Title

New numerical approaches for modeling thermochemical convection in a compositionally stratified fluid

Permalink

https://escholarship.org/uc/item/2d10b4ns

Authors

Puckett, Elbridge Gerry
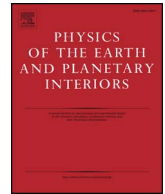Turcotte, Donald L
He, Ying
et al.

Publication Date

2018-03-01

DOI

10.1016/j.pepi.2017.10.004

Peer reviewed

New numerical approaches for modeling thermochemical convection in a compositionally stratified fluid

Elbridge Gerry Puckett[a,*], Donald L. Turcotte[b], Ying He[a], Harsha Lokavarapu[b], Jonathan M. Robey[a], Louise H. Kellogg[b,c]

[a] Department of Mathematics, U. C. Davis, Davis, CA 95616, USA
[b] Department of Earth and Planetary Sciences, U. C. Davis, Davis, CA 95616, USA
[c] Computational Infrastructure for Geodynamics, U. C. Davis, Davis, CA 95616, USA

ARTICLE INFO

ABSTRACT

Geochemical observations of mantle-derived rocks favor a nearly homogeneous upper mantle, the source of mid-ocean ridge basalts (MORB), and heterogeneous lower mantle regions. Plumes that generate ocean island basalts are thought to sample the lower mantle regions and exhibit more heterogeneity than MORB. These regions have been associated with lower mantle structures known as large low shear velocity provinces (LLSVPS) below Africa and the South Pacific. The isolation of these regions is attributed to compositional differences and density stratification that, consequently, have been the subject of computational and laboratory modeling designed to determine the parameter regime in which layering is stable and understanding how layering evolves. Mathematical models of persistent compositional interfaces in the Earth's mantle may be inherently unstable, at least in some regions of the parameter space relevant to the mantle. Computing approximations to solutions of such problems presents severe challenges, even to state-of-the-art numerical methods. Some numerical algorithms for modeling the interface between distinct compositions smear the interface at the boundary between compositions, such as methods that add numerical diffusion or 'artificial viscosity' in order to stabilize the algorithm. We present two new algorithms for maintaining high-resolution and sharp computational boundaries in computations of these types of problems: a discontinuous Galerkin method with a bound preserving limiter and a Volume-of-Fluid interface tracking algorithm. We compare these new methods with two approaches widely used for modeling the advection of two distinct thermally driven compositional fields in mantle convection computations: a high-order accurate finite element advection algorithm with entropy viscosity and a particle method that carries a scalar quantity representing the location of each compositional field. All four algorithms are implemented in the open source finite element code ASPECT, which we use to compute the velocity, pressure, and temperature associated with the underlying flow field. We compare the performance of these four algorithms on three problems, including computing an approximation to the solution of an initially compositionally stratified fluid at Ra = 10^5 with buoyancy numbers B that vary from no stratification at B = 0 to stratified flow at large B.

1. Introduction

A major unresolved question in geodynamics is whether mantle convection consists of a single layer or two separate layers. Seismic studies have shown deep penetration of subducted lithosphere which favors whole mantle convection. However isotopic studies provide strong evidence for a well mixed upper mantle but a lower mantle reservoir containing primordial material and unmixed subducted material. The near uniform isotopic ratios of most mid-ocean ridge basalts indicate that this volcanism is sampling a near homogeneous upper-mantle reservoir. The upper mantle composition is complementary to the composition of the continental crust. Depleted subducted oceanic lithosphere has been mixed in the upper mantle by mantle convection (Kellogg, 1992). In contrast ocean island basalts have heterogeneous isotopic ratios. These basalts are attributed to mantle plumes which have a deep mantle origin. The source region includes primordial material as well as a variety of subducted materials that have not been mixed into the mantle. The isolated lower mantle is attributed to chemical layering leading to a density difference that inhibits mixing (Kellogg et al., 1999; Turcotte and Schubert, 2014).

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 extending some 1000 km above the core-mantle boundary and horizontal dimensions of several thousand kilometers (Cottaar and Romanowicz, 2012; French and Romanowicz, 2015). These are known as large low shear-velocity provinces (LLSVPs). 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 studies it has been argued that LLSVPs have persisted for billions of years (Burke et al., 2008).

Mantle convection with compositional differences, together with a thermal contribution, is known as thermal-chemical convection. Tackley (2012) has given a comprehensive review of the influence of compositional buoyancy on mantle convection both in terms of evidence for buoyancy structures in the mantle and models of the influence of compositional buoyancy on thermal-chemical convection. Thermal-chemical convection with applications to the mantle requires distinct regions with different compositions and densities. Molecular diffusion that mixes regions with different compositions can occur only on very small scales in the mantle on geological time scales, centimeters to meters. Thus, mixing is predominantly kinematic with sharp compositional interfaces (Kellogg, 1992).

A number of studies of the role of compositional discontinuities on mantle convection have been carried out. Numerical studies include Montague and Kellogg (2000), McNamara and Zhong (2004), Tan and Gurnis (2005) and Galsa et al. (2015). Davaille (1999) has carried out extensive laboratory studies.

Computational modeling of persistent compositional interfaces presents severe challenges to numerical methods. Many numerical algorithms for modeling distinct compositional regions in the mantle do not maintain sharp boundaries between distinct compositions as they migrate and distort. In particular, some of these algorithms exhibit compositional diffusion, either as an unwanted *numerical artifact* of the algorithm or as an *artificial diffusion*, which is an explicit design feature of the algorithm, or both. Artificial diffusion, which is typically referred to as *artificial viscosity* (Von and Richtmyer, 1950), is a numerical quantity that is added to the numerical algorithm for modeling the advection of some quantity, such as the compositional variable C, in order to maintain the upper and lower bounds on the value of that variable, and also, at least for a Finite Element Method (FEM), to maintain the stability of the underlying numerical advection method (Kronbichler et al., 2012). This numerical technique is described in more detail in Section 3.5 below.

Note that, in the context of the present work, we prefer to use the term 'artificial diffusion' rather than 'artificial viscosity'. Numerical algorithms that allow the compositional boundary to diffuse, whether as a numerical artifact or as an intentional part of the advection method are inherently in conflict with the fact that the true compositional boundary must remain sharp for an indefinite period of time. Furthermore, these algorithms must necessarily limit the accuracy of the advection method to second-order in the grid-size h.

In this paper we examine four alternative algorithms for numerically modeling this compositional discontinuity. All of these algorithms are implemented in the open source mantle convection code ASPECT, which is described in detail in Kronbichler et al. (2012) and Heister et al. (2017). We use ASPECT to compute the velocity, pressure, and temperature associated with the underlying flow field for each of the four advection methods. In other words, the only difference between the four computational methods that we study here is the advection algorithm we use for a given computation; *the computation of all of the other parts of the problem are identical.*

Each advection algorithm is designed to model the motion of two distinct compositional regions. These four algorithms are: (1) a Finite Element compositional advection method with 'Entropy Viscosity' (Guermond et al., 2011) (FEM-EV), (2) a Discontinuous Galerkin method with a Bound Preserving limiter (DGBP) (He et al., 2017), (3) a particle or 'particle-in-cell' method (Gassmöller et al., 2016), and (4) a Volume-of-Fluid (VOF) interface tracking method.

In many cases our computations demonstrate that the entropy viscosity-based method has far too much numerical diffusion to yield meaningful results. In other words, the computations we made with this method differ substantially from the results we obtained with the other three methods, when these three methods yield essentially identical results. On the other hand, the DGBP method yields good results, although small amounts of each compositional field are numerically entrained within the other compositional field. The particle method yields yet better results than this, but some particles representing the denser fluid are entrained in the upper, less dense fluid and are advected to the top of the computational domain and, similarly, particles representing the less dense fluid are entrained in the lower, denser fluid and are advected to the bottom of the computational domain. The VOF method maintains a sharp interface between the two compositions on a subgrid scale throughout the computation.

We begin by introducing our model problem for the thermal-chemical convection of an initially stratified region in Section 2. We then describe the four advection algorithms in Section 3, examine and discuss their relative performance in Sections 4 and 5. and end with our conclusions in Section 6.

2. Thermochemical convection with density stratification

2.1. The dimensional form of the equations

In order to study alternative numerical algorithms for modeling persistent compositional interfaces we will consider 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 d. In dimensional terms our problem domain Ω has width $3d$ and depth d. At a given reference temperature T_0 the region $d/2 < y \leqslant d$ has a compositional density of ρ_0 and the region $0 \leqslant y < d/2$ 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

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

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

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

where κ is the thermal diffusivity so that inertial effects can be neglected. The fluids in the high density and low density layers are immiscible; i.e., they cannot mix by diffusion. The Lewis number

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

where D is the diffusion coefficient for the compositional variable C (Table 1). 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 compositional density. We make the assumption that the Boussinesq approximation holds; namely, that density differences associated with convection $\rho_0 \alpha (T_1 - T_0)$ and $\Delta\rho$

Table 1
A list of symbols used in this paper.

Symbol	Quantity	Unit	Symbol	Quantity	Unit
$\mathbf{u}$	Velocity	m/s	ρ	Density	$kg \cdot m^{-3}$
P	Dynamic pressure	Pa	$\Delta\rho$	Density difference	$kg \cdot m^{-3}$
T_0	Temperature at the top	K	D	Compositional diffusivity	m^2/s
T_1	Temperature at the bottom	K	α	Thermal expansion coefficient	1/K
T	Temperature	K	d	Vertical thickness of fluid layer	m
ΔT	Temperature difference	K	Pr	Prandtl number	$\frac{\mu}{\rho\kappa}$
C	Composition	–	Le	Lewis number	$\frac{\kappa}{D}$
μ	Viscosity	Pa · s	Ra	Rayleigh number	$\frac{\rho_0 \mathbf{g}\alpha\Delta T d^3}{\mu\kappa}$
κ	Thermal diffusivity	m^2/s	B	Buoyancy ratio	$\frac{\Delta\rho}{\rho_0\alpha\Delta T}$
ρ_0	Reference density	$kg \cdot m^{-3}$			

are small compared with the reference density ρ_0.

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

The governing equations have been discussed in detail by Schubert et al. (2001) (Chapter 6). 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 spacial coordinates, oriented as shown in Fig. 1, and u and v denote the horizontal and vertical velocity components, respectively. We use the Stokes equations

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

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

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

Conservation of energy requires

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

With no diffusion, i.e., $D = 0$, the composition variable C satisfies the *advection equation*

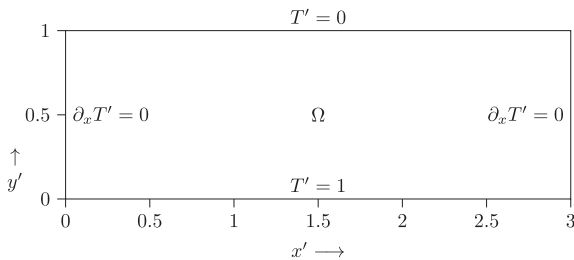$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} = 0. \tag{10}$$

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 $\partial\mathbf{u}/\partial\tau = 0$ (free slip) where $\mathbf{n}$ and τ are the unit normal and tangential vectors to the boundary respectively.

2.2. The nondimensional form of the equations

We introduce the non-dimensional variables

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

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

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

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

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

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

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

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

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

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

This is the superposition of a Rayleigh–Taylor problem and a Rayleigh-Bernard problem. In the isothermal limit ($T_0 = T_1$) it is the classic Rayleigh–Taylor problem (Turcotte and Schubert, 2014, pp. 285–286). If $\Delta\rho$ 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. The density layer will overturn. If $\Delta\rho = 0$ this is the classic Rayleigh-Bernard problem for thermal convection. The governing parameter is the Rayleigh number Ra. If $0 < \text{Ra} < \text{Ra}_c$, the critical Rayleigh number, no flow will occur. If $\text{Ra}_c < \text{Ra} < \text{Ra}_t$ steady cellular flow will occur. If $\text{Ra} > \text{Ra}_t$ the flow becomes unsteady and thermal turbulence develops.

If $\text{Ra} > \text{Ra}_c$ 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 and $C = \frac{1}{2}$. 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 2D rectangular domain $\Omega = [0,3] \times [0,1]$ shown in Fig. 1 are given by

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

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

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

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

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 (Chorin and Marsden, 1993). 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 (LeVeque, 1990, Eq. (1.3)).

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

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

$$\frac{\partial \mathbf{u}}{\partial \tau} = 0 \quad \text{(freeslip)}, \tag{27}$$

where $\mathbf{n}$ and τ 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,

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

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

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

$$\partial_x T(3,y,t) = 0, \tag{31}$$

The geometry of the computational domain together with the boundary conditions on the temperature are shown in Fig. 1.

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 highly nonlinear system of equations. To solve this nonlinear system, we apply the Implicit Pressure Explicit Saturation (IMPES) approach, which was originally developed for computing approximations to solutions of equations for modeling problems in porous media flow (Sheldon et al., 1959; Huber and Helmig, 1999), to decouple the incompressible Stokes Eqs. (19) and (20) from the temperature and compositional Eqs. (21) and (22). This leads to three discrete systems of linear equations, the Stokes equations, the temperature equation and the composition equation, thereby allowing them 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 $\Delta t^k = t^k - t^{k-1}, k = 0,1,\dots$ Given the temperature T^k and composition

C^k at time $t = t^k$, we first solve for our approximation to the Stokes Eqs. (19) and (20) to obtain the velocity $\mathbf{u}^k = (u^k,v^k)$ and pressure P^k

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

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

For the incompressible Stokes Eqs. (32) and (33), we use the standard mixed FEM method with a Taylor–Hood element (Donea and Huerta, 2005) for the spatial approximation. We refer the interested reader to Kronbichler et al. (2012) for a more detailed discussion of the spatial discretization and the choices of Stokes preconditioner.

3.3. Discretization of the temperature equation

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). This algorithm includes the entropy viscosity stabilization technique described in Guermond et al. (2011). If we introduce the inner product of two scalar functions u and v on Ω

$$(u,v)_\Omega = \int_\Omega u\,v\,dx\,dy \tag{34}$$

and $\Gamma_D = \{y = 0\}$. By multiplying the test function $\psi(x,y)$ and taking the integration, the weak form of this spatial discretization is

$$\left(\frac{\partial T}{\partial t},\psi\right)_\Omega + (\mathbf{u}\cdot\nabla T,\psi)_\Omega = -(\nabla T,\nabla\psi)_\Omega - (\nu_h(T)\nabla T,\nabla\psi)_\Omega + \left(\frac{\partial T}{\partial \mathbf{n}},\psi\right)_{\Gamma_D} \tag{35}$$

where $\nu_h(T)$ is the entropy viscosity function as defined in Kronbichler et al. (2012), except here we do not use a second-order extrapolation to treat the advection term $(\mathbf{u}\cdot\nabla T,\psi)$ and the entropy viscosity term $(\nu_h(T)\nabla T,\nabla\psi)_\Omega$ explicitly. We use the fully implicit adaptive Backward Differentiation Formula of order 2 (BDF2) (Wanner and Hairer, 1991; Kronbichler et al., 2012) to discretize the temperature equation in time. Thus, the full discretization of the temperature equation is

$$\left(\frac{1}{\Delta t^{k+1}}\left(\frac{2\Delta t^{k+1} + \Delta t^k}{\Delta t^{k+1} + \Delta t^k}T^{k+1} - \frac{\Delta t^{k+1} + \Delta t^k}{\Delta t^k}T^k + \frac{(\Delta t^{k+1})^2}{\Delta t^k(\Delta t^{k+1} + \Delta t^k)}T^{k-1}\right),\psi\right)_\Omega$$
$$= -(\mathbf{u}^k\cdot\nabla T^{k+1},\psi)_\Omega - (\nabla T^{k+1},\nabla\psi)_\Omega - (\nu_h^k(T)\nabla T^{k+1},\nabla\psi)_\Omega + (\frac{\partial T^{k+1}}{\partial \mathbf{n}},\psi)_{\Gamma_D}. \tag{36}$$

The entropy-viscosity function $\nu_h^k(T)$ is a non-negative constant within each cell that only adds artificial diffusion in cells for which the local Péclet number $\text{Pe} = \text{Ra}\cdot\text{Pr}$ is large and the solution is not smooth.

3.4. Discretization of the composition equation

We use one of the four algorithms described below to discretize the composition Eq. (22). The first two algorithms are based on a spacial discretization of the weak form of the of the composition equation:

$$\left(\frac{\partial C}{\partial t},\psi\right)_\Omega + (\mathbf{u}\cdot\nabla C,\psi)_\Omega = 0. \tag{37}$$

3.5. The finite element advection algorithm with entropy viscosity

This is the first advection algorithm that was implemented in ASPECT. It is based on the same spatial discretization as shown in Eq. (35). However, the entropy-viscosity stabilization term on the right-hand side in

$$\left(\frac{\partial C}{\partial t},\psi\right)_\Omega + (\mathbf{u}\cdot\nabla C,\psi)_\Omega = -(\nu_h(C)\nabla C,\nabla\psi)_\Omega \tag{38}$$

is computed separately for the composition field; i.e, it does *not* have the same value in each cell as does the entropy viscosity function $\nu_h(T)$

for the temperature field. We also use the adaptive BDF2 algorithm for the time discretization, leading to the following FEM Entropy Viscosity (FEM-EV) discretization of Eq. (22),

$$\frac{1}{\Delta t^{k+1}}\left(\frac{2\Delta t^{k+1} + \Delta t^k}{\Delta t^{k+1} + \Delta t^k}C^{k+1} - \frac{\Delta t^{k+1} + \Delta t^k}{\Delta t^k}C^k + \frac{(\Delta t^{k+1})^2}{\Delta t^k(\Delta t^{k+1} + \Delta t^k)}C^{k-1}, \psi\right)_\Omega$$

$$= -(\mathbf{u}^k \cdot \nabla C^{k+1}, \psi)_\Omega - (\nu_h^k(C)\nabla C^{k+1}, \nabla\psi)_\Omega. \tag{39}$$

In Eq. (39) the entropy viscosity function $\nu_h^k(C)$ has the same purpose as $\nu_h^k(T)$. We shall discuss the role the entropy viscosity $\nu_h^k(C)$ plays in the compositional advection method in Section 5.3.1. below.

3.6. The discontinuous Galerkin bound preserving advection algorithm

In this algorithm we use adaptive BDF2 to discretize the advection Eq. (22) for the composition in time as shown in Eq. (39). However we use a Discontinuous Galerkin method with a Bound Preserving limiter (DGBP) for the discretization of the spatial terms in Eq. (22).

The DG method differs from the classic continuous Galerkin FEM, since it allows for discontinuities between elements (Reed and Hill, 1973; Shu, 2016). If we denote the discretized computational domain by $\Omega = \cup_{e=1}^E \Omega_e$, where Ω_e denotes non-overlapping body-conforming quadrilateral elements, and let $V_N(\Omega)$ denote the DG element space, then the fully discretized problem is as follows. Find $C^{k+1} \in V_N(\Omega)$ such that for each $\Omega_e \subset \Omega$,

$$\left(\frac{1}{\Delta t^{k+1}}\left(\frac{2\Delta t^{k+1} + \Delta t^k}{\Delta t^{k+1} + \Delta t^k}C^{k+1} - \frac{\Delta t^{k+1} + \Delta t^k}{\Delta t^k}C^k + \frac{(\Delta t^{k+1})^2}{\Delta t^k(\Delta t^{k+1} + \Delta t^k)}C^{k-1}\right), w\right)_{\Omega_e}$$

$$= -(\mathbf{u}^k \cdot \nabla C^{k+1}, w)_{\Omega_e} + (\mathbf{u}^k \cdot \mathbf{n} C^{k+1}, w)_{\partial\Omega_e} - (\mathbf{u}^k \cdot \mathbf{n} C^{k+1,*}, w)_{\partial\Omega_e} \tag{40}$$

for any $w \in V_N$. An upwind flux is used to determine $C^{k+1,*}$,

$$C^{k+1,*} = \begin{cases} C^{k+1,-}, & \text{if} \quad \mathbf{u}^k \cdot \mathbf{n} > 0, \\ C^{k+1,+}, & \text{if} \quad \mathbf{u}^k \cdot \mathbf{n} < 0, \end{cases} \tag{41}$$

where $C^{k+1,-}$ is the local/interior solution on Ω_e, and $C^{k+1,+}$ is the neighbor/exterior solution of Ω_e (Cockburn and Shu, 1998; Hesthaven and Warburton, 2008). Although, Eq. (40) appears to be only defined locally on each element Ω_e, it also depends on the adjacent solutions through the flux term $C^{k+1,*}$, which is defined at each element interface using two side values. Generally, the flux term $C^{k+1,*}$ is the most difficult part to determine when one wants to design a DG method, since it is the essential feature of the algorithm that ensures the stability of the method.

After obtaining the DG solution C^{k+1}, we apply a Bound Preserving (BP) limiter to the newly obtained numerical solution in a post-processing procedure, which was initially developed in Zhang and Shu (2010), and further developed for approximating solutions of the equations for modeling convection in the Earth's mantle in He et al. (2017). We shall briefly introduce the idea of how to implement the BP limiter. However we refer the interested reader to the latter reference for a detailed explanation of the DGBP algorithm that we have used in this work. Three key ingredients required for constructing a high order bound preserving DG scheme were introduced in Zhang and Shu (2010):

1. The cell average of the composition at the next time step that we obtain using the forward Euler discretization in time is a monotone function with respect to certain point values (e.g., Gauss–Lobatto points in the one-dimensional case), under a suitable CFL condition.
2. On each cell Ω_e a simple scaling limiter modifies the DG polynomial $p(x)$ into $\tilde{p}(x)$ such that $\tilde{p}(x) \in [C_m, C_M]$ at these special points without changing its cell average. Moreover, it can be proven that the modified polynomial $\tilde{p}(x)$ is also a high-order approximation just as $p(x)$. Thus we have $\overline{C}^{k+1} \in [C_m, C_M]$ if all the degrees of freedom at time level k are replaced by using those of modified polynomials $\tilde{p}(x)$.
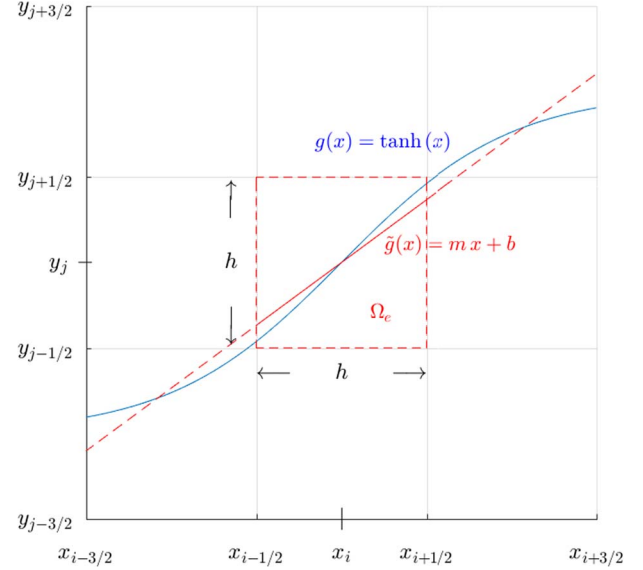
Fig. 2. In our implementation of the VOF interface reconstruction algorithm the true interface, which in this example is $g(x) = \tanh(x)$, is approximated as a line segment $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ in each cell Ω_e that has a volume fraction f_e with $0 < f_e < 1$. The approximate interface 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., $f_e^{true} = f_e$.
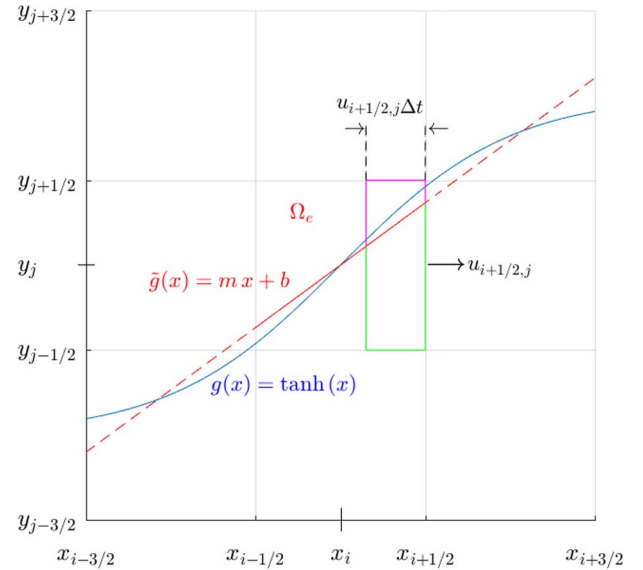
Fig. 3. The volume of the quadrilateral outlined in green on three sides and by a portion of the solid red line on top, is the volume of C_1 that will cross the right-hand edge of Ω_e during the time step from time t^k to t^{k+1}. Here $\Delta t^k = t^{k+1} - t^k$ and we have dropped the superscript k from $u_{i+1/2,j}^k$ and Δt^k for clarity. The solid red line in Ω_e is the reconstructed interface that approximates the true interface $g(x) = \tanh(x)$ in Ω_e as shown in Fig. 2.

3. The forward Euler method is replaced by a strong stability preserving (SSP) high order time discretization scheme, which is a convex combination of forward Euler and thus, will preserve the bounds; i.e., at time step k, if the DG solution is bounded by $C_m \leqslant C^k \leqslant C_M$, then after applying the limiter, the solution C^{k+1} is guaranteed to satisfy the same upper and lower bounds $C_m \leqslant C^{k+1} \leqslant C_M$. However, due to limitations in the current release of ASPECT, the DGBP limiter has been implemented in ASPECT with an implicit BDF2 time discretization algorithm, not the more effective SSP algorithm. Therefore, after applying the BP limiter the DG solution is not
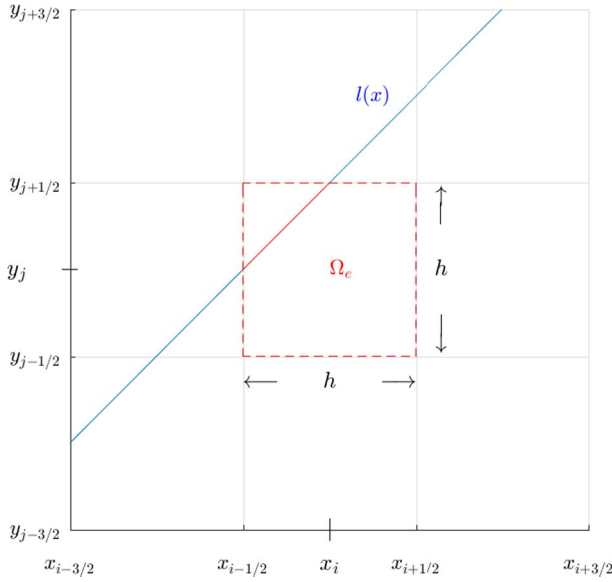
Fig. 4. In this example the true interface is the line $l(x) = mx + b$ Note that the volumes (areas) V_{i-1} and V_i under the line in the first two columns $i-1$ and i are exactly equal to the volumes due to the column sums $\widetilde{V}_{i-1} = h^2 S_{i-1}$ and $\widetilde{V}_i = h^2 S_i$ in the first 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 $\widetilde{m} = S_i - S_{i-1}$ is *exactly* equal to the slope m of the interface as shown in Eq. (59). 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 Eqs. (62) or (63) is exact and hence, one of the linear approximations to the interface in the center cell Ω_e defined in Eq. (65) will always equal the interface in that cell, *exactly*, $\widetilde{g}_{ij}(x) = m_{ij}x + b_{ij} = mx + b = l(x)$. In other words, the piecewise linear VOF approximation to $l(x)$ will *always* reconstruct the linear interface exactly.
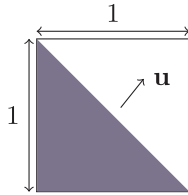

Fig. 5. Schematic of the linear interface translation benchmark problem.

Table 2
The L^1 error in the volume fractions f_e after advecting a linear interface on a uniform grid of squares with sides $h = 2^{-4}, \ldots, 2^{-7}$ in the constant velocity field $\mathbf{u} = (\frac{1}{5}, \frac{1}{4})$. Note that machine precision for variables stored as 64 bit floats is $\epsilon_{mach} \approx 10^{-16}$.

h	Error
2^{-4}	$1.23382 \cdot 10^{-16}$
2^{-5}	$1.21675 \cdot 10^{-16}$
2^{-6}	$1.96083 \cdot 10^{-16}$
2^{-7}	$1.92738 \cdot 10^{-16}$

strictly bounded within the upper and lower bounds $\widetilde{p}(x) \in [C_m, C_M]$ as shown in item (1) above. There is a small error in the bound We discuss this issue in He et al. (2017, Section 4.1.).

3.7. The particle method

Particle methods, sometimes called 'Particle-In-Cell' or 'Tracer in Cell' methods, have long been used by researchers to model problems involving convection in the Earth's Mantle; e.g., (Tackley and King,

2003; McNamara and Zhong, 2004; Trim et al., 2014; Trim and Lowman, 2016). The accuracy of high-order accurate versions of these methods have been recently studied, both in conjunction with Finite Difference Duretz et al. (2011) and Finite Element methods Thielmann et al. (2014), as well as their efficient parallel implementation in AS-PECT (Gassmöller et al., 2016). The particle algorithm in ASPECT that we use here is based on a second-order Runge–Kutta time discretization and an arithmetic averaging interpolation algorithm. It is described in detail in Gassmöller et al. (2016) and Heister et al. (2017).

All three of the algorithms discussed above are based on the Eulerian frame of reference, or Eulerian coordinates. Therefore, the flow velocity field $\mathbf{u}$ is represented as a vector function of position $\mathbf{x}$ and time t, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$. However, since the individual fluid particles are followed as they move through the grid over time it is convenient to use a Lagrangian frame of reference in order to have a convenient notation for describing the location of the particles in time. In particular, we will label each particle with a (time-independent) vector $\mathbf{x_0}$, which is the initial location of the particle at time $t = 0$. We use the vector function $\mathbf{X}(\mathbf{x_0}, t)$ to denote the location of the particle with initial position $\mathbf{x_0}$ at time t. Thus, $\mathbf{X}(\mathbf{x_0}, t)$ satisfies the following equation

$$\frac{\partial \mathbf{X}(\mathbf{x_0}, t)}{\partial t} = \mathbf{u}(\mathbf{X}(\mathbf{x_0}, t), t) \tag{42}$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity at the point $\mathbf{x}$ at time t. Therefore, given the initial positions of the particles $\mathbf{x_0}$, we can solve Eq. (42) to evolve the particle locations in time.

Now denote the discretization of the particles in time by $\mathbf{X}_{\mathbf{x_0}}^k \approx \mathbf{X}(\mathbf{x_0}, t^k)$ and $\mathbf{u}^k(\mathbf{X}_{\mathbf{x_0}}^k) \approx \mathbf{u}(\mathbf{X}_{\mathbf{x_0}}^k, t^k)$, We use a second order Runge Kutta time discretization to approximate Eq. (42)

$$\mathbf{X}_{\mathbf{x_0}}^{k+1,1/2} = \mathbf{X}_{\mathbf{x_0}}^k + \frac{1}{2}\Delta t^{k+1}\mathbf{u}^k(\mathbf{X}_{\mathbf{x_0}}^k), \tag{43}$$

$$\mathbf{X}_{\mathbf{x_0}}^{k+1} = \mathbf{X}_{\mathbf{x_0}}^k + \Delta t^{k+1}\mathbf{u}\left(\mathbf{X}_{\mathbf{x_0}}^{k+1,1/2}, t^k + \frac{1}{2}\Delta t^{k+1}\right), \tag{44}$$

where we approximate the velocity at the half time $t^k + \frac{1}{2}\Delta t^{k+1}$ by averaging the velocities at t^k and t^{k+1}

$$\mathbf{u}\left(\mathbf{X}_{\mathbf{x_0}}^{k+1,1/2}, t^k + \frac{1}{2}\Delta t^{k+1}\right) \approx \frac{1}{2}\{\mathbf{u}^{k+1}(\mathbf{X}_{\mathbf{x_0}}^{k+1,1/2}) + \mathbf{u}^k(\mathbf{X}_{\mathbf{x_0}}^{k+1,1/2})\}. \tag{45}$$

Once we obtain the new particle locations $\mathbf{X}_{\mathbf{x_0}}^{k+1}$ at time t^{k+1}, we compute the value of the composition 'carried' by that particle $C^{k+1}(\mathbf{X}_{\mathbf{x_0}}^{k+1})$ by assigning it the value it had at time $t = 0$; i.e., $C(\mathbf{x_0})$. We use ASPECT to compute the approximate velocity field $\mathbf{u}(\mathbf{x}, t^k)$ at each time step. In order to project the compositional value carried by each particle onto the quadrature points of a given finite element cell Ω_e at time t^k, we use the arithmetic average of the values of the particles in Ω_e,

$$C_e^k = \frac{\sum_p C_p}{\sum_p 1} \tag{46}$$

where p is the index of each particle located within Ω_e. The value of the compositional variable C at each of the quadrature points in Ω_e at time t^k is simply the constant C_e^k.

3.8. The volume-of-fluid interface tracking algorithm

The advection methods described in Sections 3.5–3.7 above are sometimes referred to as interface *capturing* methods, since the interface between the two compositions is not *explicitly* tracked. On the other hand, the Volume-of-Fluid (VOF) method is an interface *tracking* method in which, at each time step, the interface between the two compositions C_1 and C_2 is explicitly reconstructed *in every cell that contains a portion of the interface* and the interface is advanced in time using the explicit knowledge of the interface location and topology at the current time step. In this sense the VOF method approximates the
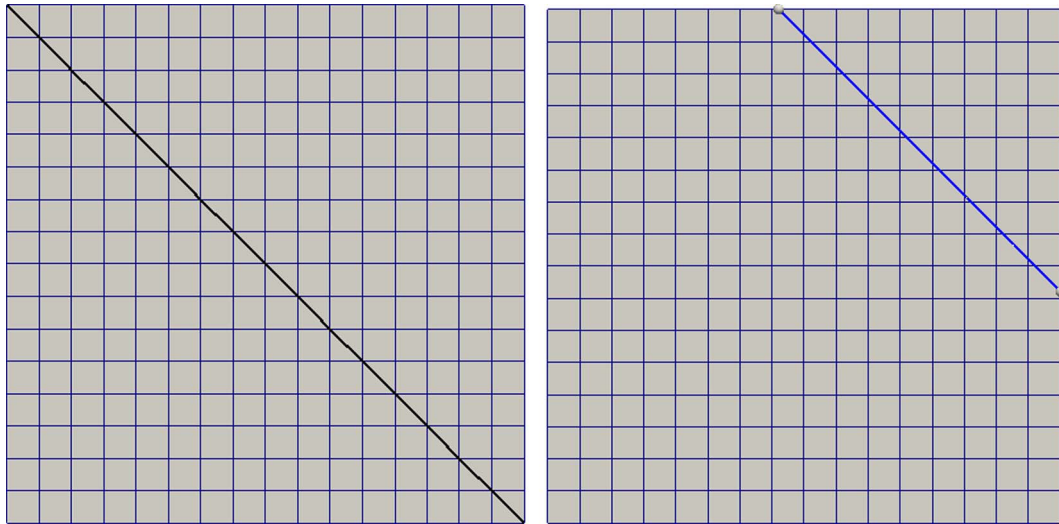
Fig. 6. The initial position $l_0(x) = 1-x$ at $t = 0$ and final position $l_1(x) = 1.45-x$ at $t = 1$ of a linear interface after having been advected in the constant velocity field $\mathbf{u} = (\frac{1}{5},\frac{1}{4})$ with the VOF method described above on a uniform grid of squares with sides $h = 2^{-4}$. In all of the linear interface benchmark computations we used a CFL number of 0.5. As one can see in Table 2 the line is advected *exactly* in the sense that the errors are on the order of machine epsilon; i.e., $error = O(\epsilon_{\mathrm{mach}}) = O(10^{-16})$ for all grid sizes h.
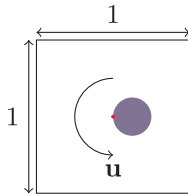

Fig. 7. Schematic of the circular interface rotation benchmark. The red dot is the center of rotation.

Table 3
The L^1 error in the volume fractions f_e and the rate at which the error converges to zero as $h \rightarrow 0$ for the rotation of a circle benchmark shown in Figs. 7 and 8. The third column shows that the error is decreasing at a rate that asymptotes to second-order in h (i.e., $O(h^2)$) as $h \rightarrow 0$, which is the design accuracy of the method.

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

compositional interface on a subgrid scale. Furthermore, in an incompressible flow both the VOF *interface reconstruction algorithm* and the VOF *advection algorithm* conserve the volume of each of the two compositions throughout the course of the computation.

3.8.1. Background

There are a wide variety of possible VOF interface reconstruction and advection algorithms. The VOF method was first developed at the U.S. National Labs in the 1970s (Noh and Woodward, 1976) and have continued to be used and developed by researchers at the Labs (Nichols et al., 1980; Hirt and Nichols, 1981; Torrey et al., 1985, 1987) as well as around the world. The advantage that VOF methods have over other interface tracking algorithms is that they are designed to naturally satisfy a conservation law; namely, Eq. (47) below. Thus, materials that should be conserved as they move with the flow are conserved, without the need to resort to additional numerical algorithms such as the *redistancing* step in a Level Set method Sethian (1999). 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 Henderson et al. (1991) and Miller and Puckett (1994, 1996), jetting in meteorite impacts Puckett and Miller (1996), *non-conservative* interface motion such as photolithography (Helmsen et al., 1996, 1997), the transition from deflagration to detonation Pilliod and Puckett (1998) and more than two materials Anbarlooei and Mazaheri (2011) and Hill and Shashkov (2013); i.e., more than one interface in a cell.

3.8.2. Description

In this article we use a two-dimensional VOF algorithm to discretize the *conservation equation*

$$\frac{\partial f}{\partial t} + \nabla \cdot \mathbf{F}(f) = 0, \tag{47}$$

where $\mathbf{u} = (u,v)$ 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', and

$$\mathbf{F}(f) = (F(f),G(f)) = (uf,vf) \tag{48}$$

is the volume fraction flux associated with C_1. In our VOF implementation in ASPECT we use the 'Efficient Least Squares VOF Interface Reconstruction Algorithm' (ELVIRA), which is described in detail in Pilliod and Puckett (2004) and is based on the ideas in Puckett (1991) and Pilliod (1992). The ELVIRA interface reconstruction algorithm reconstructs lines on a uniform grid with square cells *exactly*. We will explain this in more detail in Section 3.8.2.1 and give a example in Section 3.8.3.2 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 turns out to be true (Puckett, 2010a,b, 2014). We use a second-order accurate operator splitting advection method (Pilliod and Puckett, 2004; Strang, 1968) to update the values of the volume fractions in time.

For simplicity of exposition we will assume the finite element grid consists entirely of square cells Ω_e, of side h, indexed by e, and aligned parallel to the x and y axes. The discretization of Eq. (47) proceeds as follows. Let Ω_e denote an arbitrary finite element cell in our 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 $0 \leqslant f_e^k \leqslant 1$ such that

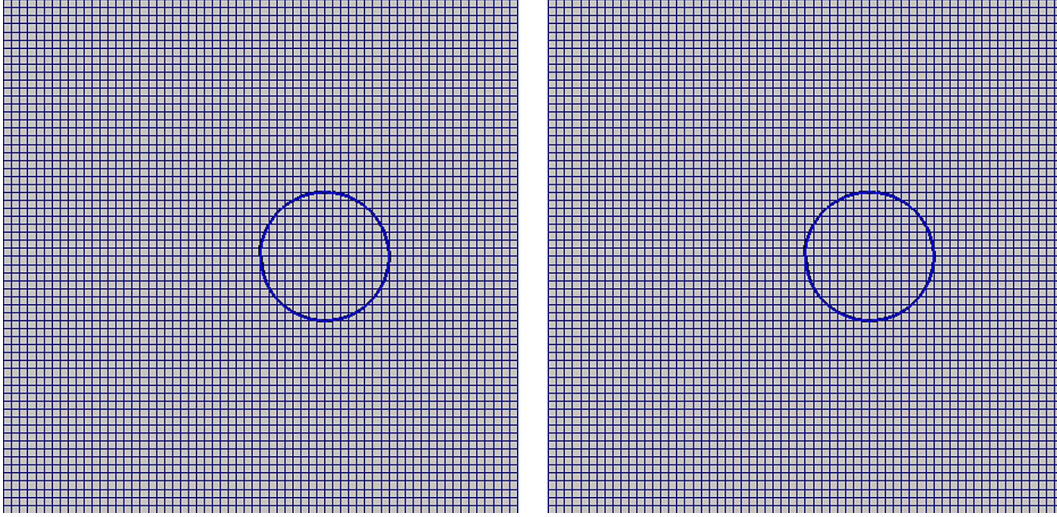$$f_e^k \cong \frac{1}{h^2} \int_{\Omega_e} f(x,y,t^k)\, dx\, dy. \tag{49}$$

Fig. 8. The Rotating Circle Benchmark: On the left is the circle at the *initial time* $t = 0$ and on the right is the circle *after* one revolution about the *center* of the 1×1 square domain – not the center of the circle – on a grid of 64×64 square cells. In this benchmark the angular velocity is π radians per unit time with an end time of $t = 2.0$ and we used a CFL number of $\sigma = \frac{1}{2}$ for a total of 1138 time steps. Note that $h = 2^{-6}$ is not the smallest value of h we tested. (See Table 3.)

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

$$V_e^k = \int_{\Omega_e} f_e^k \, dx \, dy = h^2 f_e^k. \tag{50}$$

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

From a mathematical point of view the variable $f(x,y)$ is the *characteristic function* associated with C_1. In other words,

$$f(x,y) = \begin{cases} f(x,y) = 1 & \text{if } (x,y) \text{ is occupied by composition 1,} \\ f(x,y) = 0 & \text{if } (x,y) \text{ is not occupied by composition 1.} \end{cases} \tag{51}$$

This implies $1 - f(x,y)$ is the characteristic function associated with C_2. In this article we restrict ourselves to modeling the interface between two compositions. However, there is currently a great deal of research into modeling three or more interfaces in one cell with a VOF method; e.g., see Jemison et al. (2015).

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}.

1. *The interface reconstruction step:* Given a cell Ω_e 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, 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 $\widetilde{g}_e(x) = m_e x + b_e$, which is denoted $\widetilde{g}_{ij}(x) = m_{ij} x + b_{ij}$ in the figure, to the true interface in cells Ω_e for which $0 < f_e^k < 1$ in Section 3.8.2.1 below.

2. *Computation of the fluxes:* In the computations presented in this article we use a second-order accurate operator split algorithm, often referred to as Strang Splitting (Strang, 1968) in order to advance the interface in time. However, as mentioned above, for clarity and simplicity of exposition we will only describe a first-order accurate operator split VOF advection algorithm here. See Pilliod and Puckett (2004) for the details of a second-order accurate operator split VOF advection algorithm.

For convenience and clarity of exposition for the remainder of this section we will use the index notation (i,j) 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 indexed as shown in the figure. In the EL-VIRA interface reconstruction algorithm we use the information in the 3×3 block of cells $\Omega_{i'j'}$ immediately adjacent to the cell Ω_{ij} in which we wish to reconstruct the interface. Given the reconstructed interface $\widetilde{g}_{ij}(x)$ in

$$\Omega_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{i+1/2}] \tag{52}$$

as shown in Fig. 3 and the velocity $u_{i+1/2,j}^k$ normal to the right and left edges of Ω_{ij} at time t^k, we wish to determine the *volumes* $V_{i\pm1/2,j}^k$ 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*. A diagram for how to determine the *volume* $V_{i+1/2,j}^k$ of C_1 that crosses the right-hand edge of Ω_{ij} in the time interval $[t^k, t^{k+1}]$, given the assumption that $u_{i+1/2,j}^k > 0$, is shown in Fig. 3.

3. *The conservative update:* Given the *volumes* $V_{i\pm1/2,j}^k$ of C_1 that cross the left and right-hand edges of Ω_{ij} in the time interval $[t^k, t^{k+1}]$ we use the following equation to determine an *intermediate* volume $\widehat{V}_{ij}^k$ in Ω_{ij} for the first part of the two part operator split algorithm:

$$\widehat{V}_{ij}^k = V_{ij}^k + V_{i-1/2,j}^k - V_{i+1/2,j}^k, \tag{53}$$

where $V_{ij}^k = h^2 f_{ij}^k$ and $\widehat{V}_{ij}^k$ denotes the 'intermediate' volume in Ω_{ij} after the first part of the operator split advection step from time t^k to time t^{k+1}.

Given the nine intermediate volume fractions $\widehat{f}_{i'j'}^k \equiv \widehat{V}_{i'j'}^k / h^2$ 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 $\Omega_{i'j'}$, reconstruct an intermediate interface $\widehat{g}_{i'j'}(x)$ in each cell $\Omega_{i'j'}$ and use it to geometrically determine the volumes $\widehat{V}_{i,j\pm1/2}^k$ 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. The volume of V_{ij}^{k+1} in Ω_{ij} at the new time t^{k+1} is thus,

$$V_{ij}^{k+1} = \widehat{V}_{ij}^k + \widehat{V}_{i,j-1/2}^k - \widehat{V}_{i,j+1/2}^k. \tag{54}$$

The new volume fraction in Ω_{ij} is now

$$f_{ij}^{k+1} = V_{ij}^{k+1} / h^2. \tag{55}$$

There are also *unsplit* VOF advection algorithms. See Puckett et al. (1997) and Pilliod and Puckett (2004) for examples.
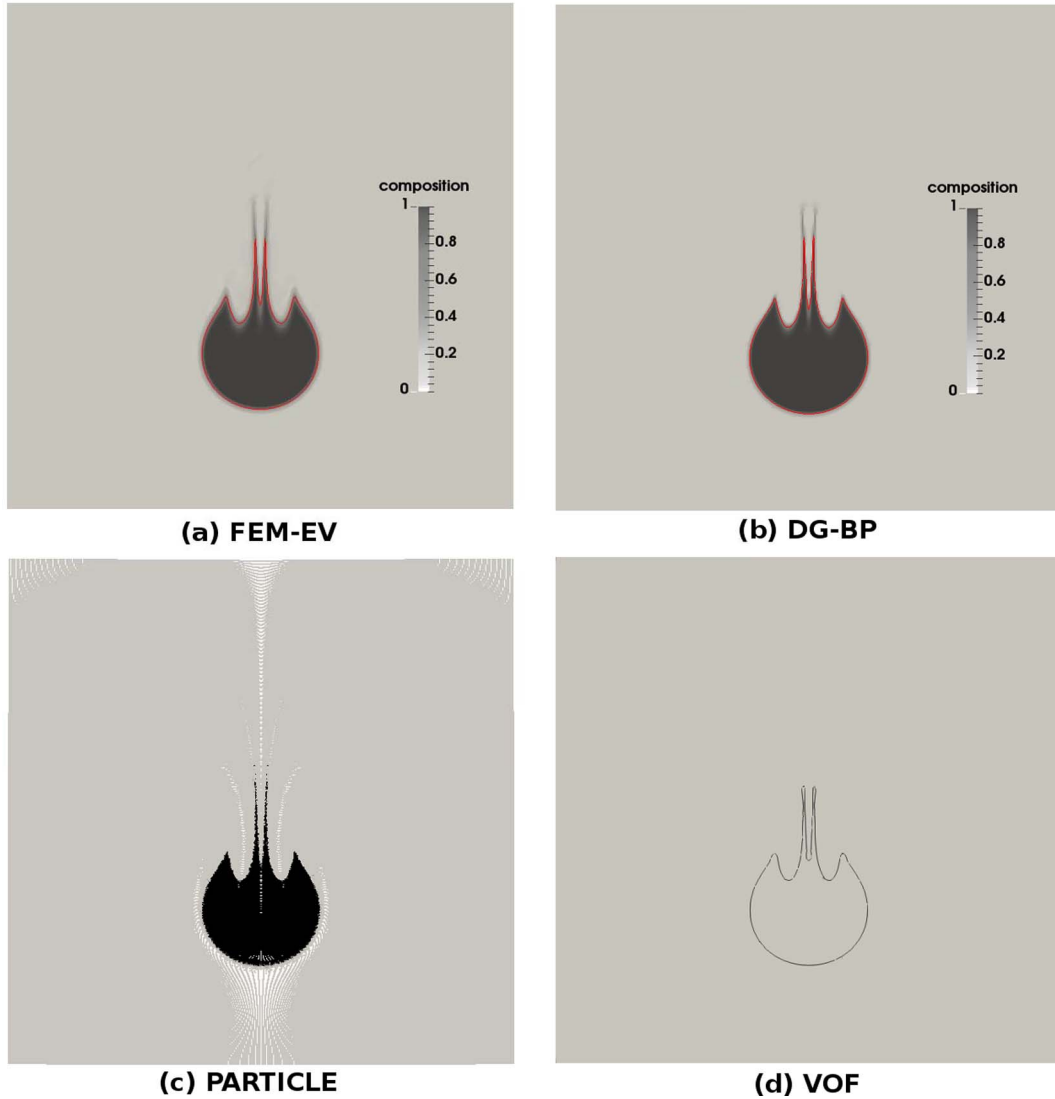
Fig. 9. The Gerya-Yuen 'Sinking Box' problem at $t = 9.81$ Myr with (a) the FEM-EV compositional field with the $C = 0.5$ contour in red, (b) the DG-BP compositional field with the $C = 0.5$ contour line in red, (c) the particles where gray particles carry the background density, black particles carry the density of the initial box and white spaces are regions where there are no particles, and (d) the reconstructed VOF interface drawn in black. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.8.2.1. The ELVIRA interface reconstruction algorithm. Here we briefly describe the ELVIRA interface reconstruction algorithm (Pilliod and Puckett, 2004) we used in this article. 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 (Torrey et al., 1985) and three dimensions (Torrey et al., 1987) and on every conceivable grid; e.g., (Kothe et al., 1999), as well as numerous hybrid VOF/ Level Set algorithms (Sussman and Puckett, 2000). See Tryggvason et al. (2011) and the references therein for a more complete overview of VOF methods.

In the ELVIRA algorithm the approximate interface will be a *piecewise linear* approximation $\widetilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ 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 $\widetilde{g}_{ij}$ 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 $l(x) = mx + b$. 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

$$S_{i-1} \equiv \sum_{j'=j-1}^{j+1} f_{i-1,j'} \qquad \text{and} \qquad S_i \equiv \sum_{j'=j-1}^{j+1} f_{i,j'} \tag{56}$$

are *exact* in the sense that

$$S_i = \frac{1}{h^2} \int_{x_{i-1/2}}^{x_{i+1/2}} (l(x) - y_{j-3/2}) \, dx \tag{57}$$

and similarly for S_{i-1}, but *not* for S_{i+1}. Thus, using Eq. (57) we find the difference in the column sums S_i and S_{i-1} is

$$h^2(S_i - S_{i-1}) = \int_{x_{i-1/2}}^{x_{i+1/2}} (mx-b) - y_{j-3/2} \, dx - \int_{x_{i-3/2}}^{x_{i-1/2}} (mx-b) - y_{j-3/2} \, dx$$
$$= \int_{x_{i-1/2}}^{x_{i+1/2}} mx \, dx - \int_{x_{i-3/2}}^{x_{i-1/2}} mx \, dx$$
$$= m \frac{x^2}{2} \Big|_{x_{i-1/2}}^{x_{i+1/2}} - m \frac{x^2}{2} \Big|_{x_{i-3/2}}^{x_{i-1/2}}$$
$$= \frac{m}{2}[(x_{i+1/2})^2 - (x_{i-3/2})^2] - \frac{m}{2}[(x_{i+1/2})^2 - (x_{i-1/2})^2]$$
$$= \frac{m}{2}h(x_{i+1/2} - x_{i-3/2})$$
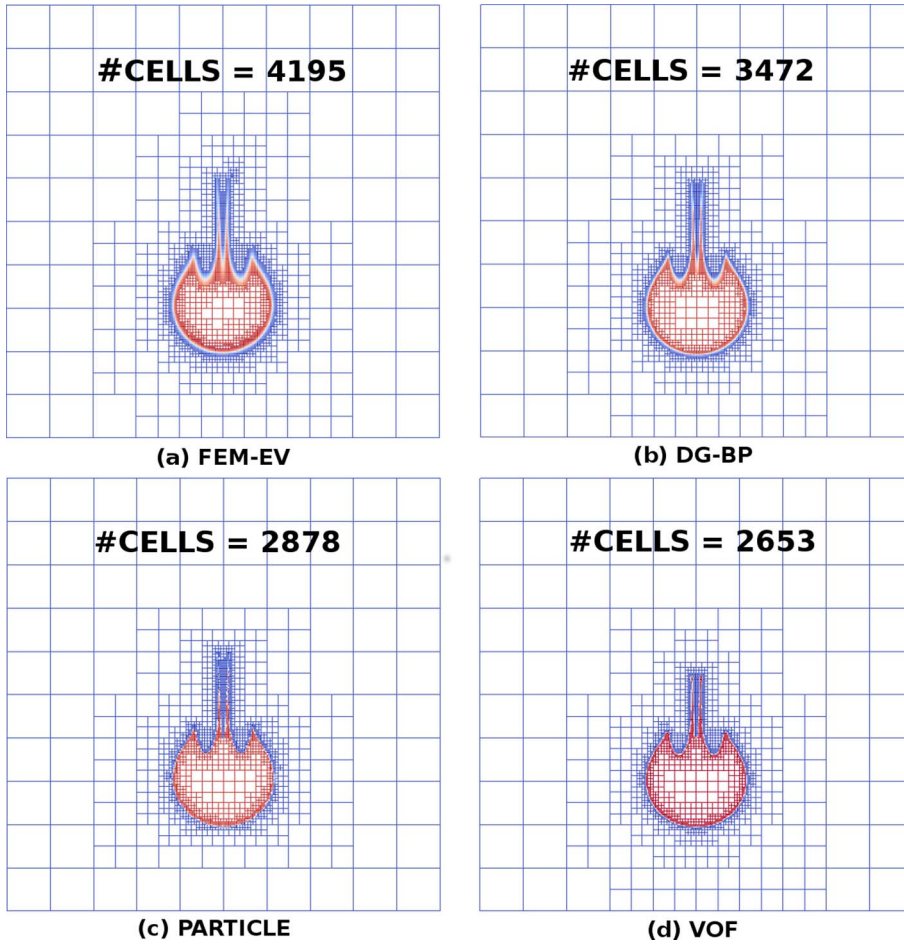$$= m \, h^2. \tag{58}$$

Fig. 10. The Gerya-Yuen Sinking Box problem results overlaid on the underlying AMR grid.

Thus,

$$m = S_i - S_{i-1} \tag{59}$$

and we have recovered the *exact* slope m of the true interface $l(x)$ in the center cell simply by differencing the the correct pair of column sums of volume fractions. A little thought will show that the constraint

$$f_{ij} = f_{ij}^{true} \tag{60}$$

determines b uniquely, thus determining the linear approximation

$$g_{ij}(x) = m x + b \tag{61}$$

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 Chorin (1985) or Pilliod and Puckett (2004). This always works on a uniform grid of square cells, each of side h.

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

$$
\begin{aligned}
m^{x,l} &= (S_i - S_{i-1}) \\
m^{x,c} &= \frac{(S_{i+1} - S_{i-1})}{2} \\
m^{x,r} &= (S_{i+1} - S_i)
\end{aligned} \tag{62}
$$

and three ways to difference the row sums

$$
\begin{aligned}
m_l^y &= (R_j - R_{j-1}) \\
m_c^y &= \frac{(R_{j+1} - R_{j-1})}{2} \\
m_l^y &= (R_{j+1} - R_j)
\end{aligned} \tag{63}
$$

where the *row sums* are defined by

$$R_{j-1} \equiv \sum_{i'=i-1}^{i+1} f_{i',j-1}, \quad R_j \equiv \sum_{i'=i-1}^{i+1} f_{i',j} \quad \text{and} \quad R_{j+1} \equiv \sum_{i'=i-1}^{i+1} f_{i',j+1}. \tag{64}$$

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

$$
\begin{aligned}
g_l^x &= m_l^x x + b_l^x, & g_l^y &= m_l^y x + b_l^y, \\
g_c^x &= m_c^x x + b_c^x, & g_c^y &= m_c^y x + b_c^y, \\
g_r^y &= m_r^x x + b_r^x, & g_r^y &= m_r^y x + b_r^y.
\end{aligned} \tag{65}
$$

We obtain from each of the six slopes in Eqs. (62) and (63) 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 Eq. (65). We now explain this procedure in a bit more detail.

3.8.2.2. Approximating an unknown interface. 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 $\widetilde{g}_{ij}(x) = m_{ij} x + b_{ij}$ that is a second-order accurate approximation to $g(x)$, in the following sense,

$$\max |g(x) - \widetilde{g}_{ij}(x)| \leqslant \widetilde{C} h^2 \quad \text{for all} \quad x \in [x_{i-1/2}, x_{i+1/2}], \tag{66}$$
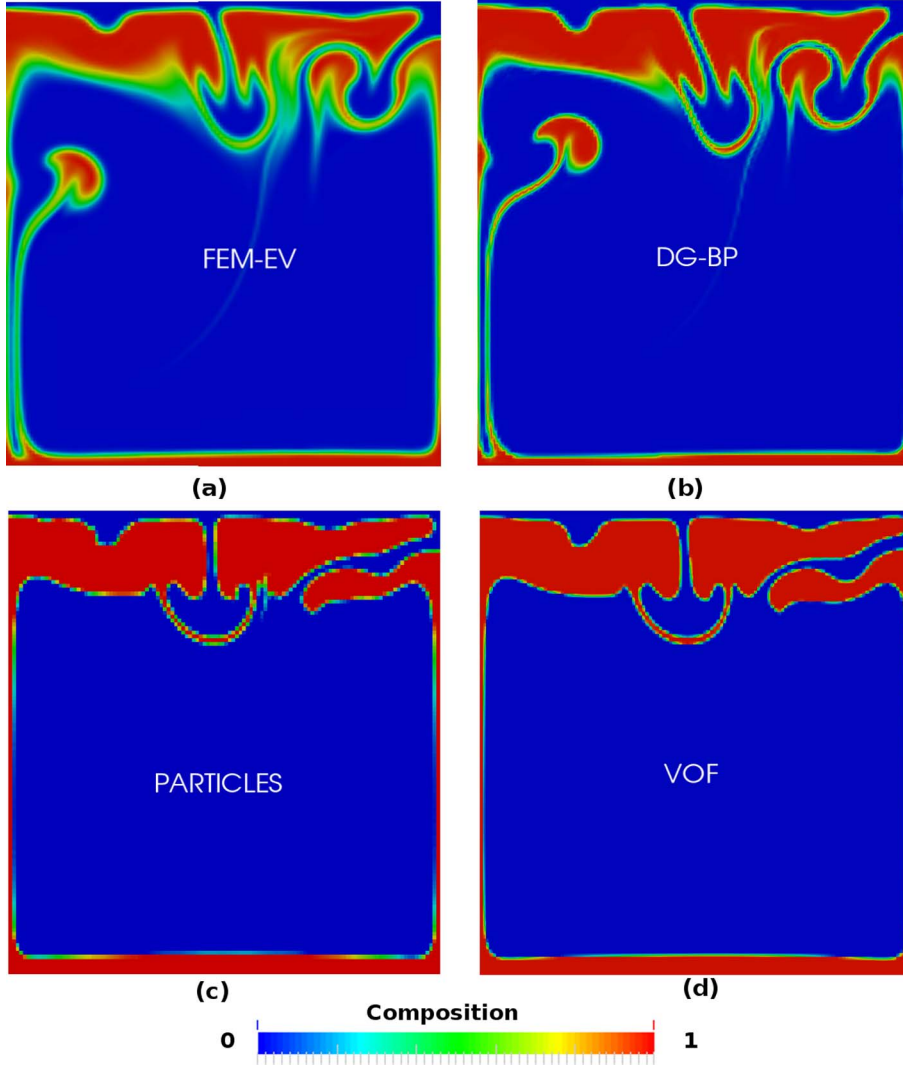
Fig. 11. The van Keken Rayleigh–Taylor problem at $T = 1500$, on a uniform grid with 120×120 cells. The composition field C computed with (a) FEM-EV, (b) DG-BP, (c) PARTICLES, and (d) VOF. Note that regions in which mixing of the two layers has occurred the most is highlighted in green, since it is in the middle of the color map.

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

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

$$E(\widetilde{m}) = \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} (f_{i'j'} - \widetilde{f}_{i'j'})^2. \tag{67}$$

Note that this is the square of the *two norm* on vector spaces R^n from linear algebra, where in our case $n = 9$, (Strang, 2016).

Now take the volume fractions we are given, namely $f_{i'j'}$, and form all six of the slopes in Eqs. (62) and (63) and the six candidate lines in Eq. (65) from these slopes. Remember that the 'y intercept' b for each of the lines in Eq. (65) is determined by the constraint $f_{ij}^{true} = f_{ij}$. Each of the six line produces nine volume fractions in the 3×3 block B_{ij}. For example, given the slope $m^{x,c}$ defined in Eq. (62) we obtain the line $g_c^x = m_c^x\,x + b_c^x$ defined in Eq. (65), 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 Eq. (65) with slopes computed as in Eqs. (62) and (63). Finally, take the line from Eq. (65) that minimizes the error defined in Eq. (67); i.e., pick the

slope from Eqs. (62) and (63), call it $\widetilde{m}$, that satisfies

$$E(\widetilde{m}) = \min\{E(m_l^x), E(m_c^x), \ldots, E(m_r^y)\}. \tag{68}$$

The line

$$\widetilde{g} = \widetilde{m}\,x + \widetilde{b} \tag{69}$$

is the linear approximation to the true interface $g(x)$ in Ω_{ij} that we use in the VOF algorithm in this article. In Puckett (2010a, 2014) it is proven that this algorithm produces a second-order accurate approximation in the sense of Eq. (66) to the interface provided that

$$h \leqslant \frac{2}{33\,\sigma_{max}} \tag{70}$$

where σ_{max} denotes the maximum curvature of the interface, h is the grid size, and the volume fractions due to the true interface are exact.

3.8.3. VOF interface tracking algorithm benchmarks

In this section we present two standard VOF benchmarks to demonstrate that:

(a) The VOF algorithm described above advects linear interfaces in a constant velocity field *exactly*.
(b) The VOF algorithm described above is second-order accurate provided the interface and velocity field is smooth; e.g., both have two
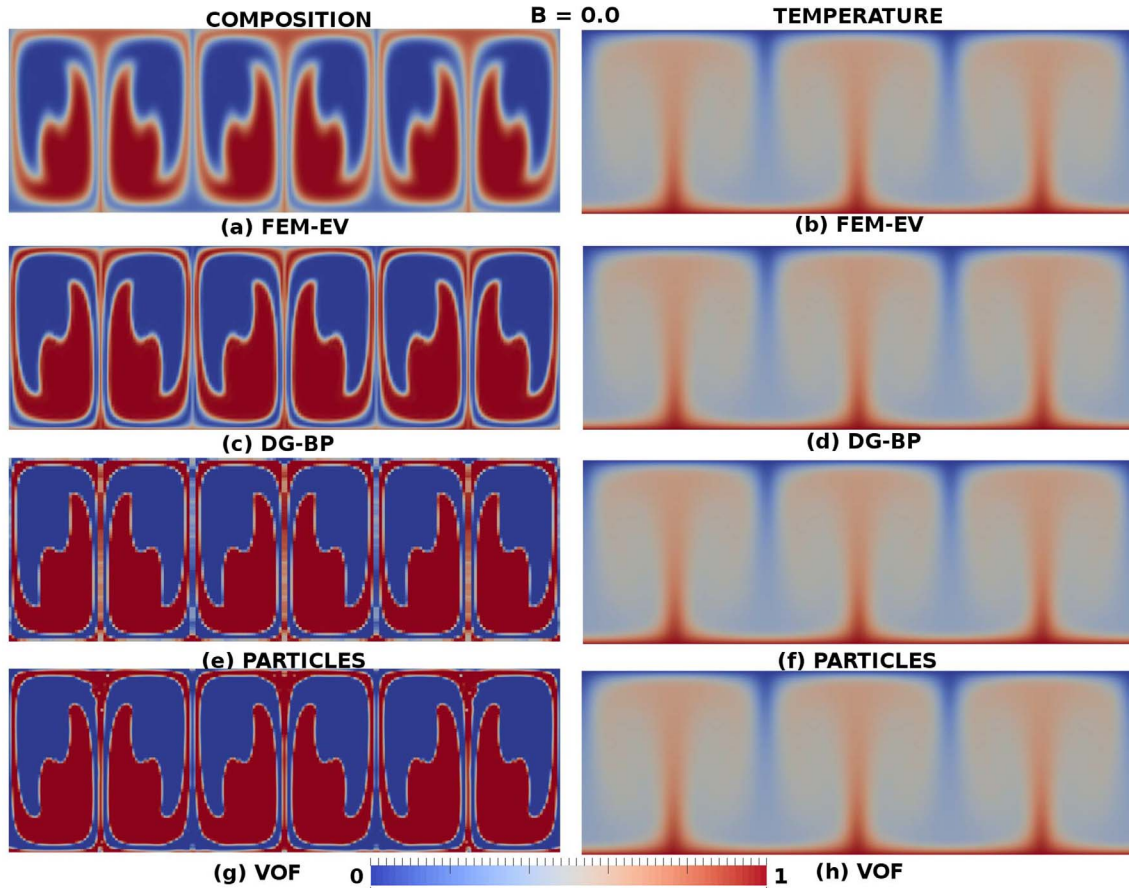
Fig. 12. The composition and temperature at $t = 0.0375$ with B = 0.0 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

continuous derivatives.

3.8.3.1. The method of measuring the error. In the two benchmarks presented below we will measure the error by computing the difference of true and approximate volume fractions $f_e^{true} - f_e$ in the discrete L^1 norm at the final time $t^{end} = t^K$,

$$error(h) = \sum_e |\int_{\Omega_e} f_e^{true}(x,y,t^{end}) dx\, dy - f_e^{end}|.$$

(71)

3.8.3.2. The linear interface advection benchmark. The first benchmark problem is the advection of a linear interface $l(x) = 1-x$ in the constant velocity field $\mathbf{u} = \left(\frac{1}{4}, \frac{1}{5}\right)$ as shown in Fig. 5. The VOF method we use in this article has been designed so that the error in advecting *any* line in *any* constant velocity field on a square grid should be accurate to machine precision, $\epsilon_{mach} \approx 10^{-16}$. The L^1 error in the volume fractions f_e for four computations with decreasing $h = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$ are shown in Table 2). Fig. 6 contains a graphical display of the results of this computation for $h = 2^{-4}$.

3.8.3.3. Circular Interface Rotation Benchmark. The second benchmark problem is the advection of a circle containing composition 1 in a rotating velocity field as shown in Fig. 7. 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 the rotation is in the center of the square domain and *not* in the center of the circle. Due to the chosen algorithms, this is expected to be correct to second order in h. The L^1 error in the volume fractions f_e for this benchmark for six computations with increasing grid resolution $h = 2^{-4}, 2^{-5}, \ldots, 2^{-9}$ are shown in Table 3. We used a CFL number of $\sigma = \frac{1}{2}$

for all of these computations. Contains a graphical display of the results of this computation for $h = 2^{-6}$.

4. Numerical results

In order to compare the four advection algorithms presented in Section 3, above we have used them to model the following three problems: (1) the Gerya-Yuen 'sinking box' problem (Gerya and Yuen, 2003); (2) the van Keken Rayleigh–Taylor problem (van Keken et al., 1997); and (3) the model problem with thermal-chemical convection and density stratification described in Section 2 above. Note that in the first two examples, the density depends only on the compositional field. Thus, the temperature does not play a role in these problems and we only need to solve the coupled system consisting of the incompressible Stokes Eqs. (19) and (20) with boundary conditions (26) and (27) and the compositional Eq. (22).

In all of our computations we use a Taylor–Hood (Q_2, Q_1) FEM element combination (Donea and Huerta, 2005) for the numerical solution of the Stokes Eqs. (32) and (33) (second-order spatial element basis for the velocity field and a first-order spacial element basis for the pressure), and a Q_2 FEM element for the numerical solution of the temperature Eq. (36). We use a second-order accurate spatial discretization for the composition equation with FEM-EV and DGBP, a first order discontinuous element with the VOF algorithm, and a piecewise constant 'composition' element for the particle method with an arithmetic cell averaging interpolation algorithm. See Kronbichler et al. (2012) for additional details concerning the implementation of the FEM-EV algorithm in ASPECT, He et al. (2017) for additional details concerning the implementation of the DGBP algorithm in ASPECT, and Gassmöller
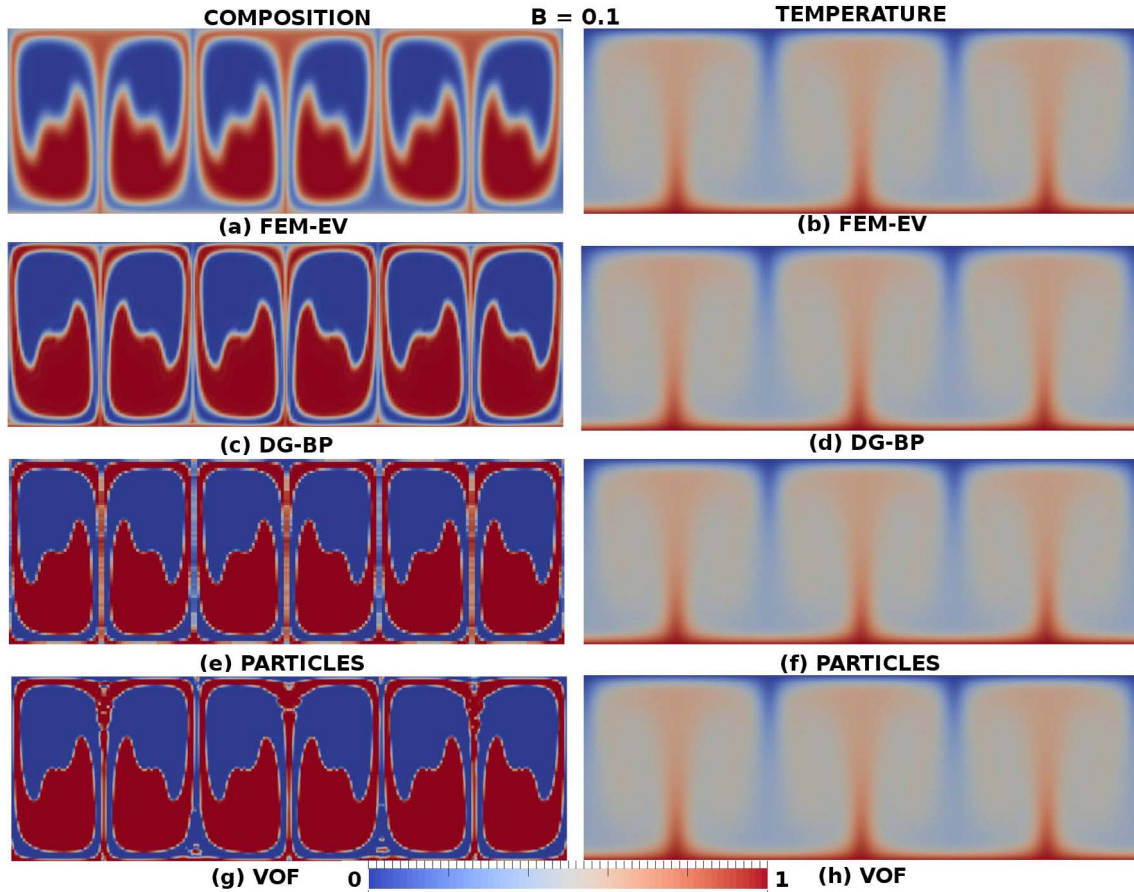
Fig. 13. The composition and temperature at $t = 0.0375$ with B = 0.1 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

et al. (2016) for additional details concerning the implementation of the particle algorithm in ASPECT.

We have chosen to present the computational results from the Gerya–Yuen 'sinking box' problem (Gerya and Yuen, 2003) first for the following reasons.

1. It is a mathematically *well-posed problem*. For the definition of a well-posed problem see, e.g., John (1982, p. 155) or Guenther and Lee (1988, p. 16 & p. 91). In particular, the 'sinking box' problem depends continuously on the initial data; i.e., it is *stable*.
2. The second problem for which we present computational results, the van Keken (viscous) Rayleigh–Taylor problem, is unstable to perturbations at all length scales (Chandrasekhar, 1961, pp. 441–445).
3. Our model problem for thermal-chemical convection with density stratification is unstable for some range of values of Ra and B.
4. Computing unstable problems poses particular difficulties when comparing computational results; even computational results computed with the same code but on different grids; i.e., different values of h.

For the Gerya-Yuen 'sinking box' problem and the van Keken isoviscous Rayleigh–Taylor problem, we computed all of the numerical results on a shared memory computer using 16 processors.

4.1. The Gerya-Yuen Sinking box problem

The Gerya-Yuen 'sinking box' problem is defined on a 500 km × 500 km two-dimensional Cartesian domain. A small block (100 km × 100 km) is placed with its top edge 50 km below the top of the domain and centered horizontally. The following parameters are held fixed:

$\mathbf{g} = (0,9.8)$ m/s^2,   acceleration due to gravity
$L = 500$ km   domain height and width
$\mu = 10^{21}$ Pa·s   viscosity
$\rho_0 = 3200$ kg/m^3,   background density
$\rho_1 = 3300$ kg/m^3,   small box density   (72)

The initial location and dimension of the small box is defined by the composition field $C(\mathbf{x},t)$ as follows:

$$C(\mathbf{x},0) = \begin{cases} 1, & \text{if } (x,y) \in [200\,\text{km}, 300\,\text{km}] \times [350\,\text{km}, 450\,\text{km}], \\ 0, & \text{otherwise.} \end{cases} \quad (73)$$

In this section, we apply each of the four advection algorithms to the sinking box problem from Gerya and Yuen (2003). We used the Adaptive Mesh Refinement (AMR) algorithm in ASPECT for all four computations with the same refinement criterion and input parameters. The initial coarse mesh is a uniform grid with 10×10 cells. Given this coarse mesh, ASPECT initially performs five additional adaptive mesh refinement steps using "composition approximate gradient" as the refinement strategy with the relevant parameters set as follows: "Coarsening fraction = 0.05" and "Refinement fraction = 0.95". After this initial step the AMR refinement and coarsening strategy in the ASPECT AMR algorithm is run every five time steps. See the ASPECT manual (Bangerth et al., 2017) for additional details.

For the PARTICLE method, we initially placed $256 = 16 \times 16$ uniformly distributed particles in each cell on the initial 10×10 coarse grid.

In Fig. 9 the computational results are shown at time $t = 9.81$ Myr: (a) and (b), the interface of the now deformed small box is reconstructed by coloring the $C = 0.5$ contour of the composition red. The interface in Fig. 9 (c) from the PARTICLE method is recovered by
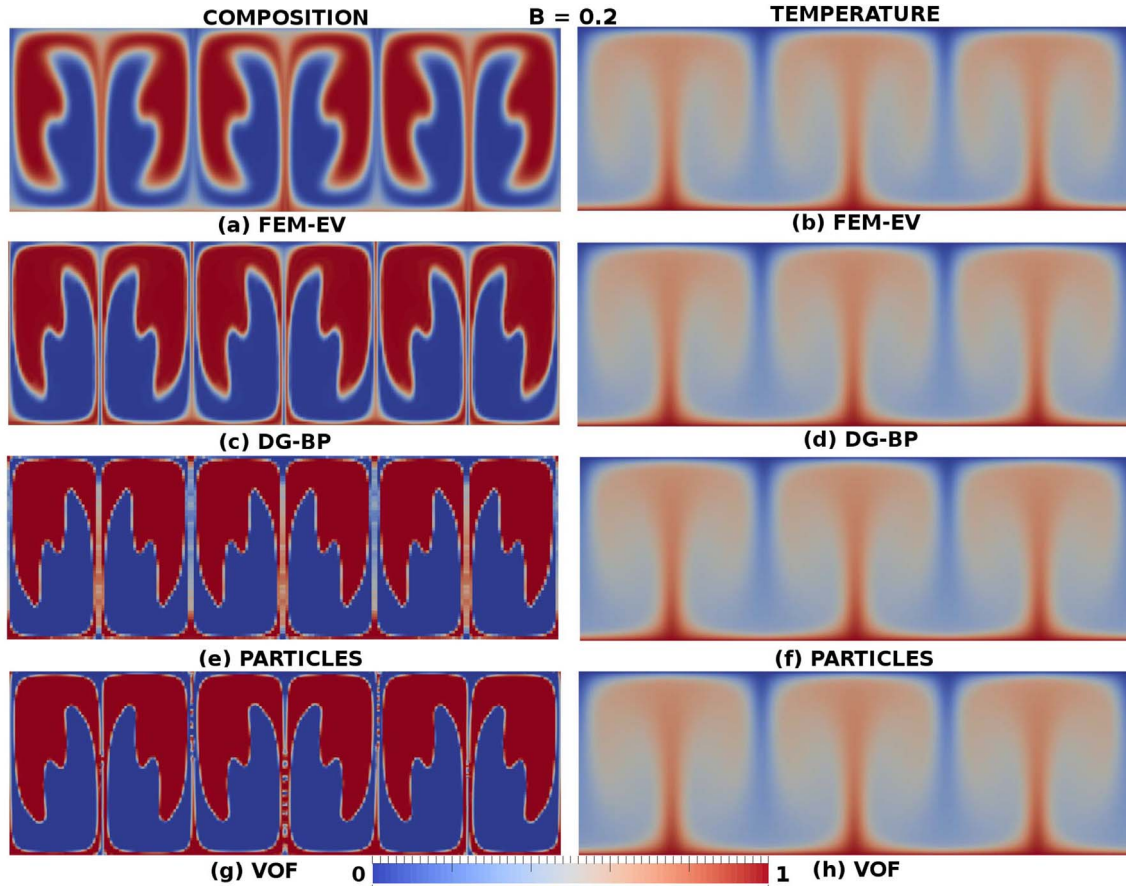
Fig. 14. The composition and temperature at $t = 0.0375$ with B = 0.2 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

looking at the contrast of gray particles, which carry the background density, and the black particles, which carry the density of the small box. The white spaces in Fig. 9 (c) are regions of the computational domain where there are no particles. In Fig. 9 (d), the interface shown is the interface that is reconstructed by VOF algorithm from the fractions at that time as described in Section 3.8. All of the numerical results displayed in Fig. 9 are looking similar, although, for this problem, we find that the interface produced by the VOF algorithm is the most simple, clean, and sharp.

In Fig. 10 (a)–(d), for each advection method we present the number of refined grid cells at the final computational time that were produced by the AMR algorithm. The number of cells required to compute the VOF results was the smallest (2653), followed by the PARTICLE algorithm (2878), DG-BP (3472) and FEM-EV (4195).

4.2. The van Keken Rayleigh–Taylor problem

In this section, we present our computations of the van Keken isoviscous Rayleigh–Taylor problem (van Keken et al., 1997). In this problem a less dense (buoyant) fluid lies beneath a denser fluid, with a perturbed interface between the two layers. In our computations we have used the parameter file van−keken−discontinuous. prm from the "The van Keken thermochemical composition benchmark" in the "Benchmarks" section of the ASPECT manual (Bangerth et al., 2017). The initial discontinuity between the two compositional/ density layers is given by

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

(74)

where $D = 0.9142$ is the width of the computational domain. This initial

condition has a sharp discontinuous interface along the curve

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

(75)

A comparison of our computational results computed with the four advection algorithms are shown in Fig. 11 (a)–(d). Note that computations made with the PARTICLE and VOF algorithms appear to be very similar and, furthermore, they *look* similar to Fig. 2(e) in (van Keken et al., 1997). In fact, given how unstable the Rayleigh–Taylor problem is, we find it surprising that the PARTICLE and VOF methods produce results that appear to be as similar to each other and to the results in the original computation of this problem as these three computational results do. In contrast, the DG-BP and FEM-EV results are similar to each other, except that the DG-BP algorithm produces less mixing of the two density layers than FEM-EV. This is highlighted by the green shading in the color map, since it is in the center of the color map and hence, displays those regions where the two density layers are being mixed via (numerical) diffusion the most. Note that these regions are at least several cell widths h in extent. In contrast, the PARTICLE and VOF computations exhibit regions in green that occupy (roughly) at most one cell. This is particularly apparent when one enlarges Figs. 11 (c) & (d) to the point at which some square cells are entirely green. It is important to recognize that this is because the numerical mechanisms that lead to particles with mixed values or volume fractions f_e with $0 < f_e < 1$, respectively, are fundamentally different than the *numerical* artificial diffusion that is a fundamental part of the FEM-EV algorithm. To the best of our knowledge these mechanisms have not been rigorously analyzed for the PARTICLE, VOF, and DG-BP methods. However, it is tempting to draw analogies with the rigorous analysis that has been conducted for finite difference methods such as *modified equation*
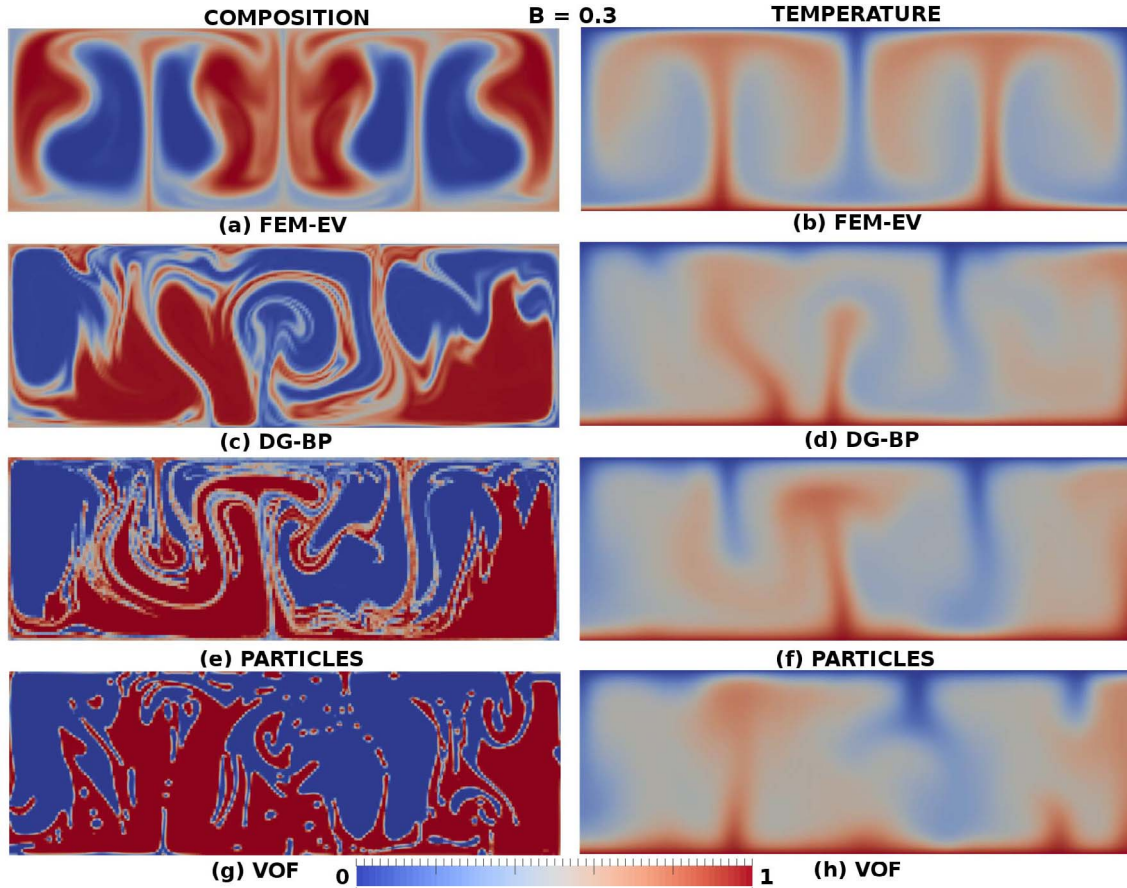
Fig. 15. The composition and temperature at $t = 0.075$ with B = 0.3 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

analysis (Warming and Hyett, 1974) or for *high-resolution advection algorithms*, such as is discussed in LeVeque (1996).

In addition, the DG-BP and FEM-EV computations differ from the PARTICLE and VOF computations in that they have a plume that begins near the bottom left-hand corner of the domain. Note that these results are very similar to those in Figs. 5 (a)–(d) in Samuel and Evonuk (2010) and the isoviscous results shown in Fig. 1 of Tackley and King (2003).

As mentioned in the discussion of this problem in Bangerth et al. (2017), we note that when one uses a finite element method with continuous piecewise basis functions to approximate the compositional field that determines the density differences that drive this flow, as is being done with the FEM-EV algorithm, the numerical method cannot represent a *discontinuous* function such as the curve in Eq. (75) with *continuous* shape functions. Therefore, although the initial condition is a discontinuous function $C(x,y,t)$ that consists only of the values zero and one as shown in Eq. (74), the initial conditions used in the computer model are in fact not discontinuous; they are the interpolated values of this discontinuous function on a finite element mesh. Furthermore, if the initial grid is not aligned with the interface of the discontinuous initial composition, then this is also true for the DG-BP method. On the other hand, both the particle and VOF methods are able to approximate an initially discontinuous compositional field of any shape *provided the perturbation is on the same scale (e.g., h) as the underlying FEM grid*.

4.3. Computations of thermochemical convection with density stratification

In order to examine the utility of the four advection algorithms described in Section 3 as tools for studying thermochemical convection we now present the results of our computations of the model problem for thermochemical convection with density stratification described in

Section 2. In these computations the Rayleigh number is *fixed* at Ra = 10^5 and we vary only the buoyancy ratio as follows: B = 0.0, 0.1, 0.2, ..., 1.0. 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.

We use one of the following two initial conditions for the temperature.

OUT-OF-PHASE TEMPERATURE INITIAL CONDITIONS

$T(\mathbf{x},0)$
$$= \begin{cases} (1-5y) + A\sin(10\pi y)\left(1-\cos\left(\frac{2}{3}k\pi x\right)\right) & \text{if } 0 \leqslant y \leqslant \frac{1}{10}, \\ (5-5y) + A\sin(10\pi y)\left(1-\cos\left(\frac{2}{3}k\pi x + \pi\right)\right) & \text{if } \frac{9}{10} \leqslant y \leqslant 1, \\ 0.5 & \text{otherwise,} \end{cases}$$
(76)

IN-PHASE TEMPERATURE INITIAL CONDITIONS

$$T(\mathbf{x},0) = \begin{cases} (1-5y) + A\sin(10\pi y)\left(1-\cos\left(\frac{2}{3}k\pi x\right)\right) & \text{if } 0 \leqslant y \leqslant \frac{1}{10}, \\ (5-5y) + A\sin(10\pi y)\left(1-\cos\left(\frac{2}{3}k\pi x\right)\right) & \text{if } \frac{9}{10} \leqslant y \leqslant 1, \\ 0.5 & \text{otherwise,} \end{cases}$$
(77)

where the period of the perturbation $k = 3$ and the amplitude of the perturbation $A = 0.05$. Note that $A = 0.05$ ensures that $0 \leqslant T(x,y;0) \leqslant 1$ throughout the entire computational domain.

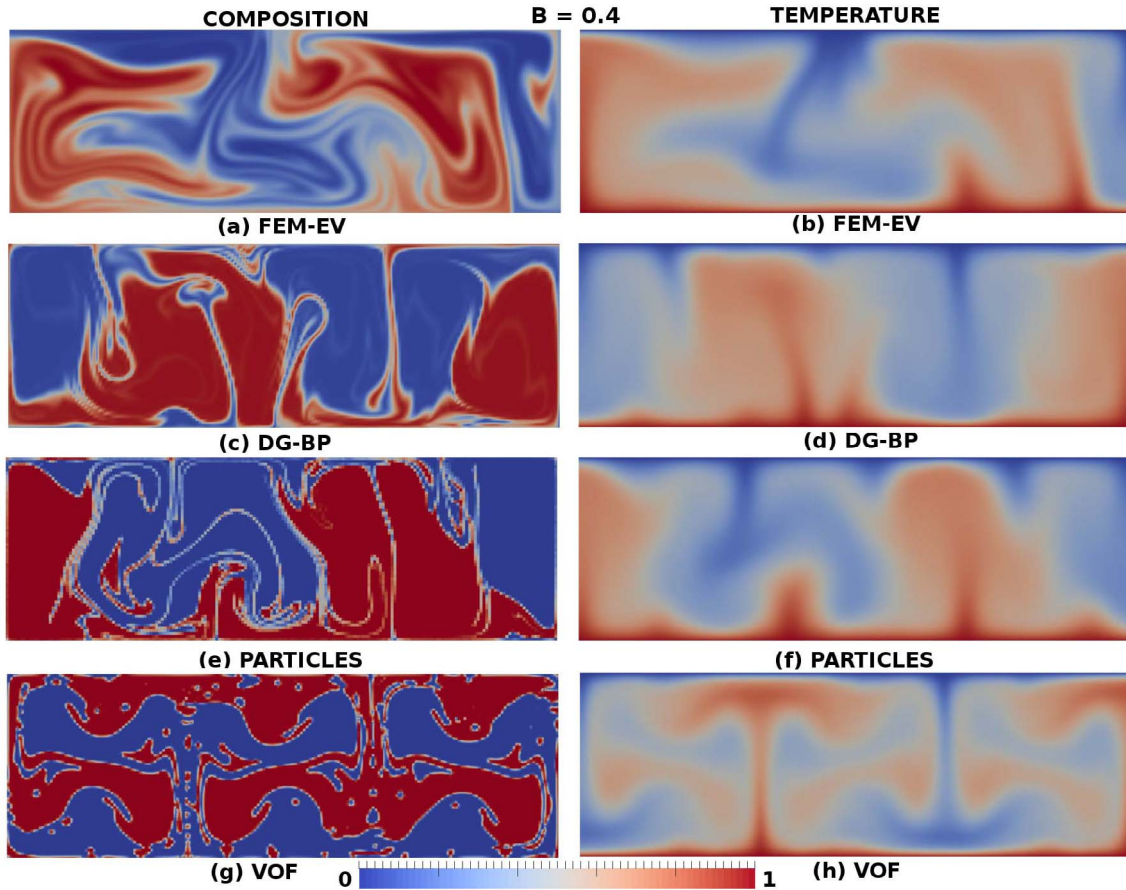The initial conditions for the composition are given by

Fig. 16. The composition and temperature at $t = 0.075$ with B = 0.4 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

$$C(x,y;t=0) = \begin{cases} 1 & \text{if} \quad 0 \leqslant y < \frac{1}{2}, \\ 0 & \text{if} \quad \frac{1}{2} \leqslant y \leqslant 1 \end{cases} \tag{78}$$

and the boundary conditions for the velocity and temperature are given by Eqs. (26)–(31).

For all of the computations shown here we use a fixed uniform grid with 192×64 square cells each with side $h = 64^{-1}$. We have also computed the same problems on a uniform grid with $h = 32^{-1}$; i.e., with 96×32 grid cells. The computational results on the coarser 96×32 grid are quite similar to those on the finer 192×64 grid, albeit at a lower resolution. We have thus determined that our computations on a uniform 192×64 grid are sufficiently well-resolved and accurate to allow us to arrive at the conclusions we discuss below.

Note that since we are using a second-order accurate (i.e., $O(h^2)$) (Q_2,Q_1) FEM element combination for the velocity, temperature and pressure, our grid resolution of $h = 64^{-1}$ *roughly* corresponds to a grid resolution of $h = 4096^{-1}$ for a first-order accurate (Q_1,Q_0) FEM element combination that is often used by researchers to approximate solutions of Eqs. (19)–(21) that govern incompressible convection in the Earth's mantle.

Based on trial computations with out-of-phase and in-phase initial conditions for the temperature (76) and (77) we used out-of-phase initial conditions for small values of B; namely, $0.0 \leqslant B \leqslant 0.2$. For B = 0 we must obtain classic Rayleigh-Bernard single layer convection cells. These have out-of-phase temperatures near the boundaries. Similar results were obtained for B = 0.1 and 0.2. At B = 0.3 our trial computations showed that the compositional boundary was influencing the flow. In-phase initial temperature conditions gave a well defined transition from single-layer to two-layer convection in the range $0.3 \leqslant B \leqslant 1$ where out-of-

phase boundary conditions tended to block the transition. Since the primary purpose of our studies is to illustrate the alternative behaviors obtained for the four numerical methods the transition was better defined with the in-phase conditions and we present these results.

4.3.1. B = 0.0,0.1,0.2 *with out-of-phase initial conditions*

In this first set of computations, we used the four advection algorithms with the out-of-phase temperature initial conditions in Eq. (76) to compute our model problem (19)–(31) with the initial composition field given by (78). The numerical results are shown at time $t = 0.0375$ in Figs. 12–14 for Ra = 10^5 with B = 0.0,B = 0.1, and B = 0.2, respectively. In each figure, we display a plot of the composition C on the left and the temperature T on the right where, in each row, the two subfigures are labeled with the advection algorithm we used to compute that particular pair of figures.

At this Rayleigh number the convection is essentially independent of time and kinematic mixing of the two layers occur in each cell.

For these small values of B; namely, B = 0.1 and B = 0.2 and relatively short times $t \leqslant 0.0375$, both the composition and temperature computed with each of the four advection algorithms appear to be nearly identical, except that the FEM-EV results for the compositional field are more diffusive than the results computed with the other three methods. In fact, if one only examines the plots of the temperature T, there is hardly any discernible difference between the four computational results at this relatively early time $t = 0.0375$. In addition, note that all of the computational results are organized into (roughly) steady (i.e, independent of time) single-layer thermal convection with three counter rotating cells. In this sense, for B $\leqslant 0.2$ we conclude that for the out-of-phase initial conditions (76) the computational results approach the limit B $\rightarrow 0.0$ in a uniform, organized manner.
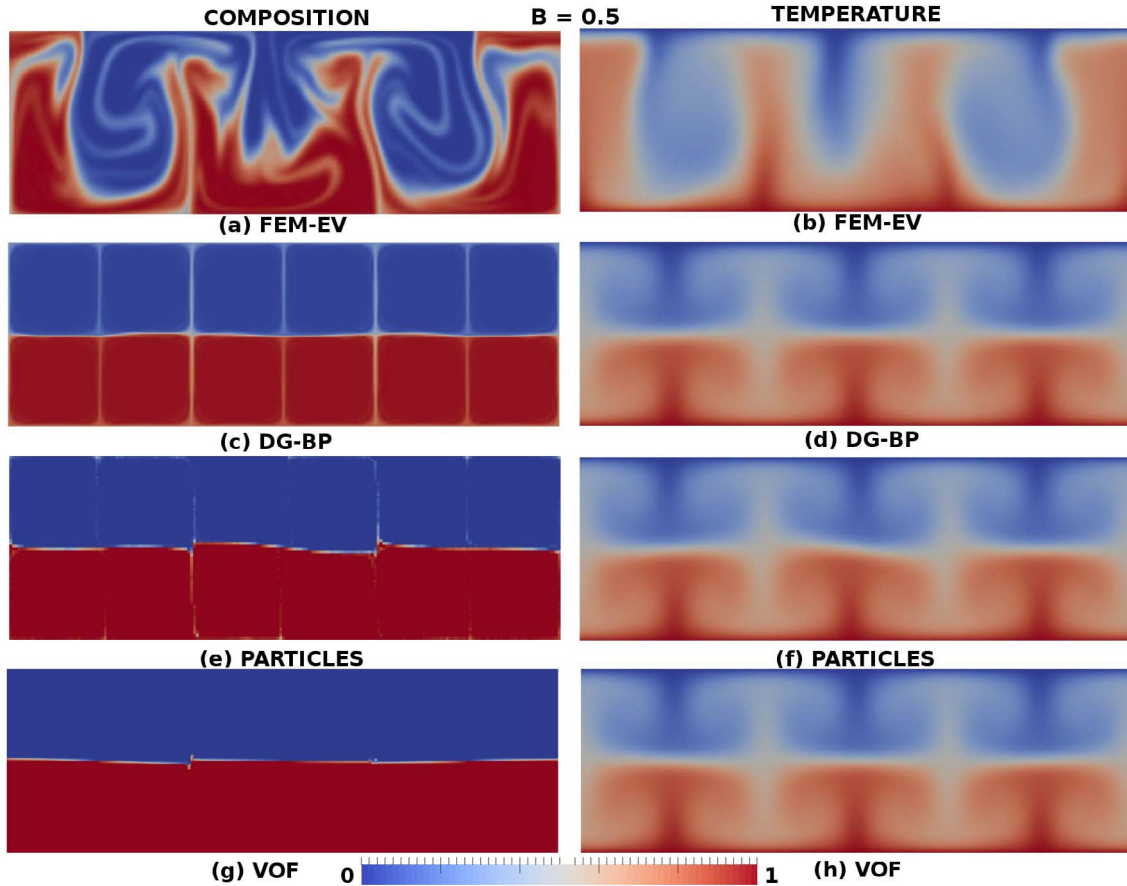
Fig. 17. The composition and temperature at $t = 0.075$ with B = 0.5 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

4.3.2. B = 0.3 and 0.4 with in-phase initial conditions

In our second set of computations – as well as in all of the remaining computations of the model problem presented in Section 2 – we use the in-phase temperature initial conditions (77) and present our computational results at the (nondimensional) time $t = 0.075$. For B = 0.3 and 0.4 our results are displayed in Figs. 15 and 16. It is apparent from the plots in these figures that the composition and temperature solutions produced by each of the four advection algorithms are quite different in appearance, both from the computations with $0.0 \leqslant B \lesssim 0.2$ in Section 4.3.1 and from one another at B = 0.3 and B = 0.4. This is because for $0.3 \lesssim B \lesssim 0.4$ the system of PDEs (19)–(31) with initial conditions (77) and (78) is unstable. In other words, for $0.3 \lesssim B \lesssim 0.4$ the problem (19)–(31) is mathematically ill-posed.

Note that, with the exception of the FEM-EV results in Figs. 15 (a)–(b), the computational results in Figs. 15 and 16 exhibit mixing on a much smaller scale than the computations shown in Section 4.3.1. The compositional barrier impedes single layer convection, but the passive kinematic mixing in individual cells continues to dominate. Eventually, the flow mixes on increasingly smaller scales but the flow is still dominantly single layer convection. We attribute this small scale mixing behavior to a transition to a time dependent flow that enhances kinematic mixing.

4.3.3. B = 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0 with in-phase initial conditions

Finally, for $0.5 \lesssim B \lesssim 1.0$ with the in-phase initial conditions our results exhibit an increasingly strongly stratified flow as $B \to 1.0$. Furthermore, once the flow has reached a stratified two layer convection pattern for some $B \geqslant 0.5$ it remains so as $B \to 1.0$.

With the exception of the FEM-EV algorithm, the plots in Fig. 17 show that the composition and temperature solutions computed with each of the other three algorithms produce a steady flow with two-layer

thermal convection. However, the FEM-EV algorithm produces computational results that are far too diffusive to accurately model this stratified flow at this grid resolution. See Section 5 for a detailed analysis of this issue.

In this regime it is possible for us to use any of the other three algorithms for modeling the location of the compositional field. However, Our computations indicate that the VOF algorithm is optimal in this regime; i.e., for the range of values of B for which the compositional density fields are stably stratified or nearly stably stratified. The DGBP and particle methods can certainly be used in this regimes, but they exhibit more numerical artifacts than the VOF algorithm in the strongly stratified regime, although, for the particle method, this may not be immediately evident from Fig. 22. We discuss this matter in detail in Section 5.3 below.

4.3.4. Summary of the computations of the model problem

These results are all for Ra = 10^5 based on the full layer thickness. In the range $0.0 \leqslant B \leqslant 1.0$ we find that there are three regimes depending on the value of the buoyancy parameter B. With no chemical buoyancy B = 0, a time independent cellular convection pattern develops – the classic Rayleigh-Bénard flow – and there is no mixing between adjacent cells. For small chemical buoyancy, B = 0.1 and 0.2, this flow is maintained. Kinematic mixing takes place in the individual cells but is relatively slow. For B = 0.3 and 0.4 the flow is quite different. The chemical buoyancy results in unsteady flows that greatly enhance the kinematic mixing but do not block the vertical flows between the upper and lower layers. For $0.5 \leqslant B \leqslant 1.0$ the chemical stabilizing buoyancy is sufficiently strong to block vertical flows between the upper and lower layers, the compositional boundary is essentially flat, and thermal convection occurs in two independent layers.
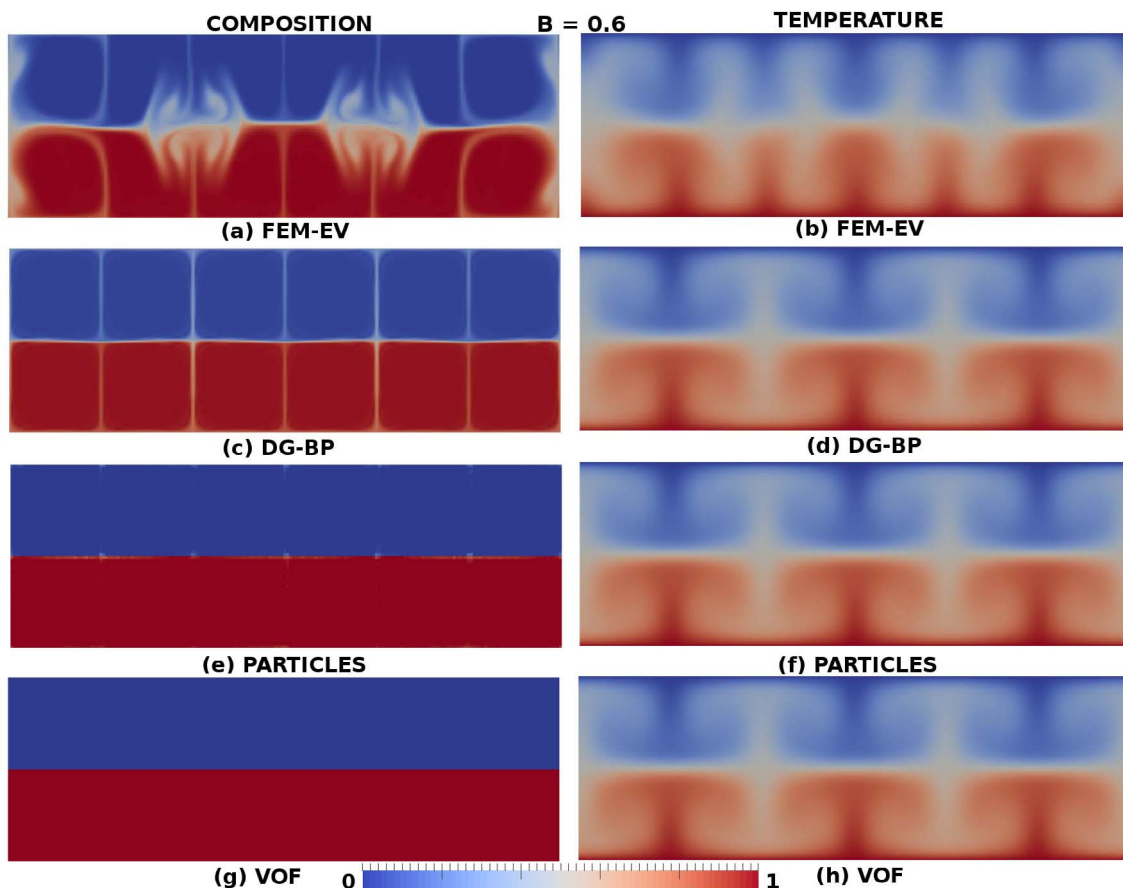
COMPOSITION   **B = 0.6**   **TEMPERATURE**

(a) FEM-EV   **(b) FEM-EV**

(c) DG-BP   **(d) DG-BP**

(e) PARTICLES   **(f) PARTICLES**

(g) VOF   0   1   **(h) VOF**

Fig. 18. The composition and temperature at $t = 0.075$ with B = 0.6 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

In the following section, we present a more detailed comparison of the efficacy of the four algorithms for computing flows with the distinct types of dynamics shown in Sections 4.3.1–4.3.3.

5. Discussion

Our numerical results in Section 4 demonstrate the capabilities and limitations of the four algorithms we have used to model the motion of the compositional interface. A detailed discussion of these capabilities and limitations follows. Again, all computations were made with Ra = 10^5 where Ra is based on the single layer thickness.

5.1. The performance of the algorithms for $0 \leqslant B \lesssim 0.2$

For B = 0.0 the compositional boundary is completely passive, since $\Delta\rho = 0$. This is the classic Rayleigh-Bénard problem. Single layer thermal convection occurs at Ra = 10^5. At this Rayleigh number the convection is independent of time and passive kinematic mixing of the two layers occur in each of three distinct cells. For B = 0.1 and 0.2 the compositional barrier impedes single layer convection, but the passive kinematic mixing in individual cells continues to dominate and the flow remains single layer and is also organized into three counter rotating cells. In summary, the flow for $0 \leqslant B \lesssim 0.2$ appears to be a continuous extension of the flow at B = 0.0.

For times $0.0 \leqslant t \leqslant 0.0375$, where $t = 0.0375$ is the time at which computational results are shown in Figs. 12–14, with the out-of-phase initial conditions, all four advection algorithms yield very similar results. The plot of the compositional field made with the FEM-EV algorithm is noticeably more diffusive than the other plots of the compositional field. However, the features of the flow field are still clearly

identifiable as being nearly identical to those found in the other plots. Except for the excessive numerical diffusivity created by FEM-EV, it is difficult to recommend using one algorithm over another in this regime.

5.2. The performance of the algorithms for $0.3 \lesssim B \lesssim 0.4$

For B = 0.3 and 0.4 the flow becomes unstable but remains dominantly single layer convection. We attribute the dynamics of this flow to the transition to a time dependent flow that enhances the kinematic mixing.

In the following discussion we will *assume* that the flow field is undergoing *non-diffusive* mixing in such a way that, as the time t increases, the two compositions can form what we will refer to as *tendrils*, defined in the sense of Ottino (1989, Section 7.2). Note that when Ottino defines a tendril he is specifically referring to two dimensional flows, such as those we are computing here. Furthermore, when we refer to a tendril here we have in mind the type of fluid structure that appears on the right-hand side of the illustration in Ottino (1989, Figure 7.3.13). Similar structures *of widely varying widths* appear in numerous photographs of experiments in his book; e.g., (Ottino, 1989, Figure 7.5.5 (a)). Many, if not all, of these tendrils tend to develop increasingly smaller cross sections as $t \to \infty$. It is apparent from Figs. 15 and 16 that, with the exception of Fig. 15 (a), this is also occurring in the images of the computations that appear in these figures.

Now, since by assumption the two compositions are undergoing non-diffusive mixing, it is apparent from Figs. 15 and 16 that the FEM-EV, DGBP, and PARTICLE algorithms all mix the compositional field within grid cells that contain a portion of both compositions *without maintaining a sharp interface* between the two compositions. This is readily apparent since for each of these methods some regions of the
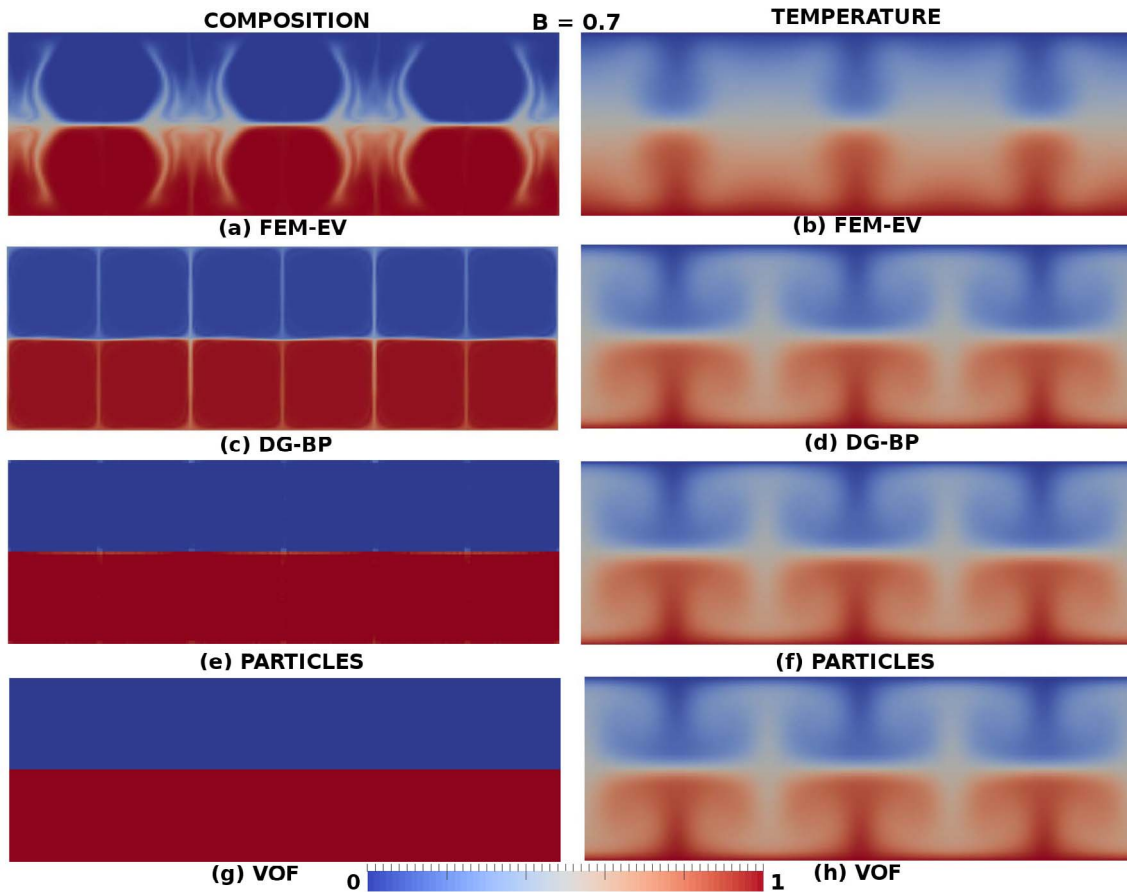
Fig. 19. The composition and temperature at $t = 0.075$ with and B = 0.7 Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

computational domain are not just red or blue, but are colored with an intermediate color. On the other hand, since it is an interface tracking algorithm, the VOF method maintains a sharp interface between each composition. This is readily apparent in Fig. 23. To summarize, the FEM-EV, DGBP, and PARTICLE methods *are not designed* to separate the two compositions with an explicit, sharp interface. They therefore mix the two compositions in some way, be it numerical diffusion or by some other numerical mechanism.

The remainder of this discussion is only concerned with the problem of computing the *interface* between the two compositions, by which we mean the boundaries or edges of the tendrils that separate the two compositions. Thus, we are *only* discussing the VOF interface tracking algorithm here..

Fig. 24 contains an illustration of a 3×3 block of cells containing the end of an idealized tendril of composition 1, say, *inside* a blue boundary. Note that both sides (boundaries) of this tendril lie in the center cell of the 3×3 block of cells; i.e., the tendril is entirely contained in the cell marked Ω_e. Thus, it is not possible to reconstruct the boundary of this tendril accurately *on this grid* with the VOF interface reconstruction algorithm described in Step 1 of the itemized list in Section 3.8. Instead, we must decrease the grid width h of the 3×3 block of cells by at least a factor of two. This is an excellent example of a problem for which AMR is particularly well-suited for use in combination with the VOF method. In Fig. 23 it is apparent that some tendrils of both compositions have formed with widths that are on the order of the cell size h. Hence, these computations require either a grid with a uniformly higher resolution or AMR in order to produce an accurate computation of this flow at this time.

In conclusion, given that the width w of some, if not all, of the tendrils in the computation go to zero as $t \to \infty$, if one wants to resolve

the interface between the two sides of each tendril in the computation then we must have the smallest grid size $h_{min} \to 0$ as the width of the thinnest tendril goes to zero. This is certainly true for the VOF method described in our paper, since it only produces a linear approximation to the interface in each cell.

5.3. The performance of the algorithms as B → 1.0

For B → 1.0 with B $\gtrsim$ 0.5 the initial compositional barrier between the upper and lower half of the domain (i.e., the initial stratification of the compositional fields) persists indefinitely. For convenience and brevity we will focus the following discussion on the case B = 1.0.

The numerical results we computed with the FEM entropy viscosity-based method are far too diffusive to produce meaningful results at a grid resolution of 192×64 square cells, a resolution at which the other three algorithms produce strongly stratified two layer flow with three rectangular cells in each layer, each of which contains two square subcells. Furthermore, the boundary of these subcells contains a counter rotating flow between each of the square subcellular structures, including at the interface between the square subcells along the boundary between compositions. Furthermore the temperature fields produced by the DGBP, PARTICLE, and VOF methods are virtually identical whereas the temperature field produced by the FEM-EV method appears to be underdeveloped. In Section 5.3.1 below we argue that this is due to the additional artificial diffusion that is added by the entropy viscosity method in order to stabilize the advection algorithm.

We note that the aforementioned cellular structure is visible in the image of the composition from the computation made with the DGBP method but not in the images from the computations we made with the PARTICLE and VOF methods in Fig. 22, although it is visible in the
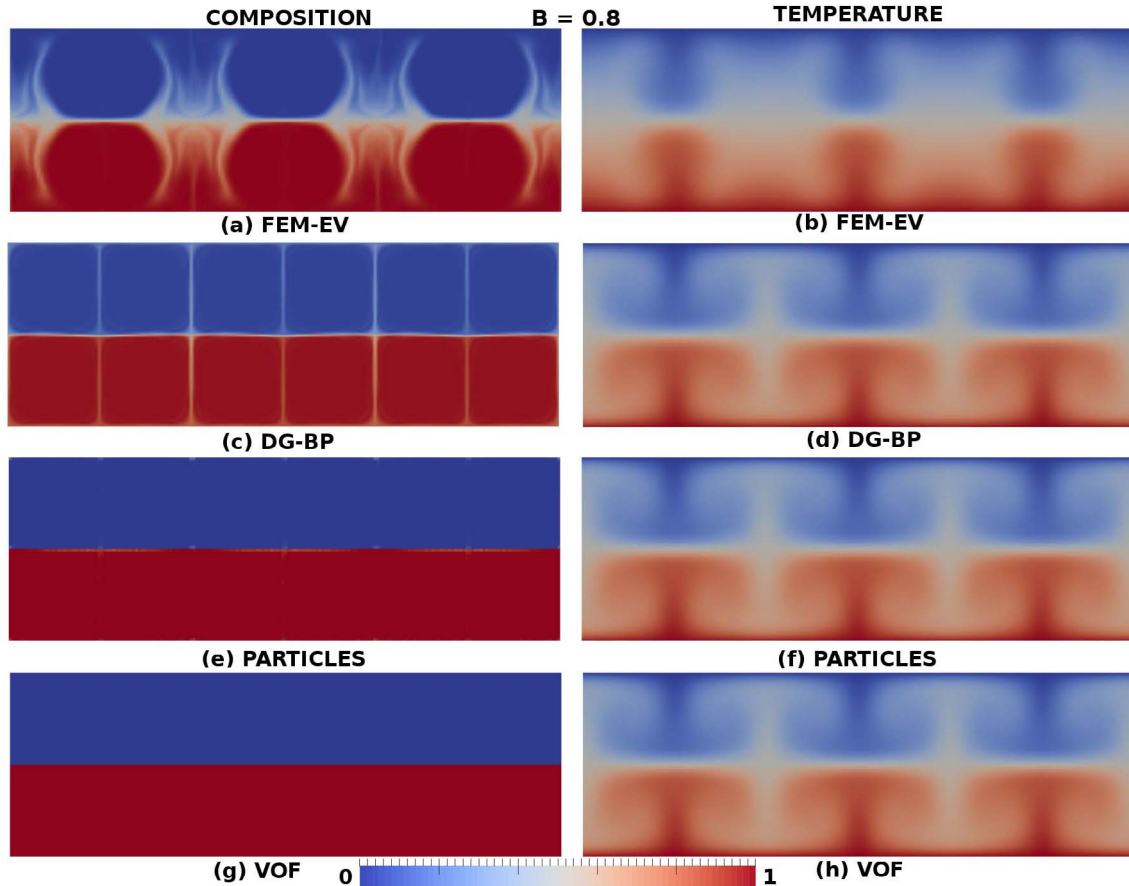
Fig. 20. The composition and temperature at $t = 0.075$ with B = 0.8 and Ra = 10^5 computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

image of the composition produced by the PARTICLE method for B = 0.5 in Fig. 17 and in Figs. 27 and 28 below.

On the other hand, the compositional field we computed with DGBP, PARTICLE, and VOF methods have distinct differences. Computations made with the DGBP method exhibit some amount of each compositional field that is (numerically) entrained within the other compositional field and advects along the boundary of the convection cells in the wrong compositional domain. Although this may not be immediately apparent in Fig. 22 (e) the particle method exhibits a similar numerical artifact, in which some particles representing the denser fluid are entrained in the upper, less dense fluid and advect along the boundary of the convection cells in the wrong compositional domain and, similarly, a small number of particles representing the less dense fluid are entrained in the lower, denser fluid and are advected along the boundary of the convection cells but in the wrong compositional domain. We will revisit this issue in detail in Section 5.3.3 below. Finally, we found that the VOF method maintains a sharp interface between the two compositions on a subgrid scale throughout the computation.

5.3.1. The effect of entropy viscosity on the FEM-EV computations

The numerical results for B = 1 at Ra = 10^5 in Fig. 22(a)–(b) show that the FEM-EV advection algorithm produces extremely diffusive temperature and composition fields on a grid with 192×64 cells, a resolution at which the other three algorithms produce well resolved results. In stark contrast with the results of the other three algorithms; the results for both the composition and temperature fields we computed with the FEM-EV algorithm are completely dissimilar from those computed with the other three algorithms. In particular, it is apparent that temperature we computed with the FEM-EV algorithm is

far more diffusive than the temperature we computed with the other three algorithms. In fact, it is difficult to tell the temperature fields computed with the other three algorithms apart, but the temperature field computed FEM-EV is very clearly different. Similarly the composition we computed with the FEM-EV algorithm is completely different from the composition field we computed with the other three algorithms. In particular, the boundaries of the convection cells in both the upper and lower regions exhibit an unacceptably large amount of diffusion.

Among the four algorithms, the FEM-EV and DGBP algorithms are both based on the Galerkin approach, which is a method for approximating the true solution in the weak formulation of the composition Eq. (37). Since we used the same time discretization for each of these two advection algorithms, the reason the results between the FEM-EV and DGBP advection algorithms are so completely different must be due to the differences in spacial discretization.

The major difference between the standard FEM and DG methods is that DG allows for discontinuities between elements. Therefore, DG is more appropriate for problems with strong discontinuities or large gradients (Cockburn et al., 2000) such as the boundary between the compositional fields. In addition, in order to stabilize the numerical method, the FEM-EV algorithm includes entropy viscosity, which adds an artificial diffusion term with a cell-wise constant artificial diffusivity $\nu_h^E(C)$ to the weak form of the advection Eq. (37). This leads to Eq. (38). Thus, the FEM-EV method more closely approximates solutions of (38) than solutions of the true compositional advection Eq. (37). With regards to the computations with $0.5 \lesssim B \leqslant 1.0$, this is probably the predominant reason for the discrepancy between the FEM-EV results and the results we obtained with the other three algorithms. We will now test this hypothesis.
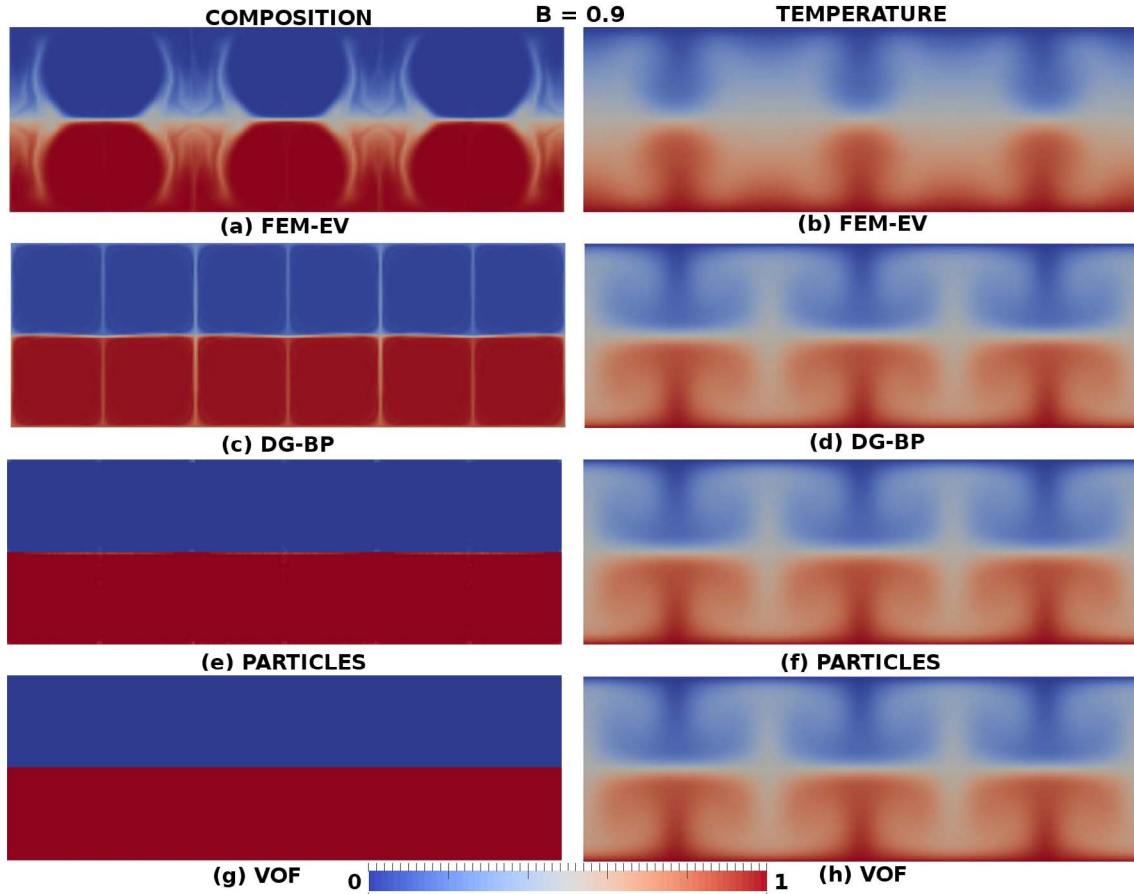
Fig. 21. The composition and temperature at $t = 0.075$ with B = 0.9 and Ra = 10^5 and computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

Here we follow the definition and discussion of the entropy viscosity in Guermond et al. (2011). On each cell Ω_e, the entropy viscosity is defined by

$$\nu_h^E|_{\Omega_e} = \frac{\gamma\, h^2 \|r_E(C)\|_{\infty,\Omega_e}}{\|E(C) - E_{avg}\|_{\infty,\Omega}} \tag{79}$$

where γ is a constant parameter that is chosen by the user,

$$E(C) = \frac{1}{2}\left(C - \frac{1}{2}C_{min} - \frac{1}{2}C_{max}\right)^2.$$

is the *composition entropy*,

$$E_{avg} = \frac{1}{|\Omega|}\int_\Omega E(C)$$

is the space-average of the compositional entropy, and $r_E(C)$ is the entropy residual function of C and $E(C)$. The residual $r_E(C)$ is zero if applied to the exact solution of the composition equation, in which case there is no artificial diffusion for Ω_e; i.e., $\nu_h^E|_{\Omega_e} = 0$. However $r_E(C)$ will be non-zero when applied to the numerical approximation and furthermore it will be large in areas where the numerical approximation is poor, such as near strong gradients.

Note that the value of entropy viscosity decreases as the grid size h decreases. Therefore, in order to reduce the amount of artificial diffusion for a fixed polynomial basis, say a Q_2 finite element, one must reduce h. See (Guermond et al., 2011) and (Kronbichler et al., 2012) for further details.

In order to demonstrate how the entropy viscosity introduced by the FEM-EV algorithm is affected by the grid size h we made two computations, one on a grid with 192×64 square cells and another on a grid with 768×256 square cells. The computational results appear in

Fig. 26. Note that we used a log scale color map for the plots in Fig. 26 (b) & (d).

The maximum value of the entropy viscosity is reduced from a maximum value of $O(10^{-3})$ on the grid with 192×64 cells (Fig. 26 (b)) to a maximum value of $O(10^{-4})$ on the grid with 768×256 cells (Fig. 26 (d)). Thus, Figs. 25 and 26 clearly show that using a finer mesh with FEM-EV will result in a compositional field that more nearly looks like the compositional field we computed with DGBP on the coarser 192×64 grid at time $t = 0.075$.

However, if we run the numerical computation to a longer time t, we must account for the accumulated error due to the artificial diffusion term on a relatively large grid size h. In other words, if the final computation time t is fixed, we can determine a small enough h such that the artificial diffusion will not significantly change the final solution as compared to DGBP. However, to have numerical solutions with the same accuracy as the other three advection algorithms, one must use a much small grid size h.

5.3.2. The performance of the DGBP algorithm

In contrast, in the DGBP advection algorithm we first discretize the problem by applying a standard DG method with an upwind monotone flux. This does not explicitly add an artificial diffusion term to the advection equation. We then use a BP limiter in a post processing step in order to reduce or eliminate overshoot and undershoot in the DG solution near discontinuities in the composition variable C. In Zhang and Shu (2010), it is shown that the BP limiter does not reduce the accuracy of the original DG solution. Also, numerical examples of the advection of non-diffusive fields in solid Earth geodynamics in He et al. (2017) show that the DGBP solutions preserve a much sharper boundary as compared to the FEM-EV method on the same mesh with the same grid size.
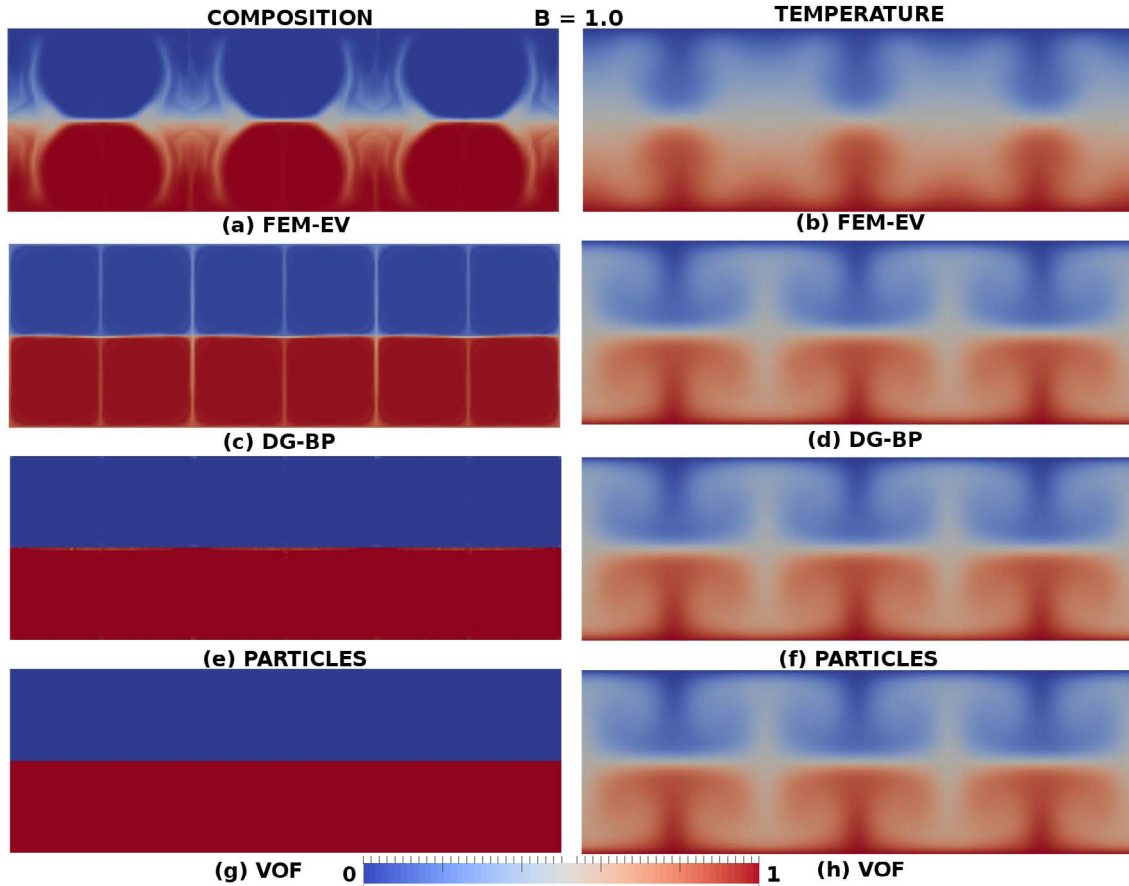
Fig. 22. The composition and temperature at $t = 0.075$ with B = 1.0 and Ra = 10^5 and computed with the (a)-(b) FEM-EV, (c)-(d) DGBP, (e)-(f) PARTICLE, and (g)-(h) VOF advection algorithms.

However, compared to the VOF interface tracking algorithm at B = 1 the DGBP advection algorithm does not maintain a sharp compositional boundary, across which the two compositional fields do not mix. The ability to maintain sharp boundaries through time in a computation is likely to influence conclusions related to the entrainment of compositional signals from boundary layers into rising thermal plumes. The computational results in He et al. (2017) indicate that coupling the DGBP method to Adaptive Mesh Refinement (AMR) will limit the computational cost as mixing proceeds, at least until the mixing leads to compositional heterogeneity at all scales.

Fig. 22 demonstrates that at the (dimensionless) time $t = 0.075$ the composition and temperature fields we computed with the DGBP, VOF and PARTICLE algorithms are quite similar to each other, especially the temperature fields. The flow is organized into steady-state, discrete cells. Furthermore, the temperature fields we computed with these three algorithms are virtually indistinguishable from one another at each of the times shown. Note, however, that the DGBP algorithm has visible white bands at the interface between each of the six counter rotating portions of the three convection cells. These are locations where the values of the composition field is approximately $C = 0.5$ indicating that some of the composition field with value $C = 1$ that was initially in the subdomain Ω_u has been entrained into the flow in Ω_l and vice versa. Although this enables one to easily see the convection cells it is a numerical artifact; i.e., due to numerical errors inherent in the DGBP advection method.

5.3.3. The performance of the PARTICLE algorithm

Recall that we denote the computational domain by Ω as shown in Fig. 1. We now divide Ω into two disjoint complementary regions Ω_l and Ω_u, defined by,

$$\Omega_l = \{(x,y): 0.0 \leqslant y < 0.5\},$$
$$\Omega_u = \{(x,y): 0.5 \leqslant y \leqslant 1.0\}, \tag{80}$$

where the subscripts 'l' and 'u' stand for 'lower' and 'upper', respectively.

In Figs. 18(e)–22(e), the particles that are initially in Ω_l are red and the particles that are initially in Ω_u are blue. Furthermore, due to the high density of $16 \times 16 = 256$ particles per cell in the computations shown in these figures the entire lower subdomain Ω_l appears to remain uniformly red and the entire upper subdomain Ω_u appears to remain uniformly blue. However, upon close examination it is possible to see a small, light colored region along the compositional boundary at $y = 0.5$ and, in addition, a lighter color centered on the stagnation points that are at the center of the regions of largest shear in these computations; namely, at the six points $(0.0, 0.5), (0.5, 0.5), (1.0, 0.5), \ldots, (3.0, 0.5)$. (See Fig. 1 for the coordinate system.) This is an indication that some particles from Ω_l, the red region, have been entrained and advected upward into Ω_u, the blue region, and vice versa.

It is very difficult to see the magnitude of this feature in Figs. 18(e)–22(e), since the color map for a particle density of 256 particles per cell overwhelms the color of the particles in the region with the opposite color. One can see this phenomenon more clearly in Fig. 27. In Fig. 27 (a) it is apparent that some of the blue particles have been advected into the red region Ω_l and vice versa. If the flow with B = 1 in Fig. 27, is strictly stratified into two layers, then physically this should not occur. One can see this more clearly in the VOF computations in Figs. 18 (g)–22 (g); in which the compositional fields in the two subdomains Ω_l and Ω_u do not mix at all.

In order to investigate whether this 'leakage' of particles is a numerical artifact we computed the same problem, but this time for B = 10, in order to determine whether the number of particles
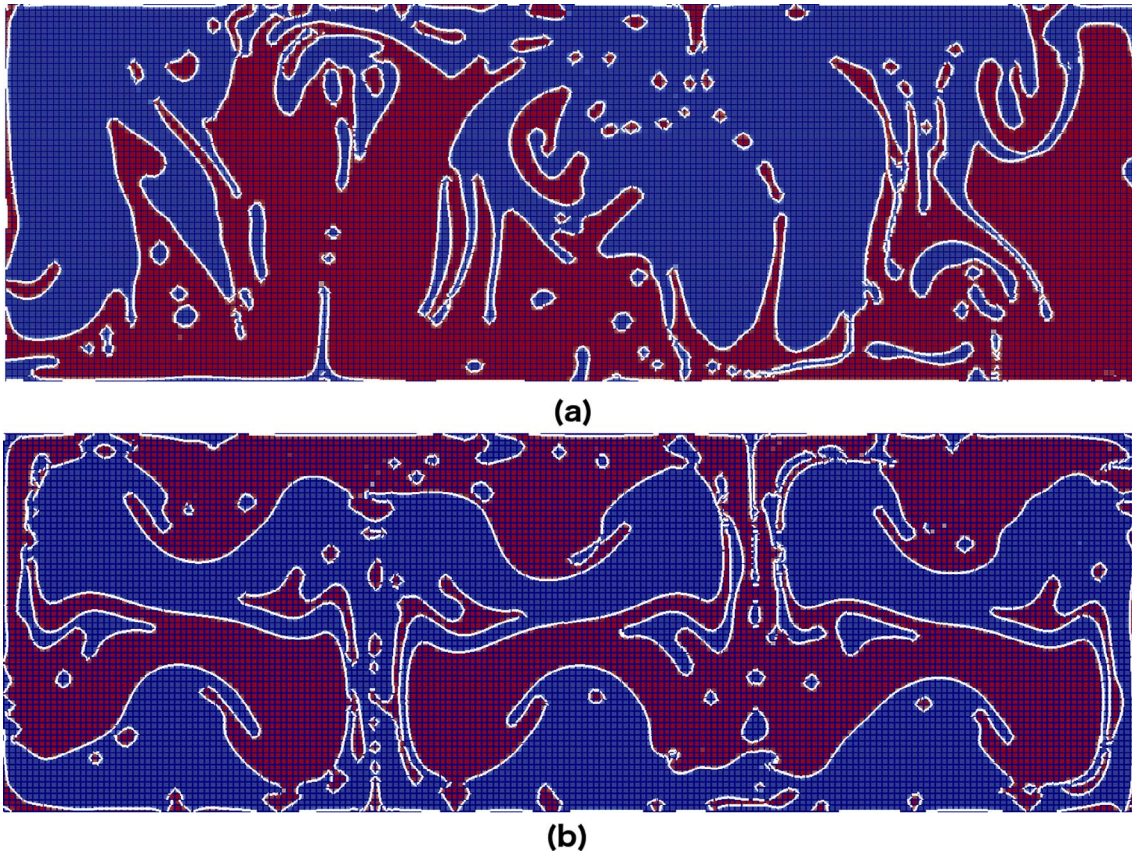
(a)

(b)

Fig. 23. The results of the VOF computations for (a) B = 0.3 and (b) B = 0.4 from Figs. 15 and 16 with the reconstructed interface in white, plotted over the underlying uniform grid of 192 × 64 square cells.
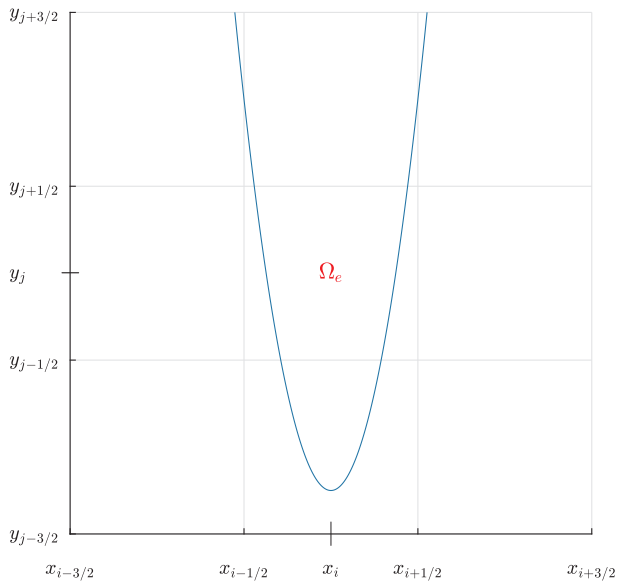

Fig. 24. An idealized depiction of the end of a tendril containing composition 1, say, *inside* a blue boundary. Note that both sides of this tendril lie in the center cell of the 3 × 3 block of cells. In other words, the tendril is entirely contained in the cell marked Ω_e, thereby requiring grid refinement (e.g., with AMR) in order to reconstruct the edges of this tendril accurately with the VOF interface reconstruction algorithm that we have used in this article.

entrained in the opposite domain depends on the strength of the flow stratification; i.e., on increasing the value of B or, equivalently, $\Delta\rho$. It is apparent from Fig. 28 and the data in Table 4 that the *total* number of particles that are entrained in regions of strong shear and are then

advected into the opposite region is only slightly dependent on the value of B; i.e., decreases slightly as B increases.

We also made the same two computations with B = 1 and B = 10 at Ra = 10^5 on a uniform grid with 192 × 64 cells, but this time with only 4 × 4 = 16 particles per cell, so as to ensure that the large number of 256 particles per cell was not the cause of the leakage of particles into the opposite domain. Our results are shown in Table 5. Based on this data we conclude that there is not a significant difference in the percentages of particles that leak into the opposite domain when we start with 16 particles per cell instead of 256 particles per cell.

Somewhat surprisingly, the number of blue particles that move down from Ω_u into Ω_l appears to be larger than the number of red particles that move up from Ω_l into Ω_u. Also, we note that the sum of blue and red particles that move into the opposite domain appears to be increasing in time. Nevertheless, we conclude that the leakage of particles is likely a numerical artifact due to small errors in the particle advection algorithm that are amplified in regions of strong shear. This feature of the particle advection algorithm is interesting and deserves study.

5.3.4. The performance of the VOF algorithm

In contrast, since it is an interface tracking method, the VOF algorithm maintains a sharp horizontal boundary at $y = 0.5$ in Figs. 18(g)–22(g) and a very nearly horizontal boundary at $y = 0.5$ in Fig. 17 (g). In our view this nearly, but not perfectly, horizontal boundary at $y = 0.5$ in the latter figure is due to the fact that B = 0.5 is close to the value of B at which the flow transitions from single layer disorganized (i.e., unstable) kinematic mixing to two layer stratified flow which occurs at some value of B with B < 0.5 when Ra = 10^5.

Furthermore, since the VOF algorithm is only tracking a set of co-dimension one in a two dimensional flow (i.e, a one-dimensional curve in a two dimensional flow) this problem is an ideal candidate for us to
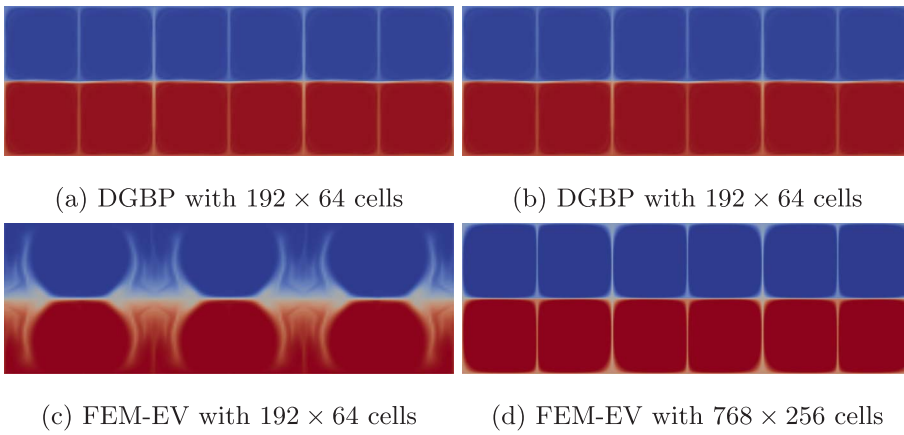
(a) DGBP with 192 × 64 cells

(b) DGBP with 192 × 64 cells

(c) FEM-EV with 192 × 64 cells

(d) FEM-EV with 768 × 256 cells

Fig. 25. The left column contains a comparison of the compositional field at $t = 0.075$ with Ra $= 10^5$ and $B = 1.0$ computed with (a) DGBP (c) FEM-EV on a grid with 192 × 64 square cells. The right column contains the same comparison except that the (d) FEM-EV result was computed on a much 16 times finer grid – 768 × 256 cells – than the DGBP result in (a) and (b).
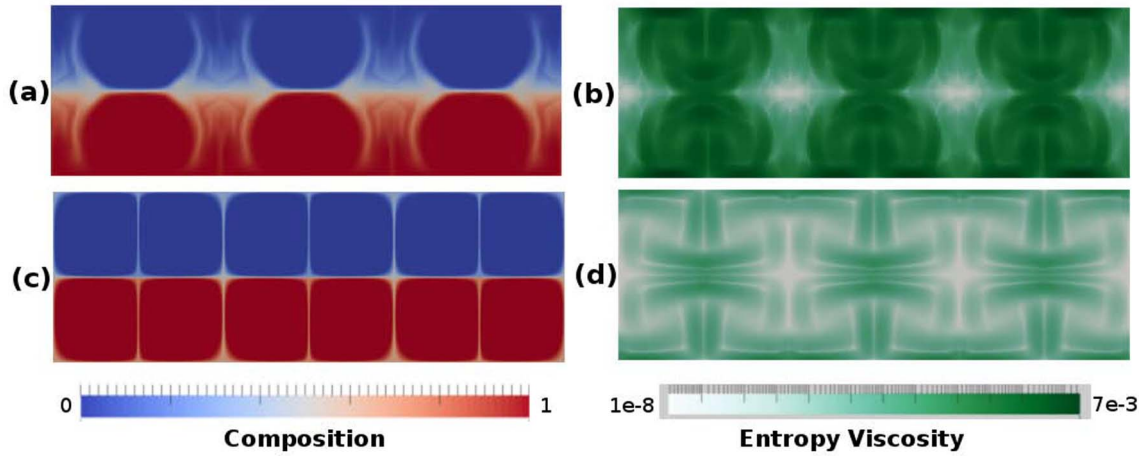

Fig. 26. The composition (left) and entropy-viscosity (right) at time $t = 0.075$, computed with FEM-EV for Ra $= 10^5$ and B $= 1$. The figures in the first row were computed on a grid with 192 × 64 cells, whereas the figures in the second row were computed on a grid with 768 × 256 cells. The FEM advection algorithm with entropy viscosity is so diffusive that one must use 16 times as many cells and 4 times as many time steps or roughly, 64 times as much work, in order to achieve approximately the same resolution as the other three advection algorithms in Fig. 22.
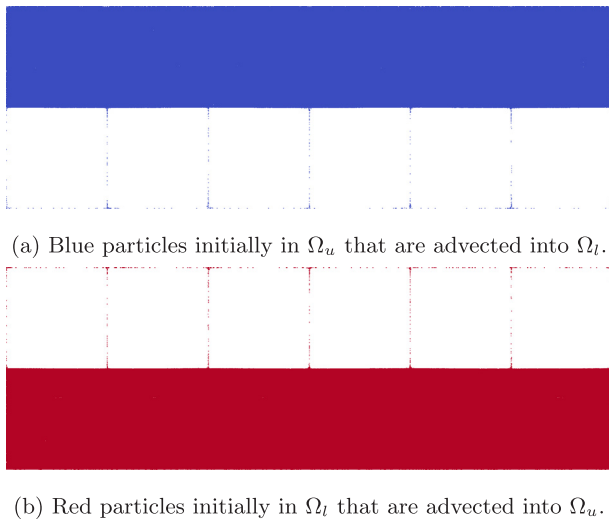

(a) Blue particles initially in Ω_u that are advected into Ω_l.

(b) Red particles initially in Ω_l that are advected into Ω_u.

Fig. 27. The particles at time $t = 0.15$ for the computation shown in Fig. 22(e), with B $= 1$ and Ra $= 10^5$ on a uniform grid with 192 × 64 cells and initially 256 particles per cell. See Table 4 for additional details.
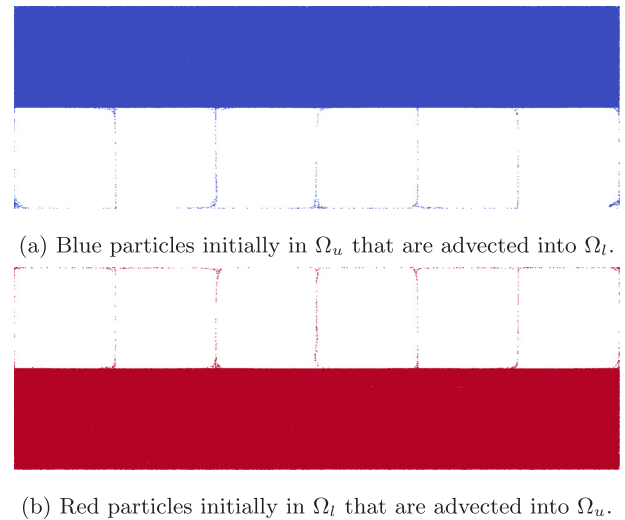

(a) Blue particles initially in Ω_u that are advected into Ω_l.

(b) Red particles initially in Ω_l that are advected into Ω_u.

Fig. 28. The particles at time $t = 0.15$ for a computation with B $= 10$ and Ra $= 10^5$ on a uniform grid with 192 × 64 cells and initially 256 particles per cell. See Table 4 for additional details.

use adaptive mesh refinement in a neighborhood of the interface, as we demonstrated in Fig. 10. This would allow us to "explore" the transition region at a much higher resolution than the 192 × 64 uniform grid we used in the computations shown in Figs. 17(g)–22(g) at much less cost than if we were to use a uniform grid at the same (finer) level of refinement.

In summary, we conclude that for computations in regimes with strongly stratified flow; e.g., $0.5 \lesssim B \leqslant 1.0$ at Ra $= 10^5$, the VOF

Table 4
The number of particles from each subdomain Ω_l and Ω_u that have been entrained and then advected into the other subdomain, with B = 1 and B = 10, respectively. All computations are made on a uniform 192×64 grid with Ra = 10^5 and initially $16 \times 16 = 256$ particles per cell for a total of 3,145,728 particles.

Time	Direction	B = 1		B = 10	
		# entrained particles	Percentage	# entrained particles	Percentage
$t = 0.05$	$\Omega_u \longrightarrow \Omega_l$	2571	0.0817%	2088	0.0664%
	$\Omega_l \longrightarrow \Omega_u$	710	0.0226%	173	0.0055%
$t = 0.10$	$\Omega_u \longrightarrow \Omega_l$	2628	0.0835%	2032	0.0646%
	$\Omega_l \longrightarrow \Omega_u$	854	0.0271%	608	0.0193%
$t = 0.15$	$\Omega_u \longrightarrow \Omega_l$	2588	0.0823%	2326	0.0739%
	$\Omega_l \longrightarrow \Omega_u$	1074	0.0341%	1126	0.0358%

Table 5
The number of particles from each subdomain Ω_l and Ω_u that have been entrained and advected into the other subdomain for B = 1 and B = 10, respectively. All computations are made on a uniform 192×64 grid with Ra = 10^5 and initially $4 \times 4 = 16$ particles per cell for a total of 196,608 particles.

Time	Direction	B = 1		B = 10	
		# entrained particles	Percentage	# entrained particles	Percentage
$t' = 0.05$	$\Omega_u \longrightarrow \Omega_l$	103	0.0524%	78	0.0397%
	$\Omega_l \longrightarrow \Omega_u$	55	0.0280%	30	0.0153%
$t' = 0.10$	$\Omega_u \longrightarrow \Omega_l$	133	0.0676%	101	0.0514%
	$\Omega_l \longrightarrow \Omega_u$	162	0.0824%	35	0.0178%
$t' = 0.15$	$\Omega_u \longrightarrow \Omega_l$	165	0.0839%	105	0.0534%
	$\Omega_l \longrightarrow \Omega_u$	190	0.0966%	52	0.0264%

algorithm or, most likely, any other high quality interface tracking algorithm, may be the most appropriate method for modeling the interface between the chemical compositions.

6. Conclusions

It is now widely accepted that compositional buoyancy plays an important role in mantle convection. Isotopic studies of mid-ocean ridge basalts provide convincing evidence that the upper mantle is well mixed and nearly homogeneous on large scales. This homogeneity is attributed to kinematic mixing. However, isotopic studies of ocean island basalts show that the lower mantle, source region of mantle plumes, is heterogeneous. This heterogeneity is attributed to a compositional barrier that segregates parts of the lower mantle from the upper mantle.

Mantle mixing is attributed to kinematic mixing due to mantle flows. The mantle does not mix by material diffusion except on scales of centimeters to meters on geological time scales. Mantle mixing is a tradeoff between the differential compositional buoyancy $\Delta \rho \, g$ that tends to segregate the two components and the thermal convective buoyancy $\rho_0 \, \alpha \, \Delta T \, g$ that promotes the kinematic mixing. There are two controlling non-dimensional parameters, the Rayleigh number Ra that determines the vigor of thermal convection and the buoyancy ratio $B = \Delta \rho / \rho_0 \, \alpha \, \Delta T$ that determines the relative strength of the stabilizing chemical buoyancy.

The purpose of this paper has been to study numerically the kinematic mixing for a range of buoyancy numbers B at a fixed Rayleigh number Ra. Previous computations of thermo-chemical convection have been carried out. However, many numerical methods for modeling the compositional interface are diffusive, resulting in artificial mixing. In this paper we compare the solutions we obtained using each of four distinct advection methods to compute a two-dimensional model

problem consisting of thermal convection in a fluid layer heated from below with an initial compositional barrier between the upper half and lower half of the layer. Results are obtained for a range of buoyancy numbers B = 0.0, 0.1, 0.2...,1.0 and B = 10.0 with a fixed Rayleigh number Ra = 10^5. These computations reveal that for the initial data we used in this range of buoyancy numbers at Ra = 10^5 there are three distinct regimes that exhibit significantly different dynamics:

1. For $0.0 \leqslant B \lesssim 0.2$ the two layers overturn and are organized into a steady single-layer thermal convection, the computational results approach the limit B → 0.0; i.e., the standard Rayleigh-Bénard problem – in a uniform manner.
2. For $0.3 \lesssim B \lesssim 0.4$ the flow exhibits unstable solutions. In particular, no two of the advection methods produce computational results, that resemble one another in the manner in which they do for the two other regimes; namely $B \lesssim 0.2$ and $0.5 \lesssim B$. In spite of this, all four algorithms exhibit kinematic mixing on (roughly) the same scale, except that – as can be seen in Fig. 15 (a)–(b) – the FEM-EV algorithm appears to still be producing computational results that are more similar to those we found in the interval $0.0 \leqslant B \lesssim 0.2$. We associate this lag in the transition to the new dynamical regime to the high degree of numerical diffusion that characterizes the FEM-EV algorithm.
3. Finally, for $0.5 \lesssim B$ our results exhibit well stratified two layer convection. Well defined cellular convection occurs in the two layers. The compositional barrier to convection is sufficiently strong to provide a near horizontal boundary across which there is no flow. Furthermore, the interface tracking method appears to produce a perfectly horizontal boundary as B → 1.0.

In closing we emphasize that these numerical results depend on the specific initial conditions we used and there is a great deal of work that remains to be done in order to quantify the nature of all possible solutions to the model problem proposed in Section 2.

References

Anbarlooei, H.R., Mazaheri, K., 2011. Moment of fluid interface reconstruction method in axisymmetric coordinates. Int. J. Numer. Methods Biomed. Eng. 27 (10), 1640–1651. http://dx.doi.org/10.1002/cnm.1426.

Bangerth, W., Dannberg, J., Gassmöller, R., Heister, T., et al., 2017. ASPECT: 1080 Advanced Solver for Problems in Earth's ConvecTion, User Manual. CIG. URL:https://geodynamics.org/cig/software/aspect/aspect-manual.pdf.

Burke, K., Steinberger, B., Torsvik, T.H., Smethurst, M.A., 2008. Plume generation zones at the margins of large low shear velocity provinces on the core-mantle boundary. Earth Planet. Sci. Lett. 265 (1), 49–60.

Chandrasekhar, S., 1961. Hydrodynamic and Hydromagnetic Stability. Dover, New York.

Chorin, A.J., 1985. Curvature and solidification. J. Comput. Phys. 57, 472–490.

Chorin, A.J., Marsden, J.E., 1993. A Mathematical Introduction to Fluid Mechanics, 4th Edition. No. 4 in Texts in Applied Mathematics. Springer-Verlag, New York, qA901. C53 1992.

Cockburn, B., Karniadakis, G.E., Shu, C.-W., 2000. The Development of Discontinuous Galerkin Methods. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–50.

Cockburn, B., Shu, C.-W., 1998. The local discontinuous galerkin method for time-dependent convection-diffusion systems. SIAM SIAM J. Numer. Anal. 35 (6), 2440–2463.

Cottaar, S., Romanowicz, B., 2012. An unusually large ULVZ at the base of the mantle near Hawaii. Earth Planet. Sci. Lett. 355, 213–222.

Davaille, A., 1999. Two-layer thermal convection in miscible viscous fluids. J. Fluid

Mech. 379, 223–253.

Donea, J., Huerta, A., 2005. Steady Transport Problems. John Wiley and Sons.

Duretz, T., May, D.A., Gerya, T.V., Tackley, P.J., 2011. Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: a numerical study. Geochem. Geophys. Geosyst. 12 (7), Q07004.

French, S.W., Romanowicz, B., 2015. Broad plumes rooted at the base of the Earth's mantle beneath major hotspots. Nature 525 (7567), 95–99.

Galsa, A., Herein, M., Lenkey, L., Farkas, M., Taller, G., 2015. Effective buoyancy ratio: a new parameter for characterizing thermo-chemical mixing in the Earth's mantle. Solid Earth 6 (1), 93–105.

Gassmöller, R., Heien, E., Puckett, E.G., Bangerth, W., 2016. Flexible and scalable particle-in-cell methods for massively parallel computations. Submitted to the ACM Transactions on Mathematical Software (2016). https://arxiv.org/pdf/1612.03369.pdf.

Gerya, T.V., Yuen, D.A., 2003. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties. Phys. Earth Planet. Inter. 140 (4), 293–318.

Guenther, R.B., Lee, J.W., 1988. Partial Differential Equations of Mathematical Physics and Integral Equations. Books on Mathematics. Dover, New York.

Guermond, J.-L., Pasquetti, R., Popov, B., 2011. Entropy viscosity method for nonlinear conservation laws. J. Comput. Phys. 230 (11), 4248–4267 special issue High Order Methods for CFD Problems.

He, Y., Puckett, E.G., Billen, M.I., 2017. A discontinuous Galerkin method with a bound preserving limiter for the advection of non-diffusive fields in solid Earth geodynamics. Phys. Earth Planet. Inter. 263, 23–37.

Heister, T., Dannberg, J., Gassmöller, R., Bangerth, W., 2017. High accuracy mantle convection simulation through modern numerical methods II: realistic models and problems. Geophys. J. Int. 210 (2), 833–851.

Helmsen, J.J., Colella, P., Puckett, E.G., 1997. Non-convex profile evolution in two dimensions using volume of fluids. Technical Report LBNL-40693, Lawrence Berkeley National Laboratory.

Helmsen, J.J., Colella, P., Puckett, E.G., Dorr, M., 1996. Two new methods for simulating photolithography development in three dimensions. In: Proceedings of the 10th SPIE Optical/Laser Microlithography Conference, vol. 2726. SPIE, San Jose, CA, pp. 253–261.

Henderson, L.F., Colella, P., Puckett, E.G., 1991. On the refraction of shock waves at a slow–fast gas interface. J. Fluid Mech. 224, 1–27.

Hesthaven, J.S., Warburton, T., 2008. Nodal discontinuous Galerkin methods. vol. 54 of Texts in Applied Mathematics. Springer, New York.

Hill, R.N., Shashkov, M., 2013. The symmetric moment-of-fluid interface reconstruction algorithm. J. Comput. Phys. 249, 180–184. URL:http://www.sciencedirect.com/science/article/pii/S0021999113003227.

Hirt, C.W., Nichols, B.D., 1981. Volume of Fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. 39, 201–225.

Huber, R., Helmig, R., 1999. 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. Fluids 29 (8), 899–920.

Jemison, M., Sussman, M., Shashkov, M., 2015. Filament capturing with the multimaterial moment-of-fluid method. J. Comput. Phys. 285, 149–172.

John, F., 1982. Partial Differential Equations. Applied Mathematical Sciences, 4th ed. Springer-Verlag, New York.

Kellogg, L.H., 1992. Mixing in the mantle. Annu. Rev. Earth Planet. Sci. 20 (1), 365–388.

Kellogg, L.H., Hager, B.H., van der Hilst, R.D., 1999. Compositional stratification in the deep mantle. Science 283 (5409), 1881–1884.

Kothe, D.B., Puckett, E.G., Williams, M.W., 1999. Robust finite volume modeling of 3-d free surface flows on unstructured meshes. In: Proceedings of the 14th AIAA Computational Fluid Dynamics Conference American Institute of Aeronautics and Astronautics, Norfolk, VA, pp. 1–6.

Kronbichler, M., Heister, T., Bangerth, W., 2012. High accuracy mantle convection simulation through modern numerical methods. Geophys. J. Int. 191 (1), 12–29.

LeVeque, R.J., 1990. Numerical Methods for Conservation Laws Lectures in Mathematics, 2nd ed. Birkhäuser, Basel; Boston; Berlin, ETH Zürich.

LeVeque, R.J., 1996. High-resolution conservative algorithms for advection in incompressible flow. SIAM J. Numer. Anal. 33 (2), 627–665.

McNamara, A.K., Zhong, S., 2004. Thermochemical structures within a spherical mantle: Superplumes or piles? J. Geophys. Res.: Solid Earth 109 (B07402).

Miller, G.H., Puckett, E.G., 1994. Edge effects in molybdenum-encapsulated molten silicate shock wave targets. J. Appl. Phys. 75 (3), 1426–1434.

Miller, G.H., Puckett, E.G., 1996. A high-order Godunov method for multiple condensed phases. J. Comput. Phys. 128 (1), 134–164.

Montague, N.L., Kellogg, L.H., 2000. Numerical models of a dense layer at the base of the mantle and implications for the geodynamics of D. J. Geophys. Res.: Solid Earth 105 (B5), 11101–11114.

Nichols, B.D., Hirt, C.W., Hotchkiss, R.S., 1980. SOLA-VOF: a solution algorithm for transient fluid flow with multiple free boundaries. Technical Report LA-8355, Los Alamos National Laboratory.

Noh, W.F., Woodward, P.R., 1976. SLIC (Simple Line Interface Calculation). In: van de Vooren, A.I., Zandbergen, P.J. (Eds.), Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics. vol. 59 of Lecture Notes in Physics. Springer-Verlag, Twente University, Enschede, pp. 330–340.

Ottino, J.M., 1989. The Kinematics of Mixing: Stretching, Chaos, and Transport. Cambridge University Press.

Pilliod, J.E., 1992. An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods. MS Thesis, Graduate Group in Applied Mathematics, University of California, Davis.

Pilliod, J.E., Puckett, E.G., 1998. An unsplit, second-order accurate Godunov method for tracking deflagrations and detonations. In: Houwing, A.F.P., Paull, A., Boyce, R.R., Danehy, P.M., Hannemann, H., Kurtz, J.J., McIntyre, T.J., McMahon, S.J., Mee, D.J., Sandeman, R.J., Tanno, H. (Eds.), Proceedings of the 21st International Symposium on Shock Waves. Vol. II. Panther Publishing, Fyshwick, Australia, pp. 1053–1058.

Pilliod, J.E., Puckett, E.G., 2004. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. J. Comput. Phys. 199 (2), 465–502.

Puckett, E.G., 1991. 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. pp. 933–938.

Puckett, E.G., 2010a. On the second-order accuracy of volume-of-fluid interface reconstruction algorithms: Convergence in the max norm. CAMCoS 5 (1), 99–148.

Puckett, E.G., 2010b. A volume-of-fluid interface reconstruction algorithm that is second-order accurate in the max norm. CAMCoS 5 (2), 199–220.

Puckett, E.G., 2014. On the second-order accuracy of volume-of-fluid interface reconstruction algorithms II: an improved constraint on the cell size. CAMCoS 8 (1), 123–158.

Puckett, E.G., Almgren, A.S., Bell, J.B., Marcus, D.L., Rider, W.J., 1997. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. J. Comput. Phys. 130 (2), 269–282.

Puckett, E.G., Miller, G.H., 1996. The numerical computation of jetting impacts. In: Sturtevant, B., Shepherd, J.E., Hornung, H. (Eds.), Proceedings of the 20th International Symposium on Shock Waves. vol. II. World Scientific, New Jersey, pp. 1467–1472.

Reed, W., Hill, T., Oct 1973. Triangular mesh methods for the neutron transport equation. Tech. Rep. LA-UR–73-479, Los Alamos Scientific Lab.

Samuel, H., Evonuk, M., 2010. Modeling advection in geophysical flows with particle level sets. Geochem. Geophys. Geosyst. 11 (8).

Schubert, G., Turcotte, D.L., Olson, P., 2001. Mantle Convection in the Earth and Planets. Cambridge University Press.

Sethian, J.A., 1999. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials sciences, 2nd ed. Cambridge Monographs on Applied and Computational Mathematics, vol. 3 Cambridge University Press, Cambridge, U.K.; New York.

Sheldon, J., Cardwell Jr, W., et al., 1959. One-dimensional, incompressible, noncapillary, two-phase fluid flow in a porous medium. Pet. Trans., AIME 216, 290–296.

Shu, C.-W., 2016. Discontinuous Galerkin methods for time-dependent convection dominated problems: Basics, recent developments and comparison with other methods. In: Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations Springer, pp. 369–397.

Strang, W.G., 1968. On the construction and comparison of difference schemes. SIAM J. Numer. Anal. 5 (3), 506–517.

Strang, W.G., 2016. Introduction to Linear Algebra, 5th ed. Cambridge Wellsely Press.

Sussman, M.S., Puckett, E.G., 2000. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. J. Comput. Phys. 162, 301–337.

Tackley, P.J., 2012. Dynamics and evolution of the deep mantle resulting from thermal, chemical, phase and melting effects. Earth Sci. Rev. 110 (1), 1–25.

Tackley, P.J., King, S.D., 2003. Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations. Geochem. Geophys. Geosyst. 4 (4), Q8302.

Tan, E., Gurnis, M., 2005. Metastable superplumes and mantle compressibility. Geophys. Res. Lett. 32 (20).

Thielmann, M., May, D., Kaus, B., 2014. Discretization errors in the hybrid finite element particle-in-cell method. Pure Appl. Geophys. 171 (9), 2165–2184.

Torrey, M.D., Cloutman, L.D., Mjolsness, R.C., Hirt, C.W., 1985. NASA-VOF2D: a computer program for incompressible flows with free surfaces. Technical Report LA-10612-MS, Los Alamos National Laboratory.

Torrey, M.D., Mjolsness, R.C., Stein, L.R., 1987. NASA-VOF3D: a three-dimensional computer program for incompressible flows with free surfaces. Technical Report LA-11009-MS, Los Alamos National Laboratory.

Trim, S., Heron, P., Stein, C., Lowman, J., 2014. The feedback between surface mobility and mantle compositional heterogeneity: Implications for the Earth and other terrestrial planets. Earth Planet. Sci. Lett. 405, 1–14.

Trim, S.J., Lowman, J.P., 2016. Interaction between the supercontinent cycle and the evolution of intrinsically dense provinces in the deep mantle. J. Geophys. Res.: Solid Earth 121 (12), 8941–8969 2016JB013285.

Tryggvason, G., Scardovelli, R., Zaleski, S., 2011. Direct Numerical Simulations of Gas-Liquid Multiphase Flows. In: 1st ed. Cambridge Monographs on Applied & Computational Mathematics Cambridge University Press, Cambridge.

Turcotte, D.L., Schubert, G., 2014. Geodynamics, 3rd ed. Cambridge University Press, Cambridge.

van Keken, P.E., King, S.D., Schmeling, H., Christensen, U.R., Neumeister, D., Doin, M.-P., 1997. A comparison of methods for the modeling of thermochemical convection. J. Geophys. Res.: Solid Earth 102 (B10), 22477–22495. http://dx.doi.org/10.1029/97JB01353.

Von Neumann, J., Richtmyer, R.D., 1950. A method for the numerical calculation of hydrodynamic shocks. J. Appl. Phys. 21 (3), 232–237.

Wanner, G., Hairer, E., 1991. Solving ordinary differential equations II. vol. 14 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin Heidelberg.

Warming, R., Hyett, B., 1974. The modified equation approach to the stability and accuracy analysis of finite-difference methods. J. Comput. Phys. 14 (2), 159–179.

Zhang, X., Shu, C.-W., 2010. On maximum-principle-satisfying high order schemes for scalar conservation laws. J. Comput. Phys. 229 (9), 3091–3120.