84] and ASPECT [29,41]. These codes, as well as others, can be downloaded from the Computational Infrastructure for Geodynamics (CIG) at U.C. Davis.[1]

There are a large number of problems associated with the Earth's mantle that contain one or more interfaces in some form or another. Although there have been some very specialized computational models of interfaces in the mantle, for example, the dynamics of bubbles and plumes [45–47], it is only recently that researchers have begun to implement interface tracking algorithms in codes designed to model convection and other processes in the entirety of the Earth's mantle; e.g., [66]. However, to our knowledge, the Volume-of-Fluid (VOF) method has not yet been implemented in a code designed to model convection in the Earth's mantle or, more generally, used by researchers to model geodynamic flows.

In this article we describe the implementation of a second-order accurate VOF interface tracking algorithm in the open source finite element code ASPECT, which is an acronym for "Advanced Solver for Problems in Earths ConvecTion" [29,41]. ASPECT is a parallel, extensible finite element code designed to model thermal convection and other processes in the Earth's mantle in two and three dimensions. It is built on the deal.II Finite Element Library [2,6], which includes adaptive mesh refinement (AMR) [10], and has been shown to scale to thousands of processors [22]. ASPECT has been extended to model other processes that occur in the mantle, such as modeling grain size evolution in the mantle [17], melt generation and migration [18], as well as other problems.

* Corresponding author.
E-mail address: egpuckett@ucdavis.edu (E.G. Puckett).

[1] CIG is an NSF funded, community driven organization that advances Earth science by developing and disseminating software for geophysics and related fields.

https://doi.org/10.1016/j.compfluid.2019.05.015
0045-7930/

Table 1
A list of symbols used in this paper.

Symbol	Quantity	Units	Symbol	Quantity	Units
$\mathbf{u}$	Velocity	m/s	C	Composition	–
p	Dynamic pressure	Pa	D	Compositional diffusivity	m^2/s
T	Temperature	K	α	Coefficient of thermal expansion	1/K
T_0	Temperature at the top	K	d	Vertical height of fluid layer	m
T_1	Temperature at the bottom	K	h_e	Characteristic size of cell e	m
ΔT	Temperature difference	K	Pr	Prandtl number	$\frac{\mu}{\rho\kappa}$
μ	Viscosity	$Pa \cdot s$	Le	Lewis number	$\frac{\kappa}{D}$
κ	Thermal diffusivity	m^2/s	Ra	Rayleigh number	$\frac{\rho_0 g \alpha \Delta T d^3}{\mu \kappa}$
ρ	Density	$kg \cdot m^{-3}$	B	Buoyancy ratio	$\frac{\Delta\rho}{\rho_0 \alpha \Delta T}$
ρ_0	Reference density	$kg \cdot m^{-3}$	Pe_e	*local* Péclet number on cell e	$\frac{h_e \|\mathbf{u}\|_\infty}{\kappa}$
$\Delta\rho$	Density difference	$kg \cdot m^{-3}$			

There is currently a very active community of researchers extending ASPECT to new problem areas and improving existing algorithms. Our VOF algorithm is fully parallelized and is designed to work efficiently with ASPECT's AMR algorithm. The User Manual for the most recent stable release of ASPECT can be found here [5].

Recent studies utilizing seismic imaging have revealed large regions with anomalous seismic properties in the lower mantle. In particular, there are two dome-like regions beneath Africa and the Pacific Ocean with low shear-wave velocities that extend some 1000 km above the core-mantle boundary and have horizontal dimensions of several thousand kilometers [15,21]. Most interpretations propose that these heterogeneities are compositional in nature, differing from the surrounding mantle, an interpretation that would be consistent with chemical geodynamic models. Based on geological and geochemical studies it has been argued that these so-called 'Large Low Shear wave Velocity Provinces' (LLSVPs) have persisted for billions of years [9]. In this article we use the VOF method that we have recently implemented in ASPECT to compute solutions to a model problem designed to understand the dynamics of plumes that form on the LLSVPs, entrain some of the material in the LLSVP that differs from the surrounding mantle, and brings it to the Earth's surface. The model problem consists of two horizontal layers, equal in height, in a rectangle, with a density difference of $\Delta\rho = \rho - \rho_0 \geq 0$, where ρ_0 is the density of the upper layer. The initial condition for the temperature is a perturbation from the well-known static temperature field, connecting the temperature boundary conditions T_0 at the top of the rectangle and T_1 at the bottom of the rectangle [78]. We study of a range of density differences $\Delta\rho$ that we characterize by the non-dimensional buoyancy number B, which is the ratio of $\Delta\rho$ to $\rho_0 \alpha \Delta T$, where $\Delta T = T_1 - T_0$, and α is the volumetric coefficient of thermal expansion. The temperature perturbation initially drives the convection and, depending on the value of B, determines the dynamics and structure of the resulting flow field (Table 1).

In Section 2 we begin by describing the equations that govern thermochemical convection in the mantle and the modification to these equations that we use to model density stratification in such flows. Then, in Section 3 we describe the numerical methodology, including the underlying Finite Element Method (FEM) and the coupling of our VOF method to this FEM. In Section 4 we begin in Section 4.2 by presenting two standard interface tracking benchmarks, each in a (different) stationary velocity field, in order to demonstrate that our VOF method attains its design rate of second order accuracy on simple flows. In Section 4.3 we then present a sequence of computations of a time-dependent problem; namely, a falling circular region of greater density than the surrounding fluid and measure the convergence rate of the VOF method in this time-dependent flow field with an interface across which there is a jump in density. In Section 4.4 we present the results of two benchmarks commonly used by researchers in the computational mantle convection community. Finally in Section 4.5 we present computational results of a model problem first proposed in [64], which is designed to provide insight into how thermal plumes, that are thought to eventually reach the Earth's surface as ocean island basalts, originate at structures on the core-mantle boundary known as LLSVPs. We discuss these latter computational results in Section 5 and, in Section 6, we present our conclusions.

2. Thermochemical convection with density stratification

In this section we present in detail the equations associated with the model problem, which we briefly described above. After giving an overview of the numerical methodology we use to approximate solutions of these equations in Section 3 and, in particular, an in depth description of how we implementation of the VOF algorithm in ASPECT in Section 3.4 we present a sequence of computations of the model problem using the VOF methodology in Section 4.5.

2.1. The dimensional form of the equations

In order to study the efficacy of our implementation of a VOF algorithm in ASPECT to model processes that occur in the Earth's mantle, we compute a problem that emphasizes the effect of a compositional density difference on thermal convection. We consider a two-dimensional flow in a horizontal fluid layer with a thickness or height d. Our problem domain Ω has width $3d$ and height d. At a given reference temperature T_0 the region $d/2 < y \leq d$ has a compositional density of ρ_0 and the region $0 \leq y < d/2$ has a compositional density of $\rho_0 + \Delta\rho$ where $\Delta\rho \ll \rho_0$.

We also introduce a composition variable $C(x, y, t)$ defined by

$$C = \frac{\rho - \rho_0}{\Delta\rho}\,. \tag{1}$$

The composition C is the concentration of the dense fluid as a function of space and time. The initial condition for C is

$$C(x, y, t = 0) = \begin{cases} 1 & \text{for} \quad 0 \leq y \leq d/2\,, \\ 0 & \text{for} \quad d/2 < y \leq d\,. \end{cases} \tag{2}$$

The upper boundary, at $y = d$, has temperature T_0 and the lower boundary at $y = 0$ has temperature T_1. The fluid is assumed to have a constant viscosity μ, which is large. The Prandtl number is assumed to be very large,

$$\mathrm{Pr} = \frac{\mu}{\rho_0 \kappa} \gg 1\,, \tag{3}$$

where κ is the thermal diffusivity, so that inertial effects can be neglected. The fluids in the high density and low density layers are

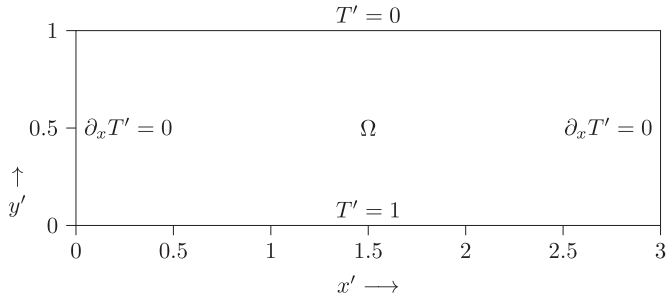


Fig. 1. The geometry of the (nondimensional) computational domain Ω shown with the temperature boundary conditions on the four side walls. The velocity boundary conditions on the side walls are $\mathbf{u} \cdot \mathbf{n} = 0$ (no flow) and $\partial \mathbf{u} / \partial \boldsymbol{\tau} = \mathbf{0}$ (free slip) where $\mathbf{n}$ and $\boldsymbol{\tau}$ are the unit normal and tangential vectors to the boundary respectively.

immiscible; i.e., they cannot mix by diffusion. Similarly, the Lewis number is also assumed to be large,

$$\mathrm{Le} = \frac{\kappa}{D} \gg 1\,, \tag{4}$$

where D is the diffusion coefficient for the compositional variable C. Thus, the discontinuous boundary between the high density and low density fluids is preserved indefinitely.

The problem we have posed requires the solution of the standard equations for thermal convection with the addition of an equation for the compositional field C that tracks the density field. The governing equations are described in detail in [69,78].

We make the assumption that the Boussinesq approximation

$$\rho(x, y, t) = \rho_0 \left(1 - \alpha \ (T - T_0)\right) + \Delta\rho\, C\,. \tag{5}$$

holds; namely, that density differences associated with convection $\rho_0\, \alpha\ (T_1 - T_0)$ and $\Delta\rho$ are small compared with the reference density ρ_0.

Conservation of mass requires

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{6}$$

where x and y denote the horizontal and vertical spatial coordinates, oriented as shown in Fig. 1, and u and v denote the horizontal and vertical velocity components, respectively. We use the Stokes equations

$$0 = \frac{-\partial P}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{7}$$

$$0 = \frac{-\partial P}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \rho_0 \alpha (T - T_0) g - \Delta\rho\ C\ g\,, \tag{8}$$

where α is the coefficient of thermal expansion, g is the gravitational acceleration in the negative (downward) y direction as shown in Fig. 1, and

$$P = p + \rho_0\, g y$$

where p is the dynamic pressure and $\rho_0 g y$ is the isostatic pressure. Conservation of energy requires

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \kappa\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \tag{9}$$

where κ is the thermal diffusivity.

When there is no compositional diffusion, i.e., $D = 0$, the composition variable C satisfies the advection equation

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} = 0\,. \tag{10}$$

2.2. The nondimensional form of the equations

We introduce the nondimensional variables

$$\begin{aligned} x' &= \frac{x}{d}, \quad y' = \frac{y}{d}, \quad t' = \frac{\kappa}{d^2}\, t, \\ u' &= \frac{d}{\kappa}\, u, \quad v' = \frac{d}{\kappa}\, v, \quad \rho' = \frac{\rho}{\rho_0}, \\ T' &= \frac{T - T_0}{T_1 - T_0}, \quad P' = \frac{d^2 P}{\mu\kappa}\,, \end{aligned} \tag{11}$$

and the two nondimensional parameters, the Rayleigh number Ra and the buoyancy ratio B

$$\mathrm{Ra} = \frac{\rho_0\, g \alpha\, (T_1 - T_0)\, d^3}{\mu\, \kappa}\,, \tag{12}$$

$$\mathrm{B} = \frac{\Delta\rho}{\rho_0\, \alpha\, (T_1 - T_0)}\,. \tag{13}$$

where $g = 9.80665\, m/s^2$ is the acceleration due to gravity.

Substitution of Eqs. (11)–(13) into Eqs. (6)–(10) gives

$$\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} = 0\,, \tag{14}$$

$$0 = \frac{-\partial P'}{\partial x'} + \frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2}\,, \tag{15}$$

$$0 = \frac{-\partial P'}{\partial y'} + \frac{\partial^2 v'}{\partial x'^2} + \frac{\partial^2 v'}{\partial y'^2} + \mathrm{Ra}\, T' - \mathrm{Ra}\, \mathrm{B}\, C\,, \tag{16}$$

$$\frac{\partial T'}{\partial t'} + u'\frac{\partial T'}{\partial x'} + v'\frac{\partial T'}{\partial y'} = \frac{\partial^2 T'}{\partial x'^2} + \frac{\partial^2 T'}{\partial y'^2}\,, \tag{17}$$

$$\frac{\partial C}{\partial t'} + u'\frac{\partial C}{\partial x'} + v'\frac{\partial C}{\partial y'} = 0\,. \tag{18}$$

This is the superposition of a Rayleigh-Taylor problem and a Rayleigh–Bénard problem [11,78]. In the isothermal limit, $T_0 = T_1$, it is the classic Rayleigh-Taylor problem. If C is positive, a light fluid is above the heavy fluid and in a downward gravity field the fluid layer is stable. If $\Delta\rho$ is negative, a heavy fluid lies over a light fluid and the layer is unstable. Flows will transfer the heavy fluid to the lower half and the light fluid to the upper half and the density layer will overturn. If $\Delta\rho = 0$ and hence, $\mathrm{B} = 0$, this is the classic Rayleigh–Bénard problem for thermal convection. The governing parameter is the Rayleigh number Ra. If $0 < \mathrm{Ra} < \mathrm{Ra}_c$, the critical Rayleigh number, no flow will occur; e.g., see [78]. If $\mathrm{Ra}_c < \mathrm{Ra} < \mathrm{Ra}_t$, where Ra_t is the Rayleigh number beyond which *thermal* turbulence develops, steady cellular flow will occur. If $\mathrm{Ra} > \mathrm{Ra}_t$, the flow becomes unsteady and *thermally* turbulent.

If $\mathrm{Ra} > \mathrm{Ra}_c$ and B is small, the boundary between the density differences will not block the flow driven by thermal convection. Kinematic mixing will occur and the composition will homogenize so that the density is constant. Whole layer convection will occur. If B is large, the density difference boundary will block the flow driven by thermal convection. The compositional boundary will be displaced vertically but will remain intact. Layered convection will occur with the compositional boundary, the boundary between the convecting layers. In this work the Rayleigh number Ra defined in Eq. (12) is based on the domain thickness d and this is the case for which we will show numerical computations.

3. The numerical methodology

In the following discussion of the numerical methodology, we will only consider the dimensionless Eqs. (14)–(18) and drop the primes associated with the dimensionless variables. The vector form of the dimensionless equations on the two dimensional rectangular domain $\Omega = [0, 3] \times [0, 1]$ shown in Fig. 1 are given by

$$-\nabla^2 \mathbf{u} + \nabla P = (-\text{Ra}\ T + \text{Ra}\ \text{B}\ C)\,\mathbf{g} \tag{19}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{20}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T \tag{21}$$

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0, \tag{22}$$

where $\mathbf{u} = (u, v)$ is the velocity and $\mathbf{g} = (0, -1)$ is the unit vector pointing downward.

Note that the composition Eq. (22) is equivalent to

$$\frac{DC}{Dt} = \frac{\partial C}{\partial t} + u\,\frac{\partial C}{\partial x} + v\,\frac{\partial C}{\partial y} = 0\,, \tag{23}$$

where

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u\,\frac{\partial}{\partial x} + v\,\frac{\partial}{\partial y} \tag{24}$$

is the *material derivative*. Eq. (23) implies that the composition C is constant on particle paths in the flow [13]. Furthermore, since by (20) the velocity $\mathbf{u}$ is divergence free, the composition Eq. (22) can be written in conservation form

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u} C) = 0\,, \tag{25}$$

implying that the composition C is a conserved quantity - it is neither created nor destroyed as it is advected in the flow field.

We assume no-flow and free-slip velocity boundary conditions on all boundaries,

$$\mathbf{u} \cdot \mathbf{n} = 0 \qquad (\text{no-flow})\,, \tag{26}$$

$$\frac{\partial \mathbf{u}}{\partial \boldsymbol{\tau}} = 0 \qquad (\text{free slip})\,, \tag{27}$$

where $\mathbf{n}$ and $\boldsymbol{\tau}$ are the unit normal and tangential vectors to the boundary respectively. We impose Dirichlet boundary conditions for the temperature on the top and bottom of the computational domain and Neumann boundary conditions (no heat flux) on the sides of the computational domain,

$$T(x, 0, t) = 1\,, \tag{28}$$

$$T(x, 1, t) = 0\,, \tag{29}$$

$$\partial_x T(0, y, t) = 0\,, \tag{30}$$

$$\partial_x T(0, y, t) = 0\,. \tag{31}$$

The geometry of the computational domain together with the boundary conditions on the temperature are shown in Fig. 1. In this work we only consider no-flow boundary conditions (26). Therefore, we do not need to specify boundary conditions on the compositional field C, since there can be no flow of the composition through the boundaries.

3.1. Decoupling of the nonlinear system

The incompressible Stokes equations can be considered as a constraint on the temperature and composition at any given time leading to a nonlinear system of equations. To solve this nonlinear system, we apply the Implicit Pressure Explicit Saturation (IMPES) approach, originally developed for computing solutions of equations for modeling problems in porous media flow [35,70], to decouple the incompressible Stokes Eqs. (14)–(16) from the temperature and compositional Eqs. (17)–(18). This leads to three discrete systems of linear equations, the Stokes equations, the temperature equation, and the composition equation, thereby allowing each equation to be solved easily and efficiently.

3.2. Discretization of the Stokes equations

Let t^k denote the discretized time at the kth time step with a time step size of $\Delta t^k = t^k - t^{k-1}$, $k = 0, 1, \ldots$ Given the temperature T^k and composition C^k at time $t = t^k$, we first solve for our approximation to the Stokes Eqs. (14)–(16) to obtain the velocity $\mathbf{u}^k = (u^k, v^k)$ and pressure P^k

$$-\nabla^2 \mathbf{u}^k + \nabla P^k = \left(-\text{Ra}\ T^k + \text{Ra}\ \text{B}\, C^k\right)\,\mathbf{g}\,, \tag{32}$$

$$\nabla \cdot \mathbf{u}^k = 0. \tag{33}$$

For the incompressible Stokes Eqs. (32) and (33), we use the standard mixed FEM method with a Taylor–Hood element [20] for the spatial approximation. We refer the interested reader to [41] for a more detailed discussion of the spatial discretization and the choice of Stokes preconditioners and solvers.

3.3. The discretization of the temperature equation

In mantle convection the thermal diffusivity κ is very small compared to the magnitude of the velocity. Thus, if we let $\|\mathbf{u}\|_{\infty,e}$ denote the maximum magnitude of the velocity on cell e and h_e denote the characteristic size of this cell, in some computations, even for very fine meshes (i.e., small h_e), the local Péclet number on cell e,

$$\text{Pe}_e \overset{\text{def}}{=} \frac{h_e\,\|\mathbf{u}\|_{\infty,e}}{\kappa}\,, \tag{34}$$

is usually in the range 10^2 to 10^4. For such high local Péclet number problems, standard finite element discretizations introduce spurious oscillations in the vicinity of steep gradients of advected quantities, even in the presence of some (relatively small) diffusion [20]. Therefore, some form of stabilization must be added to the discrete formulation of the advection-diffusion equation for the temperature.

In all of the computations presented here we use the algorithm currently implemented in ASPECT to approximate the spatial and temporal terms *in the temperature* Eq. (21) *only*. This algorithm is based on the so-called 'entropy viscosity' method, which is described in detail in [24,41]. The entropy-viscosity stabilization method adds additional (i.e., artificial) 'viscosity' where the local Péclet number is large and the solution is not smooth. In other words, we approximate solutions of the modified temperature equation

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot (\kappa + \nu_h(T))\nabla T\,, \tag{35}$$

with an artificial diffusion term $\nu_h(T)$ added to the equation. Here the entropy viscosity function $\nu_h(T)$ is a non-negative constant within each cell, which can vary from cell to cell.

Note that we have written Eq. (35) in the dimensional form (9) in order to make our discussion of the entropy-viscosity stabilization technique consistent with the discussion in [41].[2] Also note that in all of the computational results shown in this article the thermal diffusivity κ is constant, but that in general this need not be the case. Hence, we have written the advection diffusion equation for the temperature in (35) in a more general form.

Conceptually, in regions where the temperature field T is smooth ν_h should be small, and in regions with significant variability ν_h should be of a size that is roughly the same as the diffusive flux in a first-order upwind method. This nonlinear definition of the artificial viscosity ensures that the dissipation is as small as possible, while still large enough to prevent oscillations in the temperature field. In particular, the global approximation property of the method will not be affected, as would be the case with the addition of a simple linear artificial diffusion with a constant value ν_h.

The details concerning how ν_h is determined on cell e, which is denoted $\nu_h|_e$, are as follows. As in [24] and [41] we let

$$\nu_h|_e \;=\; \min\left(\nu_h^{\max}|_e,\;\; \nu_h^{E}|_e\right) \tag{36}$$

In Eq. (36) the maximum viscosity $\nu_h^{\max}|_e$ is defined by

$$\nu_h^{\max}|_e \;=\; \beta\; h_e\, \|\mathbf{u}\|_{\infty,e}\,, \tag{37}$$

where the parameter $\beta = 0.078$ is the (current) default value in ASPECT. This parameter controls the maximum dissipation of the entropy viscosity, which is the part that only scales with the cell diameter h_e and the maximum velocity $\|\mathbf{u}\|_{\infty,e}$ in cell e, but does not depend on the solution field itself or its residual.

The entropy viscosity $\nu_h^{E}|_e$ in Eq. (36) is defined by

$$\nu_h^{E}|_e \;=\; c_R\, \frac{h_e^2\;\|r_E(T)\|_{\infty,e}}{\|E(T) - E_{\text{avg}}\|_{\infty,\Omega}}\,, \tag{38}$$

where $c_R = 0.33$ is the (current) default value in ASPECT. This parameter controls the part of the entropy viscosity that depends on the solution field itself and its residual in addition to the cell diameter and the maximum velocity in the cell. See the ASPECT manual [5] for additional information.

Now, if we let

$$T_m \;=\; \frac{1}{2}\,(T_{\min} + T_{\max})\,,$$

then the function $E(T)$ in (38) is defined by

$$E(T) \;=\; \frac{1}{2}\,(T - T_m)^2\,.$$

The entropy viscosity in (38) is scaled globally by the term

$$\|E(T) - E_{\text{avg}}\|_{\infty,\Omega}\,,$$

which is the maximum deviation of $E(T)$ from its spatial average,

$$E_{\text{avg}} \;=\; \frac{1}{|\Omega|}\int_\Omega E(T)\,.$$

Also, the residual $r_E(T)$ in (38) is defined by

$$r_E(T) \;=\; \frac{\partial\, E(T)}{\partial t} \;+\; (T - T_m)\,\left(\mathbf{u}\cdot\nabla T \;-\; \kappa\,\nabla^2 T\right).$$

This residual is zero if applied to the true solution T of the temperature Eq. (9), leading to no artificial diffusion. However, it is non-zero when applied to the numerical approximation of the true solution T and will be large in areas where the numerical approximation is poor, such as close to strong gradients.

[2] There are only three differences in our notation from that in [41]. First, in [41] the authors use the letter 'K' to denote a specific cell rather than the letter 'e' as we do here. In addition, we use c_R and β instead of α_E and $\alpha_{\max}$, respectively, which are used in [41] but are now also denoted c_R and β, in the ASPECT manual [5], which is the definitive source for information concerning these parameters.

There is a detailed explanation of how the default values of the parameters β and c_R were chosen in the section entitled "Numerical experiments to determine optimal parameters" of the reference documentation for deal.II [40]. The exact value of these parameters may have been modified since this documentation was written. However, the manner in which they were chosen is similar to the procedure discussed in the above reference. As of this writing the values $\beta = 0.078$ and $c_R = 0.33$ are the default values in ASPECT. More importantly, these are the values we used in all of the computations of the model problem defined in Section 2 that are shown in Section 4.5. These are the only computations in this article that involve approximating solutions of the temperature Eq. (35).

In addition, we have studied the effect the entropy-viscosity algorithm has on the computed solution as a function of the local Péclet number in a problem that involves a rising square, which is about one-fifth the size of a larger two dimensional square in which the smaller square is hotter than the surrounding fluid [27]. In this work we demonstrated that on a 100×100 grid the local Péclet number had to be $\text{Pe}_e > 10^2$ in order for the approximate solution of the temperature equation *without* entropy viscosity to cause oscillations in the flow field, while the approximate solution of the temperature equation with entropy viscosity did not allow these oscillations. Furthermore, for computations with $\text{Pe}_e \le 10^2$ the approximate solution of the temperature equation with and without entropy viscosity were visually identical, suggesting that the entropy viscosity was zero or near zero in most if not all of the cells. In summary, we confirmed that the additional diffusion added by the entropy-viscosity algorithm for approximating solutions of the temperature equation is sufficiently small that it does not adversely affect our computed solutions until the local Péclet number was $\text{Pe}_e > 10^2$.

We compared these computational results with results obtained with the Bound Preserving Discontinuous Galerkin (DGBP) advection method [28] mentioned earlier. See Section 3.5 below for a brief description of the DGBP method and Section 4.3 for a comparison of DGBP with VOF on the "sinking ball" test problem.

In all of our computations in Section 4.5 of the model problem defined in Section 2 above, the local Péclet number is $\text{Pe}_e < 10$ for the entire time of the computation. Furthermore, in computations of this same model problem (but with different initial conditions) with all four of the advection methods that are implemented in ASPECT, including the VOF method, at the end time the temperature fields are *visually* indistinguishable for the VOF, DGBP, and Particle methods. In this work the computational results for the compositional field C were also nearly visually identical, modulo small numerical artifacts associated with the DGBP and Particle methods, for all values of B that did not yield unstable results [64]. (See Section 3.5 below for a brief description of the other three advection methods in ASPECT.)

Now let

$$(\psi\,,\phi)_\Omega = \int_\Omega \phi(x,y)\,\psi(x,y)\,dx\,dy \tag{39}$$

be the inner product of two scalar functions ϕ and ψ on the domain Ω and let $\Gamma_D = \{(x,y) : y = 0\}$ denote the bottom boundary of Ω. Multiplying Eq. (35) by the test function $\psi(x,y)$ and integrating over Ω we obtain the weak form of the spatial discretization of (35)

$$\left(\frac{\partial T}{\partial t},\,\psi\right)_\Omega + (\mathbf{u}\cdot\nabla T,\;\psi)_\Omega = -(\nabla T,\,\nabla\psi)_\Omega - ((\kappa + \nu_h(T))\,\nabla T,\,\nabla\psi)_\Omega + (\frac{\partial T}{\partial \mathbf{n}},\,\psi)_{\Gamma_D} \tag{40}$$

where $\nu_h(T)$ is the entropy viscosity function defined above.

We use the fully implicit adaptive Backward Differentiation Formula of order 2 (BDF2) [29,80] to discretize the weak form of the temperature equation with entropy-viscosity in time. Thus, the full discretization of the temperature equation is

$$\begin{aligned}\frac{1}{\Delta t^{k+1}}\Bigg(&\frac{2\Delta t^{k+1}+\Delta t^k}{\Delta t^{k+1}+\Delta t^k}T^{k+1}-\frac{\Delta t^{k+1}+\Delta t^k}{\Delta t^k}T^k\\&+\frac{(\Delta t^{k+1})^2}{\Delta t^k(\Delta t^{k+1}+\Delta t^k)}T^{k-1},\psi\Bigg)_\Omega\\=&-\left(\mathbf{u}^k\cdot\nabla T^{k+1},\psi\right)_\Omega-\left(\nabla T^{k+1},\nabla\psi\right)_\Omega\\&-\left(\left(\kappa+\nu_h^k(T)\right)\nabla T^{k+1},\nabla\psi\right)_\Omega+\left(\frac{\partial T^{k+1}}{\partial\mathbf{n}},\psi\right)_{\Gamma_D}.\end{aligned}\tag{41}$$

3.4. The Volume-of-Fluid interface tracking method

The Volume-of-Fluid (VOF) method is an interface tracking method in which, at each time step, there are two distinct steps. In the first step the interface between two fluids or compositions is explicitly reconstructed with an *interface reconstruction method* in every cell that contains a portion of the interface. For example, in our computations of the model problem described in Section 2 above, the compositional variable C will have a value of $C = 1$ in cells completely occupied by the fluid with density $\rho = \rho_0 + \Delta\rho$ and a value of $C = 0$ in cells completely occupied by the fluid with density $\rho = \rho_0$. Thus, cells in which $0 < C < 1$ contain a portion of the interface. Given the explicit (but approximate) reconstructed interface in each cell with $0 < C < 1$ at the current time step one then uses this information to advance the interface in time with an *advection method*. In this sense the VOF method approximates the compositional interface on a subgrid scale.

3.4.1. Background

There are a wide variety of possible VOF interface reconstruction and advection algorithms; e.g., see [57,65,68] and the references therein. The VOF method was first developed at the U.S. National Labs in the 1970s [53] and have continued to be used and developed by researchers at the National Labs [34,52,76,77] as well as around the world.

VOF methods can and have been used effectively to model a wide variety of moving interface problems, including interfaces in compressible flow with shock waves [32], interfaces with shock waves in materials in the limit of no strength effects [49,50], jetting in meteorite impacts [63], *nonconservative* interface motion such as photolithography [30,31], the transition from deflagration to detonation [56] and more than two materials; i.e., more than one interface in a cell [1,33].

An advantage that VOF methods have over other interface tracking methods is that they can readily (or naturally) be designed to approximate solutions of a conservation equation such as Eq. (25) for the composition C. Thus, materials that should be conserved as they move with the flow are (*theoretically*) conserved. However, in practice, i.e., when the algorithm is implemented on a computer, some VOF advection algorithms - including the one we use here - will only conserve the volume of the composition or fluid that is being tracked up to some numerical error that typically depends on the grid size h. This will depend on the design of the VOF advection algorithm. There has been considerable research into how to design VOF advection algorithms that conserve volume to machine zero; e.g., see [4,68,81]. (In this article, machine zero, also known as machine precision, will be denoted by ϵ_{mach} and we will assume $\epsilon_{\text{mach}} = O(10^{-16})$).

In Section 4.3 below we examine the degree to which the VOF advection algorithm we use in this work conserves volume and compare the results with a Bound Preserving Discontinuous Galerkin (DGBP) advection method [28], which is also implemented in ASPECT. See Section 3.5 below for a brief description of the DGBP method and Section 4.3 for the computational results.

3.4.2. Overview

In this article we use a two-dimensional VOF algorithm to discretize the conservation Eq. (25), which - when describing the VOF method - we will usually write in the following form,[3]

$$\frac{\partial f}{\partial t}+\nabla\cdot\mathbf{F}(f)=0.\tag{42}$$

Here $\mathbf{u} = (u, v)$ is the velocity field, f is the *volume fraction* of one of the compositional fields, say $C = 1$, the field with density $\rho_0 + \Delta\rho$, which we will refer to as 'Composition 1', or C_1 for short, and

$$\mathbf{F}(f)=(F(f),G(f))=(uf,vf)=\mathbf{u}f,\tag{43}$$

is the volume fraction flux associated with C_1.[4] Since $\nabla\cdot\mathbf{u} = 0$ one can rewrite Eq. (42) as a pure advection equation for f,

$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}=0.\tag{44}$$

This equation is equivalent to Eq. (23).

From a mathematical point of view the variable $f(x, y, t)$ in the conservation Eq. (42) with fluxes (43) may be regarded as the characteristic function (sometimes denoted $\chi(x, y, t)$) associated with the composition C_1. In other words,

$$f(x,y)=\begin{cases}f(x,y)=1 & \text{if } (x,y) \text{ is occupied by Composition 1},\\ f(x,y)=0 & \text{if } (x,y) \text{ is not occupied by Composition 1}.\end{cases}\tag{45}$$

This implies $1 - f(x, y)$ is the characteristic function associated with C_2, the composition with density $\rho = \rho_0$.

Our use of the variable f to represent the quantity that is advected in a VOF method is historical. In particular, in the presentation of our VOF method the variable C and f can be used interchangeably. However, in Section 3.5 below we will describe other methods in ASPECT for advecting the quantity C and in Section 4.3 we will use one of these other advection methods to compute a test problem in order to compare the results of an entirely different advection method with the results we obtain with the VOF method presented here. In these sections, as well as in Section 4.5 where we show the results of our computations of the model problem presented in Section 2, our use of the variable C is typical of the notation researchers use in the computational geodynamics literature.

In our VOF implementation in ASPECT we use the 'Efficient Least Squares VOF Interface Reconstruction Algorithm' (ELVIRA), which is described in detail in [57] and is based on the ideas in [58] and [54]. The ELVIRA interface reconstruction algorithm reconstructs lines on a uniform grid with square cells *exactly*; i.e., to machine zero. We explain this in more detail in Section 3.4.3 and Fig. 4, and also demonstrate it with a computational example in Section 4.2.1 below. Since the ELVIRA algorithm reconstructs lines in square cells exactly it is natural to assume that the algorithm is second-order accurate on a uniform grid with identical square cells. This has been proven to be true [59,61]. We use a second-order accurate dimensionally split advection method, which is described in Section 2.2.1 of [62], to update the values of the volume fractions in time. For simplicity of exposition we will

[3] In Eq. (42) and in much of what follows, depending on the context, we will use the symbol f, $f_{i,j}$, or f_e, for the fraction of the fluid or composition denoted by $C = 1$ that occupies the cell $\Omega_{i,j}$ or cell Ω_e.

[4] Throughout this section and beyond we will use the terms "volume" and "volume fraction" of C_1, etc., although it is to be understood that in two dimensions the quantity in question is an area.

assume the finite element grid consists entirely of square cells Ω_e, of side h, indexed by the variable e, and aligned parallel to the x and y axes. This is the case for all of the computations we present in this article.

However, we emphasize that VOF algorithms have been developed and applied to problems on a wide variety of unstructured grids in three dimensions [39], including tetrahedral [82], hexahedral [83] and general convex grids [44], as well as having been developed and applied to great many applications on non-rectangular grids in two dimensions. However, the interface reconstruction algorithm and advection algorithm on irregular grids in two and three dimensions must be modified or completely redesigned for such grids. This has been the focus of much research over the past 20 or 30 years. For example, one approach is to minimize the difference, say in the least-squares sense (i.e., the discrete two norm), between the given volume fractions and the volume fractions due to a linear interface in two dimensions or planar interface in three dimensions in a neighborhood of the cell of interest with a minimization algorithm such as Brent's method; e.g., see [55,57,58,65] for examples and comments on this approach and [81] for alternate approaches.

The discretization of Eq. (42) proceeds as follows. Let Ω_e denote an arbitrary cell in the computational domain Ω and let f_e^k denote the discretized volume fraction in Ω_e at time t^k. The variable f_e^k is a scalar that satisfies $0 \le f_e^k \le 1$ such that

$$f_e^k \approx \frac{1}{h^2} \int_{\Omega_e} f(x, y, t^k)\, dx\, dy\,. \tag{46}$$

Thus, the discretized volume, V_e^k, of C_1 in Ω_e at time t^k is

$$V_e^k = \int_{\Omega_e} f_e^k\, dx\, dy = h^2\, f_e^k\,. \tag{47}$$

Note that for an incompressible velocity field $\mathbf{u} = (u, v)$ we have $\nabla \cdot \mathbf{u} = 0$ and hence, for the true solution the volume of 'parcels' or regions of C_1 are constant as they evolve in time.

Cells that contain a portion of the interface (i.e., $0 < f_e^k < 1$) will have a value of

$$\rho = f_e^k(\rho_0 + \Delta\rho) + \left(1 - f_e^k\right)\rho_0\,. \tag{48}$$

In other words a weighted average of the two compositions C_1 and C_2 with weights f_e^k and $(1 - f_e^k)$. In most of our work with the VOF method to track the interface between two compositions we use the weighted average in Eq. (48), with C_1 replacing $(\rho_0 + \Delta\rho)$ and C_2 replacing ρ_0. This is common practice among researchers who use VOF methods; e.g., see Eq. (2) of [81].

In this article we restrict ourselves to modeling the interface between two compositions. However, there is currently a great deal of research into modeling two or more interfaces in one cell with a VOF method; e.g., see [36] and the references there.

In its simplest form our implementation of the VOF algorithm in ASPECT proceeds as follows. Given the values f_e^k at time t^k and the velocity field at time t^k we do the following to obtain the volume fractions f_e^{k+1} at time t^{k+1}.

For convenience and clarity of exposition, in the remainder of this section we will *usually* use the index notation (i, j), as shown in Figs. 2–4. Thus, we have nine cells with centers $(x_{i'}, y_{j'})$ for $i' = i-1, i, i+1$ and $j' = j-1, j, j+1$ with edges $x = x_{i\pm\frac{1}{2}} = x_i \pm \frac{h}{2}$ and $x = x_{i\pm\frac{3}{2}} = x_i \pm \frac{3h}{2}$ and similarly for y as shown in Fig. 2. In the ELVIRA interface reconstruction algorithm we use the information in the 3×3 block of cells $\Omega_{i'j'}$ immediately adjacent to the cell $\Omega_e \equiv \Omega_{ij}$ in which we wish to reconstruct the interface. Note that we use both Ω_e and Ω_{ij} to denote the center cell of the 3×3 block of cells. The reason for this is that in peer reviewed and technical (i.e., the manual, etc.) literature that describes ASPECT the notation Ω_e is often used to denote cells, while the notation Ω_{ij} is com-

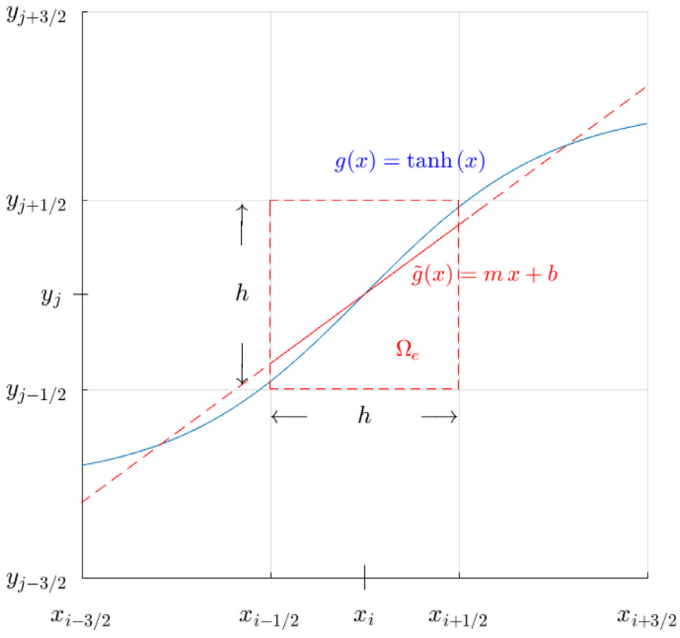


Fig. 2. In our implementation of the VOF interface reconstruction algorithm the true interface, which in this example is $g(x) = \tanh(x)$, is approximated as a line segment $\tilde{g}_e(x) = m_e\,x + b_e$ in each cell Ω_e that has a volume fraction f_e with $0 < f_e < 1$. The approximate interface in Ω_e is depicted as the solid red line segment in the center cell Ω_e. In this example, as with all VOF methods, the volume $h^2 f_e^{true}$ beneath the true interface in Ω_e is *exactly equal* to the volume $h^2 f_e$ beneath the approximate interface $\tilde{g}$ in Ω_e; i.e., $f_e^{true} = f_e$. Note that, for convenience, we have used the notation (x_i, y_j) to denote the center of the cell Ω_e, $[x_{i-1/2}, x_{i+1/2}] \times [y_{i-1/2}, y_{i+1/2}]$ to denote the cell Ω_e, etc. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

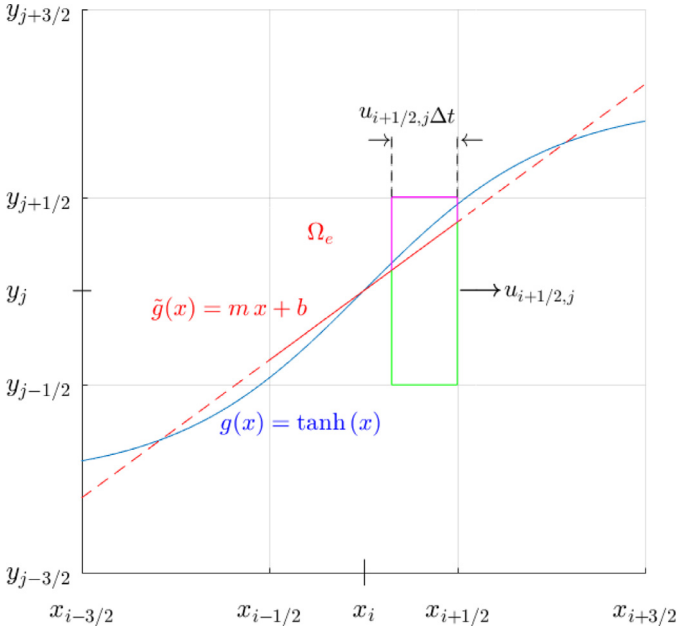


Fig. 3. The volume $V^k_{i+1/2,j,1} = \delta x\, \tilde{g}(\delta x)$ where $\delta x = u^k_{i+i/2,j}\,\Delta t^k$ of C_1 in the quadrilateral outlined in green on three sides and by a portion of the solid red line on top is the flux of C_1 that will cross the right-hand edge of Ω_e during the time step from time t^k to t^{k+1}. Here $\Delta t^k = t^{k+1} - t^k$ and we have dropped the superscript k from $u^k_{i+i/2,j}$ and Δt^k in the diagram for clarity. The solid red line in Ω_e is the reconstructed interface $\tilde{g}(x)$ that approximates the true interface $g(x) = \tanh(x)$ in Ω_e at time t^k as shown in Fig. 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

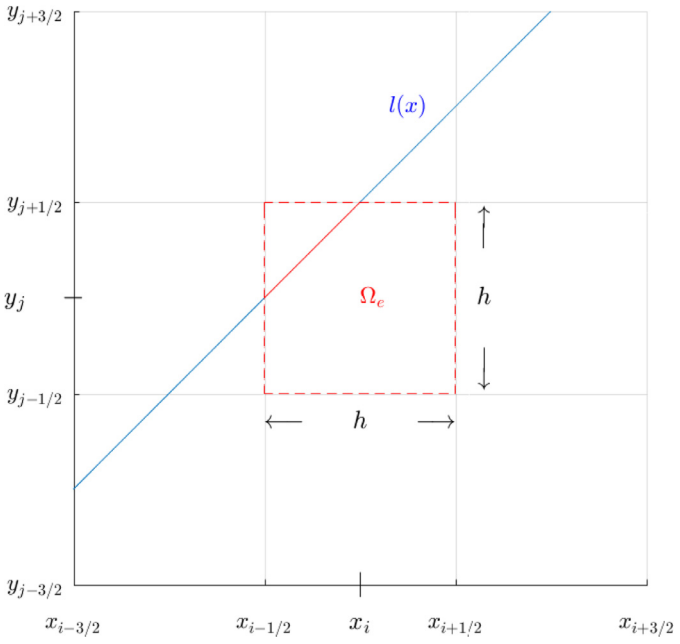


Fig. 4. In this example the true interface is the line $l(x) = mx + b$ Note that that the volumes V_{i-1} and V_i under the line in the first two columns $i-1$ and i are exactly equal to the volumes due to the column sums $\tilde{V}_{i-1} = h^2 S_{i-1}$ and $\tilde{V}_i = h^2 S_i$ in the first and second columns of the 3×3 block of cells B_{ij} centered on the center cell $\Omega_e (= \Omega_{ij})$. In this case the slope $\tilde{m} = S_i - S_{i-1}$ is *exactly* equal to the slope m of the interface as shown in (56). It is always the case that if the true interface is a line, then one of the four standard rotations of B_{ij} by a multiple of 90 degrees about its center will orient the block so at least one of the divided differences of the column sums in (59) or (60) is exact and hence, one of the linear approximations to the interface in the center cell Ω_e defined in (62) will always equal the interface in that cell, *exactly*, $\tilde{g}_{ij}(x) = m_{ij} x + b_{ij} = mx + b = l(x)$. In other words, the piecewise linear VOF approximation to $l(x)$ will *always* reconstruct the linear interface exactly.

monly used in VOF literature, especially when coupled to a finite volume or finite difference method.

1. The interface reconstruction step: Given a cell Ω_{ij} that contains a portion of the interface, so $0 < f_e^k < 1$ where f_e^k is the volume fraction in Ω_e at time t^k, we use the volume fractions $f_{e'}^k$ in the 3×3 block of cells $\Omega_{e'}$ centered on the cell Ω_e to *reconstruct* the interface in Ω_e. The reconstructed interface will be a piecewise linear approximation to the true interface as shown in Fig. 2 that preserves the given volume $h^2 f_e^k$ of C_1 in Ω_e. We give a brief description of how we determine the linear approximation $\tilde{g}_e(x) = m_e x + b_e$, to the true interface in cells Ω_e for which $0 < f_e^k < 1$ in Section 3.4.3 below.
2. Computation of the fluxes: In the computations presented in this article we use a second-order accurate dimensionally split (also known as "operator split") advection algorithm in order to advance the interface in time. However, for clarity and simplicity of exposition, in this section we will only describe *the simplest possible version* of a dimensionally split advection algorithm for updating the location of the interface. In Section 3.4.5 below we will describe an important modification to the dimensionally split advection algorithm described here. *We use this modified dimensionally split advection algorithm* described in Section 3.4.5 in all of the computations shown in this article.
   Given the reconstructed interface $\tilde{g}_e = \tilde{g}_{ij}(x)$ in
   $$\Omega_e \equiv \Omega_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{i+1/2}]$$
   as shown in Fig. 3 and the velocity $u^k_{i\pm 1/2,j}$ normal to the right and left edges of Ω_{ij} at time t^k, we wish to determine the volumes $V^k_{i\pm 1/2,j}$ of C_1 that cross the right and left edges of Ω_e in the time interval $[t^k, t^{k+1}]$. These volumes are determined *geometrically*. Since we are using a dimensionally split advection method, the total volume of both compositions C_1 and C_2 that crosses each cell edge will be that of a rectangle. This is illustrated in Fig. 3 for the right edge of the cell Ω_e where we have assumed that $u^k_{i\pm 1/2,j} > 0$. The rectangle is shown in green and pink and the volume of the rectangle is $u_{i+i/2,j}\, \Delta t\, h$. We then determine the volume $V^k_{i+1/2,j}$ of C_1 that crosses the right-hand edge of Ω_{ij} in the time interval $[t^k, t^{k+1}]$ is outlined in green on three sides and by a portion of the solid red line on top in Fig. 3.
3. The volume fraction update: Now we describe the *unmodified* dimensionally split VOF advection method, which we are referring to as here as the "Volume Fraction Update". One may also think of this as a "Volume Update"; i.e., the update $V^k_{e,1} \to V^{k+1}_{e,1}$ of the volume of C_1 in cell e, which is how we have chosen to present the algorithm here. This is the simplest possible dimensionally split VOF advection method. However, a simple modification, which we will describe in Section 3.4.5 below, greatly improves the volume conservation of the method. The update proceeds in two steps.

Step I Given the volume $V^k_{ij,1} = h^2 f^k_{ij}$ of C_1 in Ω_{ij} at time t^k and the volumes $V^k_{i\pm 1/2,j,1}$ of C_1 that cross the left and right-hand edges, respectively, of Ω_{ij} in the time interval $[t^k, t^{k+1}]$ we use the following equation to determine an *intermediate* volume $\tilde{V}^k_{ij,1}$ of C_1 in Ω_{ij} for the first step of the two step dimensionally split algorithm,

$$\tilde{V}^k_{ij,1} = V^k_{ij,1} + V^k_{i-1/2,j,1} - V^k_{i+1/2,j,1}\,. \tag{49}$$

We do this in every cell Ω_{ij} before advancing to Step II below. In Fig. 3 we illustrate how we determine the volume $V^k_{i+1/2,j,1}$ that crosses the right edge of Ω_{ij}.

Step II Now, given the nine intermediate volume fractions

$$\tilde{f}^k_{i'j'} = \frac{1}{h^2}\, \tilde{V}^k_{i'j',1}$$

in Ω_{ij} and the 3×3 block of cells $\Omega_{i'j'}$ surrounding Ω_{ij}, together with all of the intermediate volume fractions in the 3×3 block of cells surrounding each of the cells $\Omega_{i'j'}$, we reconstruct an intermediate interface $\hat{g}_{i'j'}(x)$ in each cell $\Omega_{i'j'}$. We then use this intermediate interface to geometrically determine the volumes $\tilde{V}^k_{i,j\pm 1/2,1}$ of C_1 that cross the *top* and *bottom* edges of Ω_{ij} in the time interval $[t^k, t^{k+1}]$ in the same manner as illustrated in Fig. 3, but this time in the y-direction. Now the volume $V^{k+1}_{ij,1}$ of C_1 in Ω_{ij} at the new time t^{k+1} is,

$$V^{k+1}_{ij,1} = \tilde{V}^k_{ij,1} + \tilde{V}^k_{i,j-1/2,1} - \tilde{V}^k_{i,j+1/2,1} \tag{50}$$

and the new volume fraction in Ω_{ij} is

$$f^{k+1}_{ij} = \frac{1}{h^2}\, V^{k+1}_{ij,1}\,. \tag{51}$$

This is the simplest of all dimensionally split volume fraction advection algorithms. It can be made to be second-order accurate by alternating the direction of the first volume update at each time step, a procedure that is known a 'Strang splitting" in the numerical methods community [72]. There are also *unsplit* VOF advection algorithms; e.g., see [57,62,65].

3.4.3. The ELVIRA interface reconstruction algorithm

Here we describe the ELVIRA interface reconstruction algorithm [57] in more detail. In this example we present the simplest possible case; namely, when the true interface is a line that passes through the center cell of the 3×3 block B_{ij} of cells $\Omega_{i'j'}$ centered on the cell Ω_{ij} as shown in Fig. 4. The following description is intended to be easy to understand. However, the reader should be aware that there are many VOF interface reconstruction algorithms in both two [76] and three dimensions [77] and on every conceivable grid; e.g., [39]

In the ELVIRA algorithm the approximate interface will be a *piecewise linear* approximation $\tilde{g}_{ij}(x) = m_{ij}\,x + b_{ij}$ to the true interface in Ω_{ij} as depicted in Fig. 2. Furthermore *the approximate interface is subject to the constraint that* the volume fraction in the center cell due to the true interface $g(x)$ and the approximate interface $\tilde{g}_{ij}$ are equal; i.e., $f_{ij}^{true} = f_{ij}$.

Consider the example shown in Fig. 4. In this example the true interface is a line $l(x) = m\,x + b$. Assume we are given the exact volume fractions $f_{i'j'}$ associated with the line $l(x)$, which is the true interface, in each cell $\Omega_{i'j'}$ of the 3×3 block. Then in this example the first two column sums

$$S_{i-1} \stackrel{\text{def}}{=} \sum_{j'=j-1}^{j+1} f_{i-1,j'} \qquad \text{and} \qquad S_i \stackrel{\text{def}}{=} \sum_{j'=j-1}^{j+1} f_{i,j'} \tag{52}$$

are *exact* in the sense that

$$S_i = \frac{1}{h^2} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(l(x) - y_{j-3/2} \right) dx \tag{53}$$

and similarly for S_{i-1}, but *not* for S_{i+1}, since the line leaves the 3×3 block B_{ij} of cells $\Omega_{i'j'}$ centered on the cell Ω_{ij} through the top edge, thereby rendering the sum S_{i+1} inexact in the sense that

$$S_{i+1} \neq \frac{1}{h^2} \int_{x_{i+1/2}}^{x_{i+3/2}} \left(l(x) - y_{j-3/2} \right). \tag{54}$$

Thus, using (53) we find the difference in the column sums S_i and S_{i-1} is

$$\begin{aligned} h^2 \left(S_i - S_{i-1} \right) &= \int_{x_{i-1/2}}^{x_{i+1/2}} (m\,x - b) - y_{j-3/2}\, dx \\ &\quad - \int_{x_{i-3/2}}^{x_{i-1/2}} (m\,x - b) - y_{j-3/2}\, dx \\ &= \int_{x_{i-1/2}}^{x_{i+1/2}} m\,x\,dx - \int_{x_{i-3/2}}^{x_{i-1/2}} m\,x\,dx \\ &= m\,\frac{x^2}{2}\Big|_{x_{i-1/2}}^{x_{i+1/2}} - m\,\frac{x^2}{2}\Big|_{x_{i-3/2}}^{x_{i-1/2}} \\ &= \frac{m}{2}\left[\left(x_{i+1/2} \right)^2 - \left(x_{i-3/2} \right)^2 \right] - \frac{m}{2}\left[\left(x_{i+1/2} \right)^2 - \left(x_{i-1/2} \right)^2 \right] \\ &= \frac{m}{2}\, h \left(x_{i+1/2} - x_{i-3/2} \right) \\ &= m\, h^2 . \end{aligned} \tag{55}$$

and hence,

$$m = S_i - S_{i-1} . \tag{56}$$

Thus, we have recovered the *exact* slope m of the true interface $l(x)$ in the center cell simply by differencing the correct pair of column sums of volume fractions. Note that this would not have been true if we had used $S_{i+1} - S_i$ instead, since the expression on the RHS of (54) is not identically equal to S_{i+1}.

A little thought will show that the constraint

$$f_{ij} = f_{ij}^{true} \tag{57}$$

determines b uniquely, thus determining the linear approximation

$$g_{ij}(x) = m\,x + b \tag{58}$$

which is exactly equal to the true interface $l(x)$. In actual fact one needs to know whether the region containing the composition C_1 is above, below, or to the left or right of C_2. However, there are a variety of algorithms for doing this; e.g., see [12,57,59–61]. This always works on a uniform grid of square cells with sides of side h.

However, there are a few caveats: There are three ways to difference the column sums,

$$\begin{aligned} m^{x,l} &= (S_i - S_{i-1}) \\ m^{x,c} &= \frac{(S_{i+1} - S_{i-1})}{2} \\ m^{x,r} &= (S_{i+1} - S_i) \end{aligned} \tag{59}$$

and three ways to difference the row sums

$$\begin{aligned} m_l^y &= \left(R_j - R_{j-1} \right) \\ m_c^y &= \frac{\left(R_{j+1} - R_{j-1} \right)}{2} \\ m_l^y &= \left(R_{j+1} - R_j \right) \end{aligned} \tag{60}$$

where the *row sums* are defined by

$$R_{j-1} \equiv \sum_{i'=i-1}^{i+1} f_{i',j-1}\,, \quad R_j \equiv \sum_{i'=i-1}^{i+1} f_{i',j} \quad \text{and} \quad R_{j+1} \equiv \sum_{i'=i-1}^{i+1} f_{i',j+1} \tag{61}$$

In order to determine the best linear approximation to the true interface we compare the volume fractions $f_{i'j'}^{x,l}$, $f_{i'j'}^{x,c}$, $f_{i'j'}^{x,r}$, … $f_{i'j'}^{y,r}$ due to each of the six lines

$$\begin{aligned} g_l^x &= m_l^x\,x + b_l^x \qquad g_l^y = m_l^y\,x + b_l^y \\ g_c^x &= m_c^x\,x + b_c^x \qquad g_c^y = m_c^y\,x + b_c^y \\ g_r^y &= m_r^x\,x + b_r^x \qquad g_r^y = m_r^y\,x + b_r^y \end{aligned} \tag{62}$$

we obtain from each of the six slopes in (59) and (60) in the 3×3 block B_{ij} centered on the cell of interest Ω_{ij} and use the line that minimizes the difference between the given volume fractions and the volume fractions due to the lines in (62). We now explain this procedure in a bit more detail.

3.4.4. Approximating an unknown interface from the volume fractions

Suppose $g(x)$ is an unknown interface that passes through the center cell Ω_{ij} of a 3×3 block of cells B_{ij} containing nine square cells $\Omega_{i'j'}$, each of side h, centered on Ω_{ij}. Furthermore, assume the only information we have are the nine *exact* volume fractions $f_{i'j'}$ in the cells $\Omega_{i'j'}$ due to $g(x)$. For example, in Fig. 2 the 'unknown' interface is $g(x) = \tanh(x)$, which is the blue curve, and the volume fractions are nonzero only in cells that either contain the curve or are below it. We want to find a line segment $\tilde{g}_{ij}(x) = m_{ij}\,x + b_{ij}$ that is a second-order accurate approximation to $g(x)$, in the following sense,

$$\max \left| g(x) - \tilde{g}_{ij}(x) \right| \leq \tilde{C}\,h^2 \qquad \text{for all } x \in [x_{i-1/2},\, x_{i+1/2}]\,, \tag{63}$$

where $\tilde{C}$ is a constant that is independent of h.

First we define a way to measure the error $E(\tilde{m})$ between the volume fractions $f_{i'j'}$ we are given that are due to the unknown interface and the approximate volume fractions $\tilde{f}_{i'j'}$ due to a line segment $\tilde{g}(x) = \tilde{m}\,x + \tilde{b}$ that passes through the center cell Ω_{ij} and the 3×3 block B_{ij} centered on Ω_{ij},

$$E(\tilde{m}) = \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} \left(f_{i'j'} - \tilde{f}_{i'j'} \right)^2 . \tag{64}$$

Note that this is the square of the *two norm* on vector spaces R^n from linear algebra, where in our case $n = 9$, [73].

Now take the volume fractions we are given, namely $f_{i'j'}$, and form all six of the slopes in (59) and (60) and the six candidate lines in (62) from these slopes. Remember that the 'y intercept' b for each of the lines in (62) is determined by the constraint $f_{ij}^{true} = f_{ij}$. Each of the six lines produces nine volume fractions in the 3×3 block B_{ij}. For example, given the slope $m^{x,c}$ defined in (59) we obtain the line $g_c^x = m_c^x\, x + b_c^x$ defined in (62), which in turn gives us nine volume fractions $f_{i'j'}^{x,c}$ for $i' = i-1,\ i,\ i+1$ and $j' = j-1,\ j,\ j+1$. Now compute $E(m_c^x)$ and repeat this procedure for each of the other lines in (62) with slopes computed as in (59) and (60). Finally, take the line from (62) that minimizes the error defined in (64); i.e., pick the slope from (59) and (60), call it $\tilde{m}$, that satisfies

$$E(\tilde{m}) \;=\; \min\left\{E(m_l^x),\, E(m_c^x), \ldots,\, E(m_r^y)\right\}. \tag{65}$$

The line

$$\tilde{g} = \tilde{m}\, x + \tilde{b} \tag{66}$$

is the linear approximation to the true interface $g(x)$ in Ω_{ij} that we use in the VOF algorithm in this article. In [59] and [61] it is proven that this algorithm produces a second-order accurate approximation to the true interface in the sense of (63) provided that

$$h \;\le\; \frac{2}{33\,\sigma_{max}} \tag{67}$$

where $\sigma_{\max}$ denotes the maximum curvature of the true interface, h is the grid size of a square grid, and the volume fractions due to the true interface are exact.[5]

As mentioned in Section 3.4.2 above, our current implementation of the VOF advection method in ASPECT is a dimensionally split advection method. However, as we emphasized in Item 3 of Section 3.4.2 there is an important modification we make to the algorithm described there that is necessary in order to more nearly satisfy the conservation of the (total) volumes of C_1 and C_2. We now describe this modification. (A different, yet equivalent description is given in Section 2.2.1 of [62], which the interested reader may wish to consult for a description of this algorithm from a slightly different point of view.)

To begin, recall that the advection of the volume fraction function f is governed by Eq. (44) which, in order to improve the ease of exposition, we rewrite here,

$$\frac{\partial f}{\partial t} + \mathbf{u}\cdot\nabla f = 0\,. \tag{68}$$

Using Eq. (20); i.e., the divergence free constraint on the velocity $\mathbf{u} = (u,\, v)$,

$$\nabla\cdot\mathbf{u} \;=\; 0\,, \tag{69}$$

we obtain a modified, but equivalent, form of (68),

$$\overbrace{\frac{\partial f}{\partial t} + \nabla\cdot(\mathbf{u}f)}^{\text{Conservation}} - \overbrace{f(\nabla\cdot\mathbf{u})}^{\text{Correction}} = 0\,. \tag{70}$$

[5] This result has only been proven for the *stationary* interface reconstruction problem. In other words, given a smooth (e.g., two times continuously differentiable) interface and the true volume fractions due to this interface on a square grid of side h, then the approximate interface is second-order accurate in the max norm; i.e., the bound in (63) holds. This bound is much stronger than a bound in the L^1 or L^2 norms that we use to examine convergence rates in Section 4. The second author thinks it should be possible to prove that, given the assumptions just stated above on the initial data and assuming the interface stays smooth as it moves in some flow and that (67) holds at each time step, then numerical approximation will remain second-order accurate in some norm; e.g., L^1 or L^2. Although this is only a conjecture, we use it as a 'rule of thumb'. It appears to work for the results shown in Section 4.3 below.

It will be instructive to write this equation in the following form

$$\frac{\partial f}{\partial t} + \nabla\cdot(\mathbf{u}f) \;=\; f\,\frac{\partial u}{\partial x} + f\,\frac{\partial v}{\partial y}\,. \tag{71}$$

Note that in Eq. (70) the first term is the conservation Eq. (42) for f. The key point is that if (69) is satisfied *exactly*, then the correction term in (70) will be zero and hence, the advection equation in (68) for f is equivalent to the conservation equation for f, which here we write in the following form,

$$\frac{\partial f}{\partial t} + \frac{\partial (uf)}{\partial x} + \frac{\partial (vf)}{\partial y} = 0\,. \tag{72}$$

In the dimensionally split advection algorithm that we described in Step II of the Volume Fraction Update in Section 3.4.2 above we approximated solutions of following two step dimensionally split advection method,

$$\begin{aligned}
\frac{\partial f}{\partial t} + \frac{\partial (u\,f)}{\partial x} &= 0\,,\\
\frac{\partial f}{\partial t} + \frac{\partial (v\,f)}{\partial y} &= 0\,.
\end{aligned} \tag{73}$$

However, unless each component of

$$\nabla\cdot\mathbf{u} \;=\; u_x + v_y\,,$$

equals zero separately (i.e., $u_x = 0$ and $v_y = 0$) numerical solutions of the equations in (73) above will fail to be adequate approximate solutions of the advection equation (68) for f. The reason for this is that in passing from (68) to (70) and thence to (72) we assumed that the velocity $\mathbf{u}$ was divergence free (69).

In short, we need to approximate solutions of the dimensionally split version of the *modified* Eq. (71),

$$\begin{aligned}
\frac{\partial f}{\partial t} + \frac{\partial (u\,f)}{\partial x} &= f\,\frac{\partial u}{\partial x}\,,\\
\frac{\partial f}{\partial t} + \frac{\partial (v\,f)}{\partial y} &= f\,\frac{\partial v}{\partial y}\,.
\end{aligned} \tag{74}$$

In the dimensionally split algorithm that we use in this article we approximate solutions of (74) as follows,

$$\begin{aligned}
\tilde{f}_e &= f_e^k \;-\; \Delta t\,\frac{\partial (u\,f)}{\partial x} \;+\; \Delta t\,\tilde{f}_e\,\frac{\partial u}{\partial x}\,,\\
f_e^{k+1} &= \tilde{f}_e \;-\; \Delta t\,\frac{\partial (v\,\tilde{f})}{\partial x} \;+\; \Delta t\,\tilde{f}_e\,\frac{\partial v}{\partial y}\,.
\end{aligned} \tag{75}$$

Note that we have written these equations in semi-discrete form; i.e., discretized in time but not in space. In (75) $\tilde{f}_e$ is the intermediate volume fraction of C_1 in Ω_e and $(v\,\tilde{f})$ is the intermediate volume fraction flux in the y direction obtained from the intermediate interface $\hat{g}$ that has been reconstructed from the volume fractions $\tilde{f}_{e'}$ surrounding Ω_e as explained in Step II of 3 "The Volume Fraction Update"in Section 3.4.2 above. We have written the equations in (75) in this semidiscrete form in order to emphasize that in the first equation $\tilde{f}_e$ is treated *implicitly* while in the second equation $\tilde{f}_e$ is treated explicitly.

Our complete discretization of (75) is as follows,

$$\begin{aligned}
\tilde{f}_e\,V_e &= f_e^k\,V_e \;-\; \Delta t\;(f_R\,U_R \;-\; f_L\,U_L) \;+\; \Delta t\;\tilde{f}_e\,(U_R \;-\; U_L)\,,\\
f_e^{k+1}\,V_e &= \tilde{f}_e\,V_e \;-\; \Delta t\left(\tilde{f}_T\,U_T \;-\; \tilde{f}_B\,U_B\right) \;+\; \Delta t\;\tilde{f}_e\,(U_T \;-\; U_B)\,,
\end{aligned} \tag{76}$$

where e is an index that ranges over all cells Ω_e, V_e is the volume of Ω_e, $r = L,\, R,\, B,\, T$ denotes the left, right, bottom, and top cell edges, respectively, f_r is the volume *fraction* of C_1 that will be fluxed across the rth edge as described in the caption to Fig. 5, and

$$U_r \;=\; \int_{\partial\Omega_{e,r}} \tilde{\mathbf{u}}_r\cdot\mathbf{n}_r\,ds\,, \tag{77}$$

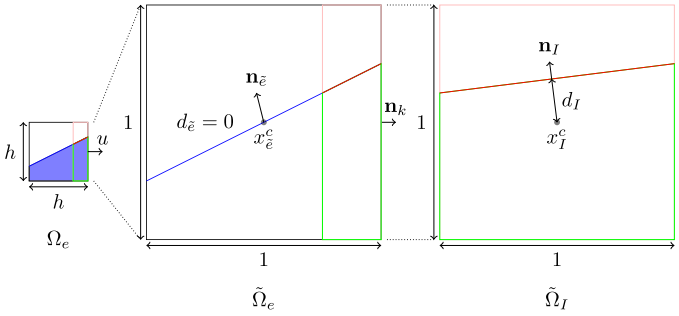


Fig. 5. A diagram of the mapping of the region (in purple) containing the compositional field C_1 in the *real* cell Ω_e to its associated unit cell $\tilde{\Omega}_e$. In this diagram we have assumed that the velocity field $\mathbf{u} = (u, 0)$ points in the x direction only so that the flux of C_1 across the right edge of $\tilde{\Omega}_e$ is a rectangular region. This allows us to compute the *total* volume V_F of C_1 and C_2 that is fluxed across the right edge of $\tilde{\Omega}_e$; namely, the rectangle on the right edge of $\tilde{\Omega}_e$. We then map this rectangle to another unit cell $\tilde{\Omega}_I$ in order to compute the volume *fraction* f_r of the (mapped) rectangle that contains the composition C_1. Since in this article, linear interfaces map to linear interfaces, we can use the unit normal $\mathbf{n}_I$ and distance d_I to calculate the volume fraction f_r of C_1 in this rectangle. Note that, since this diagram has been chosen to correspond exactly to the one in Fig. 3 in which the interface and reconstructed interface both pass through the cell center, we have $d_{\tilde{e}} = 0$. However, in general, $d_{\tilde{e}} \neq 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\tilde{\mathbf{u}}_r$ is a time centered approximation to the velocity $\mathbf{u}$ on the rth edge,

$$\tilde{\mathbf{u}}_r = \frac{\mathbf{u}^{k+1} + \mathbf{u}^k}{2} .$$

Again, note that in the dimensionally split algorithm in (76) the intermediate value $\tilde{f}_e$ in the first equation is determined via an implicit discretization, while $\tilde{f}_e$ is treated explicitly in the second equation.

There are a number of versions of VOF advection methods in the peer reviewed VOF literature that are similar - or in at least one case - identical to our modified algorithm (76). For example, on a grid of square cells of side h our method is identical to the method in Section 2.2.1. of [62]; i.e., Eqs. (22) and (23). Our algorithm is similar to the one presented in Section "*4.2 Eulerian Scheme*" in [68] and attributed to the authors of [65], except in both papers the first step of the algorithm is explicit while the second step is implicit, which is the opposite of our algorithm in (76). We also note that - assuming we understand the authors notations correctly - the monodimensional Eulerian-implicit (EI) scheme in Eq. (18) of [4] is identical to our first (implicit) step in (76) and the first (implicit) step in Eq. (22) of [62].

Finally, since we are using Strang splitting, [72] we evaluate (76) once for each spatial dimension in the problem at each time step, alternating the order of the dimensions in the subsequent time step.

3.4.5. The implementation of the VOF method in ASPECT

We now describe our implementation in ASPECT of the VOF algorithm described above on square, two-dimensional cells Ω_e in physical space (Ω_e is often referred to as the 'real' cell). In a VOF method it is natural to use the method of characteristics to calculate the flux of C_1 through each of the cell edges. This is done by tracing backward in time along a linear approximation to each characteristic that crosses the cell edge in the time interval $[t^k, t^{k+1}]$ in order to identify the *total* volume V_F of both compositions C_1 and C_2 that will cross a given edge in the time interval $[t^k, t^{k+1}]$. We then compute that portion of the volume associated with the composition C_1 that is being tracked; i.e., by computing the volume of C_1 in the total volume V_F. This procedure is depicted in Fig. 3 for a one-dimensional sweep in the x-direction, in which case the linear approximation to the characteristics that cross the right edge of Ω_e are horizontal lines of length $u_{i+i/2,j}\,\Delta t$ that fill out the rectangle (shown in green and pink) of volume $V_F = u_{i+i/2,j}\,\Delta t\, h$. See [14,43] for examples of computing a second-order accurate flux in this manner in a finite volume discretization of (25), rather than a VOF discretization of (25) or, equivalently, (42), as well as higher resolution versions of these algorithms. In our computation of the volume and volume fraction flux we make use of several algorithms that we developed for the interface reconstruction step. We will describe these algorithms is more detail below.

There are a number of approaches one can consider for obtaining the velocities on the rth edge from the approximate FEM solution of the incompressible Stokes equations. Two such approaches are:

1. A point sample of the velocity normal to the rth edge $\partial\Omega_{e,r}$ taken at some point on $\partial\Omega_{e,r}$,
2. The velocity integrated along the rth edge $\partial\Omega_{e,r}$ of Ω_e,

$$\int_{\partial\Omega_{e,r}} \mathbf{u} \cdot \mathbf{n}_r \, ds \, , \tag{78}$$

where $\mathbf{n}_r$ denotes the (outward facing) unit normal to $\partial\Omega_{e,r}$ and $k = 1, 2, 3,$ or 4.

For a finite volume method (1) and (2) are both reasonable approximations to the edge velocities. However, the latter method (78) is a closer analogue to the type of procedure one would typically choose for a finite element method.

We now describe our implementation of the computation of the volume flux of C_1 into or out of a square cell Ω_e of side h. (When we employ AMR, h denotes the length of each side of the most finely resolved cells in the FEM grid.) First note that all of the information that describes interface; namely, its distance d_e to the center of the cell and the unit normal $\mathbf{n}_e$ to the interface, is stored with respect to the center of the unit cell $\tilde{\Omega}_e$ as depicted in Fig. 5. In particular, the interface in the unit cell $\tilde{\Omega}_e$ is given by

$$\mathbf{n}_{\tilde{e}} \cdot (\mathbf{x} - \mathbf{x}^c_{\tilde{e}}) = d_{\tilde{e}} \tag{79}$$

where $\mathbf{x}^c_{\tilde{e}}$ is the center of $\tilde{\Omega}_e$, $d_{\tilde{e}}$ is the distance of the (mapped) interface from the center $\mathbf{x}^c_{\tilde{e}}$ of $\tilde{\Omega}_e$, and $\mathbf{n}_{\tilde{e}}$ is a unit vector that is perpendicular to the reconstructed linear interface in $\tilde{\Omega}_e$, with the convention that $\mathbf{n}_{\tilde{e}}$ *always points away* from the region containing C_1. The location of the interface is stored by recording $\mathbf{n}_{\tilde{e}}$ and $d_{\tilde{e}}$ for each cell Ω_e that contains a portion of the interface. For the case when the velocity field is perpendicular to a cell edge, say $\partial\Omega_{e,r}$, for some $k = 1, 2, 3, 4$, let $\tilde{\mathbf{n}}_r$ be the outward facing unit normal vector to the rth edge $\partial\tilde{\Omega}_{e,r}$ of the unit cell $\tilde{\Omega}_e$, and, as above, let V_F denote the *total* volume flux that will cross $\partial\Omega_{e,r}$; i.e., the volume flux of C_1 plus the volume flux of C_2.

As shown in Fig. 5, with only a few computationally inexpensive transformations we can use the same algorithm we used to compute the volume fraction on a cell Ω_e in the reconstruction step to compute the volume flux of C_1 across each of the edges of Ω_e. If we map V_F from $\tilde{\Omega}_e$ to another unit cell $\tilde{\Omega}_I$ and assuming the velocity is perpendicular to the rth cell edge $\partial\tilde{\Omega}_{e,r}$ of Ω_e, we find that the interface within the unit cell $\tilde{\Omega}_I$ is given by

$$\mathbf{n}_I \cdot (\mathbf{x} - \mathbf{x}^c_I) = d_I$$

where $\mathbf{x}^c_I$ is the center of $\tilde{\Omega}_I$ as shown in Fig. 5. The values of $\mathbf{n}_I$ and d_I in terms of $\mathbf{n}_e$, $\mathbf{n}_k$, and d_e are given by

$$\mathbf{n}_I = \mathbf{n}_{\tilde{e}} + \left(\frac{V_F}{V_e} - \mathbf{n}_{\tilde{e}} \cdot \mathbf{n}_k\right)\mathbf{n}_k \, ,$$

$$d_I = d_{\tilde{e}} - \left(\frac{1}{2} + \frac{V_F}{2V_e}\right)(\mathbf{n}_{\tilde{e}} \cdot \mathbf{n}_k) \, .$$

where V_e is the volume of Ω_e (the upwind cell for this edge), and $\mathbf{n}_k$ is the outward pointing unit normal to the cell edge $\partial\tilde{\Omega}_{e,r}$.

Since we are computing on a uniform square grid, we have a constant Jacobian and hence, the volumes on the unit cell and the volumes in physical space are related by a constant multiple. For a given interface, there is a simple formula to calculate the volume of C_1 on the side opposite the unit normal $\mathbf{n}$; e.g., see [67]. In our notation this formula is,

$$f(\mathbf{n}, d) = \begin{cases} 1 & \frac{1}{2} \leq \bar{d} \\ 1 - \frac{(\bar{d}-\frac{1}{2})^2}{2m(1-m)} & \frac{1}{2} - m < \bar{d} < \frac{1}{2} \\ \frac{1}{2} + \frac{\bar{d}}{(1-m)} & m - \frac{1}{2} \leq \bar{d} \leq \frac{1}{2} - m \\ \frac{(\bar{d}+\frac{1}{2})^2}{2m(1-m)} & -\frac{1}{2} < \bar{d} < m - \frac{1}{2} \\ 0 & \bar{d} \leq -\frac{1}{2} \end{cases} \tag{80}$$

where $m = 1 - \frac{\|\mathbf{n}\|_\infty}{\|\mathbf{n}\|_1}$ and the components of $\mathbf{n}$ are parallel to sides of the unit cell $\tilde{\Omega}_e$, and $\bar{d} = \frac{d}{\|\mathbf{n}\|_1}$. We use (80) to compute the volume flux of C_1 across the right edge, which is $f(\mathbf{n}_I, d_I)\, V_F$. We use an analogous procedure to compute the flux of C_1 across the other three edges $\Omega_{e,r}$ of Ω_e.

Now let $V_{F,r}$ and $f_r\, V_{F,r}$ denote the total volume flux and the volume flux of C_1 across the rth edge $\Omega_{e,r}$ of Ω_e, respectively. We can now use these quantities to write our *modified* dimensionally split update $f_e^k \to f_e^{k+1}$ of the volume fraction of C_1 in Ω_e in a slightly different form than we did in Eq. (76),

$$\begin{aligned} \tilde{f}_e \left(V_e - \sum_r V_{F,r} \right) &= f_e^k V_e + \sum_r f_r V_{F,r} \\ f_e^{k+1} V_e &= \tilde{f}_e V_e + \sum_r f_r V_{F,r} + \tilde{f}_e \sum_r V_{F,r}\,. \end{aligned} \tag{81}$$

where r only runs over the cell faces on the unit cell that are perpendicular to the direction of that particular sweep. Furthermore, as mentioned before, the order of the sweep directions are alternated at each time step in order to achieve second-order accuracy. Finally, note that the simplified version of these equations; i.e., without the terms $\tilde{f}_e \sum_r V_{F,r}$ in each equation of (81), are essentially identical to the Eqs. (49)–(51), although with a slightly different notation.

3.4.6. The model coupling procedure

Having now described our implementation the VOF method in ASPECT, it is necessary to establish how the computed fluid interface is presented as a so-called *compositional* field C in ASPECT, which will be used by the Finite Element methodology in cases where the tracked fluid is an *active* part of the problem, such as the density, or the viscosity, or both. For example, see [64] where thee density is tracked in ASPECT by placing its values on *active* tracer particles.

In order to be compatible with the existing infrastructure for advecting compositional fields in ASPECT, it proved to be most efficient to present the results of the VOF method as a traditional continuous or discontinuous Galerkin FEM field to the rest of the software. Therefore, the location of the tracked composition is presented as a finite element approximation to the characteristic function implied by the reconstructed interface (i.e., $\chi(d_e - \mathbf{n}_e \cdot (\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_c))$), as one of the "compositional fields" in ASPECT.

In order to avoid additional complexity due to interfering with the field values on neighboring cells, we require that the finite-element field used for the approximation use a discontinuous finite-element discretization; e.g., discontinuous P_q or discontinuous Q_q elements, which are typically denoted as P_{-q} and Q_{-q} elements. In this paper we explicitly consider cases suitable for use with discontinuous P_0 and discontinuous Q_1 elements, and make use of the latter approach. For a number of reasons, often relating to the physical interpretation of the quantity C, it is also desirable to ensure that the generated approximation will always be bounded; i.e., $0 \leq C \leq 1$. Among the more obvious reasons for this requirement are physical constraints, such as the density must satisfy $\rho > 0$.

A basic implementation can be done by directly copying the volume fraction data to a discontinuous P_0 element (i.e. the value of the discretized variable is constant on each cell). This is equivalent to a minimum L^2 error approximation when using the discontinuous P_0 element to approximate the indicator field implied by the reconstructed interface.

However, attempting to obtain an ideal (minimum L2 error) approximation using a higher order element such as DG Q_1 or DG P_1 is more difficult, especially when we wish to respect the bounds on the compositional fields $0 \leq C \leq 1$, since the result of a minimum L_2 error approximation for such an element is almost certain to violate the $0 \leq C \leq 1$ bounds in all non-trivial cases. Also, a basic minimum L_2 error approximation for the indicator function would require significant additional computational expense and code complexity. Thus, any approximation using a non-constant element would best be done using a heuristic approach.

In our implementation, in order to generate a DG Q_1 element approximation to the C field that is implied by the reconstructed interface, we apply the following constraints.

1. The gradient of the element is in the same direction as the normal of the interface.
2. The gradient is as large as possible while maintaining $0 \leq C \leq 1$ everywhere.
3. In order to conserve the mass in this step, the volume fraction implied by the DG Q_1 element approximation to the C field must match the volume fraction f_e in the VOF approximation to the C field; i.e.,

$$\int_{\Omega_e} C(\mathbf{x})\, d\mathbf{x} = f_e V_e$$

where V_e is the volume of Ω_e.

On a square mesh, for a cell with the reconstructed interface

$$\mathbf{n}_{\tilde{e}} \cdot (\mathbf{x} - \mathbf{x}_{\tilde{e}}^c) = d_{\tilde{e}} \tag{82}$$

the above constraints result in the approximation on the unit cell being

$$C(\mathbf{x}) = f_{\tilde{e}} - 1 - |\,2\, f_{\tilde{e}} - 0.5|\, \frac{\mathbf{n}_{\tilde{e}}}{\|\mathbf{n}_{\tilde{e}}\|_1} \cdot (\mathbf{x} - \mathbf{x}_{\tilde{e}}^c) \tag{83}$$

Since we use a DG Q_1 element the above equation produces a bilinear approximation to the VOF method's reconstructed indicator function, with little additional computational cost as compared to using a P_0 approximation.

3.4.7. Coupling with the AMR algorithm

The deal.II library [3] upon which ASPECT is built manages the AMR algorithm through the p4est library [10]. Deal.II, and hence, ASPECT provides a mechanism for setting the refinement criteria; both when to refine a cell and when to coarsen a cell. Since reconstructing and advecting the interface across different levels of refinement both increases algorithm complexity and decreases the accuracy with which the interface is resolved, in this work we ensure that the interface is always on the finest level of refinement. This approach requires that the cells that contain the interface, including the case where the interface is on a cell boundary, and any cell that shares a vertex with any of those cells must also be at the finest level of refinement.

The criteria for refining a cell that we have adopted is a two step algorithm that requires one pass over the entire mesh and one

pass over a subset of the entire mesh. In the first step we check every cell in the entire mesh making a list of all cells that contain a part of the interface. More specifically, we regard all cells Ω_e that satisfy $\epsilon_{vof} < f_e < 1 - \epsilon_{vof}$, where ϵ_{vof} is a small parameter, to contain a portion of the interface. In addition, all cells Ω_e that have a neighboring cell Ω'_e that shares a face with Ω_e and differ in volume fraction sufficiently (e.g., $|f_e - f'_e| > \epsilon_{vof}$) are also added to this list. In the computational results shown in Section 4 we use the value $\epsilon_{vof} = 10^{-6}$. In the second pass over a subset of the entire grid we make a list of all cells that share a vertex with any cell already in the list of cells that contain a portion of the interface and also flag each of these cells for refinement. These flags are then passed to deal.II and thus on to p4est [10], which handles the details of the refinement of these cells and the coarsening of those cells that no longer need to be at the finest level of refinement.

Recall that the CFL (Courant–Friedrichs–Lewy) condition is a constraint on the time step Δt that is typically given by

$$\Delta t \leq \sigma \, \frac{h_e}{\|\mathbf{u}\|_\infty} \qquad \text{for all cells } e \text{ in the computation}. \tag{84}$$

Here the dimensionless constant $\sigma \leq 1$ is the CFL number, h_e is the characteristic size of cell e, and $\|\mathbf{u}\|_\infty$ is the maximum value of the velocity over the entire domain Ω. (See [16,37,42] for more details and, in particular, for an explanation as to why $\sigma \leq 1$ for numerical approximations to the solution of advection equations.) The constraint in (84) is required to hold at each time step t^k, which implies that in most computations Δt must be recomputed at each time step, since the magnitude of the velocity $\mathbf{u}$ may have changed from time t^k to time t^{k+1}. In addition, when AMR is a part of the computation, the minimum or maximum value of h_e may also have changed during the time step.

In ASPECT the constraint in (84) is modified by dividing the right-hand side by the order p of the polynomial basis functions that are used to discretize the velocity field

$$\Delta t \leq \sigma \, \frac{h_e}{p \, \|\mathbf{u}\|_\infty} \qquad \text{for all cells } e \text{ in the computation}. \tag{85}$$

The rational behind this formula is that the number of nodes in a given cell at which a value of the velocity is specified depends on the degree of the polynomial. In other words, in a finite element method the nodes are locations at which the unknowns, such as velocity, are specified just as the corners or centers of a cell in a finite difference or a finite volume method are locations at which the unknowns are specified. Thus, the distance between the nodes $\frac{h_e}{p}$ is analogous to the distance, say Δx, between grid points in a finite difference or finite volume method, for which one would use the formula in (84). (See the section entitled "Numerical experiments to determine optimal parameters" of the reference documentation for deal.II [40] for further information.)

Given that the time step Δt is constrained by (85), the interface can move at most $\sigma \leq 1$ cell widths in one time step $t^k \to t^{k+1}$. This permits the reduction of the frequency with which we must conduct the remeshing procedure to N time steps where $N < \frac{W-2}{2\sigma}$ and W is the minimum width of the maximally refined band of cells. (See, for example, any of the AMR computations in the second (b) and fourth (d) frames in Figs. 21–32 for explicit examples of W.) For the refinement strategy described above, the safest assumption is that $W = 4$. This takes into consideration the case where the interface is at the cell boundary. A band of larger width $W > 4$ would both require a more complex algorithm to find the necessary cells to flag and would increase the number of refined cells. Thus, there is a balance between cost associated with the frequency of running the algorithm to flag cells for refinement and cost of having a larger value of W. This balance is problem dependent.

3.5. Alternate discretizations of the composition equation in ASPECT

In all of the work described in this article we use the Volume-of-Fluid interface tracking algorithm described in Section 3.4 above to approximate solutions of the composition Eq. (10).[6] However, there are three other algorithms implemented in ASPECT that one can use to approximate solutions of the advection Eq. (10) for the compositional variable C. In practice, this variable can be density, viscosity, or any other quantity that is passively transported with the flow. Users may have multiple distinct compositions that are each passively advected with the flow. For completeness we briefly describe them here.

Two of these three alternate advection methods are based on a spatial discretization of the weak form of the advection Eq. (10) for the composition; namely,

$$\left(\frac{\partial C}{\partial t}, \, \psi\right)_\Omega + (\mathbf{u} \cdot \nabla C, \, \psi)_\Omega = 0 \, . \tag{86}$$

1. The first method that was implemented in ASPECT for approximating solutions of the advection Eq. (10) for the quantity is a *continuous* Galerkin finite element method. Since Eq. (86) is the weak form of the advection Eq. (10) and our numerical approximation to solutions of (86) are based on a continuous Galerkin finite element formulation, this advection method also includes an entropy-viscosity stabilization term $\nu_h(C)$*for the compositional field* on the right-hand side of (86),

$$\left(\frac{\partial C}{\partial t}, \psi\right)_\Omega + (\mathbf{u} \cdot \nabla C, \, \psi)_\Omega = -(\nu_h(C) \, \nabla C, \, \nabla \psi)_\Omega \, . \tag{87}$$

We emphasize that the entropy-viscosity stabilization term $\nu_h(C)$ in Eq. (87)*does not* have the same value in each cell as the entropy-viscosity function $\nu_h(T)$ for the temperature that appears in Eq. (40); they are computed separately and are unlikely to have the same value on any given cell Ω_e. The time discretization of the composition equation in this advection method is also the adaptive BDF2 algorithm. This leads to the following FEM Entropy Viscosity (FEM-EV) discretization of Eq. (10),

$$\begin{aligned} &\frac{1}{\Delta t^{k+1}} \left(\frac{2\,\Delta t^{k+1} + \Delta t^k}{\Delta t^{k+1} + \Delta t^k} C^{k+1} - \frac{\Delta t^{k+1} + \Delta t^k}{\Delta t^k} C^k \right. \\ &\qquad \left. + \frac{(\Delta t^{k+1})^2}{\Delta t^k (\Delta t^{k+1} + \Delta t^k)} C^{k-1}, \, \psi \right)_\Omega \\ &= -(\mathbf{u}^k \cdot \nabla C^{k+1}, \psi)_\Omega - (\nu_h^k(C) \, \nabla C^{k+1}, \, \nabla \psi)_\Omega \, . \end{aligned} \tag{88}$$

We often refer to this advection method as the FEM-EV advection method, where 'EV' is an abbreviation for entropy-viscosity. Also, we emphasize that this is the only advection method in ASPECT that has any form of artificial viscosity or entropy-viscosity.
2. Another algorithm for modeling solutions of Eq. (10) that we have implemented in ASPECT is a Discontinuous Galerkin (DG) method with a Bound Preserving limiter (DGBP). See [28] for a detailed description of this method and a comparison with the continuous Galerkin FEM-EV method described in item (1).
3. We have also implemented a particle-in-cell method in ASPECT [22,64], which one can use to approximate the solution of the composition Eq. (22).

[6] This is true albeit with one caveat; namely, in the benchmark problem in Section 4.3 below we compare the computational results obtained with the VOF method to those obtained with the Bound Preserving Discontinuous Galerkin (DGBP) advection method described in this section.

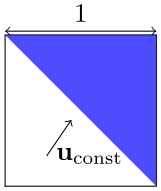


Fig. 6. A diagram of the initial condition for the "Advection of a Linear Interface in a Constant Velocity Field" benchmark.

See [64] for a detailed comparison of these three advection methods with the VOF method described here.

In closing this section we wish to emphasize that in all of the authors' previous and current work designing VOF methods we have not used any type of artificial viscosity, including entropy viscosity, to stabilize the method. In fact, the authors are not aware of any version of a VOF algorithm in which some form of stabilization other than the application of an appropriate CFL constraint was required, regardless of whether the VOF method was coupled to a finite element method or to a finite difference method.

4. Numerical results

In this section we present our numerical results. First, in Section 4.2 we compute two test problems with prescribed velocity fields to verify the accuracy of our implementation of the VOF algorithm. Next, in Section 4.3 we present a sequence of computations of a time-dependent problem; namely, a falling circular region of greater density than the surrounding fluid and measure the convergence rate of the VOF method in this time-dependent flow field with an interface across which there is a jump in density. Then, in Section 4.4, we compute two well-known benchmarks from the computational mantle convection community to verify that our VOF method has been correctly implemented in the underlying mantle convection code ASPECT. Finally, in Section 4.5, we apply the algorithm to a problem of interest in the field of geodynamics.

4.1. Definition of the error measurement

We begin by defining the norm in which we will measure the error between our computed and true solutions (Section 4.2) or estimate the error between computed solutions on successive grids in order to obtain an estimate of the convergence rate of our interface tracking method (Section 4.3).

Since each volume fraction f_e is constant on its grid cell Ω_e, we use P_0 elements to store the value of the volume fraction f_e on each Ω_e. Given a *fixed* grid with cells Ω_e indexed by e we define the error between the exact f_e^{exact} and computed f_e^{comp} volume fractions by

$$\text{Error}\left(f^{exact} - f^{comp}\right) = \sum_e \left| f_e^{exact} - f_e^{comp} \right| V(\Omega_e) \tag{89}$$

where $V(\Omega_e)$ denotes the volume of the cell Ω_e. Note that (89) is the discrete L^1 norm of the difference between f_e^{exact} and f_e^{comp} with weight $V(\Omega_e)$.

4.2. Interface tracking benchmark problems in stationary flows

In this section, we compute two test problems in stationary velocity fields with known exact solutions to ensure that our the implementation of the VOF algorithm is exact to machine zero, ϵ_{mach}, when we use it to advect a line in a constant velocity field of the form in a constant velocity field of the form $\mathbf{u}_{\text{const}} = (u_{\text{const}}, v_{\text{const}})$ and that it converges at its second-order accurate design rate when the flow field is solid body rotation and the interface is a smooth closed curve that does not intersect itself. These very simple problems are what some researchers refer to as 'sanity checks'. In other words, if we do not obtain the expected error / convergence rate, then we know something is wrong with our implementation of the VOF algorithm in the FEM code.

4.2.1. Advection of a linear interface in a constant velocity field benchmark

Our first benchmark is the advection of a linear interface in a constant velocity field $\mathbf{u}_{const} = (\frac{20}{100}, \frac{25}{100})$ as shown in Figs. 6 and 7. In these figures, the computational domain is $[0, 1] \times [0, 1]$ square covered with a grid of square cells of side $h = 2^{-4}$, and the initial interface is one of the two main diagonals of the domain, namely, $y = 1 - x$. At each time step $t^k \to t^{k+1}$ the interface is advanced the velocity field $\mathbf{u}_{\text{const}}$ and then compared with the exact solution $y = \frac{145}{100} - x$. Note that the velocity field *is not* perpendicular to the interface and that neither the interface nor the flow is aligned with the grid.

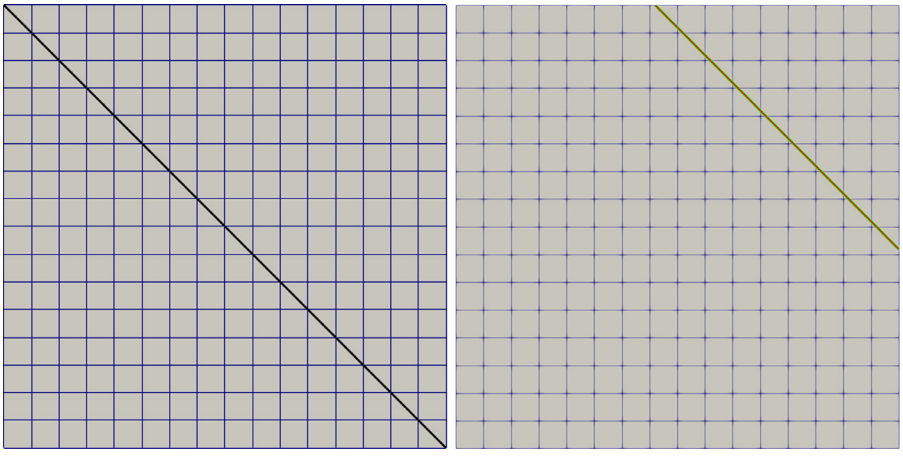


Fig. 7. On the left is the initial condition; namely, a diagonal line reconstructed by the ELVIRA interface reconstruction algorithm. On the right is a comparison between the exact and computed interface at time $t = 1.0$, with the exact interface in red and the computed interface in green. he contour for the (green) computed interface is wider so that the two may be compared visually. It is apparent that the two interfaces are visually indistinguishable. It is also evident from Table 2 that the error between the approximate and true interfaces is $O(\epsilon_{\text{mach}})$ where $\epsilon_{\text{mach}} \approx 10^{-16}$ denotes machine precision. This is because the ELVIRA interface reconstruction algorithm will *always* reconstruct a linear interface *exactly* on a grid of equally sized square cells, i.e., up to machine precision ϵ_{mach} and hence, in a constant velocity field of the form $\mathbf{u}_{\text{const}} = (u_{\text{const}}, v_{\text{const}})$ the approximate interface will remain a line for all time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
The error in advecting a linear interface in a constant velocity field $\mathbf{u}_{\text{const}} = (\frac{20}{100}, \frac{25}{100})$ that is not aligned to the mesh nor perpendicular to the interface. Note that the error is on the order of machine precision $\epsilon_{\text{mach}} \approx 10^{-16}$ and the number of cells that the interface passes through is approximately $\frac{L}{h}$ where L is the distance traveled by the interface from time $t = 0.0$ to time $t = 1$.

h	Error
2^{-4}	$1.23382 \cdot 10^{-16}$
2^{-5}	$1.21675 \cdot 10^{-16}$
2^{-6}	$2.96083 \cdot 10^{-16}$
2^{-7}	$5.92738 \cdot 10^{-16}$

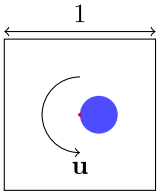


Fig. 8. Diagram of the Circular Interface Rotation Benchmark problem. Note that the red dot is the center of rotation and the circle is offset from the center of rotation by exactly one radius so that the edge of the circle just touches the center of rotation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In this computation we used a CFL number of $\sigma = \frac{1}{2}$, which resulted in, for example, a total of 23 time steps on the least refined grid of $h = 2^{-4}$. (See Eq. (84) below and the accompanying text for the definition of the CFL number σ and, in particular, its modification for advection problems in ASPECT.) Since the ELVIRA interface reconstruction algorithm reconstructs lines exactly (i.e., to $\epsilon_{\text{mach}} \approx 10^{-16}$), we expect the error in computations of a linear interface in a constant velocity field to be exact to machine precision $\epsilon_{\text{mach}} \approx 10^{-16}$. The errors from computations with $h = 2^{-4}$, 2^{-5}, 2^{-6}, and 2^{-7} shown in Table 2 confirm that this is true for our implementation of our VOF method in ASPECT; namely, in all cases the error is $O(\epsilon_{\text{mach}})$.

4.2.2. The circular interface rotation benchmark

The second benchmark problem is the advection of a circular disk containing composition 1 in a rotating velocity field as shown in Fig. 8. In this problem the angular velocity is π radians per unit time with an end time of $t = 2.0$. Note that the center of rotation is *not* at the center of the circle, but rather it lies on the boundary of the circle and is marked with a red dot. In each of these computations we used a CFL number of $\sigma = \frac{1}{2}$. The initial and final states for a computation on a grid with $h = 2^{-6}$ are shown in Fig. 9.

Table 3
The error and convergence rate after the true and approximate circular interfaces have rotated 2π radians. It is apparent that the convergence rate tends to 2.00 as $h \to 0$.

h	Error	Rate
2^{-4}	6.03897×10^{-3}	
2^{-5}	1.74516×10^{-3}	1.79
2^{-6}	3.92745×10^{-4}	2.15
2^{-7}	1.05605×10^{-4}	1.89
2^{-8}	2.63464×10^{-5}	2.00
2^{-9}	6.48952×10^{-6}	2.02

Since our interface reconstruction and advection algorithms are designed to be second-order accurate for smooth interfaces in smooth flows, in this problem we expect the approximate interface to be a second-order accurate approximation to the true interface. The (discrete) L^1 error in the volume fractions f_e and the corresponding convergence rates for six computations with increasing grid resolutions of $h = 2^{-4}, 2^{-5}, \ldots, 2^{-9}$ are shown in Table 3. It is apparent that the convergence rate asymptotes to 2.00, confirming that the VOF method produces a second-order accurate approximation to the true interface.

4.3. The sinking ball benchmark

We now present a nondimensional variation of the Gerya-Yuen [23] 'sinking box' problem in order to perform a convergence study on a non-trivial problem. Our version of the problem is defined on a 1×1 square domain in which a ball (disk) of heavier fluid of radius 0.26 is horizontally centered 0.3 units below the top edge of the domain as shown in Fig. 10. The ball's density is $\rho_1 = 110$, while the background density is $\rho_0 = 100$. The viscosity of both the ball and the background fluid is $\mu_0 = \mu_1 = 10^7$. We approximate the solution (of the nondimensional version) of the incompressible Stokes equations in (6)–(8), but with the term $\rho_0 \alpha (T - T_0) g - \Delta\rho C g$ replaced by ρg in Eq. (8), together with

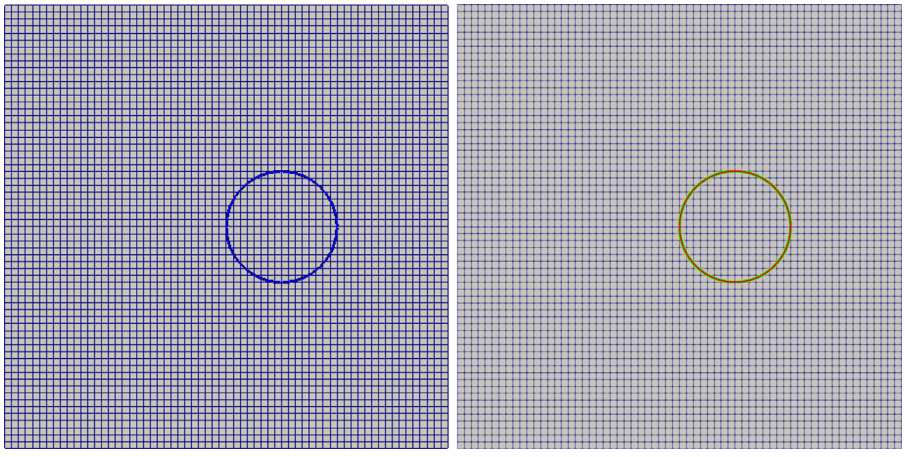


Fig. 9. The initial and final states for the Circular Interface Rotation Benchmark on a uniform grid with $h = 2^{-6}$. On the left is the initial condition as reconstructed by the ELVIRA interface reconstruction algorithm. On the right is a comparison between the true and computed interface after one full rotation ($t = 2.0$), with the true interface in red and the computed interface in green. The green contour for the computed interface is drawn wider so that the two may be compared visually. It is apparent that the two interfaces are *visually* indistinguishable. It is also apparent from Table 3 that the error between the true and computed interfaces is $O(h^2)$ as $h \to 0$ where h is the length of the side of the square cell shown in the table heading. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

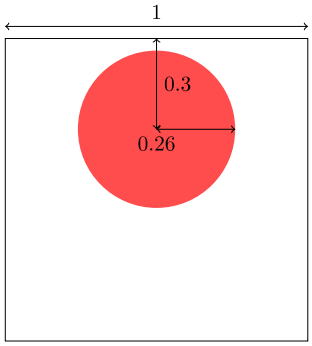


Fig. 10. Diagram of the initial condition for the Sinking Ball Benchmark problem.

the above initial conditions. Thus, we hold the following parameters fixed: (Fig. 11)

$$\begin{array}{ll} \mathbf{g} = (0, 9.8) & \text{acceleration due to gravity} \\ L = 1 & \text{domain height and width} \\ \mu_0 = 10^7 & \text{background viscosity} \\ \mu_1 = 10^7 & \text{ball viscosity} \\ \rho_0 = 100 & \text{background density} \\ \rho_1 = 110 & \text{ball density} \end{array} \tag{90}$$

For comparison, we also compute the same problem in which we use the 'Bound Preserving Discontinuous Galerkin (DGBP)' method in ASPECT [28], which we described briefly in item (2) of Section 3.5, to advect the denser material in the ball. In both cases, the velocity and pressure are discretized by Q_2 and Q_1 elements, respectively. In the DGBP computations, the fluid indicator function $\chi(x,y,t)$ is discretized in space with a discontinuous Q_2 element (often denoted by Q_{-2}) that carries a compositional field $C \stackrel{\text{def}}{=} C_{DGBP}$. This field is initialized by placing the values of χ at time $t = 0$ given by

$$\chi(x,\, y\,;\, t=0) = \begin{cases} 1 & \text{if} \quad 0 \le (x-0.50)^2 + (y-0.70)^2 \le (0.26)^2\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{91}$$

on the support points of the element.

Except for the error in the volume fractions f_e, we estimate the difference between fields w_{2h} and w_h on grids with square cells of side $2h$ and h, respectively, with the following norm,

$$E = \left(\int_{\Omega^h} |w_{2h} - w_h|^p d\,\mathbf{x} \right)^{\frac{1}{p}}, \tag{92}$$

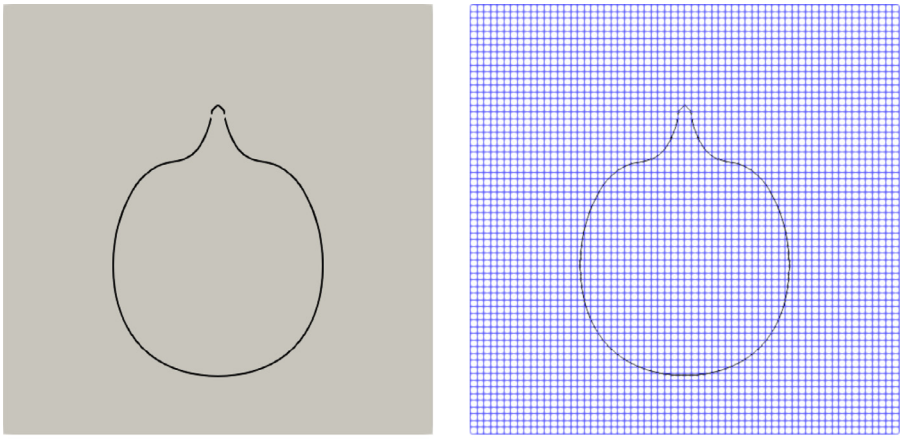
where $p = 1$ or $p = 2$ and w_h indicates that w was computed on a grid with square cells of side $2h$, and similarly for w_h. In Eq. (92) w represents quantities such as the pressure and the two velocity components. Since we use a continuous Galerkin finite element method to discretize w in space, the approximation to w is a piecewise continuous function. The integration in (92) is performed by quadrature, using points and weights generated by a standard Gauss–Legendre quadrature rule on the more refined cells; i.e., those with side h. Asymptotically, the norms of these differences are proportional to the errors on the coarser grids. This allows us to estimate the convergence rate using a formula such as the one in Eq. (13) of [4], despite not having the true solution to the problem. In all of the following convergence studies the final states of the computations were compared at $T_{end} = 5 \times 10^6$.

In the case of the volume fractions f_e, it is not appropriate to approximate them as piecewise *continuous* functions, since they are constant on each cell and typically have discontinuities at some of the edges of cells that contain a portion of the interface. Therefore we estimate the difference between the volume fractions f_e^{2h} on a grid with square cells of side $2h$ and f_e^h on a grid of with square cells of side h with the following norm

$$\text{Error}\left(f_e^{2h} - f_e^h\right) = \sum_e \left| \tilde{f}_e^h - f_e^h \right| V(\Omega_e^h)\,. \tag{93}$$

where $\tilde{f}_e^h$ are volume fractions on the fine grid Ω^h that are obtained from the reconstructed interfaces on the coarse grid Ω^{2h} via a procedure that is described below and in Fig. 12. (Fig. 14)

Just as in Eq. (92), the integration in (93) is performed on the finer grid Ω^h. Some researchers estimate the accuracy and convergence rate of a VOF interface tracking algorithm by using the difference in the values of the volume fractions on successive grids Ω^{2h} and Ω^h as an estimate of the error in the volume fractions on the coarser grid as we have done here; e.g., [4,68]. Others have chosen to estimate the error by integrating the difference between the characteristic functions χ^{2h} and χ^h associated with one of the fluids on two successive grids in order to estimate the error in the characteristic functions; e.g., [74]. However, a reasonably accurate numerical integration of this difference over all cells that contain a portion of the interface would require a relatively expensive



(a) The interface plotted against a tan background.

(b) The interface plotted on top of the uniform grid.

Fig. 11. The interface at time $t = 5 \times 10^6$ for the sinking ball test problem on a uniform grid of square cells with sides of length $h = 2^{-6}$.

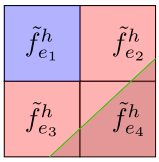


Fig. 12. An example of how the difference between the volume fractions f_e^{2h} on the coarse cells and the volume fractions f_e^h on the finer cells are computed. The reconstructed interface on the coarse cell, shown in green, produces four volume fractions on each of the four more refined subcells of the coarse cell. These subcells, each with side h, are colored pink and blue in the figure and the volume fractions on the subcells, which are derived from the coarser grid, are denoted $\tilde{f}_1^h$, $\tilde{f}_2^h$, $\tilde{f}_3^h$, $\tilde{f}_4^h$. We difference these four volume fractions with the four volume fractions f_j^h , $j = 1, 2, 3, 4$ from the finer grid that correspond to the same cells as the subcells of the coarse cell. Note that no linear interface in the coarse cell can pass through all four refined cells. Thus, for example, the compositional field $\tilde{C}^h$ associated with the volume fractions $\tilde{f}_1^h, \ldots, \tilde{f}_4^h$ on the refined cells will have an $O(1)$ jump between the value $0 < C^{2h} < 1$ on the coarse cell and the value $\tilde{C}^h$ on the blue refined cell where, say, in the figure $\tilde{C}^h_{blue} \equiv 0$. The same reasoning applies if $\tilde{C}^h_{blue} \equiv 1$. For this reason the VOF compositional field $C^h \stackrel{\text{def}}{=} C^h_{VOF}$ converges at a first-order rate whereas the volume fractions f_e^h themselves converge at a second-order rate as shown in Table 4 and Fig. 13. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

computation and, in our case, implementing such an algorithm in ASPECT would be somewhat difficult.

Unlike the functions w that are approximated with a continuous Galerkin element it is necessary to explicitly transfer the volume fraction data f_e^{2h} on the coarse grid to corresponding values $\tilde{f}_{e_j}^h$ for $j = 1, 2, 3, 4$ on the finer grid,in order compare them with the given values $f_{e_j}^h$ on the finer grid. We do this by computing the volume fractions $\tilde{f}_e^h$ on the more refined cells with side h by using the interface reconstructed from the volume fractions f_e^{2h} on the coarser grid with side $2h$ as shown in Fig. 12. We use this reconstructed interface to 'interpolate' (or project) one volume fraction on a coarse cell to four volume fractions $\tilde{f}_e^h$ on the more refined cells that are contained in that coarse cell. We then treat the volume fractions as constant fields on the refined cells and the error is estimated as in Eq. (92).

Due to the nature of the error estimation algorithm, we expect a maximum rate of first-order for the compositional field C derived from the VOF data f_e. This can be seen as follows. First, for a given level of refinement, say h, we consider the number of cells the fluid interface passes through. Assuming the length of the interface L_I is approximately constant under refinement, we can expect the number of cells the fluid interface passes through to be proportional to $\frac{L_I}{h}$, so the number of cells that contain a fluid interface is $O(h^{-1})$. Upon refinement, it is apparent from Fig. 12 that each coarse cell will have at least one refined cell that does not contain a fluid interface, and therefore will have a value

Table 4
The columns labeled C_{DGBP} and C_{VOF} contain the errors computed for the DGBP and VOF compositional fields, respectively. The column labeled VOF f_e contains the errors in the volume fraction data f_e that we use to reconstruct the fluid interface.

h	C_{DGBP} L^1 Error	Rate	C_{VOF} L^1 Error	Rate	VOF f_e L^1 Error	Rate
2^{-4}	5.03×10^{-2}		3.24×10^{-2}		1.65×10^{-2}	
2^{-5}	2.26×10^{-2}	1.16	1.96×10^{-2}	0.73	6.93×10^{-3}	1.25
2^{-6}	1.23×10^{-2}	0.88	1.03×10^{-2}	0.93	2.34×10^{-3}	1.57
2^{-7}	6.22×10^{-3}	0.98	4.87×10^{-3}	1.08	5.59×10^{-4}	2.07
2^{-8}	3.29×10^{-3}	0.92	2.28×10^{-3}	1.09	1.25×10^{-4}	2.16
2^{-9}	1.79×10^{-3}	0.88	1.14×10^{-3}	1.01	3.10×10^{-5}	2.02
2^{-10}	9.88×10^{-4}	0.86	5.82×10^{-4}	0.96	9.02×10^{-6}	1.78

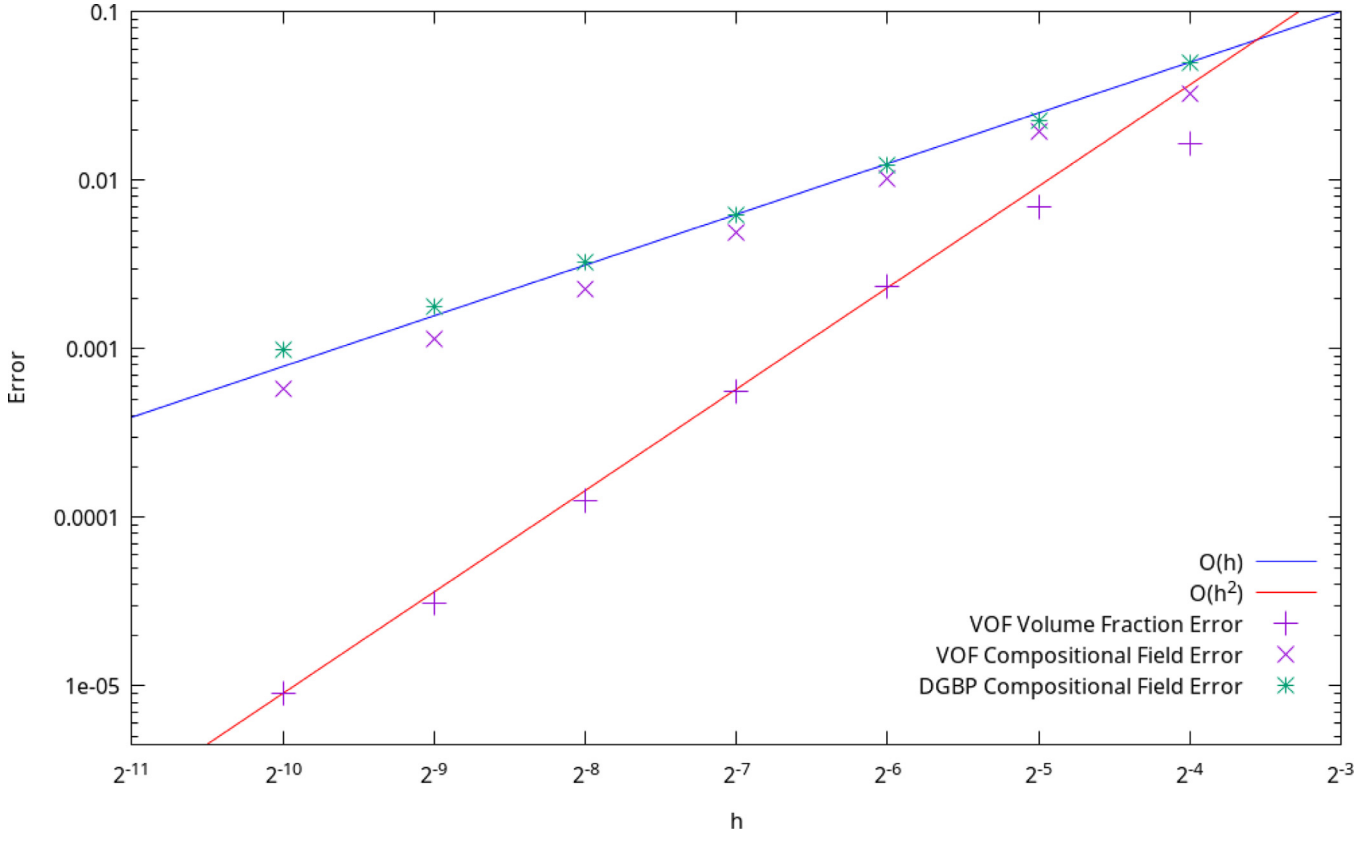


Fig. 13. Convergence rates in the discrete L^1 norm for the volume fractions f_e and the compositional fields associated with the VOF and DGBP computations of the sinking ball test problem.

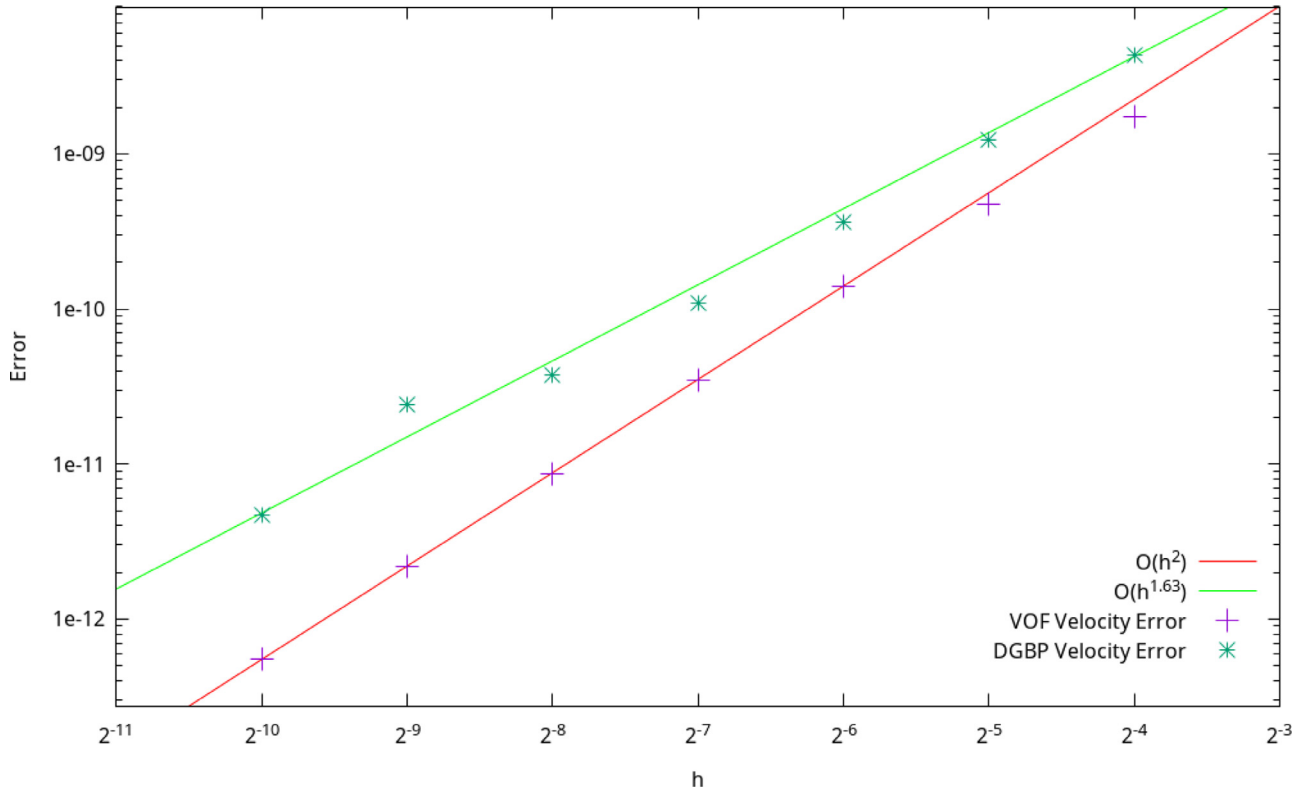


Fig. 14. Errors and convergence rates in the discrete L^2 norm for the velocity fields from the DGBP and VOF computations of the sinking ball test problem.

of either $C = 0$ or $C = 1$ on that cell. Since f_e can be expected to differ significantly from both 0 and 1 in most coarse cells that contain an interface, this results in an $O(1)$ difference in C between the coarse C^{2h} and the fine $\tilde{C}^h$ values for the compositional field on any coarse cell containing the interface, where $\tilde{C}^h$ are the compositional field values due to the volume fractions $\tilde{f}_e^h$ on the subcells that were obtained from the coarse grid volume fractions f_e^{2h} by the procedure described in Fig. 12. Since the volume of a single refined cell is $O(h^2)$, the maximum convergence rate of the C_{VOF} field should therefore be $O(h^{-1}) \times O(h^2) \times O(1) = O(h)$. This analysis agrees with the computational results shown in Fig. 13 and Table 4 (Fig. 15).

We note that the curvature of the initial interface is $\frac{50}{13}$ and hence, according to Eq. (67), the VOF computation will be underresolved as long as $h > \frac{13}{825} \approx 0.015625 = 2^{-6}$. Furthermore, one can see in Fig. 13 and Table 4 that the volume fractions f_e begin to converge at the full, second-order design rate once $h \leq 2^{-6}$ is satisfied.

4.3.1. A study of the effectiveness of AMR on this problem

The primary benefit of AMR is reduced computation time, which is due to a reduced problem size while still permitting a finer mesh in areas of interest. The precise trade-off is dependent on the problem under consideration and the strategies used for refining the grid.

This problem, as with many problems involving fluid interfaces, is especially well suited to AMR. The sole feature in the sinking ball problem is an (initially) smooth region with a higher density than the surrounding fluid and the velocity field is largely only affected in a neighborhood of the fluid interface. In this case the AMR strategy we use, which is to only refine the grid to the maximum level of refinement in a neighborhood of the fluid interface, is very well suited to the problem (Table 5).

Table 5
Errors and convergence rates in the discrete L^2 norm for the velocity. Note that the second order convergence rates for the velocity are what one expects for the $Q_2 \times Q_1$ element combination we have used to approximate the solution of the underlying incompressible Stokes flow. Note also that an error of $O(10^{-12})$ is roughly the smallest error we expect to be able to compute accurately given the tolerance set for the iterative solver of the Stokes matrix equation.

h	DGBP Velocity L^2 Error	Rate	VOF Velocity L^2 Error	Rate
2^{-4}	4.38×10^{-9}		1.74×10^{-9}	
2^{-5}	1.23×10^{-9}	1.83	4.76×10^{-10}	1.87
2^{-6}	3.64×10^{-10}	1.76	1.41×10^{-10}	1.76
2^{-7}	1.09×10^{-10}	1.74	3.47×10^{-11}	2.02
2^{-8}	3.78×10^{-11}	1.53	8.63×10^{-12}	2.01
2^{-9}	2.44×10^{-11}	0.64	2.16×10^{-12}	2.00
2^{-10}	4.69×10^{-12}	2.38	5.49×10^{-13}	1.98

In order to examine the performance of the AMR algorithm versus computing on a uniform grid with cell size $h_{\min} \times h_{\min}$, where $h_{\min}$ denotes the size of the most refined cell in the AMR computation, we compare the performance of the two grid strategies when we apply them to the sinking ball problem. In all of these computations we use the AMR grid refinement strategy described in Section 3.4.7. Thus, the fluid interface is the *only* feature of importance in this comparison. The AMR algorithm is configured so that the coarsest (square) cell size possible is $h = 2^{-3}$ and we recalculate the mesh at every time step (Table 6).

In order to examine the efficacy of the AMR computations, the solutions of the uniform and AMR computations are compared using the following value for the error

$$E = \left(\int_{\Omega^h} |w_{uniform} - w_{AMR}|^p \, d\mathbf{x} \right)^{\frac{1}{p}} \tag{94}$$

for $p = 1$ or $p = 2$ where, as in Eq. (92), the integration is performed by quadrature, with points and weights generated by a

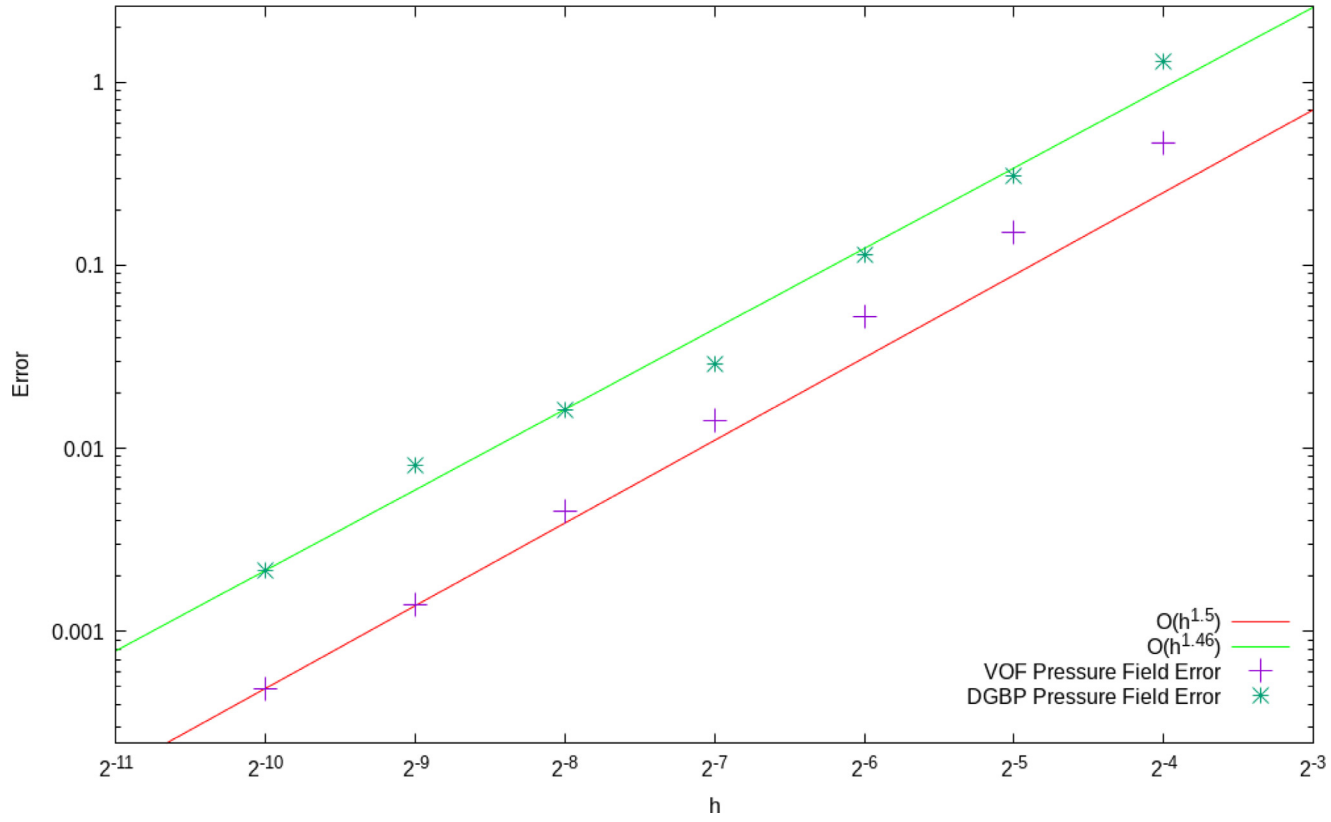


Fig. 15. Convergence rates in the discrete L^2 norm for the pressure from the DGBP and VOF computations of the sinking ball test problem.

Table 6
Errors and convergence rates for the pressure in the discrete L^2 norm. Note that the convergence rates for the pressure are essentially what one expects for the $Q_2 \times Q_1$ element combination that we have used to approximate the solution of the underlying incompressible Stokes flow.

h	DGBP Pressure L^2 Error	Rate	VOF Pressure L^2 Error	Rate
2^{-4}	1.30×10^{0}		4.67×10^{-1}	
2^{-5}	3.10×10^{-1}	2.06	1.52×10^{-1}	1.62
2^{-6}	1.14×10^{-1}	1.44	5.22×10^{-2}	1.55
2^{-7}	2.89×10^{-2}	1.98	1.42×10^{-2}	1.88
2^{-8}	1.61×10^{-2}	0.85	4.54×10^{-3}	1.64
2^{-9}	8.11×10^{-3}	0.99	1.40×10^{-3}	1.69
2^{-10}	2.13×10^{-3}	1.93	4.88×10^{-4}	1.52

Table 7
The number of cells at the final time T_{end} for our AMR computations and the wall clock runtimes with growth rates for both AMR and uniform grids. Here $n = h^{-1}$ is the number of grid cells on a side of the computational domain when the computation is made at the maximum permitted level of refinement.

n	AMR cells	AMR(sec)	Rate	Uniform(sec)	Rate
2^3	6.40000×10^{1}	1.56×10^{0}		8.19×10^{-1}	
2^4	1.84000×10^{2}	6.99×10^{0}	2.16	4.43×10^{0}	2.44
2^5	4.60000×10^{2}	3.02×10^{1}	2.11	2.96×10^{1}	2.74
2^6	1.04800×10^{3}	1.14×10^{2}	1.92	2.11×10^{2}	2.83
2^7	2.26000×10^{3}	4.14×10^{2}	1.86	1.51×10^{3}	2.84
2^8	4.66000×10^{3}	1.59×10^{3}	1.94	1.16×10^{4}	2.94

Table 8
The L^1 norm of the difference between the VOF compositional field $C = C_{VOF}$, (i.e., the integral of C_{VOF} integrated over the entire computational domain Ω), the volume fractions f_e, and the L^2 norm of the difference of the velocity field $\mathbf{u}$, each computed on a uniform grid versus an AMR grid at time, T_{end}.

n	$\|C - C_{AMR}\|_1$	$\|f_e - f_{eAMR}\|_1$	$\|\mathbf{u} - \mathbf{u}_{AMR}\|_2$
2^4	3.80×10^{-6}	3.80×10^{-6}	4.30×10^{-11}
2^5	1.41×10^{-4}	1.41×10^{-4}	4.40×10^{-11}
2^6	5.41×10^{-5}	5.26×10^{-5}	4.10×10^{-11}
2^7	2.36×10^{-5}	2.34×10^{-5}	4.16×10^{-11}
2^8	1.28×10^{-5}	1.27×10^{-5}	4.10×10^{-11}

standard Gauss–Legendre quadrature rule on the uniform grid. Since the refinement strategy requires that the fluid interface always be at the maximum level of refinement, it is sufficient to compare the f_e values by treating them as a field that is constant over the corresponding cell.

In Fig. 16 and Table 7 it is apparent that AMR reduces the computational cost from $O(h^{-3}) = O(n^3)$ to $O(h^{-2}) = O(n^2)$ where $n = h^{-1}$ is the number of grid cells on a side of the computational domain when the computation is made at the maximum permitted level of refinement. This is a significant benefit.

Additionally, in Table 8 we show the difference in C_{VOF}, f_e, and the velocity $\mathbf{u}$ for a computation with AMR versus a computation on a uniform grid. It is apparent from this table that the norm of the differences at the final time for C_{VOF} and f_e are both small and decreasing when $h \leq 2^{-6}$, which, according to (67), is the grid size at which the interface is "well-resolved". Note that since $(2^{-8})^2 = 2^{-16} \approx 1.53 \times 10^{-5}$, only for the most refined case (i.e., $h = 2^{-8}$) do we have a difference in the volume of the advected fluid that is equivalent in magnitude to the volume of a single cell. For all other refinement levels it is *less than* the volume of a single cell. On the other hand the L^2 norm of the difference in the velocities is $O(10^{-11})$ for all refinement levels, which is near the limit of the accuracy that we can expect for our computations.

Tables 9 and 10 demonstrate that when we use AMR instead of a uniform grid it does not have a significant effect on the computed (total) volume of either C_{VOF} (Table 9) or C_{DGBP} (Table 10). Note that in Table 10 the magnitude of the difference between the

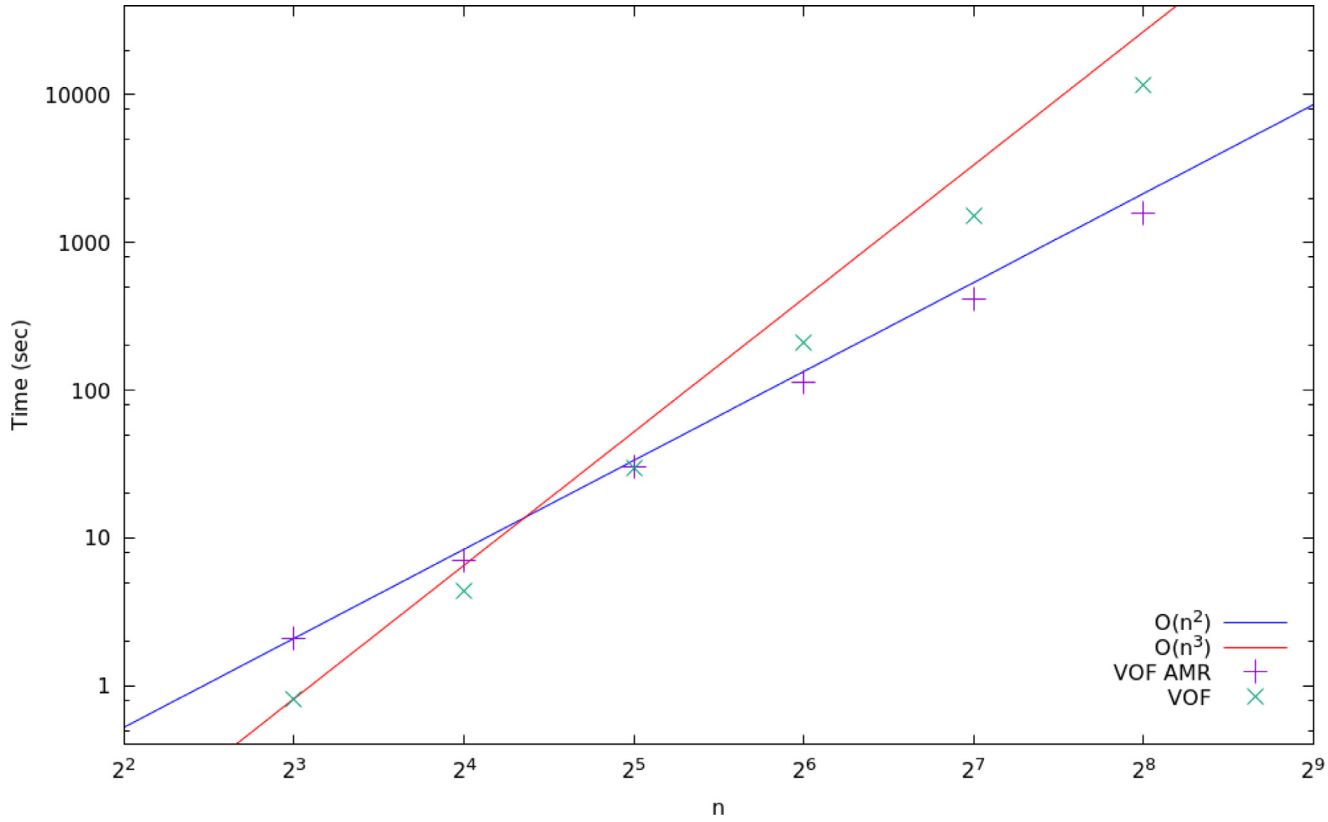


Fig. 16. Wall clock run times for AMR versus a uniform grid.

Table 9
The total change in the volume of the composition C_{VOF} between the initial and final state when we use the VOF advection method. The correct volume is approximately 0.21238. For the AMR computation h is the maximum level of refinement; i.e., the smallest size a refined cell is allowed to be.

$h = 2^{-k}$	Uniform mesh	Relative change	AMR	Relative change
2^{-3}	-5.81×10^{-4}	2.73×10^{-3}	-5.81×10^{-4}	2.73×10^{-3}
2^{-4}	-1.59×10^{-4}	7.48×10^{-4}	-1.60×10^{-4}	7.55×10^{-4}
2^{-5}	-3.94×10^{-5}	1.85×10^{-4}	-3.90×10^{-5}	1.83×10^{-4}
2^{-6}	-9.77×10^{-6}	4.60×10^{-5}	-9.51×10^{-6}	4.48×10^{-5}
2^{-7}	-2.45×10^{-6}	1.15×10^{-5}	-2.33×10^{-6}	1.10×10^{-5}
2^{-8}	-6.11×10^{-7}	2.88×10^{-6}	-5.64×10^{-7}	2.66×10^{-6}
2^{-9}	-1.53×10^{-7}	7.20×10^{-7}	-1.42×10^{-7}	6.69×10^{-7}
2^{-10}	-3.80×10^{-8}	1.79×10^{-7}	-4.40×10^{-8}	2.07×10^{-7}

Table 10
The total change in the volume between the initial and final state of the composition C_{DGBP} when we used the DGBP advection method. The correct volume is approximately 0.21238. For the AMR computation h is the maximum refinement level; i.e., the smallest size a refined cell is allowed to be. Note that for the DGBP advection algorithm, the difference in the correct volume and the computed volume is approximately $O(h)$. The output from ASPECT of the total value of the composition C over the domain Ω is limited to an accuracy of about $O(10^{-10})$ and hence, for $h = 10^{-10}$ we could not accurately subtract the computed values of these quantities from the true value. Thus, we have omitted the last row of this table since the difference between the computed and true volumes were $O(10^{-10})$.

$h = 2^{-k}$	Uniform mesh	Relative change	AMR	Relative change
2^{-3}	-4.21×10^{-5}	1.98×10^{-4}	-4.21×10^{-5}	1.98×10^{-4}
2^{-4}	-1.64×10^{-5}	7.70×10^{-5}	-2.13×10^{-5}	1.00×10^{-4}
2^{-5}	-3.18×10^{-6}	1.50×10^{-5}	-2.98×10^{-6}	1.40×10^{-5}
2^{-6}	-4.23×10^{-7}	1.99×10^{-6}	-3.31×10^{-7}	1.56×10^{-6}
2^{-7}	-6.10×10^{-8}	2.87×10^{-7}	-4.00×10^{-8}	1.88×10^{-7}
2^{-8}	-8.00×10^{-9}	3.77×10^{-8}	-4.00×10^{-9}	1.88×10^{-8}
2^{-9}	-2.00×10^{-9}	9.42×10^{-9}	-1.00×10^{-9}	4.71×10^{-9}

true value of approximately 0.21238 for the integral of C_{DGBP} over the domain Ω and the computed value is $O(h)$. Also note that in Table 10 we have not displayed the results of our computations with $h = 2^{-10}$, which are $O(10^{-10})$, since the difference between the true and computed values of this quantity is $O(10^{-10})$, which is too small to be reliable given that ASPECT only outputs the computed values to an accuracy of $O(10^{-10})$. Finally, as we mentioned in Section 3.4 there are operator split VOF advection algorithms that conserve the volume of the fluid to machine zero, which would negate the errors shown in Table 9; e.g., see [4,68,81].

4.4. Mantle convection benchmark problems

In this section we compute two 'benchmark' problems that are well-known and frequently used in the computational mantle convection community to demonstrate that our VOF interface tracking algorithm can reproduce previously published computational results of the same problem. In our view the first problem, commonly known as the "van Keken problem" or the "van Keken isoviscous Rayleigh-Taylor problem" is not a reasonable 'benchmark', since the problem is mathematically ill-posed. In other words, it is unstable [11] and perturbations due to different numerical methods can yield vastly differing results. In fact, in [64] we demonstrated that it suffices to change only the algorithm with which the composition variable C is advected in order to obtain clearly different results at the same output time. For example, see Fig. 11 of [64] or compare Fig. 5(c) and (d) of [66] to our results here or in [64] or to the results in [79]. (Fig. 17)

4.4.1. The van Keken isoviscous Rayleigh–Taylor Problem

In this section, we present our computation of the van Keken isoviscous Rayleigh–Taylor problem [79]. In spite of the fact that the problem is unstable and hence ill-posed, it has become a

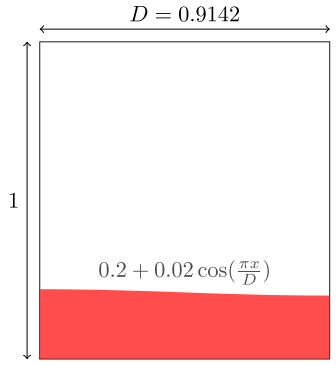


Fig. 17. A diagram of the initial condition for the "van Keken" test problem [79]. Note that although the interface appears to be a straight line it actually has a very small (0.02) amplitude perturbation that is barely visible on this scale. This perturbation produces a Rayleigh-Taylor instability, the results of which can be seen in Fig. 18. In the formula for the perturbation D = 0.9142 is the width of the domain.

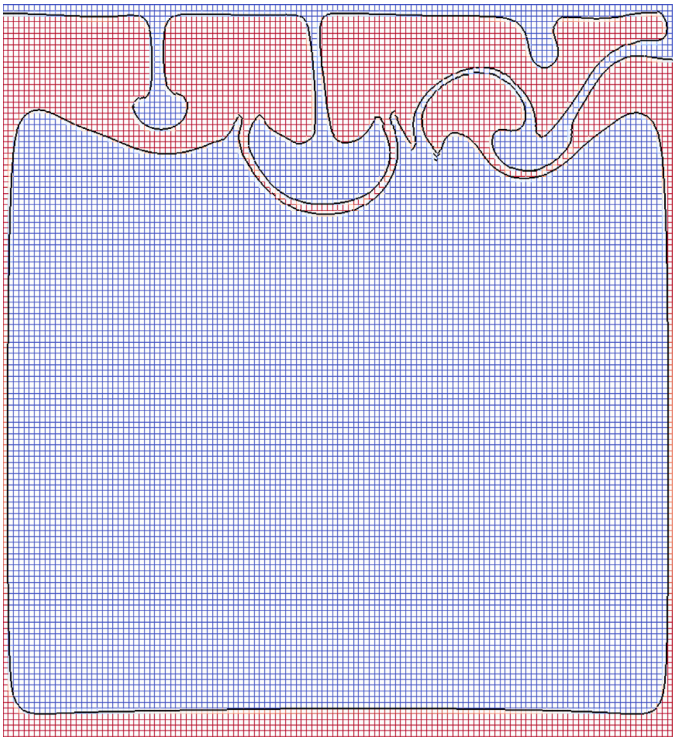


Fig. 18. Computed solution of the van Keken isoviscous Rayleigh–Taylor problem at time $t = 2000$ on a uniform grid of 128×128 cells. Compare with the computational results in [41,64,66,79].

standard "benchmark" in the computational geodynamics community. In this problem a less dense (buoyant) fluid lies beneath a denser fluid, with a perturbed interface between the two layers. The problem is computed in a $[D, 1]$ computational domain where $D = 0.9142$ is the width of the domain. The initial discontinuity between the two compositional / density layers is given by

$$C(x, y, t = 0) = \begin{cases} 0, & \text{if} \quad 0 \le y < 0.2 + 0.02\cos(\pi x / D), \\ 1, & \text{otherwise}. \end{cases} \tag{95}$$

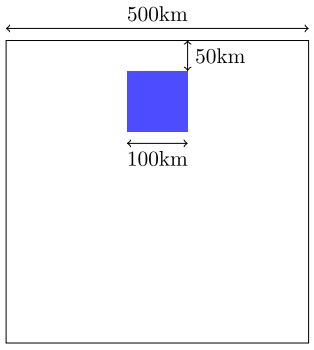


Fig. 19. Initial conditions for the Gerya-Yuen "Sinking-Box" test problem.

This initial condition has a (discontinuous) interface along the curve

$$y = 0.2 + 0.02\cos\left(\frac{\pi x}{D}\right). \tag{96}$$

4.4.2. The Gerya-Yuen sinking box benchmark

Following the original authors, we pose the Gerya-Yuen 'sinking box' problem [23] in dimensional form. The problem is defined on a 500 km × 500 km two-dimensional Cartesian computational domain. A small horizontally centered 100 km × 100 km square is placed with its top edge 50 km below the top of the domain so that the initial location and dimension of the box is defined by the composition field $C(\mathbf{x}, t)$ as follows:

$$C(\mathbf{x}, 0) = \begin{cases} 1, & \text{if } (x, y) \in [200\text{ km}, 300\text{ km}] \times [350\text{ km}, 450\text{ km}], \\ 0, & \text{otherwise}. \end{cases} \tag{97}$$

The block's density is $\rho_1 = 3300$ kg/m^3, while the background density is $\rho_0 = 3200$ kg/m^3. We approximate the solution of the incompressible Stokes equations (i.e.,Eqs. (6)–(8) without the term $\rho_0\alpha(T - T_0)g$ in Eq. (8)) with these initial conditions and holding the following parameters fixed: (Figs. 19 and 20)

$$\begin{array}{ll} \mathbf{g} = (\mathbf{0}, \mathbf{9.8})\ \text{m/s}^2, & \text{acceleration due to gravity} \\ L = 500\ \text{km} & \text{domain height and width} \\ \mu = 10^{21}\ \text{Pa}\cdot\text{s} & \text{viscosity} \\ \rho_0 = 3200\ \text{kg/m}^3, & \text{background density} \\ \rho_1 = 3300\ \text{kg/m}^3, & \text{small box density} \end{array} \tag{98}$$

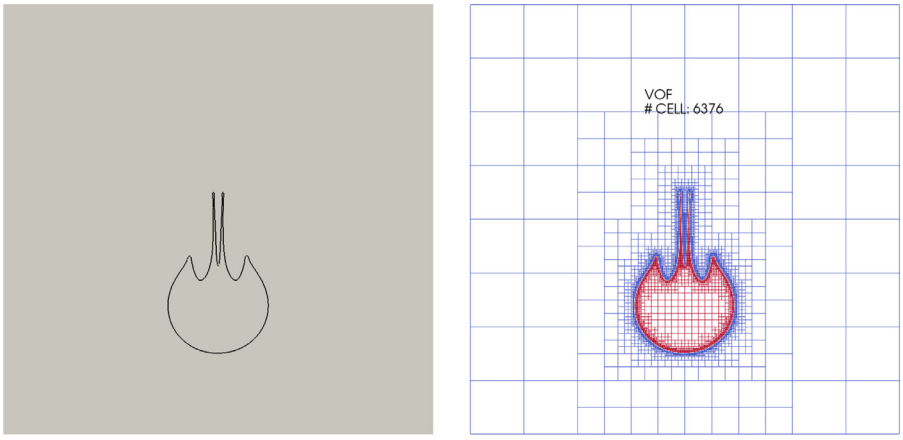
4.5. Computations of thermochemical convection in a density stratified fluid

We now present the results of our computations of the model problem for thermochemical convection with density stratification, the equations for which were presented in Section 2. In these computations the Rayleigh number is fixed at $\text{Ra} = 10^5$ and we vary only the buoyancy ratio as follows: B = 0.0, 0.1, 0.2, ..., 1.0 and B = 2.0. The domain for all of the computational results shown below is a two-dimensional rectangular region that we denote by $\Omega = [0, 3] \times [0, 1]$ as shown in Fig. 1.

The initial conditions for the temperature T are,

$$T(\mathbf{x}, 0) = \begin{cases} (1 - 5y) + A\sin(10\pi y)\,(1 - \cos(\frac{2}{3}k\pi x)) & \text{if} \quad 0 \le y \le \frac{1}{10}, \\ (5 - 5y) + A\sin(10\pi y)\,(1 - \cos(\frac{2}{3}k\pi x + \pi)) & \text{if} \quad \frac{9}{10} \le y \le 1, \\ 0.5 & \text{otherwise}, \end{cases} \tag{99}$$

where the period of the perturbation $k = 1.5$ and the amplitude of the perturbation $A = 0.05$. Note that $A = 0.05$ ensures that $0 \le T(x, y; 0) \le 1$ throughout the entire computational domain. The



(a) The interface against a tan background.

(b) The interface together with the underlying AMR grid.

Fig. 20. Fluid interface for the Gerya-Yuen [23] 'sinking box' problem at time $t = 9.81$ Myr computed with AMR as shown on the right.

initial conditions for the composition are,

$$C(x, y; t = 0) = \begin{cases} 1 & \text{if} \quad 0 \le y < \frac{1}{2}, \\ 0 & \text{if} \quad \frac{1}{2} \le y \le 1 \end{cases} \tag{100}$$

and the boundary conditions for the velocity and temperature are as specified in (26)–(31).

All of the results shown below were computed twice: once on a fixed, uniform grid with 192×64 square cells each with side $h = 64^{-1}$ and then on the same underlying grid but with the addition of two levels of an adaptively refined mesh in a neighborhood of the interface. Each level of refinement increases the grid resolution by a factor of two; i.e., $h \to \frac{h}{4}$ with two levels of refinement.

Finally, as noted in Section 3.3 above, in all of the computations in this section the local Péclet number $\mathrm{Pe}_e < 10$. And furthermore, we have conducted several extensive studies (e.g., see [27,64]) that have lead us to conclude that using the algorithm described in Section 3.3 with the entropy-viscosity stabilization technique for approximating solutions of the advection-diffusion equation for the temperature (21) does not change these computational results.

5. Discussion

In Section 4.2 we demonstrated that our implementation of the VOF method in ASPECT is exact to machine zero, ϵ_{mach}, when we use it to advect a line in a constant velocity field of the form $\mathbf{u}_{\text{const}} = (u_{\text{const}}, v_{\text{const}})$ and that it converges at its second-order accurate design rate when the flow field is solid body rotation and the interface is a smooth closed curve that does not intersect itself.

Then in Section 4.3 in order to examine the convergence rate of our VOF methodology on a more difficult time-dependent problem we introduced a problem in which a circular region with a higher density than the surrounding fluid falls. We also use this problem to carefully assess the efficacy of computing with the interface refined using AMR versus computing the same problem on a uniform grid with the (uniform) cell size being the same as the smallest cell size we allowed in the AMR computations. The results of our tests confirm that the algorithm converges at the full design rate. We also confirmed that the AMR strategy yields a significant increase in computational efficiency while remaining close to the uniform mesh result.

In Section 4.4 we demonstrated that the method reproduces (visually) two benchmarks from the computational mantle convection literature.

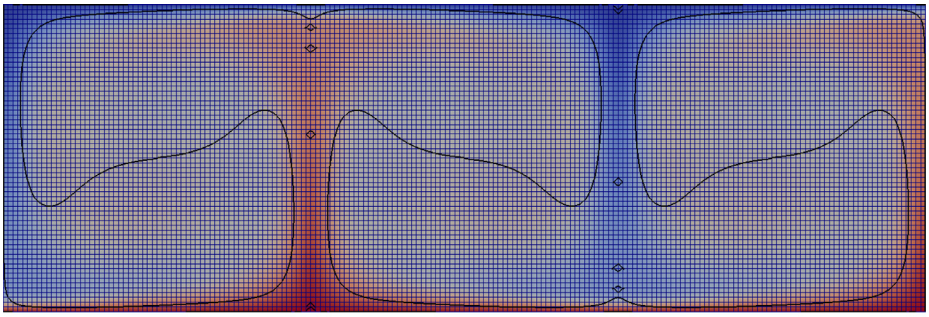
We now present a detailed discussion of the results of our computations of thermochemical convection in density stratified flow shown in Section 4.5. This model problem is designed to study the basic physics underlying the formation of thermal plumes that form at LLSVPs, entrain some of the material in the LLSVP, and bring it to the Earth's surface. It is also a two dimensional analog of the experimental results of Davaille [19] and Le Bars and Davaille [7,8].

5.1. Computations of thermochemical convection in a density stratified fluid

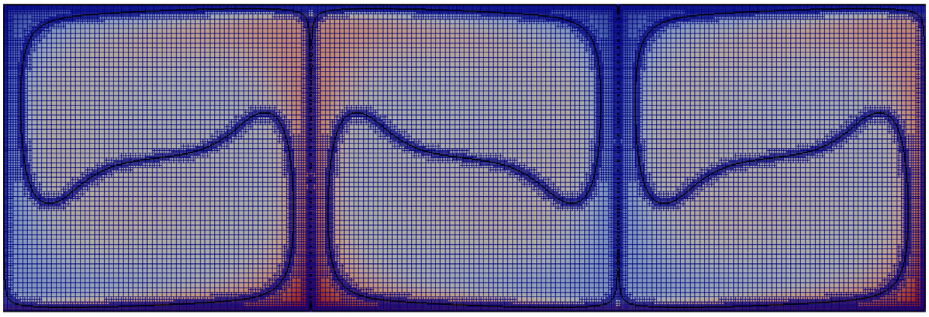
Examining the results in Section 4.5 of our computations of thermochemical convection in a density stratified fluid for values of the nondimensional buoyancy parameter $\mathrm{B} = 0.0, 0.1, 0.2, \ldots, 1.0$ and $\mathrm{B} = 2.0$ at Rayleigh number $\mathrm{Ra} = 10^5$, we note a fundamental change in the dynamics and structure of the flow field as B increases from $\mathrm{B} = 0.0$ to $\mathrm{B} = 2.0$. First, considering only the extreme values $\mathrm{B} = 0.0$ and $\mathrm{B} = 2.0$, we observe the following difference in the qualitative behavior of the interface. For $\mathrm{B} = 0.0$ (Fig. 21), which is the classic Rayleigh-Bénard problem in which there is no difference in the densities of the two fluids (i.e., $\Delta\rho = 0$), the height of the convection cells is equal to the height of the domain Ω and we observe the steady cellular convection structure with three 1×1 counter rotating cells as predicted by the analysis in Section 6.21 of [78]. That the flow is steady, (i.e., independent of time) in Fig. 21 is apparent after comparing the temperature fields at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$.

Due to the variations in the velocity solution, the thermal Péclet numbers are dependent on the regime in which the solution falls. Because the VOF scheme is inherently non-diffusive, the compositional Péclet numbers are always $\mathrm{Pe} = +\infty$ For the whole layer convection regime $\mathrm{Pe} = 3.602 \times 10^2$, while in the stratified regime $\mathrm{Pe} = 1.004 \times 10^2$. In both cases, the Péclet number is sufficiently small to justify the use of the entropy viscosity stabilized continuous Galerkin advection scheme for the temperature advection.

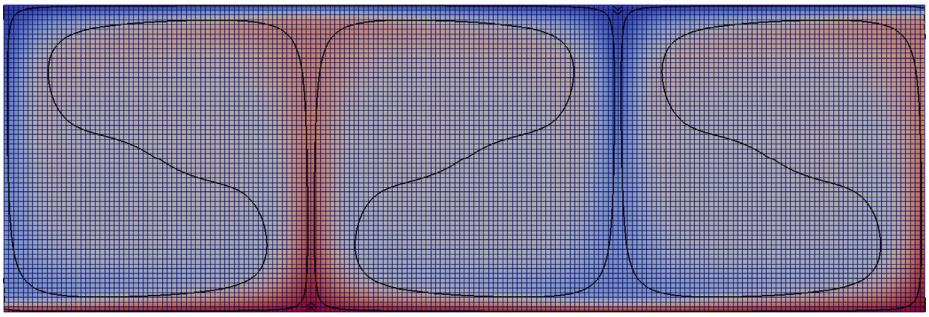
Note that for $\mathrm{B} = 0.0$ each of the three 1×1 convection cells overturn at the same fixed rate. On the other hand, for $\mathrm{B} = 2.0$ the magnitude of $\Delta\rho$ prevents the denser fluid from reaching the



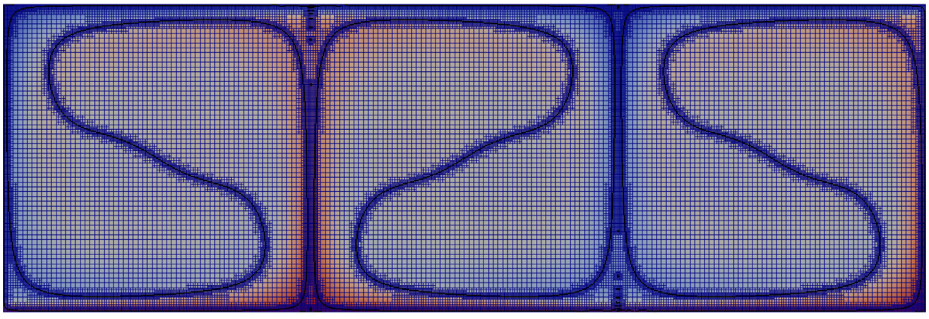
(a) $\mathrm{B} = 0.0$ at $t' = 1.97 \cdot 10^{-2}$ ON A UNIFORM GRID of 196×64 cells.



(b) $\mathrm{B} = 0.0$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.0$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

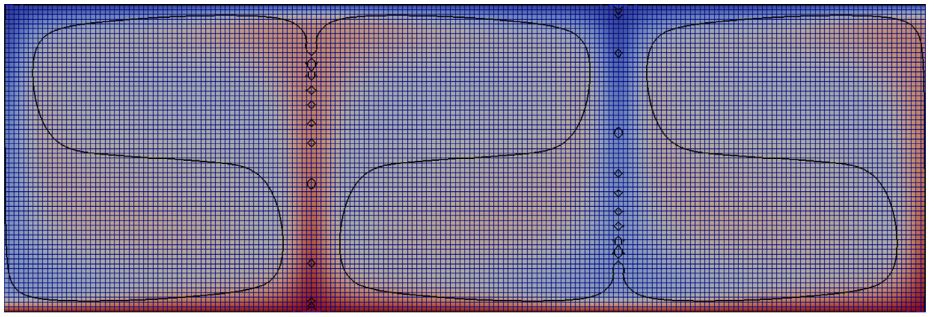


(d) $\mathrm{B} = 0.0$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

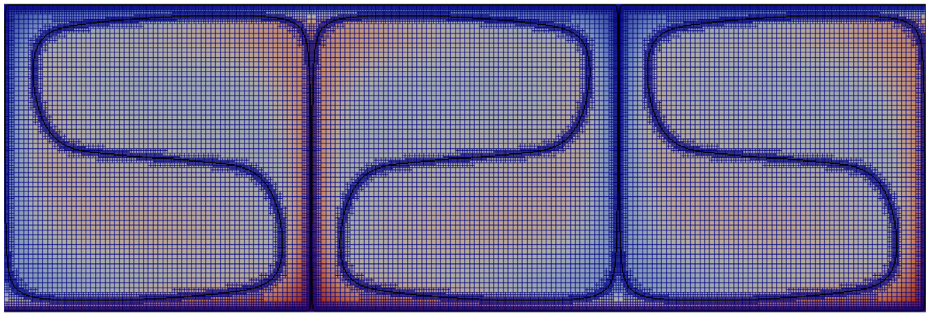
Fig. 21. Computations with $\mathrm{B} = 0.0$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

top of the domain and producing overturns, and hence convection cells, on the scale of the height of the domain. Rather, the structure of the flow shown in Fig. 32 consists of six (roughly) square counter rotating $\frac{1}{2} \times \frac{1}{2}$ cells below $y = 0.5$ and a similar structure above $y = 0.5$. Thus, for $\mathrm{B} = 2.0$ we observe a permanently stratified convection structure. Furthermore, from $\mathrm{B} = 0.7$ in Fig. 28 and, *perhaps*, from $\mathrm{B} = 0.4$ in Fig. 25 or $\mathrm{B} = 0.5$ in Fig. 26, on; i.e., as $\mathrm{B} \to 2.0$ from below with $\mathrm{B} > \mathrm{B}_c$ where $0.3 < \mathrm{B}_c \leq 0.7$, it appears that *at the times shown* the flow is tending *continuously* toward the stratified convection pattern shown in Fig. 32.

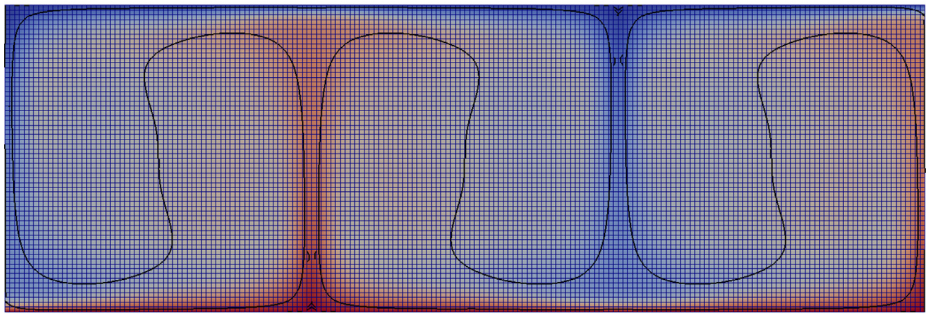
The features at either end of the interval $\mathrm{B} = [0.0\,.\,2.0]$ are consistent with the diagrams - obtained from experiments - on the left and right of Fig. 1 in [8], although in the diagram on the right the authors have only drawn three cells above and



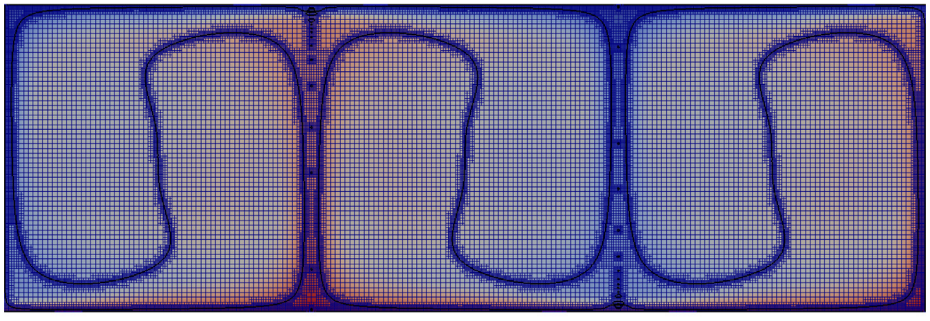
(a) $\mathrm{B} = 0.1$ at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) $\mathrm{B} = 0.1$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



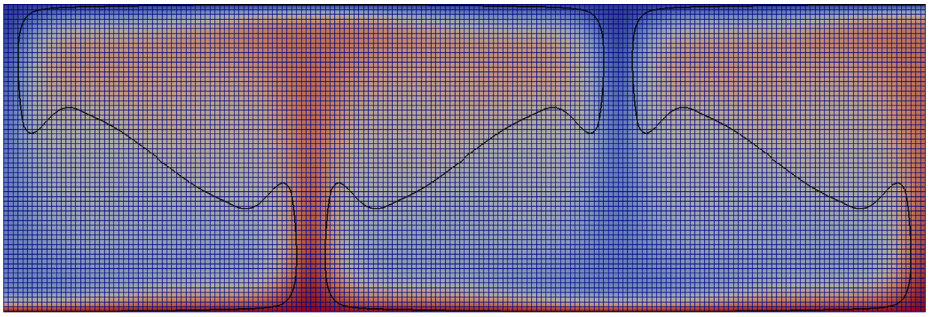
(c) $B = 0.1$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



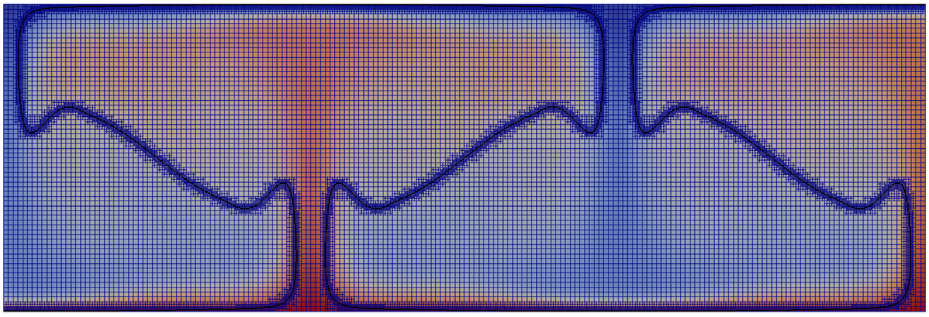
(d) $\mathrm{B} = 0.1$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

Fig. 22. Computations with $\mathrm{B} = 0.1$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

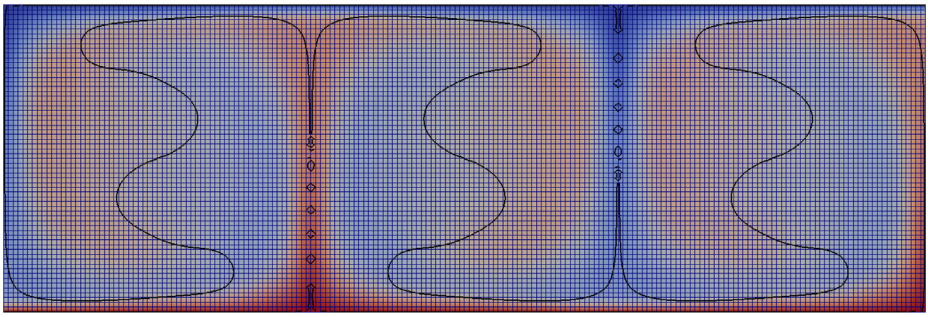
three cells below the centerline and, in both drawings, the cells appear to be more rectangular than square in shape. We assume that these diagrams are simply rough sketches of the dynamics of what the authors of [8] refer to as “Whole Layer” (left) and “Stratified” (right) convection. Perhaps, also, these diagrams are for different values of the other two nondimensional parameters the authors varied in the work described in the sequence of papers [7,8] and [19]; namely, the ratio a of the height of the lower layer to the height of the entire domain and the ratio γ of viscosity of the lower layer to that of the upper layer. In the work we present in Section 4.5 we did not vary these other two parameters; they were held fixed at $a = 0.5$ and $\gamma = 1.0$. In short, we conclude that our computational results correctly correspond *qualitatively* to what the authors of [8] observe in their



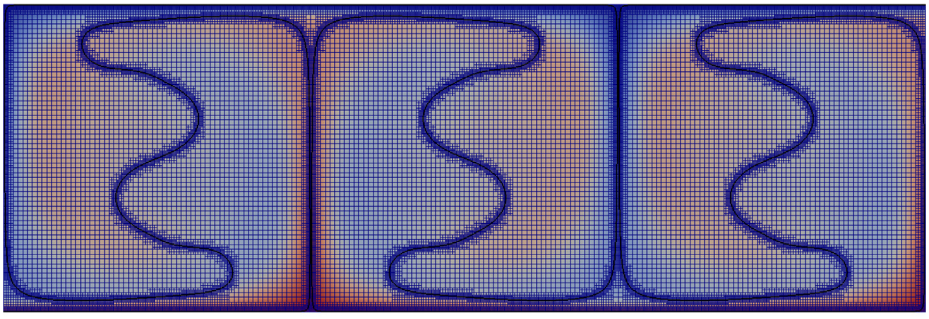
(a) B = 0.2 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 0.2 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.2$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

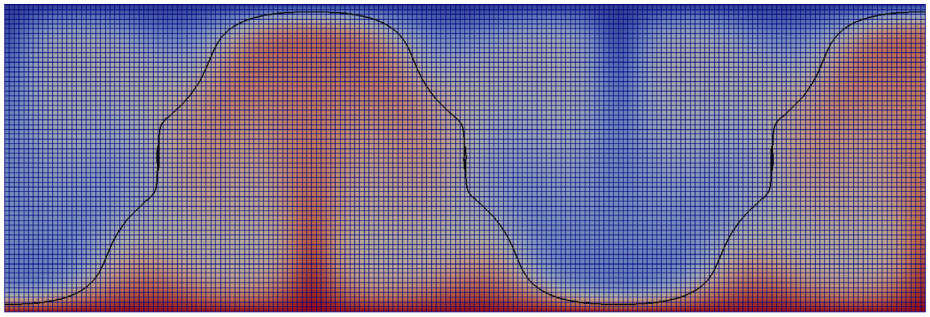


(d) B = 0.2 at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

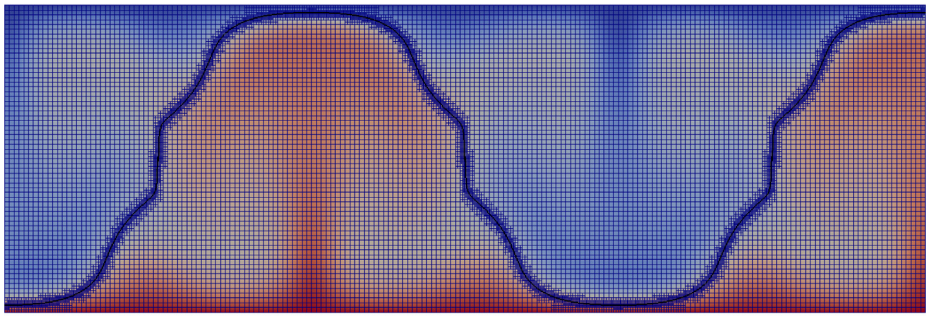
Fig. 23. Computations with B = 0.2 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

experiments when the nondimensional parameters a and γ are held fixed at $a = 0.5$ and $\gamma = 1.0$. Finally, note the similarity of the two counter rotating convection cells on the right in Figs. 21c–d and 22c and d to the structure of the flow in Fig. 4(a) of [7].

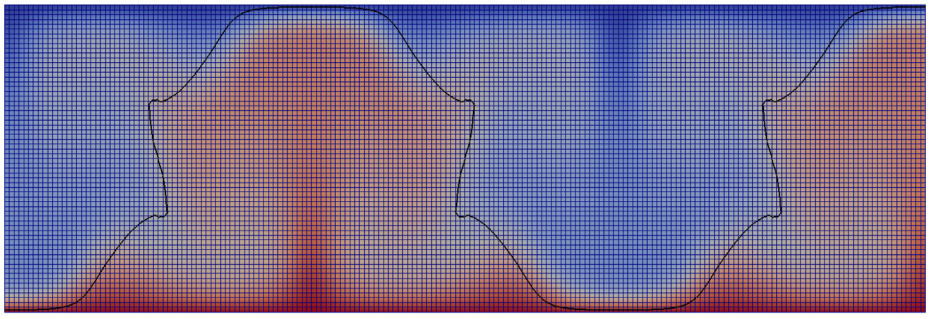
It is possible to obtain additional insight into the structure and dynamics of the flow for various values of B from the results shown in Figs. 21–32. As B increases from 0.0 to 0.1, 0.2, and 0.3 in Figs. 21–32 we observe that the rate of overturn decreases, until for B = 0.3 the denser material has just reached the top of the domain at $t' = 2.36 \cdot 10^{-2}$ (Fig. 24c and d), whereas for smaller values of B the overturn has passed beyond the top of the domain by $t' = 2.36 \cdot 10^{-2}$. For B = 0.4 we can see from Fig. 25c and d, that the fluid does not reach a full overturn by $t' = 2.36 \times 10^{-2}$ suggesting that there may be a transition between the qualitative dynamics of the flow at some B_c in the range $0.3 \leq B \leq 0.4$.



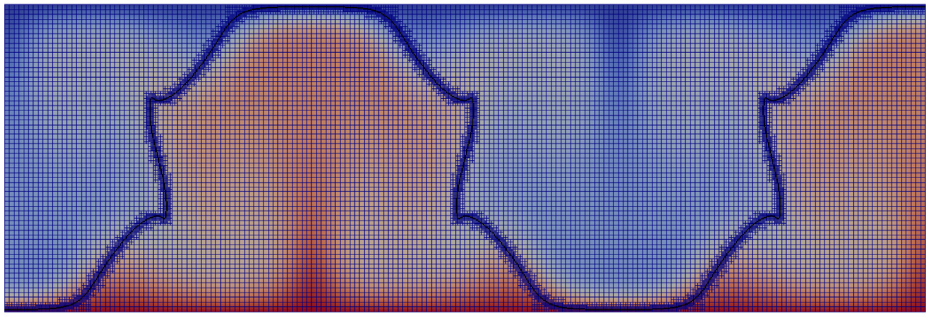
(a) $\mathrm{B} = 0.3$ at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) $\mathrm{B} = 0.3$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.3$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

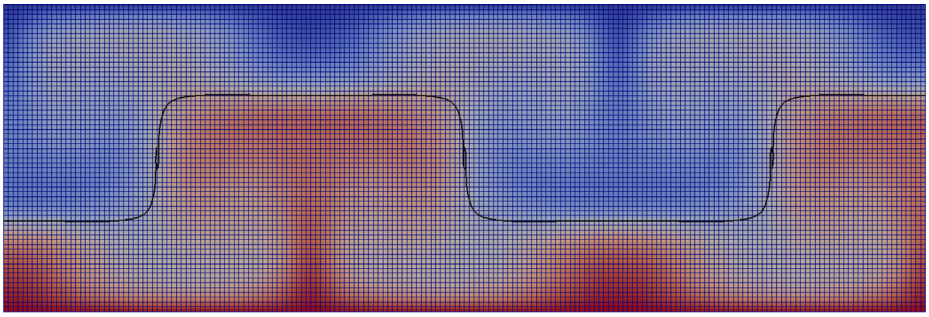


(d) $\mathrm{B} = 0.3$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

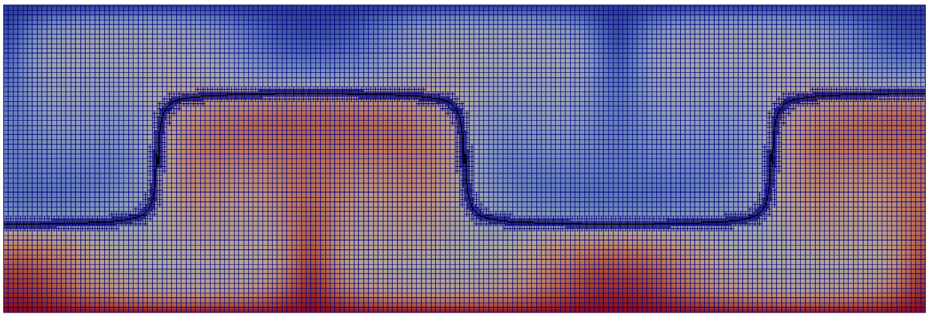
Fig. 24. Computations with $\mathrm{B} = 0.3$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In [8] the authors find $\mathrm{B}_c = 0.302$ when the viscosity ratio is $\gamma = 6.7$.

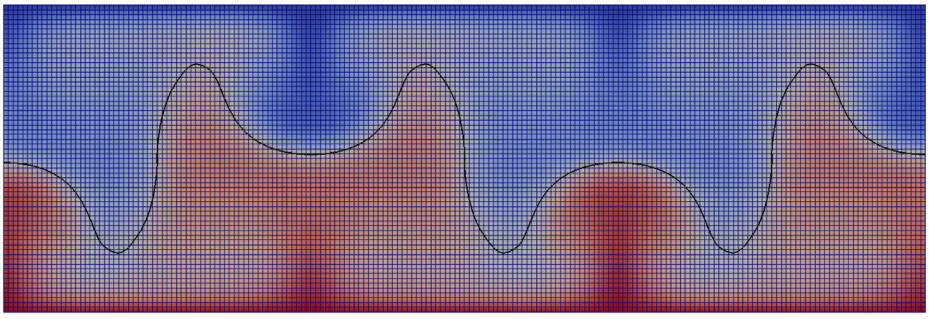
For $0.5 \leq \mathrm{B} \leq 1.0$ in Figs. 26–31 the general interface structures are similar, although with smaller volumes for the "pinched" regions that are produced during the transition from "Whole Layer" convection to "Stratified" flow. As shown in Fig. 32, for $\mathrm{B} = 2.0$, the stratification is sufficiently strong that the pinched structures do not form, although a standing wave does form as a slight perturbation from the initial location of the interface at $y = \frac{1}{2}$ with boundaries at $x \cong 0.5, 1.5, 2.5$.



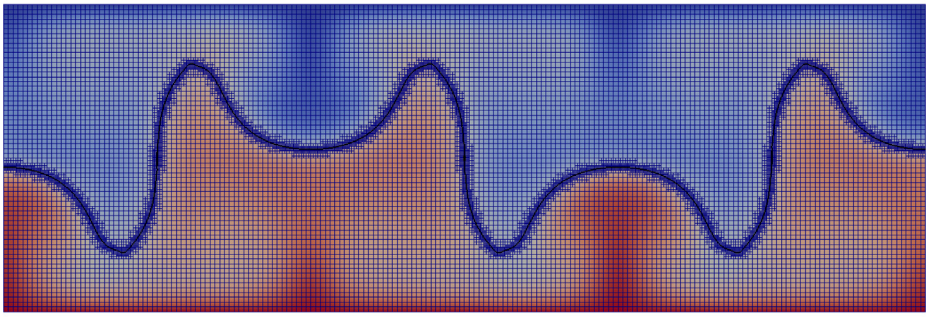
(a) B = 0.4 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 0.4 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.4$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

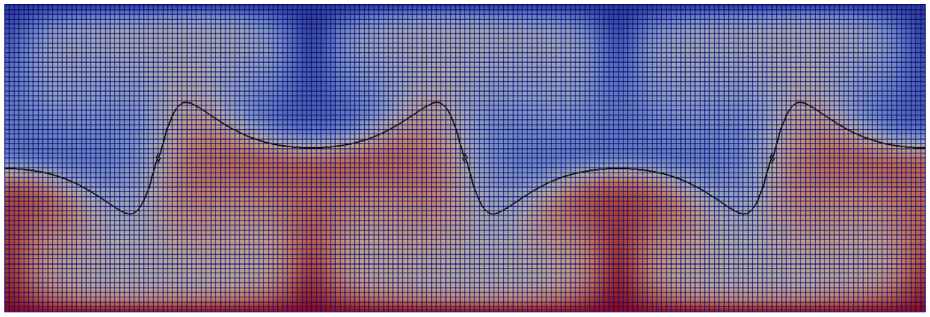


(d) B = 0.4 at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

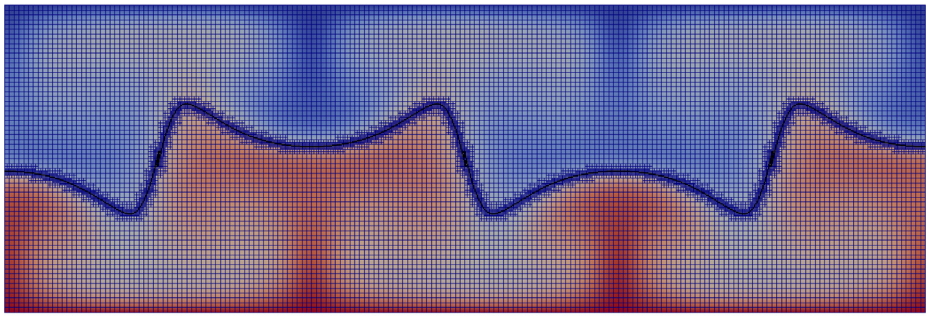
Fig. 25. Computations with B = 0.4 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1.1. A qualitative comparison to the experiments Davaille and Le Bars

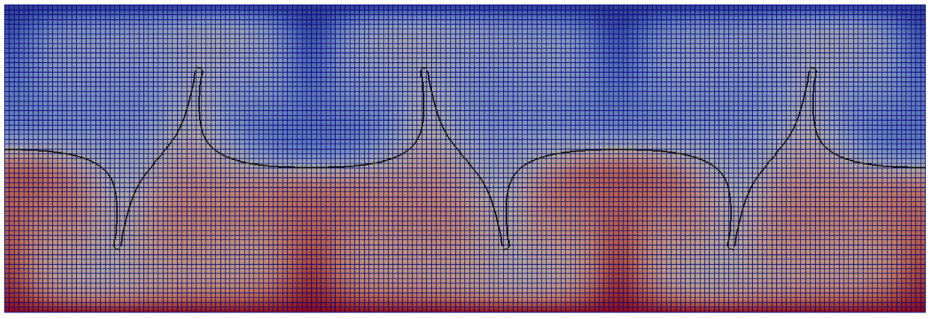
In this section we briefly make some additional *qualitative* comparisons of our computational results to the experimental results of Davaille [19] and Le Bars & Davaille [7,8]. Before doing so however, it is first necessary to make several caveats concerning this comparison. First, as we mentioned above, in the experiments the authors varied two additional nondimensional parameters; namely, (1) the ratio a of the height of the lower layer to the height of the entire domain and (2) the ratio γ of the viscosity of the fluid that initially occupies the lower layer to the viscosity that initially occupies the upper layer. In our computations, shown in Section 4.5, we kept these parameters fixed at $a = 0.5$ and $\gamma = 1.0$. Second, in the experiments the two fluids are *miscible*, whereas in



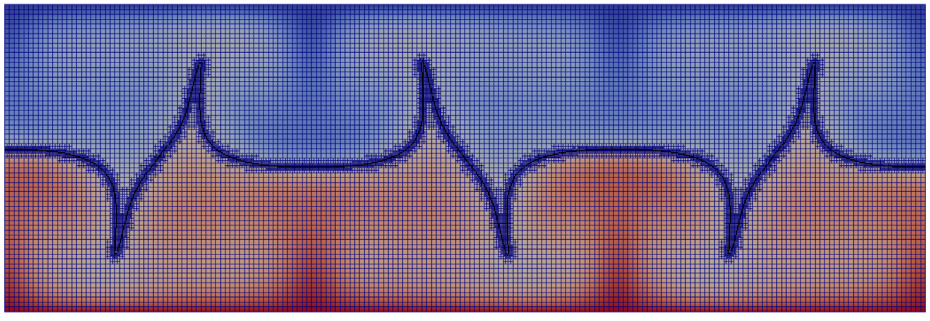
(a) $\mathrm{B} = 0.5$ at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) $\mathrm{B} = 0.5$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.5$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

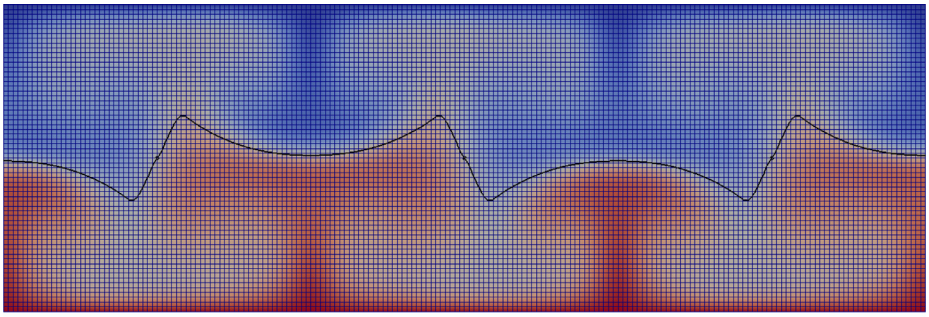


(d) $\mathrm{B} = 0.5$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

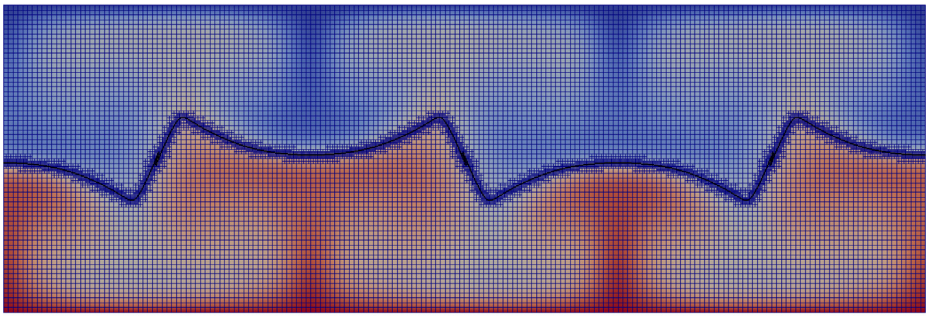
Fig. 26. Computations with $\mathrm{B} = 0.5$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

our computations the two fluids are *immiscible*. In both cases there is no surface tension at the boundary between the two fluids.

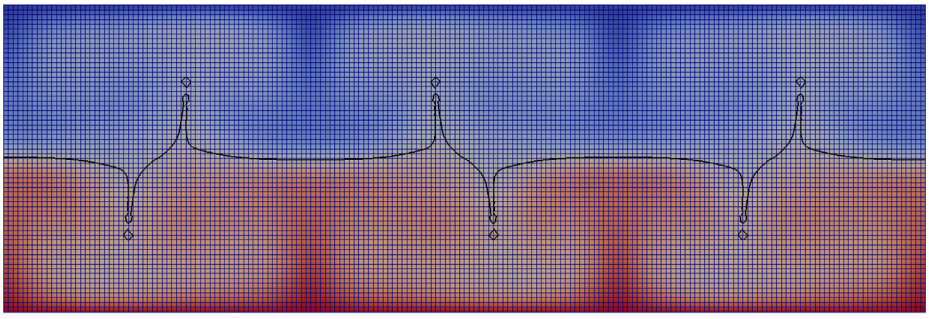
The general transition between one type of structure and another (e.g., "Whole Layer" convection to "Stratified Convection") is similar to that found in the experiments shown in [8], although the precise location of the transition may differ. A rough comparison is show in Fig. 33. The different grayscale backgrounds in Fig. 33 correspond to the grayscale regions in Fig. 3 of [19] and Fig. 2 of [8] for $a = 0.5$. In the results presented in this paper we do not continue the computation for a sufficiently long times to confirm that in the $0.3 \leq \mathrm{B} \leq 0.5$ regime the flow oscillates before beginning an overturn. However, the observed behavior does produce structures that match those described in [8] for the length of time for which we do have computational results. This difference may be in part due to the fact that in [8] the two fluids also vary in viscosity ratio γ, and Rayleigh number Ra.



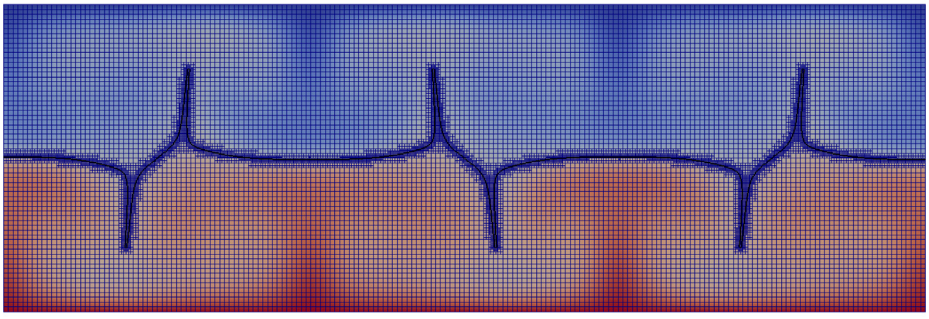
(a) B = 0.6 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 0.6 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.6$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

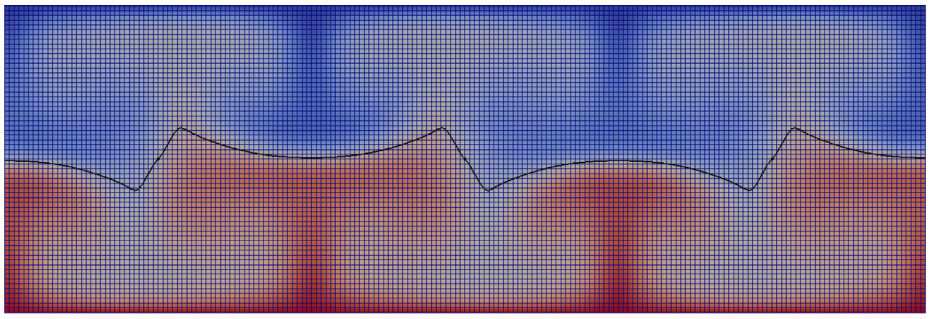


(d) B = 0.6 at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

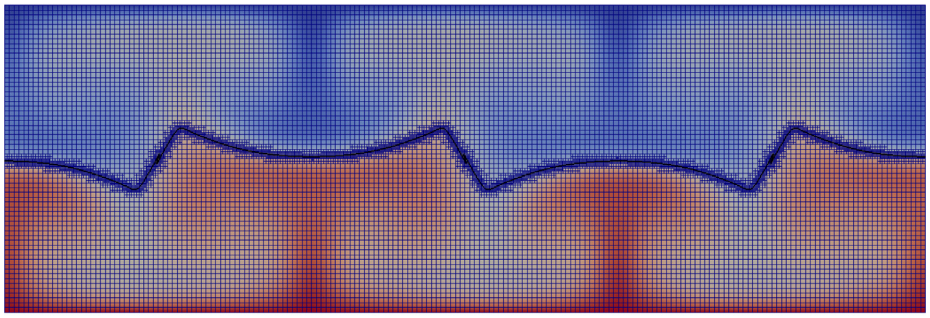
Fig. 27. Computations with B = 0.6 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1.2. Numerical artifacts that occur when the interface is underresolved

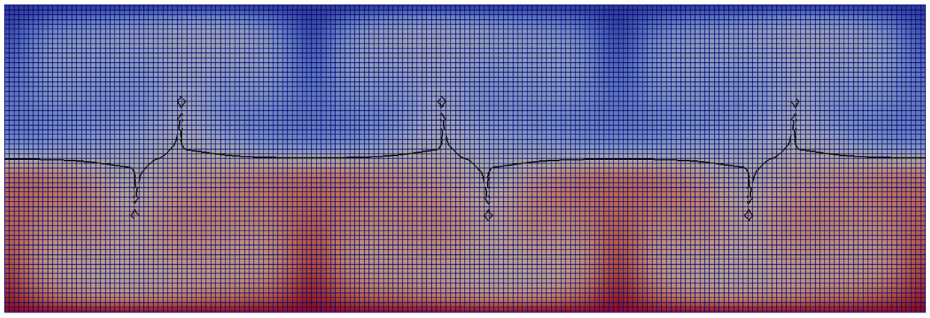
Since the VOF method maintains a sharp interface between the two compositional fields, it is able to capture features that are approximately on the order of the grid scale h. However, in cases where the structures formed by the interface become sufficiently small, for example, a thin column of fluid of width $2h$, the interface reconstruction algorithm might produce numerical artifacts that are "characteristic" of the combination of the particular reconstruction algorithm and advection algorithm one chooses to use in



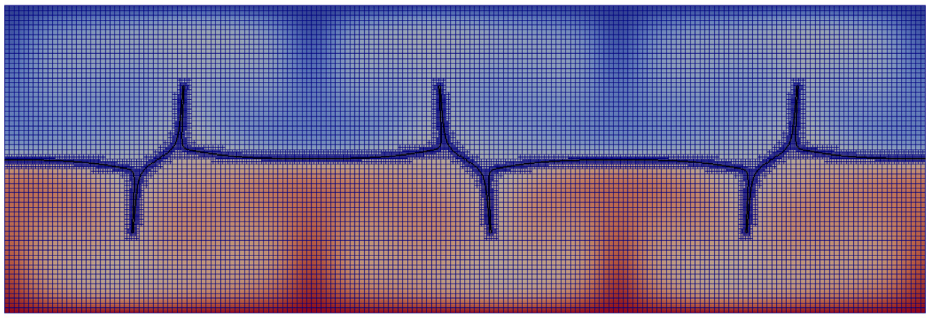
(a) $\mathrm{B} = 0.7$ at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) $\mathrm{B} = 0.7$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR

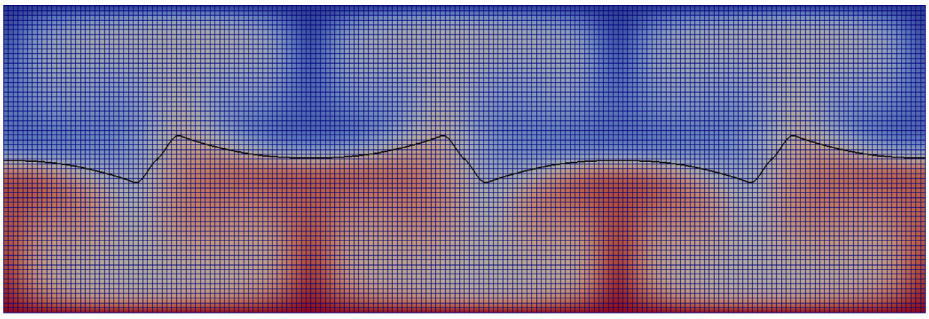


(c) $B = 0.7$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

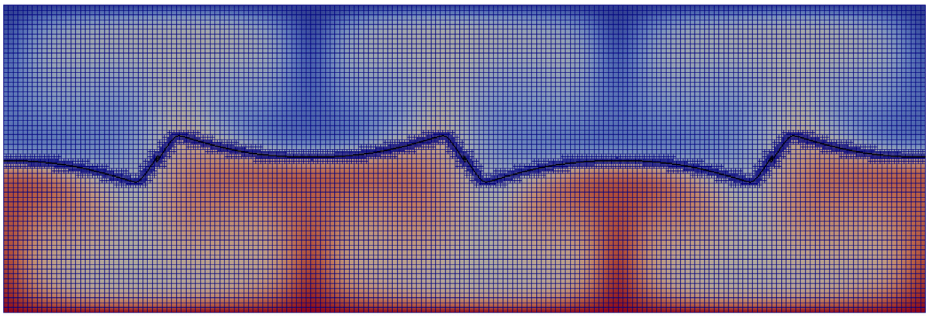


(d) $\mathrm{B} = 0.7$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

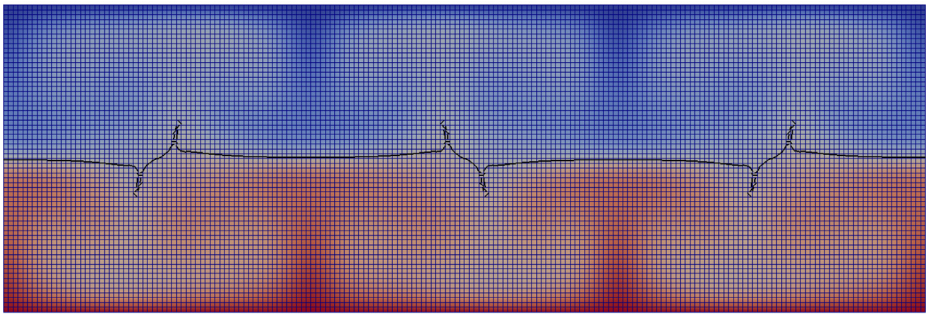
Fig. 28. Computations with $\mathrm{B} = 0.7$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



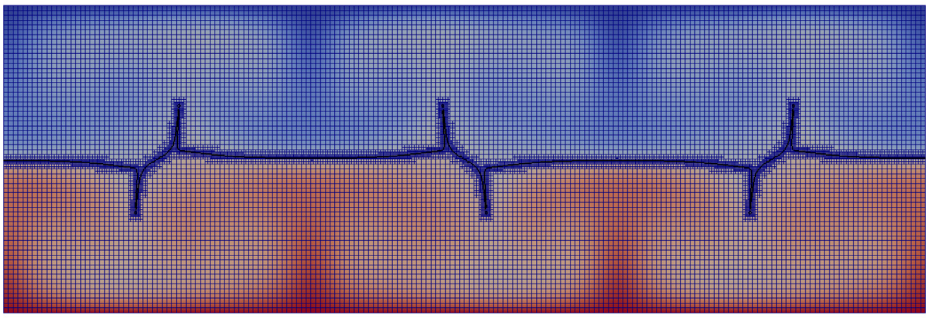
(a) B = 0.8 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 0.8 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR

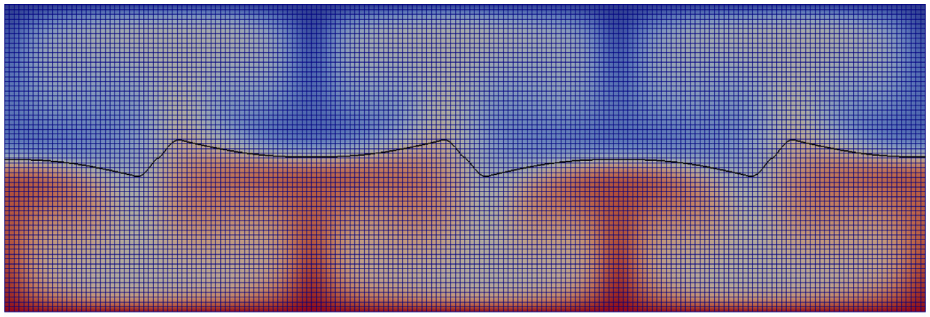


(c) $B = 0.8$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

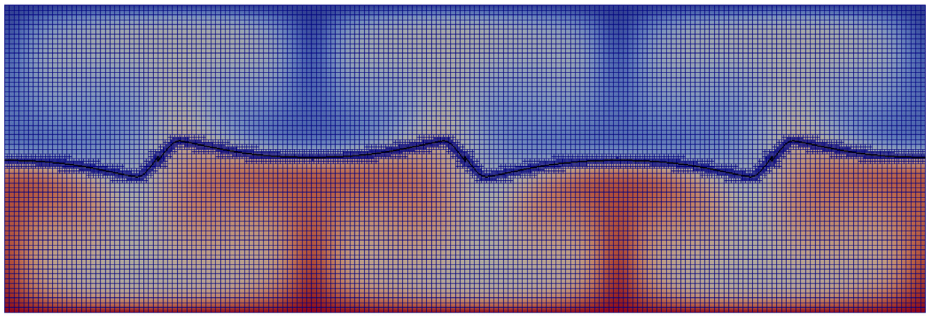


(d) B = 0.8 at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

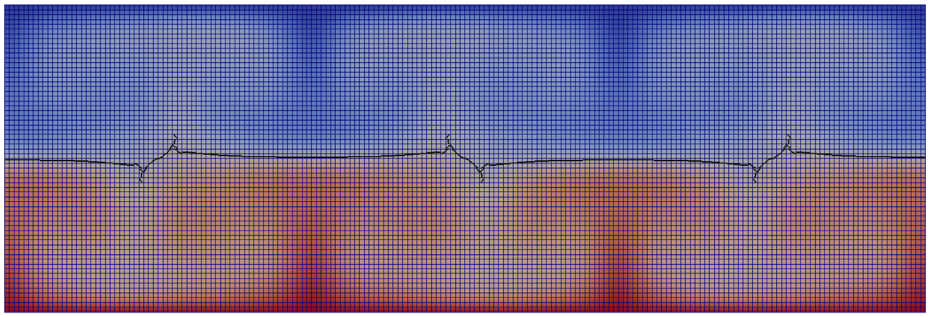
Fig. 29. Computations with B = 0.8 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



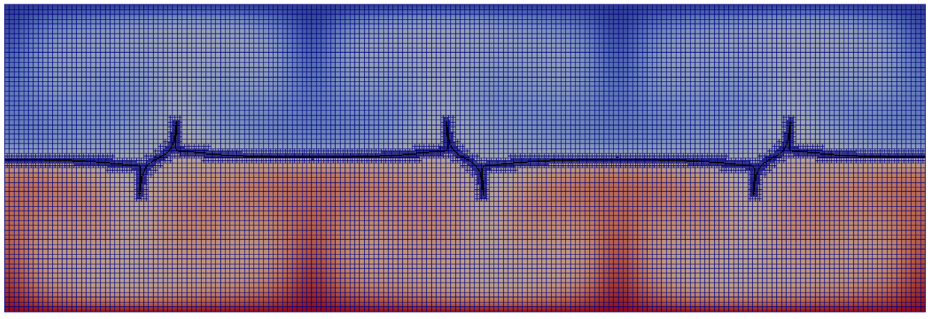
(a) $\mathrm{B} = 0.9$ at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) $\mathrm{B} = 0.9$ at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 0.9$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



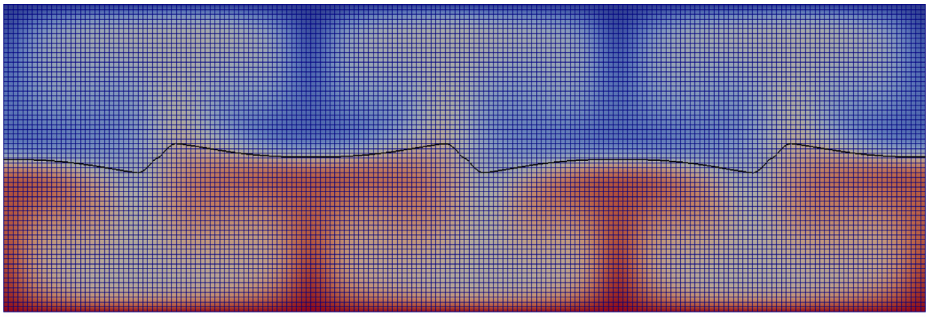
(d) $\mathrm{B} = 0.9$ at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

Fig. 30. Computations with $\mathrm{B} = 0.9$ and $\mathrm{Ra} = 10^5$ on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

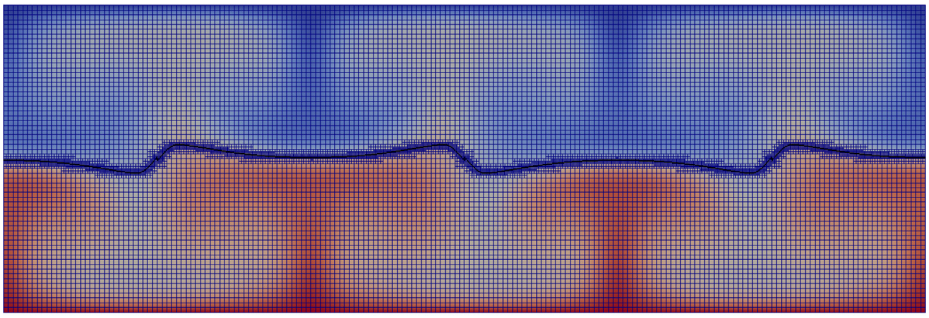
the VOF method.[7] Here we briefly examine of the nature of one particular numerical artifact that appears frequently in Section 4.5.

[7] It is important to recognize that this is not a failing of the VOF method in general or of the specific interface reconstruction and advection algorithms we have chosen for our work here, since whenever a computation is underresolved, *all* numerical methods will exhibit some sort of numerical artifact or artifacts that are "characteristic" of that particular method.

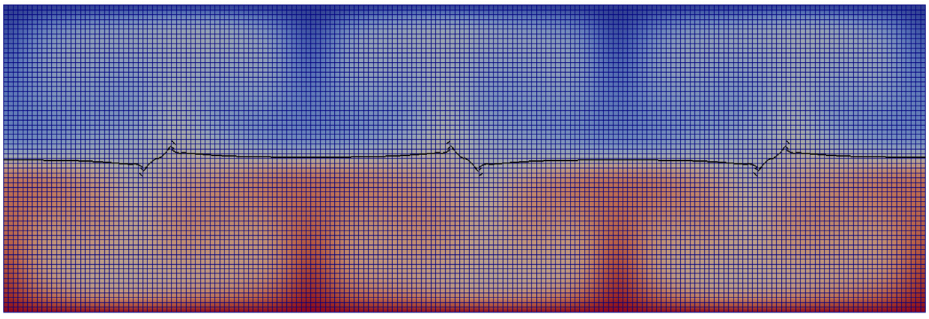
The most common numerical artifact in the computational results shown in Section 4.5 is the tendency for the reconstructed interface to form 'droplets' that are diamond shaped and generally occupy a square of four cells, each edge having two square cells of side h. For example, droplets such as these appear in Fig. 22a. In the computations shown in Section 4.5 these droplets typically resolve into a thin vertical column of fluid of approximately $2h - 4h$ in width with a length that is nearly the entire height of



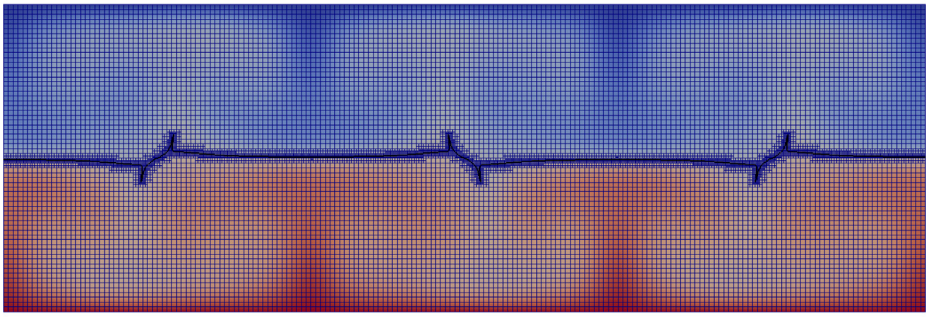
(a) B = 1.0 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 1.0 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) $B = 1.0$ at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.

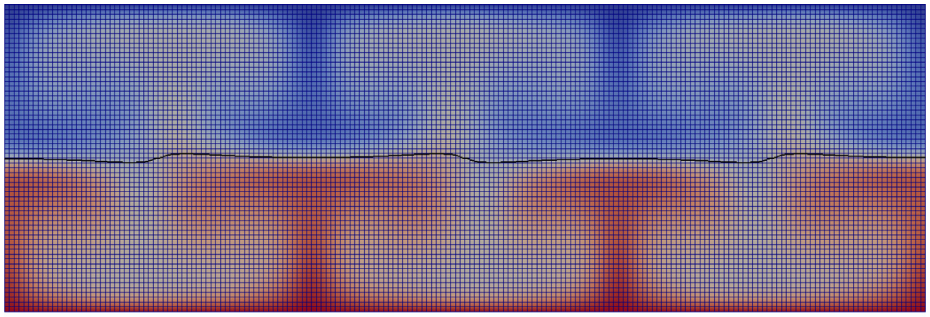


(d) B = 1.0 at $t' = 2.36 \cdot 10^{-2}$ with two additional levels of AMR

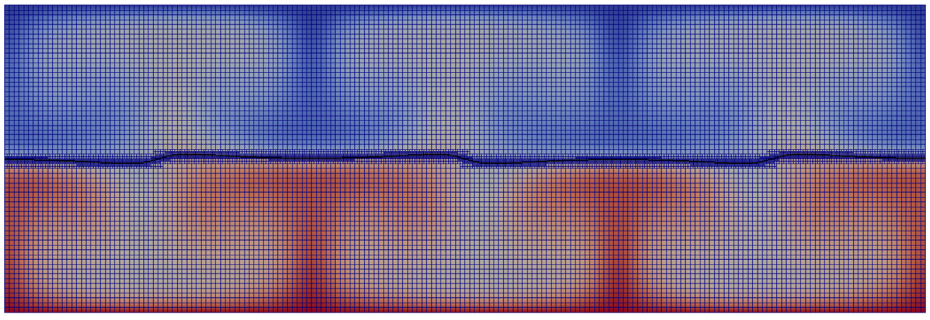
Fig. 31. Computations with B = 1.0 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the computational domain. For example Fig. 22a and b, in which the more refined computation in Fig. 22b appears to be sufficiently well-resolved to draw the conclusion that a thin column of fluid is forming in the locations where in Fig. 22a there are only a few droplets and no real indication of what the flow "should" look like. Or the droplets may resolve into a thin finger that is shorter than the height of the computational domain such as in Figs. 27c, d, 28c and d.

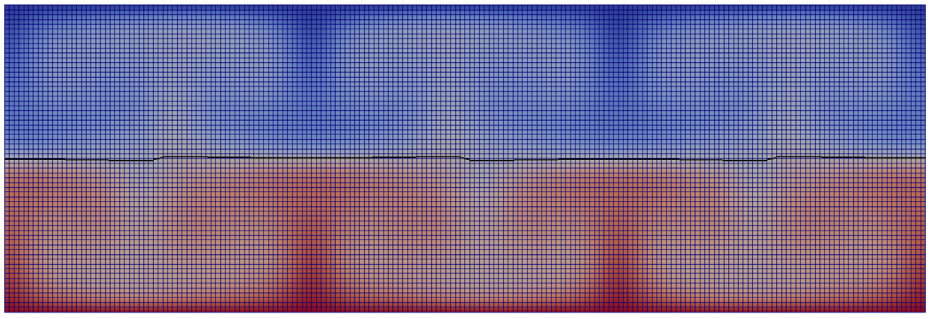
We note that if a feature of the interface is underresolved, it can help the user determine if additional refinement is required. In some instances, perhaps after making a second, more refined computation, it will be clear that additional refinement is necessary, sometimes even more refined than the second computation was. For example see Figs. 22c and d, neither of which appear sufficiently well resolved to accept the computation in Fig. 22d as well resolved enough to determine the true nature of the



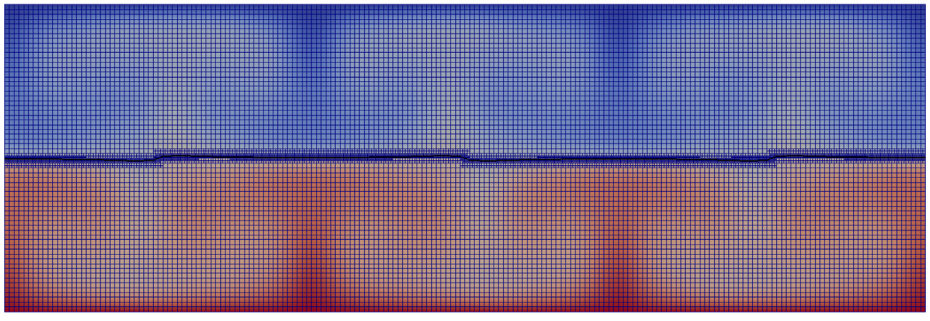
(a) B = 2.0 at $t' = 1.97 \cdot 10^{-2}$ on a uniform grid of 196×64 cells.



(b) B = 2.0 at $t' = 1.97 \cdot 10^{-2}$ with two additional levels of AMR



(c) B = 2.0 at $t' = 2.36 \cdot 10^{-2}$ on a uniform grid of 196×64 square cells.



(d) B = 2.0 at $t' = 2.36 \cdot 10^{-2}$ on a uniform 196×64 grid and two levels of AMR

Fig. 32. Computations with B = 2.0 and Ra = 10^5 on an underlying uniform grid of 196×64 square cells at $t' = 1.97 \times 10^{-2}$ and $t' = 2.36 \times 10^{-2}$. The background color is the temperature, which varies from $T = 0.0$ (dark blue) to $T = 1.0$ (dark red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

flow. On the other hand, there are instances when the numerical artifact is sufficiently small so as not to affect the dynamics of the interface that are of interest and additional resolution might not be required. For example, depending on the user and the underlying scientific application, this might be the case for Figs. 29c, d, 30 c and d, even though under magnification the fingers in the refined computations shown in Figs. 29d and 30 d do not yet appear fully resolved. In other words, depending on the application, these computations may or may not be well resolved enough for the user to arrive at conclusions *appropriate for their application* concerning the flow at this point in time.

In conclusion, we emphasize that the required degree of resolution for a given computation will depend on the purpose of the computation and the user's need for fine detail as opposed to general qualitative information concerning the flow.

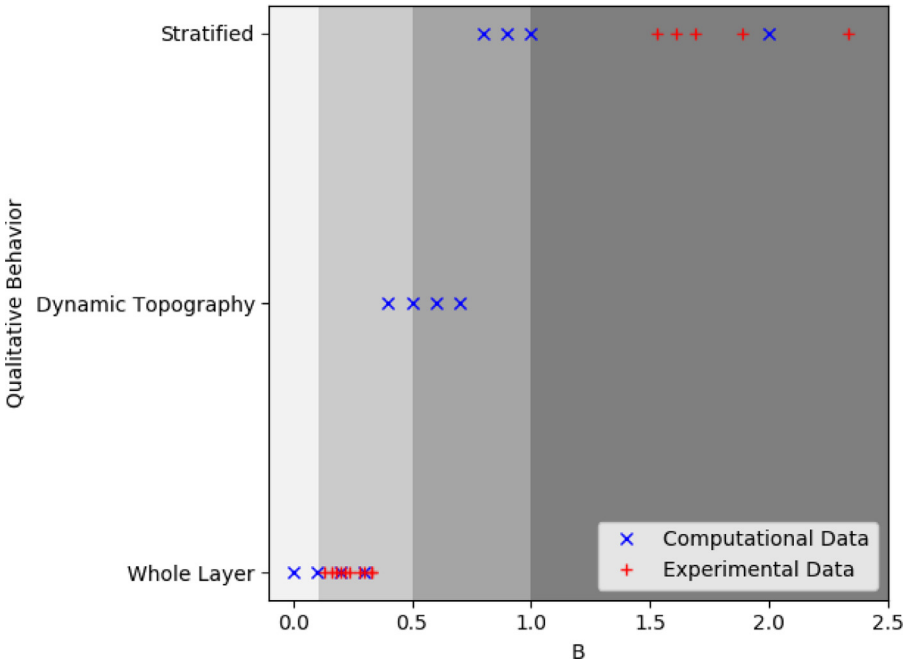


Fig. 33. A *qualitative* comparison of the computations presented in this paper to the experimental results of Davaille [19] and Le Bars & Davaille [7,8]. The grayscale regions correspond to boundaries of the qualitative regions shown in Fig. 3 of [19] and Fig. 2 of [8] for $a = 0.5$. The experimental data is from Table 3 of [19] and Table 3 of [7] with $a = 0.5$ as is the case for all of the computations in this article. The terms "Stratified", "Dynamic Topography", and "Whole Layer" used to describe the qualitative state of the flow are the same as those used by the authors of [7,8] and [19].

6. Conclusions

We have implemented a Volume-of-Fluid (VOF) interface tracking method in the open source finite element code ASPECT, which is designed to model convection and other processes in the Earth's mantle. We have demonstrated that this VOF method works efficiently and accurately in ASPECT's parallel environment and with its adaptive mesh refinement (AMR) algorithm.

In two simple benchmark computations we have demonstrated that the VOF method translates linear interfaces in a constant transitional flow to machine precision and is second-order accurate when we use it to compute the motion of a circular interface in solid body rotation. These results were as we expected.

We have also used the VOF method to compute the approximate solution of a 2D ball (disk) falling in an incompressible Stokes flow and demonstrated that the velocity and pressure converge at the expected second-order and one to one-and-a-half order rate, respectively, for a $Q_2 \times Q_1$ element combination while the volume fractions converge at a second-order rate. In addition, we compared the use of the VOF method to model the interface in this problem with a computation in which the problem is computed with a Bound Preserving Discontinuous Galerkin method and we obtain similar if not better results with the VOF method. We used this same test problem to demonstrate that the AMR strategy developed for the VOF field yields significantly greater computational efficiency, without any loss in the accuracy of the computed result.

We then demonstrated that the method shows excellent (visual) agreement with two standard benchmark problems from the computational mantle convection literature. In particular, in the second of these benchmarks we use AMR to allow us to compute at a much higher effective resolution at lower computational cost than would otherwise be possible.

Finally, we used the new interface tracking methodology to study a problem involving thermochemical convection in density stratified flow. This model problem is relevant to the study of structures at the core mantle boundary known as Large Low Shear Velocity Provinces (LLSVPs). Recent studies utilizing seismic imaging have revealed large regions with anomalous seismic properties in the lower mantle. There are two dome-like regions beneath Africa and the Pacific with low shear-wave velocities that extend some 1000 km above the core-mantle boundary and have horizontal dimensions of several thousand kilometers [15,21]. Most interpretations propose that the heterogeneities are compositional in nature, differing from the surrounding mantle, an interpretation that would be consistent with chemical geodynamic models. Based on geological and geochemical studies it has been argued that LLSVPs have persisted for billions of years [9].

The model problem is designed to study the basic physics underlying the formation of thermal plumes that bring some of this material to the Earth's surface. In our computations of we use AMR to obtain an effective grid resolution of 768×256 square cells overlaying the fluid interface on an underlying grid of 192×64 square cells. This increase in resolution confirms that for a certain range of the nondimensional buoyancy parameter B at Rayleigh number $\mathrm{Ra} = 10^5$ our computations of the interface have converged well enough to interpret with confidence the large scale dynamics of the two regions of differing densities.

In conclusion, the results of the work presented here demonstrate that our VOF interface tracking method should perform well on a number of problems of interest to the computational mantle convection community.

Acknowledgments

The authors would like to thank both reviewers for their comments and recommendations. Our paper has benefited greatly from their efforts.

This work was supported by the National Science Foundation's (NSF) SI2-SSE Program under Award number 1440811. The development of ASPECT was supported by the Computational Infrastructure for Geodynamics (CIG) under NSF Award numbers 0949446 and 1550901. The computations were made under the auspices of CIG on the U.C. Davis Division of Mathematical and Physical Sciences distributed computing cluster Peloton.

References

[1] Anbarlooei HR, Mazaheri K. Moment of fluid interface reconstruction method in axisymmetric coordinates. Int J Numer MethodsBiomed Eng 2011;27(10):1640–51. doi:10.1002/cnm.1426.

[2] Arndt D, Bangerth W, Davydov D, Heister T, Heltai L, Kronbichler M, et al. The `deal.II` library, version 8.5. J Numer Math 2017;25(3):137–46. doi:10.1515/jnma-2016-1045.

[3] Arndt D, Bangerth W, Davydov D, Heister T, Heltai L, Kronbichler M, et al. The `deal.II` library, version 8.5. J Numer Math 2017. doi:10.1515/jnma-2016-1045.

[4] Aulisa E, Manservisi S, Scardovelli R, Zaleski S. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. J Comput Phys 2007;225(2):2301–19. doi:10.1016/j.jcp.2007.03.015.

[5] Bangerth W., Dannberg J., Gassmöller R., Heister T., et al. ASPECT: advanced solver for problems in Earth's convection user manual. Computational Infrastructure for Geodynamics; 2019. https://geodynamics.org/cig/software/aspect/aspect-manual.pdf.

[6] Bangerth W, Hartmann R, Kanschat G. deal.II – a general purpose object oriented finite element library. ACM Trans Math Softw 2007;33(4) 24/1-24/27.

[7] Bars ML, Davaille A. Large interface deformation in two-layer thermal convection of miscible viscous fluids. J Fluid Mech 2004;499:75110. doi:10.1017/S0022112003006931.

[8] Bars ML, Davaille A. Thermochemical convection in two superimposed miscible viscous fluids. In: Gutkowski W, Kowalewski T, editors. Mechanics of the 21st century, Proceedings of the 21st international congress of theoretical and applied mechanics. Springer Verlag; 2005. FM7–12126.

[9] Burke K, Steinberger B, Torsvik TH, Smethurst MA. Plume generation zones at the margins of large low shear velocity provinces on the core–mantle boundary. Earth Planet Sci Lett 2008;265(1):49–60.

[10] Burstedde C, Wilcox LC, Ghattas O. `p4est`: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. SIAM J Sci Comput 2011;33(3):1103–33. doi:10.1137/100791634.

[11] Chandrasekhar S. Hydrodynamic and hydromagnetic stability. New York: Dover; 1961.

[12] Chorin AJ. Curvature and solidification. J Comput Phys 1985;57:472–90.

[13] Chorin AJ, Marsden JE. A mathematical introduction to fluid mechanics. No. 4. Texts in applied mathematics. 4th. New York: Springer-Verlag; 1993. QA901.C53 1992.

[14] Colella P. Multidimensional upwind methods for hyperbolic conservation laws. J Comput Phys 1990;87:171–200.

[15] Cottaar S, Romanowicz B. An unusually large ULVZ at the base of the mantle near Hawaii. Earth Planet Sci Lett 2012;355:213–22.

[16] Courant R, Friedrichs KO, Lewy H. On the partial difference equations of mathematical physics. IBM J Res Dev 1967;11(2):215–34. (English translation of the orignal work, "Uber die Partielien Differenzenglrichungen der Mathematischen Physik", Math. Ann. 100, 32–74 (1928)).

[17] Dannberg J, Eilon Z, Faul U, Gassmöller R, Moulik P, Myhill R. The importance of grain size to mantle dynamics and seismological observations. Geochem Geophy Geosy 2017;18(8):3034–61. doi:10.1002/2017GC006944.

[18] Dannberg J, Heister T. Compressible magma/mantle dynamics: 3D, adaptive simulations in ASPECT. Geophys J Int 2016;207(3):1343–66. doi:10.1093/gji/ggw329.

[19] Davaille A. Two-layer thermal convection in miscible viscous fluids. J Fluid Mech 1999;379:223253. doi:10.1017/S0022112098003322.

[20] Donea J, Huerta A. Steady Transport Problems. John Wiley and Sons; 2005. doi:10.1002/0470013826.ch2.

[21] French SW, Romanowicz B. Broad plumes rooted at the base of the Earth's mantle beneath major hotspots. Nature 2015;525(7567):95–9.

[22] Gassmöller R, Lokavarapu H, Heien E, Puckett EG, Bangerth W. Flexible and scalable particle-in-cell methods with adaptive mesh refinement for geodynamic computations. Geochem Geophy Geosy 2018;19:472–90. doi:10.1029/2018GC007508.

[23] Gerya TV, Yuen DA. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties. Phys Earth Planet In 2003;140(4):293–318. doi:10.1016/j.pepi.2003.09.006.

[24] Guermond J-L, Pasquetti R, Popov B. Entropy viscosity method for nonlinear conservation laws. J Comput Phys 2011;230(11):4248–67. Special issue High Order Methods for CFD Problems.

[25] Hager B, O'Connell R. A simple global model of plate dynamics and mantle convection. J Geophys Res-Sol Ea 1981;86(B6):4843–67. doi:10.1029/JB086iB06p04843.

[26] Hager BH, Clayton RW. Constraints on the structure of mantle convection using seismic observations, flow models, and the geoid. In: Mantle convection, plate tectonics and global dynamics. Gordon and Breach Science Publishers; 1989. p. 657–763.

[27] He Y, Billen MI, Puckett EG, Kellogg LH. A Discontinuous Galerkin method for solving time-dependent convection-diffusion temperature equations : Demonstration and comparison with other methods in the mantle convection code ASPECT; 2016. Talk given at the annual AGU Fall Meeting.

[28] He Y, Puckett EG, Billen MI. A discontinuous Galerkin method with a bound preserving limiter for the advection of non-diffusive fields in solid Earth geodynamics. Phys Earth Planet In 2017;263:23–37. doi:10.1016/j.pepi.2016.12.001.

[29] Heister T, Dannberg J, Gassmöller R, Bangerth W. High accuracy mantle convection simulation through modern numerical methods. II: realistic models and problems. Geophys J Int 2017;210(2):833–51. doi:10.1093/gji/ggx195.

[30] Helmsen JJ, Colella P, Puckett EG. Non-convex profile evolution in two dimensions using volume of fluids. Technical Report LBNL-40693. Lawrence Berkeley National Laboratory; 1997.

[31] Helmsen JJ, Colella P, Puckett EG, Dorr M. Two new methods for simulating photolithography development in three dimensions. In: Proceeedings of the 10th SPIE Optical/Laser Microlithography Conference, 2726. San Jose, CA: SPIE; 1996. p. 253–61.

[32] Henderson LF, Colella P, Puckett EG. On the refraction of shock waves at a slowfast gas interface. J Fluid Mech 1991;224:1–27. doi:10.1017/S0022112091001623.

[33] Hill RN, Shashkov M. The symmetric moment-of-fluid interface reconstruction algorithm. J Comput Phys 2013;249:180–4. doi:10.1016/j.jcp.2013.04.037.

[34] Hirt CW, Nichols BD. Volume of Fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 1981;39:201–25.

[35] Huber R, Helmig R. Multiphase flow in heterogeneous porous media: a classical finite element method versus an implicit pressure–explicit saturation-based mixed finite element–finite volume approach. Int J Numer Meth Fl 1999;29(8):899–920.

[36] Jemison M, Sussman M, Shashkov M. Filament capturing with the multimaterial moment-of-fluid method. J Comput Phys 2015;285:149–72. doi:10.1016/j.jcp.2015.01.014.

[37] John F. Partial differential equations. Applied Mathematical Sciences, 1. 4th. New York: Springer-Verlag; 1978.

[38] King S, Raefsky A, Hager B. ConMan: vectorizing a finite element code for incompressible two-dimensional convection in the Earth's mantle. Phys Earth Planet In 1990;59(3):195–207. doi:10.1016/0031-9201(90)90225-M.

[39] Korzekwa DR, Kothe DB, Lam KL, Puckett EG, Tubesing PK, Williams MW. A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-d unstructured meshes. In: Proceedings of the 3rd ASME /JSME joint fluids engineering conference. San Francisco, CA: American Society of Mechanical Engineers; 1999. p. 1–6.

[40] Kronbichler M, Bangerth W. Numerical experiments to determine optimal parameters; 2011. Reference documentation for deal.II, The step-31 tutorial program.

[41] Kronbichler M, Heister T, Bangerth W. High accuracy mantle convection simulation through modern numerical methods. Geophys J Int 2012;191(1):12–29. doi:10.1111/j.1365-246X.2012.05609.x.

[42] Lax PD. Hyperbolic difference equations: A Review of the Courant–Friedrichs–Lewy paper in light of recent developments. IBM J Res Dev 1967;11(2):235–8.

[43] LeVeque RJ. High-resolution conservative algorithms for advection in incompressible flow. SIAM J Numer Anal 1996;33(2):627–65.

[44] López J, Hernández J, Gómez P, Faura F. A new volume conservation enforcement method for PLIC reconstruction in general convex grids. J Comput Phys 2016;316:338–59. doi:10.1016/j.jcp.2016.04.018.

[45] Manga M. Mixing of heterogeneities in the mantle: effect of viscosity differences. Geophys Res Lett 1996;23(4):403–6. doi:10.1029/96GL00242.

[46] Manga M, Stone H. Interactions between bubbles in magmas and lavas: effects of bubble deformation. J Volcanol Geotherm Res 1994;63(3):267–79. doi:10.1016/0377-0273(94)90079-5.

[47] Manga M, Stone HA, O'Connell RJ. The interaction of plume heads with compositional discontinuities in the Earth's mantle. J Geophys Res-Sol Ea 1993;98(B11):19979–90. doi:10.1029/93JB00441.

[48] McNamara A, Zhong S. Thermochemical structures within a spherical mantle: superplumes or piles?: thermochemical structures. J Geophys Res-Sol Ea 2004;109(B7). doi:10.1029/2003JB002847.

[49] Miller GH, Puckett EG. Edge effects in molybdenum-encapsulated molten silicate shock wave targets. J Appl Phys 1994;75(3):1426–34.

[50] Miller GH, Puckett EG. A high-order Godunov method for multiple condensed phases. J Comput Phys 1996;128(1):134–64.

[51] Moresi L, Gurnis M. Constraints on the lateral strength of slabs from three--dimensional dynamic flow models. Earth Planet Sci Lett 1996;138(1):15–28.

[52] Nichols BD, Hirt CW, Hotchkiss RS. SOLA-VOF: a solution algorithm for transient fluid flow with multiple free boundaries. Technical Report LA-8355. Los Alamos National Laboratory; 1980.

[53] Noh WF, Woodward PR. SLIC (Simple Line Interface Calculation). In: van de Vooren AI, Zandbergen PJ, editors. Proceedings of the fifth international conference on numerical methods in fluid dynamics. Lecture notes in physics, 59. Twente University, Enschede: Springer-Verlag; 1976. p. 330–40.

[54] Pilliod JE. An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods. Graduate Group in Applied Mathematics, University of California, Davis; 1992. MS Thesis.

[55] Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. Technical Report LBNL-40745. Lawrence Berkeley National Laboratory; 1997.

[56] Pilliod JE, Puckett EG. An unsplit, second-order accurate Godunov method for tracking deflagrations and detonations. In: Houwing AFP, Paull A, Boyce RR, Danehy PM, Hannemann H, Kurtz JJ, et al., editors. Proceedings of the 21st international symposium on shock waves, II. Fyshwick, Australia: Panther Publishing; 1998. p. 1053–8.

[57] Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. J Comput Phys 2004;199(2):465–502.

[58] Puckett EG. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In: Proceedings of the fourth international symposium on computational fluid dynamics; 1991. p. 933–8.
[59] Puckett EG. On the second-order accuracy of volume-of-fluid interface reconstruction algorithms: convergence in the max norm. CAMCoS 2010;5(1):99–148.
[60] Puckett EG. A volume-of-fluid interface reconstruction algorithm that is second-order accurate in the max norm. CAMCoS 2010;5(2):199–220.
[61] Puckett EG. On the second-order accuracy of volume-of-fluid interface reconstruction algorithms II: an improved constraint on the cell size. CAMCoS 2014;8(1):123–58.
[62] Puckett EG, Almgren AS, Bell JB, Marcus DL, Rider WJ. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. J Comput Phys 1997;130(2):269–82.
[63] Puckett EG, Miller GH. The numerical computation of jetting impacts. In: Sturtevant B, Shepherd JE, Hornung H, editors. Proceedings of the 20th international symposium on shock waves, II. New Jersey: World Scientific; 1996. p. 1467–72.
[64] Puckett EG, Turcotte DL, He Y, Lokavarapu H, Robey JM, Kellogg LH. New numerical approaches for modeling thermochemical convection in a compositionally stratified fluid. Phys Earth Planet Inter 2018;276:10–35. doi:10.1016/j.pepi.2017.10.004. Special Issue: 15th SEDI Conference.
[65] Rider WJ, Kothe DB. Reconstructing volume tracking. J Comput Phys 1998;141(2):112–52. doi:10.1006/jcph.1998.5906.
[66] Samuel H, Evonuk M. Modeling advection in geophysical flows with particle level sets. Geochem Geophy Geosy 2010;11(8). doi:10.1029/2010GC003081.
[67] Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. J Comput Phys 2000;164(1):228–37. doi:10.1006/jcph.2000.6567.
[68] Scardovelli R, Zaleski S. Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. Int J Numer Meth Fl 2003;41(3):251–74. doi:10.1002/fld.431.
[69] Schubert G, Turcotte DL, Olson P. Mantle convection in the Earth and planets. Cambridge University Press; 2001.
[70] Sheldon J, Cardwell Jr W, et al. One-dimensional, incompressible, noncapillary, two-phase fluid flow in a porous medium. Petrol Trans, AIME 1959;216:290–6.
[71] Steinberger B. Plumes in a convecting mantle: models and observations for individual hotspots. J Geophys Res-Sol Ea 2000;105(B5):11127–52.
[72] Strang WG. On the construction and comparison of difference schemes. SIAM J Numer Anal 1968;5(3):506–17. doi:10.1137/0705041.
[73] Strang WG. Introduction to linear algebra. 5th. Cambridge Wellsely Press; 2016.
[74] Sussman MS, Puckett EG. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. J Comput Phys 2000;162:301–37.
[75] Tan E, Choi E, Thoutireddy P, Gurnis M, Aivazis M. Geoframework: coupling multiple models of mantle convection within a computational framework: geoframework-mantle convection models. Geochem Geophy Geosy 2006;7(6). doi:10.1029/2005GC001155.
[76] Torrey MD, Cloutman LD, Mjolsness RC, Hirt CW. NASA-VOF2D: a computer program for incompressible flows with free surfaces. Technical Report LA-10612-MS. Los Alamos National Laboratory; 1985.
[77] Torrey MD, Mjolsness RC, Stein LR. NASA-VOF3D: a three-dimensonal computer program for incompressible flows with free surfaces. Technical Report LA-11009-MS. Los Alamos National Laboratory; 1987.
[78] Turcotte DL, Schubert G. Geodynamics. third. Cambridge University Press; 2014.
[79] van Keken PE, King SD, Schmeling H, Christensen UR, Neumeister D, Doin M-P. A comparison of methods for the modeling of thermochemical convection. J Geophys Res-Sol Ea 1997;102(B10):22477–95. doi:10.1029/97JB01353.
[80] Wanner G, Hairer E. Solving ordinary differential equations II. Springer series in computational mathematics, 14. Springer-Verlag Berlin Heidelberg; 1991.
[81] Weymouth G, Yue DK-P. Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids. J Comput Phys 2010;229(8):2853–65. doi:10.1016/j.jcp.2009.12.018.
[82] Williams MW, Kothe DB, Puckett EG. Approximating interfacial topologies with applications for interface tracking algorithms. In: Proceedings of the 37th Aerospace Sciences Meeting and Exhibit. Reno, NV: American Institute of Aeronautics and Astronautics; 1999. p. 1–9. doi:10.2514/6.1999-1076. AIAA-99-1076.
[83] Williams MW, Kothe DB, Puckett EG. Robust finite volume modeling of 3-D free surface flows on unstructured meshes. In: Proceedings of the 14th AIAA computational fluid dynamics conference. Norfolk, VA: American Institute of Aeronautics and Astronautics; 1999. p. 1–6. doi:10.2514/6.1999-3320. A99-993567.
[84] Zhong S. Constraints on thermochemical convection of the mantle from plume heat flux, plume excess temperature, and upper mantle temperature. J Geophys Res 2006;111(B4). doi:10.1029/2005JB003972.
[85] Zhong S, Zuber M, Moresi L, Gurnis M. Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection. J Geophys Res-Sol Ea 2000;105(B5):11063–82. doi:10.1029/2000JB900003.