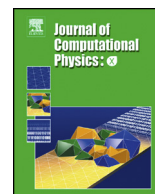




ELSEVIER

Contents lists available at ScienceDirect

Journal of Computational Physics: X

www.elsevier.com/locate/jcpX


Modelling binary alloy solidification with adaptive mesh refinement


 James R.G. Parkinson ^{a,*}, Daniel F. Martin ^b, Andrew J. Wells ^a, Richard F. Katz ^c
^a Atmospheric, Oceanic and Planetary Physics, Department of Physics, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, UK

^b Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

^c Department of Earth Sciences, University of Oxford, South Parks Road, Oxford, OX1 3AN, UK

ARTICLE INFO

Article history:

Received 15 March 2019

Received in revised form 25 September 2019

Accepted 3 November 2019

Available online 21 November 2019

Keywords:

 Binary alloy solidification
 Adaptive mesh refinement
 Mushy layer convection
 Finite volume

ABSTRACT

The solidification of a binary alloy results in the formation of a porous mushy layer, within which spontaneous localisation of fluid flow can lead to the emergence of features over a range of spatial scales. We describe a finite volume method for simulating binary alloy solidification in two dimensions with local mesh refinement in space and time. The coupled heat, solute, and mass transport is described using an enthalpy method with flow described by a Darcy-Brinkman equation for flow across porous and liquid regions. The resulting equations are solved on a hierarchy of block-structured adaptive grids. A projection method is used to compute the fluid velocity, whilst the viscous and nonlinear diffusive terms are calculated using a semi-implicit scheme. A series of synchronization steps ensure that the scheme is flux-conservative and correct for errors that arise at the boundaries between different levels of refinement. We also develop a corresponding method using Darcy's law for flow in a porous medium/narrow Hele-Shaw cell. We demonstrate the accuracy and efficiency of our method using established benchmarks for solidification without flow and convection in a fixed porous medium, along with convergence tests for the fully coupled code. Finally, we demonstrate the ability of our method to simulate transient mushy layer growth with narrow liquid channels which evolve over time.

© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A wide variety of physical processes involve binary alloy solidification. Within industrial settings, understanding this phenomena is relevant to metal casting [1] whilst, in a geophysical context, notable applications include the study of the Earth's core [2] and sea ice formation [3]. Solidification of a multi-component melt from a cold boundary generally leads to the formation of a solid phase with a different composition from the liquid. Heat diffuses away from the solid-liquid interface faster than the solute, causing liquid adjacent to the freezing interface to be constitutionally supercooled [4]. Under these conditions, the solid-liquid interface is unstable to dendritic growth [5] and a mushy layer will form: a porous solid matrix bathed in its melt.

* Corresponding author.

E-mail address: james.parkinson@physics.ox.ac.uk (J.R.G. Parkinson).

Variations in solute concentration within a mushy layer control variations in density, which can initiate natural convection [6]. Flow through the porous matrix transports heat and solute, promoting local dissolution and internal solidification which modify the porosity. In certain circumstances, dissolution leads to the formation of entirely liquid regions through which the flow is focused. Within sea ice, a binary alloy of salt and water, these regions are known as brine channels and efficiently transport brine into the ocean [3]. In industrial settings, such as metal casting, these ‘chimneys’ or ‘freckles’ are undesirable defects [1]. Henceforth, we shall refer to these features as channels.

The solidification of mushy layers and formation of channels has been considered in laboratory studies with a range of binary alloys, the most common being aqueous ammonium chloride. Experiments typically involve either cooling a tank of liquid from either above or below, known as ‘fixed chill’ [7,8], or translating a Hele-Shaw cell between two fixed heat exchangers; so called ‘directional solidification’ [9–11]. Fixed chill experiments reveal that the location of channels evolves transiently, with the spacing between channels increasing as the mushy layer grows deeper [12,13].

Theoretical descriptions of mushy layers have been developed based on mixture theory [14] and volume averaging techniques [15]. Steady-state solutions can be found in the absence of fluid motion [16], which provide the base state for linear [e.g. 17] and weakly nonlinear [e.g. 18–20] stability analyses. These predict that mushy layers are unstable to small perturbations above a critical Rayleigh number, which characterises the significance of buoyancy forcing relative to thermal and viscous dissipation. Such perturbations result in patterns of varying flow and porosity within the mushy layer, which may give rise to the liquid channels observed in experiments.

However, channel formation is strongly nonlinear; to accurately model the feedbacks involved, the governing equations must be integrated numerically. A split domain approach involves solving separately for liquid, solid, and ‘mushy’ regions along with the free boundaries which separate them. Various authors have used this method to find steady-state solutions [21–23] for a periodic array of channels, given a prescribed channel spacing. An alternative is the single domain ‘enthalpy method’ [24,25] in which a single set of equations are solved throughout the entire domain. This approach removes the need to explicitly solve for the free boundaries, which are often convoluted. Instead, mush-liquid boundaries are implicitly captured via variations of the porosity field. It is therefore a good choice for computing transient solutions where the channels and mush-liquid boundaries evolve over time. The enthalpy method formulation has been used extensively to simulate binary alloy solidification in a variety of physical settings, using both finite volume methods [10,26] and finite element methods [27,28].

Fully coupled nonlinear simulations are computationally expensive as they feature multiple scales. Narrow channels and diffusive boundary layers develop over time and require high resolution, but are restricted to a small portion of the domain. One approach for efficiently solving problems with multiple spatial scales is the use of adaptive mesh refinement (AMR). Various approaches to AMR exist, which can be broadly divided into either finite element methods on unstructured meshes [e.g. 29], or finite volume methods on structured meshes (following [30,31]). Adaptive mesh refinement has been used to study binary alloy solidification without fluid flow in the context of microstructural dendritic growth [32], and directional solidification of mushy layers [33]. Recently, various authors have studied channel formation and mushy layer convection using a similar model to the one considered here [34–36].

We develop an AMR code to simulate the solidification of binary alloys. In contrast to previous studies, which used finite elements, we implement a finite volume method based on block-structured meshes. Supported by the Chombo framework [37], which scales efficiently on thousands of processors [38], we implement adaptivity in time (so called ‘subcycling’) as well as space: coarser levels can be advanced by fewer, larger, timesteps as each refinement level is advanced at roughly the same Courant-Friedrichs-Lewy (CFL) number.

Maintaining conservation laws with subcycling is not trivial. The solution on regions covered by a fine grid is calculated after the coarse grid, meaning that the more accurate fine fluxes cannot directly influence the coarse solution. Previous authors have accounted for this via ‘reflux’ corrections for hyperbolic problems [31] and elliptic problems with linear operators [39]. For nonlinear elliptic equations, such as those encountered here, Pretorius et al. [40] applied Berger–Oliger timestepping to the hyperbolic parts of the equations, with the elliptic parts found by extrapolation, before performing an implicit solve for the elliptic parts over the entire hierarchy of grids. We use a different approach; by linearising the elliptic operators, we follow Almgren et al. [39] and solve implicitly for the small diffusive flux corrections which emerge at coarse-fine boundaries.

In section 2 we introduce the mathematical model which is solved using the numerical method presented in section 3. We verify the accuracy of the method using benchmark problems for diffusive solidification (section 4.1) and convection in a fixed porous medium (section 4.2). Example solutions of the full model of binary alloy solidification are calculated in section 4.3 using both a uniform mesh and an adaptive mesh, which demonstrate the advantages of using an adaptive mesh for this problem. Finally, in section 5, we summarize our results and suggest future studies which may benefit from the methods presented here.

2. Problem formulation

We consider a volume-averaged model of a mushy layer [15] in two spatial dimensions (x, z). Whilst it is straightforward to extend the model and our method to three dimensions, we choose to focus on flow and solidification in a 2-D rectangular box of width ℓ and height h . We will consider various physical settings within this geometry, which might describe flow in

a thin Hele-Shaw cell [10], concentrating primarily on a fluid which is cooled from the top boundary. Solidification increases the solute concentration in the fluid, and therefore the density, which can drive convection.

Adopting the concept of an 'ideal' mushy layer [41], we assume local thermodynamic equilibrium and can therefore determine the local temperature T , liquid fraction (or porosity) χ , liquid composition C_l , and solid composition C_s from the enthalpy \hat{H} , bulk composition $C = \chi C_l + (1 - \chi)C_s$, and an idealized phase diagram (Fig. 1). We denote the liquid (solid) thermal conductivity k_l (k_s), specific heat capacity $c_{p,l}$ ($c_{p,s}$), species diffusivity D_l (D_s), and density ρ_l (ρ_s). For simplicity, we assume that solute diffusion is negligible in the solid with $D_s = 0$. We make the Boussinesq approximation [43], such that the liquid density is approximated as a constant ρ_l , except where it affects the buoyancy. We make the additional approximation that the solid phase has density $\rho_s \approx \rho_l$ approximately equal to the liquid, with the result that we neglect flow driven by the expansion or contraction of a fluid upon solidification (see [42] for a discussion of this effect). The net effect of these approximations is that the small density differences in the liquid due to compositional and temperature variations control the magnitude of the buoyancy force, whilst the other material properties are approximated by the constant reference density ρ_l in all other terms [41].

Given an initial composition C_i , we choose characteristic scales for composition $\Delta C = C_e - C_i$, temperature, $\Delta T = T_L(C_i) - T_e$, and enthalpy $\Delta H = \rho_l c_{p,l} \Delta T$, where T_e and C_e are the eutectic temperature and composition, and $T_L(C_i)$ is the liquidus temperature at the initial composition (see the phase diagram Fig. 1). We scale lengths by h , use the diffusive timescale h^2/κ_l , and scale velocities by κ_l/h , where $\kappa_l = k_l/\rho_l c_{p,l}$ is the thermal diffusivity. Finally we scale pressure with $\rho_l \kappa_l^2/h^2$ and permeability with a reference value K_0 . The dimensionless temperature and composition are defined relative to the eutectic point,

$$\theta = \frac{T - T_e}{\Delta T}, \quad \Theta = \frac{C - C_e}{\Delta C}, \quad (1)$$

whilst the dimensionless enthalpy is given by

$$H = \mathcal{S}\chi + [\chi + (1 - \chi)c_p]\theta, \quad (2)$$

where the specific heat ratio $c_p = c_{p,s}/c_{p,l}$, whilst the Stefan number $\mathcal{S} = L/(c_{p,l}\Delta T)$ gives the ratio between the latent heat release per unit mass of liquid upon freezing L , and typical variations in the sensible heat.

2.1. Conservation equations

Conservation of energy and solute are given by [10, eqs. 2-4] and can be written in dimensionless form as

$$\frac{\partial H}{\partial t} + \mathbf{V}_{frame} \cdot \nabla H + \mathbf{U} \cdot \nabla \theta(H, \Theta) = \nabla \cdot [\chi + k(1 - \chi)] \nabla \theta(H, \Theta), \quad (3)$$

$$\frac{\partial \Theta}{\partial t} + \mathbf{V}_{frame} \cdot \nabla \Theta + \mathbf{U} \cdot \nabla \Theta_l(H, \Theta) = \frac{1}{Le} \nabla \cdot \chi \nabla \Theta_l(H, \Theta), \quad (4)$$

where \mathbf{U} is the Darcy velocity, $Le = \kappa_l/D_l$ is the Lewis number, $k = k_s/k_l$ is the heat conductivity ratio, and the dimensionless temperature $\theta(H, \Theta)$ and liquid concentration $\Theta_l(H, \Theta)$ are functions of enthalpy and bulk concentration determined below from the phase diagram. The frame translation velocity \mathbf{V}_{frame} is zero for fixed chill settings, and a constant vector aligned with the direction of sample translation for directional solidification. Recalling that we have made the Boussinesq approximation, and are assuming that $\rho_s = \rho_l$, mass conservation is

$$\nabla \cdot \mathbf{U} = 0. \quad (5)$$

We assume that the fluid density is well described by a linear equation of state, and that momentum conservation can be characterised by a Darcy-Brinkman equation [28] which transitions between the Navier-Stokes equation in pure liquid ($\chi = 1$) and Darcy flow in a porous medium when $\chi \ll 1$. Non-dimensionalising [28, eqn 2.13] yields

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \left(\frac{\mathbf{U}}{\chi} \right) = -\chi \nabla p + Pr \nabla^2 \mathbf{U} + \chi Pr (Ra_T \theta - Ra_C \Theta_l) \mathbf{e}_z - \frac{Pr \chi \mathbf{U}}{Da \Pi}, \quad (6)$$

where the dimensionless permeability $\Pi = \Pi(\chi)$ is a function of the porosity, and we have defined the Prandtl number, Darcy number, and thermal and solutal Rayleigh numbers,

$$Pr = \frac{\eta}{\rho_l \kappa_l}, \quad Da = \frac{K_0}{h^2}, \quad Ra_T = \frac{\rho_l g \alpha \Delta T h^3}{\kappa_l \eta}, \quad Ra_C = \frac{\rho_l g \beta \Delta C h^3}{\kappa_l \eta}. \quad (7)$$

Here, η is the (dynamic) fluid viscosity, α and β are the thermal and solutal expansivities, and g is the acceleration due to gravity. Following [10] we consider a modified Carman-Kozeny permeability function appropriate to solidification in a thin Hele-Shaw cell,

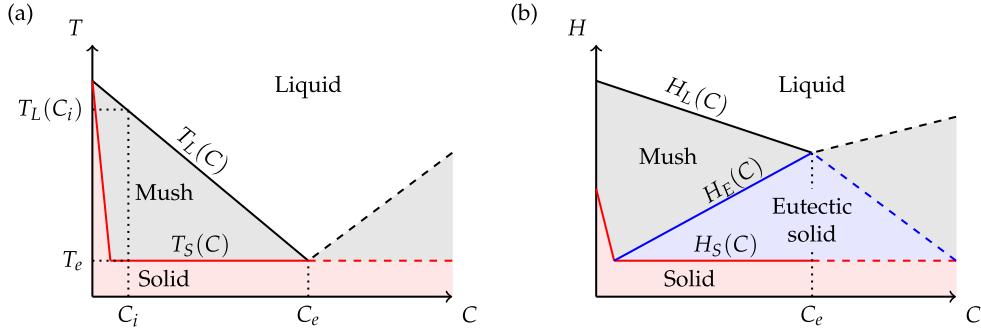


Fig. 1. Simplified phase diagram for a binary alloy in (a) T - C space and (b) H - C space. Below the eutectic composition C_e (region where phase boundaries are marked by solid lines), the solid phase of the mushy-layer is composed of the solvent. For example, for an NaCl- H_2O system with concentration below 230 g/kg the mushy-layer consists of water ice crystals and salty brine. Other alloys, such as NH_4Cl -water, are typically considered to the right of the eutectic composition (shown with dashed lines). For temperature and composition above the liquidus $T_L(C)$ the alloy is purely liquid, and below the solidus $T_S(C)$ there is only solid. In between a mushy region forms, comprising a mixture of pure solid crystals on the solidus curve and solute enriched liquid on the liquidus curve, with liquid fraction χ and phase-weighted bulk concentration C . Considering H - C space, with liquidus $H_L(C)$ and solidus $H_S(C)$, highlights that there is an additional region of eutectic solid in addition to the mush. On the mush-eutectic phase boundary ($H_E(C)$ in (b)) $T = T_e$ and the porosity depends on the bulk concentration via $\chi = \chi_e(C)$. The eutectic solid is a phase the system often passes through during the transition from the mushy layer phase to the solid phase; the enthalpy difference $H_E(C) - H_S(C)$ represents the energy which must be extracted for this transition to occur, with $T = T_e$ and $\chi = 0$.

$$\Pi(\chi) = \left\{ \Pi_c^{-1} + \left[\frac{\chi^3}{(1-\chi)^2} \right]^{-1} \right\}^{-1}. \quad (8)$$

In the pure liquid region $\chi \rightarrow 1$ and the permeability takes the value $\Pi_c = d^2/(12K_0)$, which is the dimensionless permeability for a thin Hele-Shaw cell with narrow gap-width d and sets a limit on the total permeability of the system. In section 3.7 we consider the modifications to the algorithm if the Darcy-Brinkman equation (6) is replaced with the simpler Darcy flow law.

2.2. Phase diagram

Depending on the local enthalpy H and bulk composition Θ , a particular region of the domain may be in one of four phases; solid, liquid, eutectic solid, or mushy layer. Before the local temperature, liquid fraction, and liquid composition can be calculated, we must first determine which phase the system is in [24] using a phase diagram (see Fig. 1). Following previous authors [e.g. 10] we linearise the liquidus $H_L(\Theta)$ and solidus $H_S(\Theta)$ which, along with the enthalpy at the top of the eutectic region $H_E(\Theta)$, are sufficient to describe the phase diagram:

$$H_L(\Theta) = \mathcal{S} - \Theta, \quad (9)$$

$$H_S(\Theta) = c_p \max[0, -p_s^{-1}(\Theta + \mathcal{C})], \quad (10)$$

$$H_E(\Theta) = \chi_e(\Theta)\mathcal{S} = \left(1 + \frac{\Theta}{\mathcal{C}}\right)\mathcal{S}, \quad (11)$$

where $\chi_e(\Theta)$ is the porosity on the mush-eutectic phase boundary and the partition coefficient $p_s = C_s/C_l$ is taken as 10^{-5} in all our simulations. We have also introduced the so-called concentration ratio, $\mathcal{C} = (1 - p_s)C_e/\Delta C$, which describes the size of the greatest solute concentration in the system relative to typical concentration variations. In writing eqs. (9) to (11) we have used the linear liquidus and eq. (1) to note that $\Theta_l = -\theta$ in the mushy layer, $\Theta = \Theta_l$ and $\chi = 1$ along the liquidus, and $\theta = 0$ in the eutectic solid phase. For full details of how we calculate θ , χ , Θ_l and Θ_s from H and Θ , see Appendix A.

3. Method

We solve the governing equations eqs. (3) to (6) on a hierarchy of block-structured, non-uniform meshes using a finite volume discretization, facilitated by the Chombo framework [37]. We extend the AMR algorithm presented in [44] for solving the Navier-Stokes equations to include the Darcy-Brinkman correction, for describing porous media flow, and the effects of buoyancy. Additionally, we extend the Chombo toolbox to solve the nonlinear evolution equations for the enthalpy and bulk composition implicitly using the Full Approximation Scheme (FAS) [see e.g. 45]. Advection terms in the momentum equation and advection-diffusion equations are computed using a conservative unsplit Godunov method [46,47]. The scheme is second-order in space and first-order in time. We introduce our choice of AMR notation in section 3.1. We then introduce our algorithm (section 3.2), which consists of recursive updates on individual levels of refinement (section 3.3) followed by

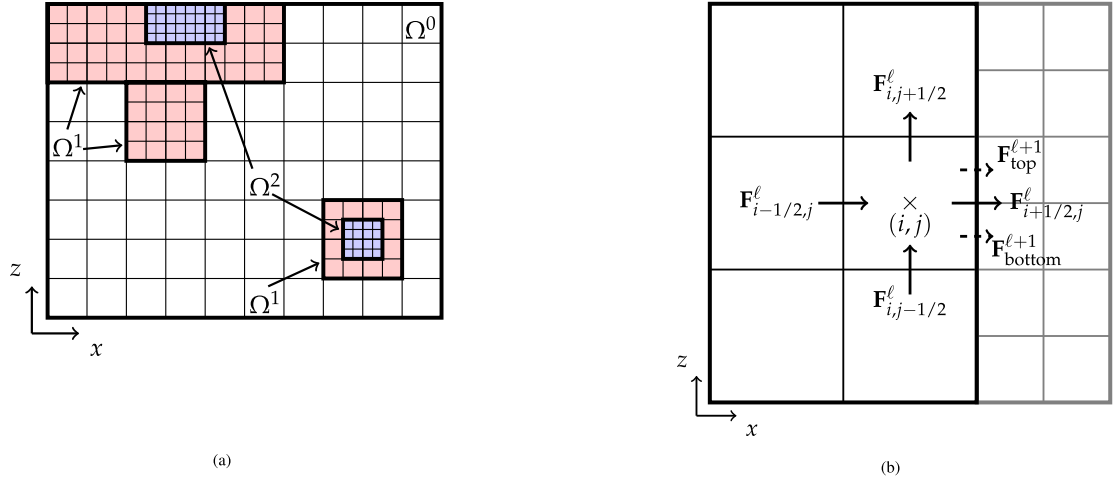


Fig. 2. (a) An example AMR hierarchy consisting of three levels $\ell = 0, 1, 2$ whose meshes consist of between 1 and 3 rectangular subsets of the full domain. (b) Cells and fluxes at a coarse-fine boundary. Differences between the coarse and fine fluxes, $\delta \mathbf{F}^\ell = \mathbf{F}_{i+1/2,j}^\ell - (\mathbf{F}_{i+1/2,j}^{\ell+1} + \mathbf{F}_{i+1/2,j}^{\ell+1})$, require a 'reflux' correction to maintain conservation by ensuring that the fluxes used to update both coarse- and fine-level solutions are consistent.

synchronization operations once a timestep has been completed (section 3.4) in order to maintain conservation and stability as needed.

3.1. AMR structure and notation

Our notation follows several previous works on AMR algorithms with Berger-Oliger timestepping [31,44,48], which we summarise here for a self contained presentation. A rectangular domain is discretized on a hierarchy of cell-centred grids Γ^ℓ with grid spacing Δx^ℓ , where the level of refinement has index $\ell = 0, 1, 2, \dots, \ell_{max}$. On each level ℓ we define the *level domain* Ω^ℓ , which consists of one or more rectangular subsets of Γ^ℓ (see Fig. 2(a)); it is on the level domains that we solve our equations. The coarsest level domain covers the entire problem domain $\Omega^0 = \Gamma^0$. Each level domain is *properly nested* [31]: no interfaces exist between Ω^ℓ and $\Omega^{\ell \pm 2}$, only between Ω^ℓ and $\Omega^{\ell+1}$ and possibly Ω^ℓ and the domain boundary $\delta\Omega$.

The grid spacing is chosen such that the refinement ratio $n_{ref}^\ell = \Delta x^\ell / \Delta x^{\ell+1}$ satisfies $n_{ref}^\ell = 2^m$, $m = 1, 2, 3, \dots$. We write the refinement ratios for an entire AMR hierarchy as $n_{ref} = (n_{ref}^0, n_{ref}^1, \dots, n_{ref}^{\ell_{max}-1})$. The timestep on each level Δt^ℓ scales with the grid spacing, $n_{ref}^\ell = \Delta t^\ell / \Delta t^{\ell+1}$, and satisfies a Courant-Friedrichs-Lewy (CFL) condition which ensures that $\Delta t^\ell < \sigma \Delta x^\ell / \text{Max}|\mathbf{U}^\ell|$ where the velocity \mathbf{U}^ℓ is taken from the previous timestep and $0 < \sigma < 1$. When solving problems where the fluid acceleration $a \sim Pr \text{Max}(|\mathbf{U}^\ell|/Da, Ra_T, Rac)$ is large, but flow has not yet fully developed, we also ensure that $\Delta t^\ell < \sigma \sqrt{\Delta x^\ell / a^\ell}$ for stability. The level ℓ on which a variable is computed is denoted by a superscript ℓ , whilst the timestep t^n is indicated by a second superscript, e.g. $\psi^{\ell,n+1/2} = \psi(t^n + \Delta t^\ell/2)$. Unless otherwise stated, time-centred fields are found by linear interpolation e.g. $\theta^{\ell,n+1/2} = (\theta^{\ell,n} + \theta^{\ell,n+1})/2$. Variables are cell-centred unless otherwise indicated; a subscript f denotes face-centred values. Where required, conversion from cell-centred to face-centred values, or vice versa, is achieved by linear interpolation and denoted by $\text{Av}^{C \rightarrow F}$, e.g. $\theta_f = \text{Av}^{C \rightarrow F} \theta$.

The *level domain* Ω^ℓ can be divided into *valid* and *invalid* regions, where Ω_{valid}^ℓ is the part of Ω^ℓ not covered by $\Omega^{\ell+1}$, and we assume that the solution on the finer level $\Omega^{\ell+1}$ is more accurate than the solution on $\Omega_{invalid}^\ell$, the part of Ω^ℓ covered by the finer level mesh $\Omega^{\ell+1}$. We define single level operators as those which are applied to both *valid* and *invalid* regions of ℓ using a superscript, e.g. ∇^ℓ . In contrast, *composite* operators are applied to the solution on a union of *valid* regions and do not have a superscript.

3.2. AMR update

Following [44], we implement adaptivity in time via Berger-Oliger timestepping [30,31] as illustrated in Fig. 3. To advance the AMR solution from time t to $t + \Delta t$, we employ a recursive timestepping scheme. Starting with the coarsest level $\ell = 0$, we advance the solution on level ℓ from time t^ℓ to $t^\ell + \Delta t^\ell$. The next finer level is then advanced n_{ref}^ℓ times with a timestep $\Delta t^{\ell+1} = \Delta t^\ell / n_{ref}^\ell$, and so on recursively, where the algorithm used on each level is identical and described in section 3.3. Once all levels finer than ℓ have reached $t^\ell + \Delta t^\ell$, we perform a series of synchronization steps.

The solution on a fine level $\ell + 1$ will have been updated using more accurate fluxes than were used to update ℓ , which can lead to a discrepancy between the fluxes across a coarse-fine boundary as shown in Fig. 2(b), with a resulting loss of conservation. We compute any difference between the coarse fluxes \mathbf{F}^ℓ and the sum of the fine fluxes $\mathbf{F}^{\ell+1}$, $\delta \mathbf{F}^\ell =$

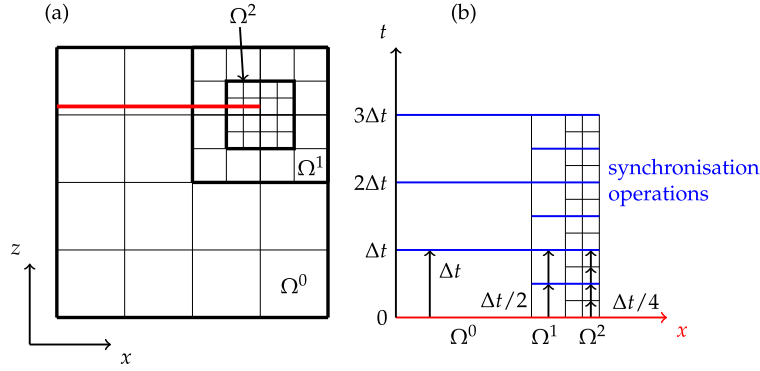


Fig. 3. Berger-Oliger timestepping. A cross section (red) through an example AMR hierarchy traverses three levels of refinement, which are advanced from coarsest (Ω^0) to finest (Ω^2) as shown on the right. Finer levels have a smaller timestep for stability, and hence require more updates for each coarse grid timestep. Once two or more levels have reached the same time (in this example, $t = n \Delta t/2$, where $n = 1, 2, 3, \dots$), synchronization operations (blue) are applied to ensure conservation.

$\mathbf{F}^\ell - \sum \mathbf{F}^{\ell+1}$, and correct our solution on the coarse level grid Ω^ℓ , to account for this as described in section 3.4. This step is called *refluxing*.

It is desirable for a numerical scheme to be *freestream preserving*; an initially constant scalar field λ should remain constant when advected by a divergence-free velocity field. However this is not the case in our scheme following refluxing, because the advection velocities are not discretely divergence-free across coarse-fine interfaces. Therefore we use the strategy suggested by [48] and evolve a scalar field λ according to

$$\frac{\partial \lambda}{\partial t} + \mathbf{U} \cdot \nabla \lambda = 0, \quad (12)$$

where $\lambda(\mathbf{x}, t=0) = 1$. Deviations in λ from its initial value are then used to compute a lagged correction to the velocity field, \mathbf{U}_p , as part of the synchronisation step (see section 3.4 for more details).

The pressure is computed using a projection method, as in [48] section 1.1 but modified to account for the porosity factor in the pressure gradient term. In general, below we solve problems of the form

$$\nabla \cdot (\chi \nabla p) = \nabla \cdot \mathbf{U} \quad (13)$$

for a pressure p given the initial estimate of the velocity field \mathbf{U} which does not, in general, satisfy the divergence constraint.

3.3. Single level update

On a single level of refinement ℓ , we begin a timestep at time t and wish to evolve the primary cell-centred variables, namely $H^\ell(t)$, $\Theta^\ell(t)$, and the two velocity components $\mathbf{U}^\ell(t) = [U^\ell(t), V^\ell(t)]$, by a time interval Δt^ℓ . In addition to the primary variables we have a scalar $\lambda^\ell(t)$ and a vector $\mathbf{U}_p^{\ell,n}(t)$ which we use to correct for errors in freestream preservation at coarse-fine boundaries, as discussed later (section 3.4). The pressure is cell-centred, computed at the half time step ($t^\ell \pm \frac{1}{2} \Delta t^\ell$), and decomposed into two parts, $p^\ell = \pi^\ell + e$, where π^ℓ is the pressure computed during a single level update and e is a small correction, computed when synchronizing multiple levels of refinement to ensure that the velocity is divergence-free across coarse-fine boundaries. A separate cell-centred pressure field, ϕ^ℓ , is computed to correct the face-centred advection velocities, as described in step 1 below.

Boundary conditions at the interface between a fine level $\ell + 1$ and coarse level ℓ are computed via interpolation in space and time, as described in Appendix A of [44]. Elliptic operators require quadratic spatial interpolation, whilst for advection operators we use bilinear interpolation. In both cases, we use linear interpolation in time following [39,48]. Domain boundary conditions are first order in space and implemented via ghost cells. Because the domain edge is one dimension less than the domain itself, it is possible to drop one order of accuracy along the boundary, using a first order boundary-condition discretization whilst retaining $O(\Delta x^2)$ global accuracy [49,50].

The update has 5 main steps. Step 1 follows the work of [48], whilst steps 3 and 4 are based on [44].

1. *Compute advection velocities.* We require a face-centred velocity $\mathbf{U}_{AD}^{\ell,n+1/2}$, in order to compute the advective terms $(\mathbf{U} \cdot \nabla)$ in eqs. (3), (4) and (6). Rewriting the advection operator as $\mathbf{U} \cdot \nabla (\mathbf{U}/\chi) = (\mathbf{U}/\chi) \cdot \nabla \mathbf{U} - (\mathbf{U}/\chi^2) \mathbf{U} \cdot \nabla \chi$, the momentum equation becomes

$$\left(\frac{\partial}{\partial t} + \frac{\mathbf{U}}{\chi} \cdot \nabla \right) \mathbf{U} = \underbrace{Pr \nabla^2 \mathbf{U} + \chi Pr (Ra_T \theta - Ra_C \Theta) \mathbf{e}_z - Pr \chi \mathbf{U} / (Da \Pi) + (\mathbf{U}/\chi^2) \mathbf{U} \cdot \nabla \chi}_{S}. \quad (14)$$

Following Appendix A.1 of [44], we treat the right hand side of eq. (14) as a source term evaluated at time t^n and predict a face-centred velocity $\mathbf{U}_f^{\ell,n+1/2}$ using a hyperbolic tracing scheme. Note that advection on the LHS of eq. (14) used in the hyperbolic tracing scheme is transporting with an effective velocity \mathbf{U}/χ . This ‘advection’ velocity is found by averaging from cell-centres to edges; $(\mathbf{U}/\chi)_f^{\ell,n} = \text{Av}^{C \rightarrow F}(\mathbf{U}/\chi)^{\ell,n}$. In principle, if \mathbf{U}/χ is large enough it may be necessary to use a CFL condition based on $(\mathbf{U}/\chi)\Delta t/\Delta x < 1$, however this situation was not encountered here. To ensure the advection velocities are divergence free, we solve for the pressure field ϕ^ℓ ,

$$\nabla^\ell \cdot (\chi_f^{\ell,n} \nabla^\ell \phi^\ell) = \nabla^\ell \cdot \mathbf{U}_f^{\ell,n+1/2}, \quad (15)$$

which is used along with the freestream preservation correction from the previous timestep to construct the final advection velocity:

$$\mathbf{U}_{AD}^\ell = \mathbf{U}_f^{\ell,n+1/2} - \chi_f^{\ell,n} \nabla^\ell \phi^\ell + \mathbf{U}_p^\ell. \quad (16)$$

2. *Update scalar fields H, Θ, λ .* For each scalar field $s \in \{H, \Theta, \lambda\}$, upwinded face-centred scalar values $s^{\ell,n+1/2}$ are predicted using the hyperbolic tracing scheme described in [48], from which we find the source term $S_s^{\ell,n+1/2}(\mathbf{u}) = \nabla^\ell \cdot (\mathbf{u} s^{\ell,n+1/2})$ due to the divergence of advected fluxes. The scalar λ is updated by solving eq. (12) explicitly,

$$\lambda^{\ell,n+1} = \lambda^{\ell,n} + \Delta t^\ell S_\lambda^{\ell,n+1/2}(\mathbf{U}_{AD}^{\ell,n+1/2}). \quad (17)$$

We treat the diffusion terms in the enthalpy and bulk composition evolution equations implicitly,

$$H^{\ell,n+1} - \Delta t^\ell N_H(H^{\ell,n+1}, \Theta^{\ell,n+1}) = H^{\ell,n} - \Delta t^\ell \left[S_H^{\ell,n+1/2}(\mathbf{V}_{frame}) + S_\theta^{\ell,n+1/2}(\mathbf{U}_{AD}^{\ell,n+1/2}) \right], \quad (18)$$

$$\Theta^{\ell,n+1} - \Delta t^\ell N_\Theta(H^{\ell,n+1}, \Theta^{\ell,n+1}) = \Theta^{\ell,n} - \Delta t^\ell \left[S_\Theta^{\ell,n+1/2}(\mathbf{V}_{frame}) + S_{\Theta_I}^{\ell,n+1/2}(\mathbf{U}_{AD}^{\ell,n+1/2}) \right], \quad (19)$$

where N_H and N_Θ are nonlinear operators for the diffusive terms.

$$N_H(H, \Theta) = \nabla^\ell \cdot \left\{ [(1-k)\chi(H, \Theta) + k] \nabla^\ell \theta(H, \Theta) \right\}; \quad N_\Theta(H, \Theta) = \frac{1}{Le} \nabla^\ell \cdot \left[\chi(H, \Theta) \nabla^\ell \Theta_I(H, \Theta) \right]. \quad (20)$$

The coupled set of eqs. (18) and (19) are solved by the nonlinear FAS multigrid algorithm [45], using a Gauss-Seidel routine for both the smoothing steps and for solving on the coarsest level. When evaluating the nonlinear operators (eq. (20)), and following the solve, we compute $\theta, \Theta_I, \Theta_S$ and χ using the phase diagram described by eqs. (9) to (11).

3. *Compute new velocity $\mathbf{U}^{\ell,n+1}$.* We first compute an unprojected velocity $\mathbf{U}_*^{\ell,n+1}$, treating the viscous and Darcy terms implicitly;

$$\left\{ 1 - \Delta t^\ell Pr \left[\left(\nabla^\ell \right)^2 - \frac{1}{Da} \frac{\chi^{\ell,n+1}}{\Pi^{\ell,n+1}} \right] \right\} \mathbf{U}_*^{\ell,n+1} = \mathbf{U}^{\ell,n} + \Delta t^\ell \left\{ -\chi^{\ell,n+1/2} \nabla^\ell \pi^{\ell,n-1/2} + \chi^{\ell,n+1/2} \left[Pr \left(Ra_T \theta^{\ell,n+1/2} - Ra_C \Theta^{\ell,n+1/2} \right) - \mathbf{U}_{AD,CC}^{\ell,n+1/2} \cdot \nabla^\ell \left(\frac{\mathbf{U}}{\chi} \right)^{\ell,n+1/2} \right] \right\}, \quad (21)$$

where $\mathbf{U}_{AD,CC}^{\ell,n+1/2} = \text{Av}^{F \rightarrow C} \mathbf{U}_{AD}^{\ell,n+1/2}$ and $(\mathbf{U}/\chi)^{\ell,n+1/2}$ is found using the same hyperbolic tracing scheme used in step 2 for scalar fields.

Following the procedure detailed in step 6 (section 2.2) of [44], we then apply a cell-centred projection operator to $\mathbf{U}_*^{\ell,n+1}$ in order to obtain the approximately divergence-free $\mathbf{U}^{\ell,n+1}$ [48];

$$\nabla^\ell \cdot \chi^{\ell,n+1/2} \nabla^\ell \pi^{\ell,n+1/2} = \frac{1}{\Delta t^\ell} \nabla^\ell \cdot \left(\mathbf{U}_*^{\ell,n+1} + \Delta t^\ell \chi^{\ell,n+1/2} \nabla^\ell \pi^{\ell,n-1/2} \right), \quad (22)$$

$$\mathbf{U}^{\ell,n+1} = \mathbf{U}_*^{\ell,n+1} + \Delta t^\ell \chi^{\ell,n+1/2} \left(\nabla^\ell \pi^{\ell,n-1/2} - \nabla^\ell \pi^{\ell,n+1/2} \right), \quad (23)$$

where eq. (22) is solved first for $\pi^{\ell,n+1/2}$ to yield the pressure correction in eq. (23). Due to the approximate nature of this projection, we find that it is necessary to manually enforce the physical condition that cells with zero liquid fraction cannot have any fluid flow; $\mathbf{U}^{\ell,n+1} = 0$ if $\chi^{\ell,n+1} = 0$.

4. *Recursively update finer levels.* If a finer level $\ell + 1$ exists, it is updated with a fractional timestep $\Delta t^{\ell+1} = \Delta t^\ell / n_{ref}^\ell$ until it reaches the time of the current level, $t_\ell + \Delta t^\ell$. Where required, boundary conditions for the solve on level $\ell + 1$ are computed by interpolation in space and time using the solution just found on level ℓ .
5. *Synchronize with finer levels.* If a finer level $\ell + 1$ exists, the current level ℓ is synchronized with all finer levels as described in section 3.4.

3.4. Synchronization

Because coarse- and fine-level solutions have been evolved independently, our solution will be continuous but will not preserve properties like conservation, freestream preservation, and the divergence constraint (eq. (5)) on the composite multilevel mesh. To ensure that our scheme is conservative, and that the velocity field is divergence free at coarse-fine interfaces, we must perform a series of synchronization steps once all levels finer than some base level ℓ_{base} , have reached the same time, t^{sync} .

1. *Refluxing.* To ensure conservation, we correct for the flux mismatch $\delta\mathbf{F}^\ell$ at the interface between a coarse level ℓ and the fine level, $\ell + 1$. Following [48,44], we introduce this mismatch via a source term in our original evolution equations and decompose the flux-corrected solution into the portion already computed and a small correction, e.g. $\mathbf{U}_{\text{Corr}}^{\ell,n+1} = \mathbf{U}^{\ell,n+1} + \delta\mathbf{U}^{\ell,n+1}$. The discretized equation for \mathbf{U} , eq. (21), becomes

$$\left\{ 1 - \Delta t^\ell Pr \left[(\nabla^\ell)^2 - \frac{\chi^{\ell,n+1}}{Da \Pi^{\ell,n+1}} \right] \right\} (\mathbf{U}^{\ell,n+1} + \delta\mathbf{U}^{\ell,n+1}) = \mathbf{U}^{\ell,n} + \Delta t^\ell \nabla \cdot (\mathbf{F}_O - \delta\mathbf{F}_U^\ell), \quad (24)$$

where \mathbf{F}_O is the original source term and we have used eq. (23) to eliminate \mathbf{U}_* . Using eq. (21) we simplify eq. (24) to

$$\left\{ 1 - \Delta t^\ell Pr \left[(\nabla^\ell)^2 - \frac{\chi^{\ell,n+1}}{Da \Pi^{\ell,n+1}} \right] \right\} \delta\mathbf{U}^{\ell,n+1} = -\Delta t^\ell D_R \delta\mathbf{F}_U^\ell \quad \text{for } \ell \geq \ell_{\text{base}}, \quad (25)$$

which is solved implicitly for $\delta\mathbf{U}^{\ell,n+1}$ where D_R , the ‘‘reflux divergence’’, is the part of the coarse-level divergence operator operating on the flux mismatch on the coarse-fine interface [48]. For enthalpy and bulk composition, we must also solve implicitly for the correction $(\delta H, \delta\Theta)$ to ensure that our scheme is stable [39,44]. In order to diffuse the correction in a physically appropriate manner, we note that $(\delta H, \delta\Theta)$ is of the same order of magnitude as the flux mismatches $(\delta\mathbf{F}_H^\ell, \delta\mathbf{F}_\Theta^\ell)$, which are themselves $O(\Delta x^2)$ [51]. Hence, we make the approximation $\chi(H + \delta H, \Theta + \delta\Theta) \approx \chi(H, \Theta)$ and use the linearised forms of the nonlinear diffusion operators defined in eq. (20),

$$N_H(H + \delta H, \Theta + \delta\Theta) \approx N_H(H, \Theta) + \nabla \cdot [(1 - k)\chi + k] \nabla \delta\theta, \quad (26)$$

$$N_\Theta(H + \delta H, \Theta + \delta\Theta) \approx N_\Theta(H, \Theta) + \frac{1}{Le} \nabla \cdot \chi \nabla \delta\theta_l, \quad (27)$$

in order to compute the composite corrections

$$\delta H^{\ell,n+1} - \Delta t \nabla^\ell \cdot \left[(1 - k)\chi^{\ell,n+1} + k \right] \nabla^\ell \delta\theta (\delta H^{\ell,n+1}, \delta\Theta^{\ell,n+1}) = -\Delta t^\ell D_R \delta\mathbf{F}_H^\ell \quad \text{for } \ell \geq \ell_{\text{base}}, \quad (28)$$

$$\delta\Theta^{\ell,n+1} - \Delta t \frac{1}{Le} \nabla^\ell \cdot \chi^{\ell,n+1} \nabla^\ell \delta\theta_l (\delta H^{\ell,n+1}, \delta\Theta^{\ell,n+1}) = -\Delta t^\ell D_R \delta\mathbf{F}_\Theta^\ell \quad \text{for } \ell \geq \ell_{\text{base}}. \quad (29)$$

Finally, the reflux correction for λ can simply be applied explicitly:

$$\lambda^{\ell,n+1} = \lambda^{\ell,n+1} - \Delta t^\ell D_R \delta\mathbf{F}_\lambda^\ell \quad \text{for } \ell \geq \ell_{\text{base}}. \quad (30)$$

2. *Apply multilevel projection.* To ensure the velocity is divergence free at coarse-fine boundaries, we apply a composite projection as in [48]. A correction to the pressure e_s is found by solving across the entire AMR hierarchy,

$$\nabla^\ell \cdot \chi^\ell \nabla^\ell e_s = \nabla^\ell \cdot \mathbf{U}^\ell \quad \text{for } \ell \geq \ell_{\text{base}}, \quad (31)$$

where boundary conditions are as in [44]. The velocity is then corrected:

$$\mathbf{U}^\ell := \mathbf{U}^\ell - \chi^\ell \nabla^\ell e_s \quad \text{for } \ell \geq \ell_{\text{base}}. \quad (32)$$

3. *Compute freestream preservation correction.* The freestream preservation correction is computed in a composite sense for use in the next timestep. As in [48], we solve for the correction e_λ^ℓ ,

$$\nabla^\ell \cdot \left(\chi^\ell \nabla^\ell e_\lambda^\ell \right) = \zeta \frac{\lambda^\ell - 1}{\Delta t^{\ell_{\text{base}}}} \quad \text{for } \ell \geq \ell_{\text{base}}. \quad (33)$$

For stability, the constant satisfies $\zeta < 1$ such that the correction velocity,

$$\mathbf{U}_p^\ell = \chi^\ell \nabla^\ell e_\lambda^\ell \quad \text{for } \ell \geq \ell_{\text{base}}, \quad (34)$$

only partially corrects the solution in a given timestep. Additionally, we find from a linear stability analysis (Appendix B) that the condition $\zeta < 2(1 - \sigma)$ must also be met, in contrast to [44] who instead found that $\zeta\sigma < 0.5$ was a sufficient condition for stability. We typically choose $\zeta = 0.95, \sigma = 0.4$.

4. *Replace coarse grid solution with average of fine grid solution.* For consistency, the solution on covered regions is replaced by the average of the solution on the overlying fine region, in a process applied recursively from ℓ_{max} down to ℓ_{base} .

3.5. Initialisation

At the start of a computation, or after the meshes have changed due to regridding, we must initialise the solution. We follow the steps described in [48] (section 3.7) and [44] (section 2.4), repeating the key details here for convenience.

1. At the start of a computation, the primary variables H , Θ and \mathbf{U} are given some initial value. An approximation to the lagged pressure π is then found by taking a single non-subcycled timestep across all levels.
2. If regridding during a computation, newly defined fine meshes are filled by linear interpolation from the solution on the underlying coarse mesh.
3. Following any change of the grids, including at the start of a computation, the velocity field is projected in a composite sense to ensure that it is divergence free, and the freestream preservation correction \mathbf{U}_p is recalculated based on the newly-initialized λ field.

3.6. Refinement

Regridding occurs every n_r timesteps. During this process, the solution at each grid cell on each level, whether valid or currently covered, is tested against some specified criteria to determine if refinement is required, in which case the cell is *tagged* for refinement. A new set of grids is then found where these tagged cells are covered by a finer level, whilst still satisfying the rules introduced in section 3.1 regarding proper nesting. This procedure enables the refinement or coarsening of the grids over time.

The appropriate refinement criteria will vary between different problems. Typical choices we make are to tag cells which lie on the mush-liquid phase boundary, where $\chi < 1$ and $\chi = 1$ in adjacent cells, or to tag cells where the undivided gradient of the enthalpy field is above some threshold value. The strategy employed for each problem we consider will be explained where appropriate.

3.7. Algorithm for Darcy flow

For flow in a Hele-Shaw cell with a narrow gap width, such that $\Pi_c Da \ll 1$, the momentum equation (eq. (6)) is well approximated by Darcy's law

$$\mathbf{U} = \Pi \left[-\nabla \tilde{p} + Da (Ra_T \theta - Ra_C \Theta_l) \mathbf{e}_z \right], \quad (35)$$

where we have rescaled the pressure $\tilde{p} = (Da/Pr)p$. In this case, the momentum \mathbf{U} becomes a diagnostic rather than prognostic variable, and is entirely determined by the instantaneous enthalpy and bulk concentration. Solving eq. (35) instead of eq. (6) can be achieved via the changes to the algorithm described below. The resulting method requires less computational effort, making it advantageous in settings where eq. (35) is a good description of the physics such as flow in a narrow Hele-Shaw cell.

In the single level update (section 3.3) steps 2, 4, and 5 are unchanged, whilst step 3 is no longer required. During step 1, where we compute advection velocities, the predicted face-centred velocity is now computed at time t^n by averaging the known permeability, temperature, and liquid concentration from cell centres to faces;

$$\mathbf{U}_f^{\ell,n} = Av^{C \rightarrow F} \left[\Pi^{\ell,n} Da \left(Ra_T \theta^{\ell,n} - Ra_C \Theta_l^{\ell,n} \right) \mathbf{e}_z \right], \quad (36)$$

whilst the pressure correction is found by solving

$$\nabla \cdot \left(\Pi_f^{\ell,n} \nabla \phi^\ell \right) = \nabla \cdot \mathbf{U}_f^{\ell,n}. \quad (37)$$

The final advection velocity is given by $\mathbf{U}_{AD}^\ell = \mathbf{U}_f^{\ell,n} - \Pi_f^{\ell,n} \nabla \phi^\ell + \mathbf{U}_p^\ell$.

During synchronization (section 3.4) we no longer need to compute a reflux correction for the momentum and the multilevel projection (step 2) is redundant. During initialisation we no longer need to compute an approximation to the lagged pressure in step 1. Following a change of grids (step 3) we do not need to project the velocity field.

4. Results

We next demonstrate the accuracy and efficiency of our method. First, we will consider two benchmark problems which separately test the two key components of the algorithm: solving for the nonlinear coupling of enthalpy and bulk composition (section 4.1), and the solution of the Darcy-Brinkman momentum equation (section 4.2). Then we consider a series of transient solutions to the fully coupled problem on both uniform and adaptive meshes, which reveal the computational savings provided by the implementation of AMR.

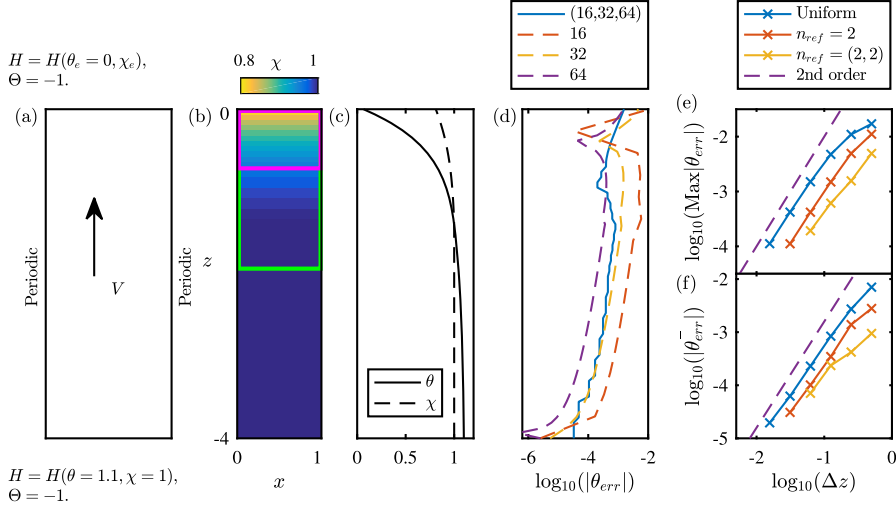


Fig. 4. Diffusive solidification benchmark (section 4.1). (a) Boundary conditions, (b) typical porosity field for an AMR simulation with $\mathcal{C} = 5$, $\mathcal{S} = 5$, and (c) 1D exact porosity and temperature profiles. Refinement in (b) is added where undivided temperature gradients are large; the borders of the refined regions $\ell = 1$ (green) and $\ell = 2$ (magenta) are highlighted. Defining $\theta_{err} = \theta - \theta_{analytic}$, in (d) we plot 1D error profiles for three uniform mesh solutions with $N_z = 16$ (red), 32 (gold), and 64 (purple) alongside an AMR solution (blue) with a coarse grid $N_z = 16$ and two refined levels. Convergence plots (e, f) confirm that our scheme is second order accurate in space, and that AMR enhances the accuracy of the solution. The refinement ratio between successive levels n_{ref} is either a single value for the case with one refined level (red) or two values for the case of two refined levels (yellow).

4.1. Benchmark 1: diffusive solidification

An analytic solution exists for purely diffusive solidification in a semi-infinite domain [17], which we use to validate our solution to the thermodynamic aspect of the problem. Specifically, we solve our governing equations with no salt diffusion ($Le = \infty$), equal solid and liquid properties ($k = c_p = 1$), no fluid motion ($Ra_T = Ra_C = 0$), and constant vertical frame advection $\mathbf{V}_{frame} = V \mathbf{e}_z$ with $V = 1$ on the domain illustrated in Fig. 4(a). The bottom boundary is held at a fixed temperature above the liquidus $\theta = 1.1$; we then numerically invert the analytic solution to determine the far-field temperature $\theta(z = -\infty)$ and the corresponding solution within the domain. The top boundary is held at the eutectic temperature and porosity, $H_{top} = H(\theta_e = 0, \chi_e)$, and the x -direction is periodic. Initially, the domain is entirely liquid $\chi = 1$ and at temperature $\theta = \theta_{analytic}$. Simulations are stopped once a steady state is reached, defined by

$$\max \left| \theta^{\ell, n+1} - \theta^{\ell, n} \right| < 10^{-4} \Delta t, \quad \text{for } \ell_{min} \leq \ell \leq \ell_{max}. \quad (38)$$

We make use of the adaptive mesh to refine around regions where we expect the error to be largest: the mush-liquid interface and wherever the undivided temperature gradient is large ($\Delta x^\ell |\nabla^\ell \theta^\ell| > 1/2$). A typical steady state porosity field is shown in Fig. 4(b) for a simulation with three levels of refinement ($\ell_{max} = 2$) and refinement ratio of 2 between each level, which we denote by $n_{ref} = (2, 2)$. A one-dimensional temperature profile is plotted in Fig. 4(c), along with the error $\theta_{err} = \theta - \theta_{analytic}$ for a selection of AMR and uniform mesh simulations in Fig. 4(d) which reveal that the refined regions greatly reduce the local error compared to the coarser grid. We perform a convergence test using a uniform mesh, an adaptive mesh with one level of refinement ($n_{ref} = 2$) and an adaptive mesh with two levels of refinement ($n_{ref} = (2, 2)$), which confirms that the scheme has second order spatial accuracy (Fig. 4(e,f)).

4.2. Benchmark 2: convection in a porous medium

To test the coupled evolution of the momentum and heat transfer, we consider convective flow in a porous medium with isothermal heated and cooled sidewalls, and without phase change ($\mathcal{S} = 0$). Following previous studies [28,53,52] we simulate a fluid with equal solid and liquid properties ($k = c_p = 1$), a Prandtl number $Pr = 1$, and consider a variety of values for the thermal Rayleigh number Ra_T and modified Darcy number $\bar{Da} = Da \chi^3 / (1 - \chi)^2$ whilst $Ra_C = 0$. For the first test case we use a square domain ($0 < x < 1$, $0 < z < 1$), with imposed porosity $\chi = 0.4$ and boundary conditions as illustrated by case b in Fig. 5(a).

Following [39], we use this problem as a convergence test, and use a fixed variable mesh rather than an adaptive mesh. This choice ensures that all simulations have the same grid hierarchy at all timesteps, permitting an accurate test of the convergence properties. We choose a mesh with enhanced resolution on the left and right walls of the domain to better resolve the boundary layers (see Fig. 5(b)). Once simulations have reached steady state, we compute the Nusselt number,

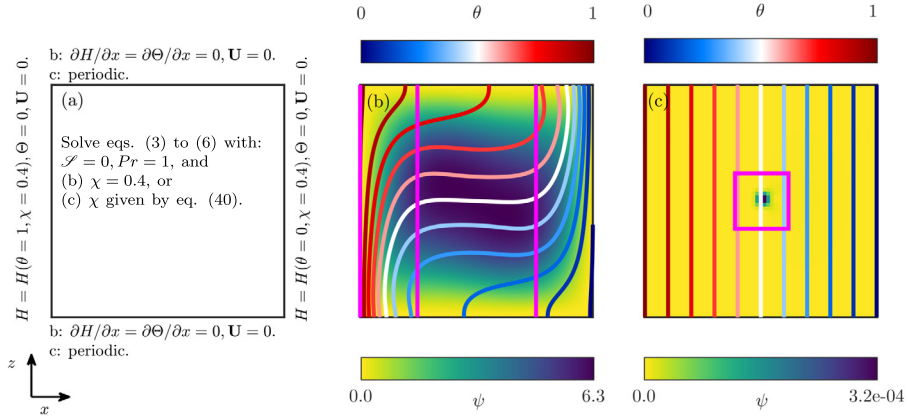


Fig. 5. (a) Boundary conditions for the problem of convection in a fixed porous medium with isothermal sidewalls of differing temperature. (b) Steady state solution with a uniform porosity $\chi = 0.4$ and $Ra_T = 10^5$, $\bar{D}a = 10^{-2}$. We implement a fixed variable mesh with coarse resolution in the interior and higher resolution along the left/right boundaries, as indicated by the magenta rectangles. Coloured lines show isotherms (top scale) whilst background yellow-green-blue color shading shows the streamfunction (lower scale). (c) Solution at time $t = 10^{-3}$ with a variable porosity field given by eq. (40), other parameters as in (a), and an adaptive mesh which has refined around the region of stronger flow at the centre.

Table 1

Comparison of the averaged Nusselt number Nu (given by eq. (39)) with previous studies for $\chi = 0.4, k = 1, c_p = 1, Pr = 1$ and $\bar{D}a \equiv Da \chi^3 / (1 - \chi)^2$. Simulations run on a variable mesh with a coarse resolution Δx^{coarse} and a fine resolution $\Delta x^{coarse}/2$ occupying the left and right quarters of the domain. Simulations are pursued at increasing resolution until Nu has converged to 3 significant figures.

$\bar{D}a$	Ra_T	Δx^{coarse}					[28]	[52]
		1/32	1/64	1/128	1/256	1/512		
10^{-2}	10^3	1.01	1.01	1.01			1.01	1.01
	10^4	1.41	1.41	1.41			1.41	1.41
	10^5	3.18	3.17	3.17			3.17	3.17
10^{-6}	5×10^5	5.28	5.26	5.25	5.24	5.24	5.24	5.24
	10^7	1.06	1.07	1.08	1.08		1.08	1.08
	10^8	2.93	3.01	3.07	3.07		3.06	3.07
	10^9	13.0	13.3	13.2	13.0	13.0	12.9	12.85

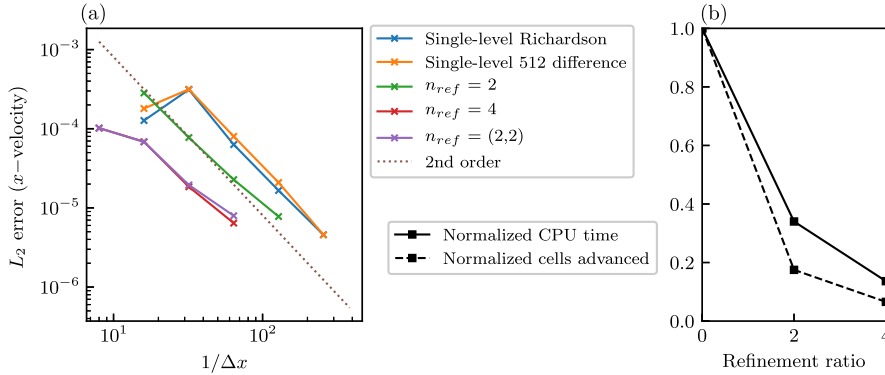


Fig. 6. (a) L_2 convergence of x -velocity and (b) AMR performance for the problem of convection in a porous medium with spatially varying porosity (section 4.2). “Single-level Richardson” errors are calculated as the difference between computations where the grid resolution differs by a factor of 2. The “Single-level 512” error is the difference between each computation and an equivalent calculation performed on a uniform 512×512 mesh. Note that $n_{ref} = 4$ differs only slightly from $n_{ref} = (2, 2)$. In (b), we compare three simulations with the same finest resolution ($\Delta x = 1/256$) and a refinement ratio of 0 (no refinement), 2, or 4. Normalized values are relative to the single-level case, where the number of cells advanced was 2.10×10^7 and the CPU time was 450 seconds. For ease of comparison of timings, simulations were performed on a single 2.6 GHz processor, with GNU compilers and Ubuntu 14.04 operating system.

$$Nu = \frac{\partial \theta}{\partial x}, \tag{39}$$

averaged over the left and right boundaries of the domain. As seen in Table 1, our results agree well with [28] and [52] over a wide range of parameters. For the cases with $\bar{D}a = 10^{-6}$, our solutions indicate the presence of a fine scale boundary layer; consequently the quadratic convergence regime is only achieved on the finest grids.

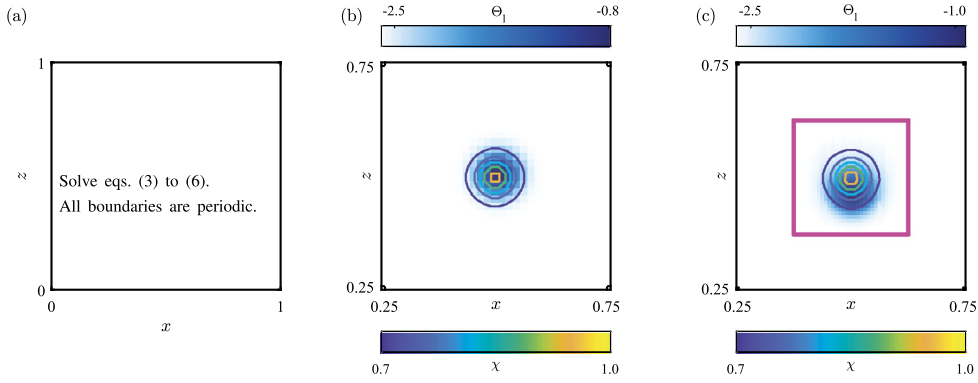


Fig. 7. (a) Domain for simulations of a high porosity inclusion (section 4.3.1). (b) Initial solution as described by eqs. (41) to (43), showing the liquid concentration Θ_l (color, top scale) and porosity χ (contours, lower scale) for a uniform mesh with $\Delta x = 1/32$. (c) Solution at $t = 0.5$ for an AMR simulation with one level of refinement ($n_{\text{ref}} = 2$) and a coarse resolution $\Delta x = 1/32$. The refined region is outlined by the magenta box.

To test the capability of the code to handle regions which are entirely solid ($\chi = 0$) and to adaptively refine around regions of interest, we now impose a porosity which varies spatially from 1 at the centre of the domain $(x_c, z_c) = (0.5, 0.5)$ to $\sim 10^{-45}$ at the boundaries according to

$$\chi = \min \left\{ \exp \left[-\frac{(x - x_c)^2 + (z - z_c)^2}{a} \right], 1.0 \right\}; \quad a = 1/500. \quad (40)$$

The minimum value of eq. (40) ($\chi \sim 10^{-45}$) yields a permeability $\Pi = 0$ to within machine precision. The resulting convection is confined to the interior of the domain, as illustrated in Fig. 5(c), and we choose to refine around regions where the dimensionless velocity satisfies $|\mathbf{U}| > 15$. The solution is advanced from an initial state given by $\theta = x$, $\mathbf{U} = 0$ until a dimensionless time $t = 2 \times 10^{-4}$, using a fixed timestep $\Delta t = 1.6 \times 10^{-4} \Delta x$.

Following [44] we first demonstrate that our scheme is second-order accurate via Richardson error estimation, where the error is estimated as the difference between grids with successive resolutions. Then, we compute the error for both uniform and AMR simulations by considering the difference between them and a uniform 512×512 mesh. These results are shown in Fig. 6(a), where the grid spacing on the x -axis indicates the coarsest resolution used in a particular simulation. We see that the error for the 64×64 single level case is similar in magnitude to the 32×32 case with $n_{\text{ref}} = 2$, and the 16×16 case with $n_{\text{ref}} = 4$ or $n_{\text{ref}} = (2, 2)$, demonstrating the accuracy of our mesh refinement.

We now illustrate the computational performance of our algorithm by measuring the runtime and total number of grid cells advanced during the 256×256 single-level simulation, and equivalent AMR calculations. As shown in Fig. 6(b), the use of AMR provides significant savings both in terms of memory (approximately indicated by the number of cells advanced) and runtime.

4.3. Fully coupled system

We now apply our method to the fully coupled problem (eqs. (3) to (6)) featuring flow and phase change. First we consider the evolution of a highly porous inclusion within a relatively impermeable material (section 4.3.1), which confirms the accuracy and performance benefits of our scheme. Second, we simulate the solidification of an initially liquid solution from above (“fixed chill”) (section 4.3.2), which demonstrates the capability of our method to solve more complex problems.

4.3.1. High porosity inclusion

Here, we simulate the evolution of a high porosity region within a relatively impermeable mushy layer on a periodic domain $0 < x < 1, 0 < z < 1$. Such inclusions can act as the precursor to fully developed channels [54], and provide a simple test case for the full coupled system of equations (eqs. (3) to (6)).

We consider a material with differing solid and liquid properties ($k = 2$, $c_p = 0.5$) which is described by the dimensionless parameters $\mathcal{C} = 10$, $Pr = 10$, $Ra_T = 10$, $Ra_C = 10^6$, $Da = 10^{-3}$, $\mathcal{S} = 5$, $Le = 100$, $\mathbf{V}_{\text{frame}} = 0$ on a doubly periodic domain. An initial state is given by

$$H = \mathcal{S} + 0.5, \quad (41)$$

$$\Theta = -\mathcal{C}/2 + (\mathcal{C}/2 - 1.0) \exp \left[-\frac{(x - 0.5)^2 + (z - 0.6)^2}{0.04^2} \right], \quad (42)$$

$$\mathbf{U} = 10^{-3} \sin(\pi x) \mathbf{e}_z, \quad (43)$$

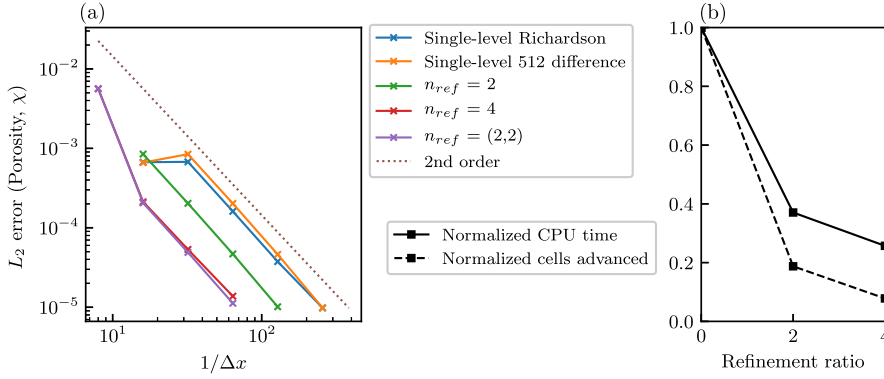


Fig. 8. (a) L_2 convergence of porosity and (b) AMR performance for the problem of buoyancy-driven flow from a high porosity inclusion in a reactive porous medium (section 4.3.1). “Single-level Richardson” errors are calculated as the difference between computations where the grid resolution differs by a factor of 2. The “Single-level 512” error is the difference between each computation and an equivalent calculation performed on a uniform 512×512 mesh. In (b), we compare three simulations with the same finest resolution ($\Delta x = 1/256$) and a refinement ratio of 0 (no refinement), 2, or 4. Normalized values are relative to the single-level case, where the number of cells advanced was 3.15×10^6 and the CPU time was 180 seconds. For ease of comparison of timings, simulations were performed on a single 2.6 GHz processor, with GNU compilers and Ubuntu 14.04 operating system.

which generates a small inclusion of higher porosity and higher fluid concentration, with a weak initial flow perturbation. This is evolved in time until $t = 1.5 \times 10^{-4}$ with a fixed timestep $\Delta t = 8 \times 10^{-4} \Delta x$. In order to replicate a hydrostatic pressure gradient on a periodic domain, we introduce the additional body force

$$\mathbf{F}_B = \chi_b Pr (Ra_C \Theta_{l,b} - Ra_T \theta_b) \mathbf{e}_z, \quad (44)$$

where χ_b , θ_b , and $\Theta_{l,b}$ are computed from the background enthalpy and bulk concentration $(H_b, \Theta_b) = (\mathcal{S} + 0.5, -\mathcal{C}/2)$. We plot the initial and final states in Fig. 7. As the dense concentrated fluid sinks under gravity, it increases the local porosity by dissolution of the solid matrix.

As in section 4.2, we consider simulations on both a uniform mesh and a variety of adaptive meshes. The refinement criteria is chosen to focus on regions where the undivided porosity gradients are large, i.e. $\Delta x |\nabla \chi| > 0.1$. As seen in Fig. 8, we achieve second order spatial accuracy on a uniform mesh, whilst the AMR solution reduces the error to roughly the same magnitude as an equivalent uniform fine-level solution. The use of AMR with a refinement ratio $n_{ref} = 4$ reduces the memory requirement by a factor of ≈ 10 , and the run time by up a factor of ≈ 3 (Fig. 8(b)).

4.3.2. Fixed chill in a Hele-Shaw cell

To demonstrate the ability of our method to handle more complicated problems, we now simulate the solidification of an initially liquid region from above with $\mathbf{V}_{frame} = 0$. We consider a unit domain $0 \leq x \leq 1, 0 \leq z \leq 1$, with symmetry boundary conditions on the side walls (i.e. no lateral gradients) and fixed temperatures applied at the no slip top and bottom surfaces which are impermeable to salt according to:

$$H(z=0, t) = \mathcal{S} + 1.3, \quad H(z=1, t) = \mathcal{S}/2, \quad (45)$$

$$\frac{\partial \Theta}{\partial z}(z=0, t) = \frac{\partial \Theta}{\partial z}(z=1, t) = 0. \quad (46)$$

The initial state is defined by $H(t=0) = \mathcal{S} + 1.3$, $\Theta(t=0) = -1.0$, $\mathbf{U}(t=0) = 0$, and we solve for the resulting flow and solidification given the parameters

$$Ra_C = 10^6, \quad Ra_T = 0, \quad \mathcal{C} = 2, \quad Da = 10^{-4}, \quad Pr = 10, \quad \mathcal{S} = 5, \quad \Pi_c = 10^4, \quad k = 1, \quad c_p = 1. \quad (47)$$

In general, the choice of mesh refinement criteria will vary depending on the nature of the problem at hand. Here, simulations were run on a base mesh 32×32 , with 2 levels of refinement using $n_{ref}^{\ell=0} = 4$, $n_{ref}^{\ell=1} = 2$. The first level of refinement is added wherever the porosity $\chi < 1$, in order to capture dynamics in the mushy layer where the instability begins. The second level of refinement is added wherever the domain has already been refined to level 1 and

$$\Delta x |\nabla \chi| (\Theta + \mathcal{C}) (\mathbf{U} \cdot \hat{\mathbf{z}}) < -0.5, \quad (48)$$

which is chosen to ensure we add further levels of refinement around the channels where porosity gradients are large (hence the $\Delta x |\nabla \chi|$ term), bulk solute concentration is high (hence the $\Theta + \mathcal{C}$ term) and there is significant downwelling (hence the $\mathbf{U} \cdot \hat{\mathbf{z}}$ term). As we are primarily interested in the mushy-layer dynamics we choose not to refine around the plumes once they leave the mushy region, which are left to mix and diffuse as they are advected downward through the liquid. Further levels of refinement could be added using eq. (48) to increase the resolution around the narrow channels.

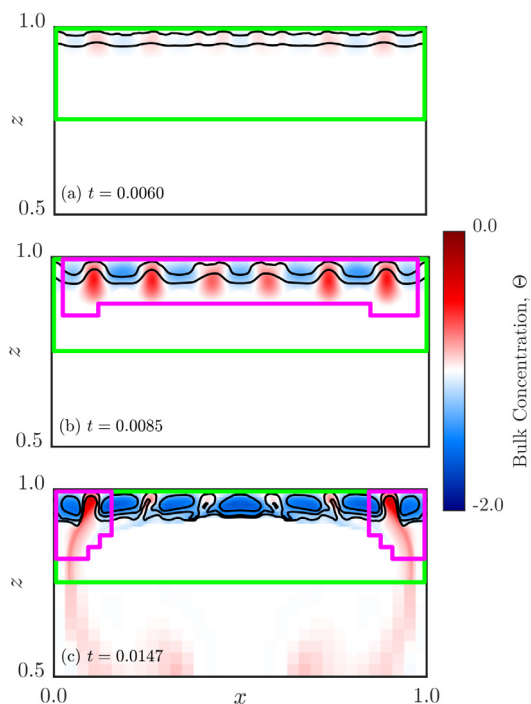


Fig. 9. Bulk concentration (color), porosity (contours, $\chi = 1 - 0.2n$, $n = 0, 1, 2, 3$) and AMR levels ($\ell = 1$: green, $\ell = 2$: magenta) for a simulation of convection during solidification of a binary alloy from a fixed cold upper boundary in a Hele-Shaw cell (parameters given in eq. (47)). Only the top half of the domain is shown, as this contains the most interesting dynamics.

The evolution of the bulk concentration and porosity, along with the AMR grids, is shown in Fig. 9. The mushy-layer grows diffusively at early times, before the convective instability triggers the formation of channels through which solute is transported into the underlying fluid. The transport of solute out of the mushy layer reduces the local porosity, whilst the salty plumes sink into the liquid. As solidification continues some channels become extinct while others merge with their neighbours, causing the typical channel spacing to increase. The highest level of refinement tracks the channels across the domain, ensuring that they are well resolved with around 10 grid cells across each channel. This provides an efficient means to overcome the limitations on resolution for resolving flow in channels experienced in previous simulations in wide domains with many channels.

5. Conclusions

We have developed a method for simulating the transport of heat, mass, and solute during binary alloy solidification, using a finite volume approach with adaptive mesh refinement. Our algorithm features local refinement in space and time on block structured Cartesian grids, implemented via the Chombo framework, along with a series of synchronization steps to ensure flux conservation. We have extended the Chombo framework to include a solver for nonlinear elliptic equations using the Full Approximation Storage multigrid scheme. Using a hierarchy of test cases, we have demonstrated the second-order convergence of our method and the performance benefits which AMR can provide. The ability of our scheme to handle transient problems, with features evolving in time, has also been demonstrated.

Future work will look to extend the current algorithm to three dimensions, where we expect the benefits of using AMR to be amplified. In the current two-dimensional setting, this method could be particularly useful for efficiently simulating flows in a Hele-Shaw cell, as demonstrated in section 4.3.2. Here, we expect that our AMR scheme will be most useful for problems with strongly multiscale features, such as solidification of mushy layers with many high porosity channels where the channel width is small relative to their spacing. We anticipate that potential applications may include the studies of sea ice growth, metal casting, and processes in the Earth's interior. The ideas developed here may also be useful in designing algorithms for other related problems, such as flow through different types of porous materials.

Supplementary material

The code used to implement this method and produce the figures in this paper can be found at <https://github.com/jrgparkinson/mushy-layer/tree/v1.0>.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

JRGP acknowledges a PhD studentship funded by NERC award NE/L002612/1. AJW acknowledges European Union FP7 award PCIG13-GA-2013-618610 SEA-ICE-CFD. Work at LBL was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research (ASCR), of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, as a part of the FASTMath SciDAC Institute. All authors acknowledge the Royal Society International Exchanges Award IE141071. We thank two anonymous reviewers who helped to improve this manuscript.

Appendix A. Enthalpy method

Given our idealised phase diagram, described by eq. (9) to (11), we can compute the temperature θ , porosity χ , liquid concentration Θ_l and solid concentration Θ_s for a given enthalpy H and bulk concentration Θ :

$$\chi = \begin{cases} 0 & H \leq H_S \\ (H - \theta_e c_p) / (\mathcal{S} + \theta_e(1 - c_p)) & H_S < H \leq H_E \\ \left[-B - \sqrt{B^2 - 4AC} \right] / (2A) & H_E < H < H_L \\ 1 & H \geq H_L \end{cases} \quad (49)$$

$$\text{where } A = \mathcal{C}(c_p - 1) + \mathcal{S}(p_s - 1) \quad (50)$$

$$B = H(1 - p_s) - \mathcal{S}p_s + \mathcal{C}(1 - 2c_p) - \Theta(c_p - 1) \quad (51)$$

$$C = Hp_s + c_p(\mathcal{C} + \Theta) \quad (52)$$

$$\theta = \begin{cases} \frac{H}{c_p} & H \leq H_S \\ \theta_e & H_S < H \leq H_E \\ -\Theta_l & H_E < H < H_L \\ H - \mathcal{S} & H \geq H_L \end{cases} \quad (53)$$

$$\Theta_l = \begin{cases} N/A & H \leq H_S \\ \Theta_e & H_S < H \leq H_E \\ \frac{\Theta + \mathcal{C}(1 - \chi)}{\chi + (1 - \chi)p_s} & H_E < H < H_L \\ \Theta & H \geq H_L \end{cases} \quad (54)$$

$$\Theta_s = \begin{cases} \Theta & H \leq H_S \\ \frac{\Theta}{1 - \chi} & H_S < H \leq H_E \\ \frac{p_s \Theta - \mathcal{C}\chi}{\chi + (1 - \chi)p_s} & H_E < H < H_L \\ N/A & H \geq H_L \end{cases} \quad (55)$$

Here θ_e and Θ_e are the eutectic temperature and concentration, which are both 0 in our nondimensionalisation.

Appendix B. Stability of the freestream preservation correction

As described in the main text, the freestream preservation correction \mathbf{U}_p is calculated according to the method in [48] where, given an initial divergence-free estimate $\tilde{\mathbf{U}}$, the revised advection velocity is $\mathbf{U}_{AD} = \tilde{\mathbf{U}} + \mathbf{U}_p$. From eq. (49) of [48] the update for λ , which we then use to compute \mathbf{U}_p via eq. (34), can be written as

$$\frac{\partial \lambda}{\partial t} = -\mathbf{U}_{AD} \cdot \nabla \lambda - \lambda \zeta \frac{\lambda - 1}{\Delta t}, \quad (56)$$

where we have used eq. (33) and eq. (34) to substitute for $\nabla \cdot \mathbf{U}_p$. To analyse the stability of the correction, we consider an upwind advection scheme and replace \mathbf{U}_{AD} with a typical velocity scale U ,

$$\lambda_j^{n+1} = \lambda_j^n - \Delta t \frac{U}{\Delta x} (\lambda_j^n - \lambda_{j-1}^n) - \zeta \lambda_j^n (\lambda_j^n - 1). \quad (57)$$

We now consider the evolution of small deviations $\lambda_j^n = 1 + \epsilon A^n e^{ikx_j}$,

$$1 + \epsilon A^{n+1} e^{ikx_j} = 1 + \epsilon A^n e^{ikx_j} - \sigma \epsilon A^n \left(e^{ikx_j} - e^{ikx_{j-1}} \right) - \zeta \left(1 + \epsilon A^n e^{ikx_j} \right) \epsilon A^n e^{ikx_j}, \quad (58)$$

where we have defined the CFL number $\sigma = (U \Delta t) / \Delta x$. Dropping terms of order ϵ^2 and dividing through by $A^n e^{ikx_j}$ leads to

$$A = 1 - \sigma \left(1 - e^{-ik\Delta x} \right) - \zeta. \quad (59)$$

Taking the absolute value, we find

$$|A|^2 = 1 + \zeta(\zeta - 2) + 2\sigma(1 - \sigma - \zeta) [\cos(k\Delta x) - 1]. \quad (60)$$

We require $|A|^2 < 1$ for the scheme to be stable. By definition, $\zeta > 0$ and $\sigma > 0$, whilst $-2 < \cos(k\Delta x) - 1 < 0$. Therefore $|A|^2$ will be largest when $\zeta + \sigma > 1$, such that the final term of eq. (60) is positive. In this case, the most unstable wavelength gives $\cos(k\Delta x) = -1$ and the criteria for stability becomes

$$\zeta(\zeta - 2) - 4\sigma(1 - \sigma - \zeta) < 0. \quad (61)$$

Solving the resulting quadratic equation for ζ , and taking the root which gives $\zeta > 0$, we find the stability condition:

$$\zeta < 2(1 - \sigma). \quad (62)$$

References

- [1] S.M. Copley, A.F. Giamei, S.M. Johnson, M.F. Hornbecker, The origin of freckles in unidirectionally solidified castings, *Metal. Trans.* 1 (1970) 2193–2204.
- [2] M.I. Bergman, D.R. Fearn, Chimneys on the Earth's inner-outer core boundary? *Geophys. Res. Lett.* 21 (1994) 477–480.
- [3] J.S. Wettlaufer, M.G. Worster, H.E. Huppert, The phase evolution of young sea ice, *Geophys. Res. Lett.* 24 (1997) 1251.
- [4] M.G. Worster, Solidification of an alloy from a cooled boundary, *J. Fluid Mech.* 167 (1986) 481–501.
- [5] W.W. Mullins, R.F. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, *J. Appl. Phys.* 35 (1964) 444–451.
- [6] M.G. Worster, Convection in mushy layers, *Annu. Rev. Fluid Mech.* 29 (1997) 91–122.
- [7] S. Tait, C. Jaupart, Compositional convection in a reactive crystalline mush and melt differentiation, *J. Geophys. Res.* 97 (1992) 6735–6756.
- [8] C.F. Chen, Experimental study of convection in a mushy layer during directional solidification, *J. Fluid Mech.* 293 (1995) 81–98.
- [9] S.S.L. Peppin, P. Aussillous, H.E. Huppert, M.G. Worster, Steady-state mushy layers: experiments and theory, *J. Fluid Mech.* 570 (2007) 69–77.
- [10] R.F. Katz, M.G. Worster, Simulation of directional solidification, thermochemical convection, and chimney formation in a Hele-Shaw cell, *J. Comput. Phys.* 227 (2008) 9823–9840.
- [11] S.H. Whiteoak, H.E. Huppert, M.G. Worster, Conditions for defect-free solidification of aqueous ammonium chloride in a quasi two-dimensional directional solidification facility, *J. Cryst. Growth* 310 (2008) 3545–3551.
- [12] T.H. Solomon, R.R. Hartley, Measurements of the temperature field of mushy and liquid regions during solidification of aqueous ammonium chloride, *J. Fluid Mech.* 358 (1998) 87–106.
- [13] C.A. Middleton, C. Thomas, A. De Wit, J.-L. Tison, Visualizing brine channel development and convective processes during artificial sea-ice growth using Schlieren optical methods, *J. Glaciol.* 62 (2016) 1–17.
- [14] R.N. Hills, D.E. Loper, P.H. Roberts, A thermodynamically consistent model of a mushy zone, *Q. J. Mech. Appl. Math.* 36 (1983).
- [15] C. Beckermann, R. Viskanta, Natural convection solid/liquid phase change in porous media, *Int. J. Heat Mass Transf.* 31 (1988) 35–46.
- [16] A.C. Fowler, The formation of freckles in binary alloys, *IMA J. Appl. Math.* 35 (1985) 159–174.
- [17] M.G. Worster, Instabilities of the liquid and mushy regions during solidification of alloys, *J. Fluid Mech.* 237 (1992) 649–669.
- [18] G. Amberg, G.M. Homsy, Nonlinear-analysis of buoyant convection in binary solidification with application to channel formation, *J. Fluid Mech.* 252 (1993) 79–98.
- [19] D.M. Anderson, M.G. Worster, Weakly nonlinear analysis of convection in mushy layers during the solidification of binary alloys, *J. Fluid Mech.* 302 (1995) 307–331.
- [20] B.S. Okhuysen, D.N. Riahi, On weakly nonlinear convection in mushy layers during solidification of alloys, *J. Fluid Mech.* 596 (2008) 143–167.
- [21] T.P. Schulze, M.G. Worster, A numerical investigation of steady convection in mushy layers during the directional solidification of binary alloys, *J. Fluid Mech.* 356 (1998) 199–220.
- [22] C.A. Chung, M.G. Worster, Steady-state chimneys in a mushy layer, *J. Fluid Mech.* 455 (2002) 387–411.
- [23] A.J. Wells, J.S. Wettlaufer, S.A. Orszag, Nonlinear mushy-layer convection with chimneys: stability and optimal solute fluxes, *J. Fluid Mech.* 716 (2013) 203–227.
- [24] V. Alexiades, H. Solomon, *Mathematical Modelling of Melting and Freezing Processes*, Hemisphere Publishing Corporation, 1993.
- [25] C. Beckermann, H.J. Diepers, I. Steinbach, A. Karma, X. Tong, Modeling melt convection in phase-field simulations of solidification, *J. Comput. Phys.* 154 (1999) 468–496.
- [26] A.B. Oertling, R.G. Watts, Growth of and brine drainage from NaClH₂O freezing: a simulation of young sea ice, *J. Geophys. Res., Oceans* 109 (2004) 1–10.
- [27] S.D. Felicelli, J.C. Heinrich, D.R. Poirier, Finite element analysis of directional solidification of multicomponent alloys, *Int. J. Numer. Methods Fluids* 27 (1998) 207–227.
- [28] M. Le Bars, M.G. Worster, Solidification of a binary alloy: finite-element, single-domain simulation and new benchmark solutions, *J. Comput. Phys.* 216 (2006) 247–263.
- [29] R. Löhner, An adaptive finite element scheme for transient problems in CFD, *Comput. Methods Appl. Mech. Eng.* 61 (1987) 323–338.
- [30] M.J. Berger, J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512.
- [31] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [32] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, *Phys. Rev. Lett.* 80 (1998) 3308–3311.
- [33] N. Provatas, Q. Wang, M. Haataja, M. Grant, Seaweed to dendrite transition in directional solidification, *Phys. Rev. Lett.* 91 (2003) 1–4.
- [34] T.U. Kaempfer, M. Rappaz, Modelling of macrosegregation during solidification processes using an adaptive domain decomposition method, *Model. Simul. Mater. Sci. Eng.* 11 (2003) 575–597.

- [35] W. Liu, C. Xie, M. Bellet, H. Combeau, 2-dimensional FEM modeling of macrosegregation in the directional solidification with mesh adaptation, *Acta Metall. Sin. (English Letters)* 22 (2009) 233–240.
- [36] U.K. Sajja, S.D. Felicelli, Modeling freckle segregation with mesh adaptation, *Metall. Mater. Trans., B Process Metall. Mater. Proc. Sci.* 42 (2011) 1118–1129.
- [37] M. Adams, P. Colella, D.T. Graves, J.N. Johnson, N.D. Keen, T.J. Ligocki, D.F. Martin, P.W. McCorquodale, D. Modiano, P.O. Schwartz, T.D. Sternberg, B. Van Straalen, Chombo Software Package for AMR Applications, Design Document, 2016.
- [38] B. Van Straalen, J. Shalf, T. Ligocki, N. Keen, W.S. Yang, Scalability challenges for massively parallel AMR applications, in: *IPDPS 2009 - Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium*, 2009.
- [39] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations, *J. Comput. Phys.* 142 (1998) 1–46.
- [40] F. Pretorius, M.W. Choptuik, Adaptive mesh refinement for coupled elliptic-hyperbolic systems, *J. Comput. Phys.* 218 (2006) 246–274.
- [41] M.G. Worster, Natural convection in a mushy layer, *J. Fluid Mech.* 224 (1991) 335–359.
- [42] D. Notz, M.G. Worster, Desalination processes of sea ice revisited, *J. Geophys. Res.* 114 (2009) C05006.
- [43] P.K. Kundu, I.M. Cohen, D.R. Dowling, Chapter 4 - Conservation laws, in: P.K. Kundu, I.M. Cohen, D.R. Dowling (Eds.), *Fluid Mechanics (Fifth Edition)*, Academic Press, Boston, 2012, pp. 95–169.
- [44] D.F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions, *J. Comput. Phys.* 227 (2008) 1863–1886.
- [45] W.L. Briggs, S.F. McCormick, V.E. Henson, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, 2000.
- [46] P. Colella, Multidimensional upwind for hyperbolic conservation laws, *J. Comput. Phys.* 200 (1990) 171–200.
- [47] J. Saltzman, An unsplit 3D upwind method for hyperbolic conservation laws, *J. Comput. Phys.* 115 (1994) 153–168.
- [48] D.F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, *J. Comput. Phys.* 163 (2000) 271–312.
- [49] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, *J. Comput. Phys.* 147 (1998) 60–85.
- [50] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, *Math. Comput.* 29 (1975) 396–406.
- [51] D.F. Martin, An Adaptive Cell-Centered Projection Method for the Incompressible Euler Equations, Ph.D. thesis, University of California at Berkeley, 1998.
- [52] R.B. Yedder, F. Erchiqui, Convective flow and heat transfer in a tall porous cavity side-cooled with temperature profile, *Int. J. Heat Mass Transf.* 52 (2009) 5712–5718.
- [53] P. Nithiarasu, K.N. Seetharamu, T. Sundararajan, Natural convective heat transfer in a fluid saturated variable porosity medium, *Int. J. Heat Mass Transf.* 40 (1997) 3955–3967.
- [54] T.P. Schulze, M.G. Worster, Weak convection, liquid inclusions and the formation of chimneys in mushy layers, *J. Fluid Mech.* 388 (1999) 197–215.