Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **4**, No. 6, pp. 665-684

DOI: 10.4208/aamm.12-12S07
December 2012

On the Fully Implicit Solution of a Phase-Field Model for Binary Alloy Solidification in Three Dimensions

Christopher E. Goodyer[1], Peter K. Jimack[1,*], Andrew M. Mullis[2], Hongbiao Dong[3] and Yu Xie[3]

[1] *School of Computing, University of Leeds, Woodhouse Lane, Leeds, LS2 9JT, UK*
[2] *School of Process, Environmental and Materials Engineering, University of Leeds, Woodhouse Lane, Leeds, LS2 9JT, UK*
[3] *Department of Engineering, University of Leicester, University Road, Leicester, LE1 7RH, UK*

Received 10 March 2012; Accepted (in revised version) 28 August 2012

Available online 9 November 2012

Abstract. A fully implicit numerical method, based upon a combination of adaptively refined hierarchical meshes and geometric multigrid, is presented for the simulation of binary alloy solidification in three space dimensions. The computational techniques are presented for a particular mathematical model, based upon the phase-field approach, however their applicability is of greater generality than for the specific phase-field model used here. In particular, an implicit second order time discretization is combined with the use of second order spatial differences to yield a large nonlinear system of algebraic equations as each time step. It is demonstrated that these equations may be solved reliably and efficiently through the use of a nonlinear multigrid scheme for locally refined grids. In effect this paper presents an extension of earlier research in two space dimensions (J. Comput. Phys., 225 (2007), pp. 1271–1287) to fully three-dimensional problems. This extension is validated against earlier two-dimensional results and against some of the limited results available in three dimensions, obtained using an explicit scheme. The efficiency of the implicit approach and the multigrid solver are then demonstrated and some sample computational results for the simulation of the growth of dendrite structures are presented.

AMS subject classifications: 65M06, 65M22, 65H10, 65M55

Key words: Phase-field simulations, binary alloys, mesh adaptivity, implicit methods, nonlinear multigrid.

*Corresponding author.
URL: http://www.engineering.leeds.ac.uk/people/speme/staff/a.m.mullis
Email: c.e.goodyer@leeds.ac.uk (C. Goodyer), p.k.jimack@leeds.ac.uk (P. Jimack), a.m.mullis@leeds.ac.uk (A. Mullis), h.dong@le.ac.uk (H. Dong), yx38@leicester.ac.uk (Y. Xie)

1 Introduction

The modelling of solidification structures, in particular the growth of dendritic crystals, is a subject of intense and enduring interest within the scientific community, both because dendrites are a prime example of spontaneous pattern formation and due to their pervasive influence on the engineering properties of metals. In all but the most restrictive of cases, analytical solutions to the equations of motion for the solid-liquid interface, using techniques such as boundary integral methods (microscopic solvability theory [1, 4, 26, 40]), cannot be found and recourse must be made to numerical techniques. These include cellular automaton [13, 33], front-tracking [7], one-domain multiphase models [10,32,55] and level set techniques [9,18,27,35]. However, the technique which over the last few years has received the most attention is that of phase-field simulation [8, 28, 29, 39], in which a non-conserved order parameter ϕ is defined over the whole domain, which encodes the phase state of the material. By assuming the interface between the solid and liquid (or different solid phases in multi-phase modelling) to be diffuse, ϕ is rendered continuous, wherein standard techniques for partial differential equations (PDEs) may be used. This allows a regular Eulerian mesh to be used and avoids many of the topological complexities involved with front tracking methods.

However, the application of phase-field modelling leads to a number of issues. The resulting set of coupled PDEs is unsteady, highly non-linear and may moreover suffer from significant multi-scale problems. The latter arises because although the phase-field equations are formulated such that in the asymptotic limit of the diffuse interface width, δ, tending to zero, the corresponding sharp interface equations are recovered exactly, this is not sufficient to ensure that the solutions do not have a dependence upon δ. Such limitations may be overcome by formulating the model in the so-called "thin interface limit" [22–25], whereby asymptotic expansions of the solution on the inner and outer regions of the solid-liquid interface are matched to obtain an equation set in which the solution is independent of the width of the diffuse interface. However, in order to perform the asymptotic matching highly restrictive assumptions need to be made about the thermodynamics governing the phase transformation, which can restrict the applicability of such models. Consequently, in many cases phase-field models are constructed such that δ is much smaller than the other length scales characteristic of the problem. In particular, there is a growing body of opinion that "the sharp interface limit of a phase-field model is not the only meaningful physical limit" [12]. This view draws on the Gibbs [15] interpretation of understanding all interfacial boundaries as being of finite width. In the context of the crystallisation of metals this finite width interface can be understood physically as the number of atom widths over which the long range order characteristic of the crystalline solid is lost, and represents a tendency towards using interface widths in phase-field modelling which may be of the order of the capillary length, typically $2-5\times10^{-10}$m. This compares with typical microstructural length scales which are of the order $10^{-6}-10^{-5}$m.

Due to this multi-scale nature phase-field simulations tend to be highly computationally intensive, requiring very significant spatial resolution in the vicinity of the (moving) phase interface. Consequently, much of the literature on phase-field simulation has tended historically to focus on two-dimensional problems, partly because such problems are generally tractable using simple numerical techniques such as explicit time stepping and uniform spatial meshing. Even in two dimensions however the limitations of such naive numerical approaches are well known and the advantages of using more sophisticated techniques, such as mesh adaptivity [41, 56] and implicit time stepping [47], have been clearly demonstrated.

Recently there have also been an increasing number of three-dimensional phase-field simulations reported. These typically have to employ either extensive parallel implementation, e.g., [14, 43], or dynamically adaptive meshing in order to achieve sufficient resolution in the diffuse interface region without introducing excessive numbers of degrees of freedom into the model, e.g., [2, 30, 34, 53, 56]. However, there are still significant challenges in utilising such models. It is difficult to combine adaptive remeshing with parallel implementation, due to the consequent need to undertake dynamic load balancing between processors as the mesh adapts [51].

Any use of adaptive mesh refinement will have its efficiency and accuracy governed by the methods used to drive it. For regular grid techniques, such as are employed in this work, the monitor functions that select areas for refinement are important, and the choice of these will necessarily need to balance faster computation against accuracy of the solution achieved. More sophisticated methods, such as mesh movement [3, 50], can be used with fully unstructured meshes, as described in [56] for example, where a significant reduction in the number of mesh points needed for both 2-d and 3-d thermal-only problems was achieved.

If explicit temporal schemes are used the time step is restricted by the size of the smallest elements in the mesh. While adaptive remeshing is thus an attractive tool for reducing the total number of elements within a mesh, this does not necessarily mean that it can always be used effectively to give better spatial resolution, since finer mesh elements may give rise to unfeasibly small time steps. This limitation can be overcome by using implicit temporal discretization [47, 54], although at the expense of greater computation time per time step. However, there are currently only a limited number of examples [11, 17] of implicit time-stepping being applied to 3-dimensional phase-field problems. In [11] a preconditioned Newton-Krylov approach is used to solve the nonlinear systems that arise at each time step (using multigrid as the preconditioner) whereas in [17] a nonlinear multigrid scheme [5, 6, 52] is applied directly at each time step.

Another important feature of the efficient solution of these problems is the use of adaptive timestepping. This enables smaller timesteps to be used during periods of great change in the solution, such as the initial transients, but allows this to grow as the simulation progresses. As with the spatial adaptivity there are many options for controlling this. A heuristic monitor function, that looks at the rate of convergence of the solution over each timestep, is applied here although more complicated measures can be calculated, e.g., [16, 44, 47], and employed to control the local error per step as well as providing input into the choice of the appropriate next timestep.

In this paper we extend the scheme described in [17] so that it may be applied to the simulation of an isothermal binary alloy using a phase-field model that features a nonlinear evolution equation for the concentration of the solutal component of the alloy in addition to the nonlinear phase-field equation. Such isothermal alloy solidification models have potentially wide applicability in the simulation of microstructure formation during conventional casting processes as the thermal diffusivity is typically 10^4 times larger then the solutal diffusivity, meaning the system is, to a good approximation, isothermal. This model, based upon [22,24], is described fully in the following section. Unlike the phase-field model for a pure melt, used in [17] for example, there have been only a very small number of three-dimensional codes developed for this alloy solidification model. Hence for validation of our results we have contrasted with corresponding two-dimensional results from [47] and with three-dimensional results obtained using the explicit code described in [21, 42]. Before this however, in section 3, we provide a complete description of the computational techniques and tools that have been used and developed in this work. The remainder of Section 4 then presents some further numerical results which demonstrate the efficiency of our approach and shows some selected outputs from a number of sample simulations. The paper concludes with a discussion of further developments that are required.

2 Mathematical model

The phase-field model used in this paper is a non-dimensional isothermal system featuring two dependent variables: a phase field, ϕ, and a dimensionless solute concentration, U [22, 24]. The phase variable takes values of $+1$ and -1 in the solid and liquid phases respectively, and in the narrow interface region it varies smoothly between these bulk values. The driving force for the solidification comes from applying a fixed undercooling throughout the melt (i.e., the bulk liquid is below its melting temperature). Initially, a small spherical seed of solid phase is created around the origin in order to initiate the solidification process, which is given preferred growth directions via the definition of an anisotropy function $A(\vartheta,\psi)$ within the phase-field model (where ϑ and ψ are the spherical angles of each point on the interface: see (2.8) below). This anisotropy function appears in the phase evolution equation which, for this model, takes the form:

$$\begin{aligned} A^2(\vartheta,\psi)\frac{\partial\phi}{\partial t} =&\underline{\nabla}\cdot\left(A^2(\vartheta,\psi)\nabla\phi\right)+\phi(1-\phi^2)+\lambda(1-\phi^2)^2(-\Omega+Mc_\infty U)\\ &+\frac{\partial}{\partial x}\Big[A(\vartheta,\psi)\Big(-\frac{\partial A(\vartheta,\psi)}{\partial\vartheta}\frac{\phi_y|\nabla\phi|^2}{\phi_x^2+\phi_y^2}+\frac{\partial A(\vartheta,\psi)}{\partial\psi}\frac{\phi_z\phi_x}{\sqrt{\phi_x^2+\phi_y^2}}\Big)\Big]\\ &+\frac{\partial}{\partial y}\Big[A(\vartheta,\psi)\Big(\frac{\partial A(\vartheta,\psi)}{\partial\vartheta}\frac{\phi_x|\nabla\phi|^2}{\phi_x^2+\phi_y^2}+\frac{\partial A(\vartheta,\psi)}{\partial\psi}\frac{\phi_z\phi_y}{\sqrt{\phi_x^2+\phi_y^2}}\Big)\Big]\\ &-\frac{\partial}{\partial z}\Big[A(\vartheta,\psi)\Big(\frac{\partial A(\vartheta,\psi)}{\partial\psi}\sqrt{\phi_x^2+\phi_y^2}\Big)\Big]. \end{aligned} \tag{2.1}$$

Throughout all of the simulations in this paper we assume that the function $A(\vartheta,\psi)$, is expressed as

$$A(\vartheta,\psi) = A_0\big\{1+\varepsilon[\cos^4\psi + \sin^4\psi(1-2\sin^2\vartheta\cos^2\vartheta)]\big\}, \tag{2.2}$$

which corresponds to a preference for growth along the Cartesian coordinate axis (i.e., a four-fold anisotropy). The small parameter ε governs the strength of the anisotropy, whilst the parameters Mc_∞ (where M is the scaled magnitude of the liquidus slope and c_∞ is the solute concentration far from the interface) and λ (a coupling parameter) are assumed to be known (see (2.6) below). The additional parameter Ω is a scaled supersaturation given by

$$\Omega = \frac{c_\ell^0 - c_\infty}{(1-k)c_\ell^0}, \tag{2.3}$$

where c_ℓ^0 is the solute concentration in the liquid at the solid-liquid interface and k is the equilibrium partition coefficient.

The evolution equation for the dimensionless concentration field is given by

$$\begin{aligned}\Big(\frac{1+k}{2} - \frac{1-k}{2}\phi\Big)\frac{\partial U}{\partial t} =& \underline{\nabla}\cdot\Big\{D\frac{1-\phi}{2}\nabla U + \frac{1}{2\sqrt{2}}[1+(1-k)U]\frac{\partial\phi}{\partial t}\frac{\nabla\phi}{|\nabla\phi|}\Big\} \\ &+\frac{1}{2}[1+(1-k)U]\frac{\partial\phi}{\partial t},\end{aligned} \tag{2.4}$$

where D is the dimensionless diffusivity. The non-dimensional concentration field U is related to the concentration c via

$$U = \frac{\Big(\frac{2c/c_\infty}{1+k-(1-k)\phi}\Big)}{1-k}, \tag{2.5}$$

where k and c_∞ are defined above.

In order to fully specify each computational run it is necessary to prescribe numerical values to all of the parameters that appear in the governing equations and to select an appropriate domain and boundary conditions. Not all of the numerical parameters are free however since we have the following two constraints:

$$\lambda = D/a_2 \quad \text{(see [22])} \quad \text{and} \quad Mc_\infty = 1-(1-k)\Omega \quad \text{(see [46])} \tag{2.6}$$

where $a_2 = 0.6267$ is a constant arising from the thin-interface analysis [22]. Unless stated to the contrary, for the results shown in this paper we have selected default values of $\Omega = 0.55$, $D = 1.5$, $\varepsilon = 0.02$ and $k = 0.15$ for the remaining parameters (though results are observed to be independent of k for any choice within the open interval $(0,1)$). Furthermore, for our domain we may select boundaries which are sufficiently far from the origin so that they have no influence on the evolution of the solid-liquid interface. At these far-field boundaries we apply the conditions:

$$\frac{\partial\phi}{\partial\underline{\hat{n}}} = 0 \quad \text{and} \quad \frac{\partial U}{\partial\underline{\hat{n}}} = 0 \tag{2.7}$$

for normal directions $\underline{\hat{n}}$ to the boundaries. Unless otherwise stated we find that a default domain size of $(-400,400)\times(-400,400)\times(-400,400)$ is sufficient for the runs undertaken here – though clearly this domain size is insufficient if either of the solution fields become non-flat at the far-field boundary (in which case a larger domain must be selected). Note however that, due to our choice of anisotropy function (2.2), the particular solutions that we are simulating here have a four-fold symmetry, with preferred growth along each Cartesian axis. Hence it is possible to exploit this symmetry in our solver by only using one eighth of the spatial domain (for example $(0,400)\times(0,400)\times(0,400)$) and use symmetry boundary conditions (still (2.7)) on the x-y, x-z and y-z planes. Indeed, all of the results presented in this paper have been obtained using this symmetry – however we have undertaken many simulations, not reproduced here, to ensure that results obtained through this simplification are identical to those obtained when running on a full domain.

Finally, note that (2.4) is easily expressed in terms of Cartesian coordinates by evaluating the divergence and the gradient operators in this system. For our finite difference discretization it is also convenient to express (2.1) in Cartesian coordinates as well. This may be achieved by noting that

$$\tan\vartheta=\frac{\phi_y}{\phi_x},\quad \cos\psi=\frac{\phi_z}{|\nabla\phi|}\quad\text{and}\quad \tan\psi=\frac{\sqrt{\phi_x^2+\phi_y^2}}{\phi_z}, \tag{2.8}$$

so that the anisotropy function A may be rewritten as

$$A(x,y,z)=A_0\Big[1+\varepsilon\Big(\frac{\phi_x^4+\phi_y^4+\phi_z^4}{|\nabla\phi|^4}\Big)\Big]. \tag{2.9}$$

Following [20], for example, the phase evolution equation may then be written as:

$$\begin{aligned}
A^2(x,y,z)\frac{\partial\phi}{\partial t} =&A^2(x,y,z)\nabla^2\phi+\phi(1-\phi^2)+\lambda(1-\phi^2)^2(-\Omega+Mc_\infty U)\\
&+2A(x,y,z)\Big(\phi_x\frac{\partial A}{\partial x}+\phi_y\frac{\partial A}{\partial y}+\phi_z\frac{\partial A}{\partial z}\Big)\\
&+\frac{\partial A}{\partial x}(\phi_x^2+\phi_y^2+\phi_z^2)\frac{\partial A}{\partial\phi_x}+2A(x,y,z)(\phi_x\phi_{xx}+\phi_y\phi_{xy}+\phi_z\phi_{xz})\frac{\partial A}{\partial\phi_x}\\
&+\frac{4A_0A(x,y,z)\varepsilon}{(\phi_x^2+\phi_y^2+\phi_z^2)^2}\Big[3\phi_{xx}\phi_x\phi_x(\phi_x^2+\phi_y^2+\phi_z^2)-\phi_{xx}(\phi_x^4+\phi_y^4+\phi_z^4)\\
&+(\phi_x\phi_{xx}+\phi_y\phi_{xy}+\phi_z\phi_{xz})\Big(-4\phi_x^3+6\phi_x\frac{\phi_x^4+\phi_y^4+\phi_z^4}{\phi_x^2+\phi_y^2+\phi_z^2}\Big)\\
&-4\phi_x(\phi_x^3\phi_{xx}+\phi_y^3\phi_{xy}+\phi_z^3\phi_{xz})\Big]\\
&+\frac{\partial A}{\partial y}(\phi_x^2+\phi_y^2+\phi_z^2)\frac{\partial A}{\partial\phi_y}+2A(x,y,z)(\phi_x\phi_{xy}+\phi_y\phi_{yy}+\phi_z\phi_{yz})\frac{\partial A}{\partial\phi_y}\\
&+\frac{4A_0A(x,y,z)\varepsilon}{(\phi_x^2+\phi_y^2+\phi_z^2)^2}\Big[3\phi_{yy}\phi_y\phi_y(\phi_x^2+\phi_y^2+\phi_z^2)-\phi_{yy}(\phi_x^4+\phi_y^4+\phi_z^4)
\end{aligned}$$

$$
\begin{aligned}
&+(\phi_x\phi_{xy}+\phi_y\phi_{yy}+\phi_z\phi_{yz})\Big(-4\phi_y^3+6\phi_y\frac{\phi_x^4+\phi_y^4+\phi_z^4}{\phi_x^2+\phi_y^2+\phi_z^2}\Big)\\
&-4\phi_y(\phi_x^3\phi_{xy}+\phi_y^3\phi_{yy}+\phi_z^3\phi_{yz})\Big]\\
&+\frac{\partial A}{\partial z}(\phi_x^2+\phi_y^2+\phi_z^2)\frac{\partial A}{\partial \phi_z}+2A(x,y,z)(\phi_x\phi_{xz}+\phi_y\phi_{yz}+\phi_z\phi_{zz})\frac{\partial A}{\partial \phi_z}\\
&+\frac{4A_0A(x,y,z)\varepsilon}{(\phi_x^2+\phi_y^2+\phi_z^2)^2}\Big[3\phi_{zz}\phi_z\phi_z(\phi_x^2+\phi_y^2+\phi_z^2)-\phi_{zz}(\phi_x^4+\phi_y^4+\phi_z^4)\\
&+(\phi_x\phi_{xz}+\phi_y\phi_{yz}+\phi_z\phi_{zz})\Big(-4\phi_z^3+6\phi_z\frac{\phi_x^4+\phi_y^4+\phi_z^4}{\phi_x^2+\phi_y^2+\phi_z^2}\Big)\\
&-4\phi_z(\phi_x^3\phi_{xz}+\phi_y^3\phi_{yz}+\phi_z^3\phi_{zz})\Big].
\end{aligned}
\tag{2.10}
$$

Not only does this formulation permit the (relatively) straightforward use of Cartesian finite differences but it is also found to be less susceptible to numerical anisotropy than direct discretizations of (2.1).

3 Numerical methods

Having defined the mathematical model that we consider in this paper we now describe the discretization and solution methods that have been employed. It is important to emphasise that none of the scientific computing techniques that we have developed here are specific to this particular mathematical model, so the proposed approach is equally applicable to other phase-field models. There are a number of components to this approach, however the overall solution technique may be summarised as follows. Select an appropriate spatial discretization (in this case we use second order finite differences) in order to semi-discretize the governing PDEs into a large system of initial value ordinary differential equations (ODEs); select an unconditionally stable implicit time-stepping scheme of equal order to the spatial discretization (here we choose BDF2, which can be shown to be A-stable [19]) which reduces the problem at each time step to that of solving a large nonlinear algebraic system; employ a nonlinear multigrid solver (we use Brandt's full approximation scheme (FAS), [5,52]) in order to obtain the fast solution of each of these algebraic systems of equations with an initial guess based upon the solution from the previous time level. In order to apply the FAS solver it is necessary to have a hierarchy of finite difference meshes so as to be able to resolve the solution at different length scales: we achieve this using nested hexahedral meshes which allow local mesh refinement and de-refinement, [31,37,38]. This local adaptivity provides the necessary spatial resolution throughout the computational domain without requiring unnecessary degrees of freedom.

A number of different second order finite difference discretizations of Eqs. (2.10) and (2.4) are possible and clearly any consistent schemes should converge to the same solution as the mesh size is reduced. This is indeed what we find when comparing the use of the minimal 7-point stencil with a compact 27-point stencil for example.

An important constraint that we have put on our choice of finite difference stencil is to avoid the use of any points around cell (i,j,k) that are not of the form $(i\pm 1, j\pm 1, k\pm 1)$. This ensures that our parallel implementation only needs a single layer of guard cells between blocks of the mesh that are stored on different processors, which reduces the memory and communication overhead significantly. In this work we have also chosen to implement the finite difference discretization in a cell-centred manner-with unknown values for both ϕ and U stored at the centre of each hexahedral cell (furthermore, we use isotropic meshes so each cell is in fact a cube, rather than a more general hexahedron). The reason for this choice stems from our adoption of an open source library, PARAMESH [31,37,38], in order to control the three-dimensional mesh refinement and de-refinement. This library provides functions to generate meshes in an oct-tree structure of mesh blocks. Starting with a base block (of $8\times 8\times 8$ cubic cells for example) it is possible to refine this into up to 8 child blocks (with each block always being of the same dimension as the base block) and then to refine any of these child blocks successively. Functions are also provided to undo regions of this local refinement (i.e., de-refinement) and to interpolate or restrict solution fields between meshes. A further capability of PARAMESH is that it is able to undertake this meshing in parallel in a manner that is hidden from the user – each block is simply treated as independent of its neighbours and PARAMESH takes care of which process owns each block, using its own rudimentary dynamic load balancing scheme. A price that has to be paid for this simplicity is that every block is required to store guard cells in each dimension regardless of whether its neighbouring blocks are actually owned by a different process: PARAMESH's guard cell update routines then take care of all of the transfer of data between neighbouring blocks, regardless of their location in memory.

The local refinement and de-refinement capability provided by PARAMESH is essential for this work since our phase-field models require very fine meshes around the solid-liquid interface in order to ensure that the interface is resolved with sufficient accuracy. The non-dimensionalization used to derive the systems introduced in Section 2 is such that the interface width is $\mathcal{O}(1)$ and so our mesh spacing cannot be greater than $\Delta x = 1$ around the interface. Hence the finest grid resolution needs to be at least this size (for a domain of dimension $(0,400)\times(0,400)\times(0,400)$ at least nine levels of refinement are required – to give $\Delta x = 0.78125$, though a tenth level is necessary if we wish to ensure that the interface is even moderately well resolved in its normal direction). Without the use of local mesh refinement and de-refinement their would need to be an excessive number of cells, creating a computational load that would be unmanageable without the very largest supercomputing resources.

Unfortunately the PARAMESH library does not provide all of the functionality necessary to perform multigrid solution, even though the mesh hierarchy that is held is ideal for multigrid. As outlined above, the need for multigrid arises from our use of an unconditionally stable time-stepping scheme which results in a large system of nonlinear algebraic equations at each time step. It has already been demonstrated in [47] that the use of implicit timestepping for this particular phase-field model is essential for fine spatial resolution to be achieved, even in two space dimensions. This is because the stability constraints imposed on the time-step size by the small spatial mesh size at the phase boundary mean that explicit time-stepping is prohibitively slow. This is equally true in three space dimensions. Hence we have employed the same second order backward difference formula, BDF2, for time integration as in [47]. Adaptive time stepping is also used, based upon the rate of convergence of the multigrid cycle at the previous step: in particular if the multigrid defect calculated after $20V$-cycles exceeds 10^{-7} the step size is reduced to 75% and the step is retaken; if it converges taking more than $10V$-cycles then the stepsize for the next step is reduced to 90% of the current size; and, if the timestep converges within $5V$-cycles the next timestep is set to be $10/9$ of the previous one. We have also a maximum time step, Δt, beyond which we do not permit the size to grow (with ten levels of refinement the default setting for this is $\Delta t = 0.4$). To assess convergence at each time step the Shampine convergence test [49] has been used.

The extension of the PARAMESH capability to include nonlinear multigrid is explained in [17]. That work describes the implementation of the multigrid itself and gives some preliminary results on the parallel scalability for a simpler phase field model, based upon the thermally-driven solidification of an undercooled pure melt. The essential ingredients are the extension of the restriction and prolongation operators for the FAS scheme and for the use of the multi-step BDF formula (requiring data from previous time steps to be used at each multigrid level). In this paper our focus is on the extension of this solution approach to a three-dimensional alloy model for the first time and so we have not considered issues of parallel scalability at all. Instead we use the application on parallel architectures as a capability mechanism: to allow sufficiently large runs to be undertaken, for which there would not be sufficient memory on a desktop workstation. Future research will focus on undertaking the research necessary to improve the scalability to large numbers of cores, and as part of this we have already implemented an improved load balancing approach for each multigrid mesh than is available within the default PARAMESH framework. Consequently the 3-d results given in the next section have been produced with typically 64 cores and 64 Gbytes of RAM. Even with this combination of mesh adaptivity, implicit time-stepping, multigrid and moderate parallelism the computational work is still considerable and so each of the reported three-dimensional runs still takes of the order of 24 hours to complete (until the dendrite tip is observed to have an almost constant velocity and tip radius). This is not unreasonable however since single core results for similar problems in two space dimensions are reported as being of the order of 11 hours in [47], for example. For more challenging two dimensional models of alloy solidification, in which a thermal field is present in addition to the solute field, reported run times are even greater [36,48].

4 Results

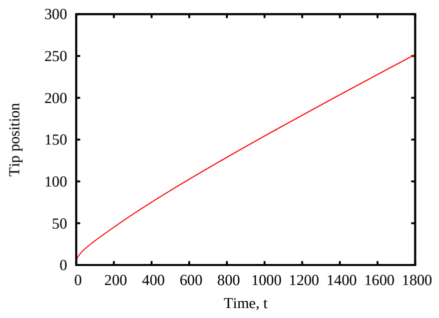
In this section we consider both the validation and verification of the solver developed. We have firstly validated the our new solver when run in 2-d mode against a test case from Rosam et al. [47]. Since we know of no other implicit 3-d results in the literature we have then run a set of test cases and compared them against equivalent results obtained using an explicit time-stepping code, which is described in [21, 42]. These initial quantitative comparisons, which demonstrate the correctness of our code, are then followed by a more detailed investigation of the numerical properties of our new solver to assess its robustness and efficiency.

In order to compare our results against those of other codes two quantities associated with the numerical solutions are considered, namely the tip radius and tip velocity of the computed dendrites. These are chosen since, once the solution progresses long past the initial transient, they should reach a steady state representing the steady growth of the dendrite. For our cell-centred solver we store the computed solution values for ϕ and U at the centre of each hexahedral element. This means that we have no solution values for ϕ on the x-, y- or z-axis. Hence the position of the dendrite tip, x_{tip}, which is the intersection of the centre of the phase boundary ($\phi = 0$) with any axis (for the symmetry used here), needs to be reconstructed through interpolation. We have achieved this, using the x-axis for example, by interpolating values in the $z = 0$ plane perpendicular to the x-axis in the vicinity of the tip. This provides interpolated values of ϕ on the axis. These can then be interpolated in the x direction to find where $\phi = 0$: the tip location. All interpolants used are cubic splines. In order to estimate the tip radius further interpolation is required to estimate where $\phi = 0$ on the $z = 0$ plane to either side of the x-axis (i.e., where $y = \Delta x$ and $y = -\Delta x$). These points are then used to form a final cubic spline, denoted here as χ, passing through $(x_{tip}, 0, 0)$. This can then be used to calculate the tip radius, R, since the radius of curvature of any function can be given by

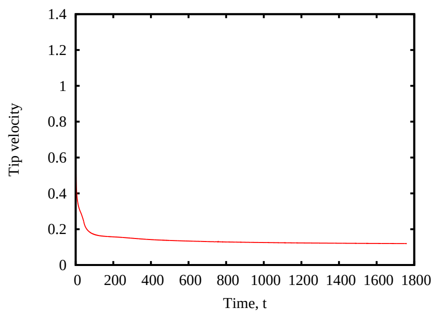
$$R = \frac{(1 + (\chi')^2)^{1.5}}{|\chi''|}, \tag{4.1}$$

noting that at the tip $\chi' = 0$. Such a calculation is inherently noisy since we are essentially recovering second derivatives from cell-centred data. This noise shows itself in the form of temporal oscillations in the calculated value, although their magnitude is diminished as the mesh gets finer. The tip radius computed here is thus averaged over 200 timesteps, reported at the middle of this interval. In order to calculate the tip velocity, V_{tip}, we have simply used the distance that the tip has moved in this same time interval.

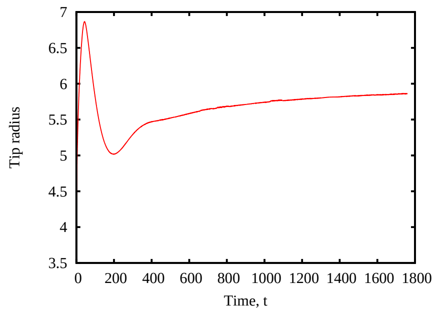
In [47], a 2-d test case is considered that is defined by the parameters $D = 2$, $k = 0.15$, $\varepsilon = 0.02$ and $\Omega = 0.55$ with an initial circular seed taken as $R_0 = 14$. Our software, developed using PARAMESH, allows us to solve a genuinely 2-d problem (rather than just considering a very thin test geometry) with minimal code modifications: the only differences inside the code are that the discretizations have all z derivatives or components removed. In [47] it was shown how a finest mesh spacing of $\Delta x = 0.39$ is appropriate for obtaining converged results when using BDF2 timestepping. We have therefore used that in this paper, unless otherwise stated.



(a) Tip location



(b) Tip velocity



(c) Tip radius

Figure 1: 2-d simulation results showing the evolution of tip location, velocity and radius against non-dimensional time.

The results achieved from our solver are summarised in Fig. 1, which is presented so as to permit a direct comparison against Fig. 11 and 9c of [47]. It should be noted that in Fig. 1 we have plotted the non-dimensional values of the R and V_{tip} since these are the quantities plotted in [47]. The agreement between these results and [47] is excellent with a steady tip velocity of 0.11 and a steady tip radius approaching 6.0 in each case. Similarly the solution shown here in Fig. 2, which has grey-scale values for phase, ϕ, [white:black] over $[-1:1]$, and for solute, U, [white:black] over [0:1], may be compared to Fig. 6 of [47]. For a final comparison it can be noted that at time $t = 1800$ our new software calculates the value of $V_{tip}d_0/D$ to be 0.0166 which is in excellent agreement with Fig. 12 of [47].

Having verified that the 2-d restriction of our code matches published 2-d results we now seek to verify that we are able to match 3-d results produced by an explicit code [21,42]. This explicit code was kindly provided by its authors and so we are able to verify that the results of these two codes do match. The explicit solver [21,42] is rather expensive in 3-d due to the small time steps necessitated by the finest mesh elements. Hence the finest mesh resolution possible in a reasonable execution time (days rather than weeks or months) is $\Delta x = 0.8$ using a maximum stable time step of $\Delta t = 0.08$. In addition the mesh domain size is optimized by using a contracted domain to only grow a single dendrite, ignoring any effects from the other tips interacting with the sides of the domain. As such, the domain chosen for this comparison was $(0,204.8) \times (0,102.4) \times (0,102.4)$. A fully developed dendrite for this case is shown in Fig. 3 with the comparison of phase and solute profiles given in Fig. 4.

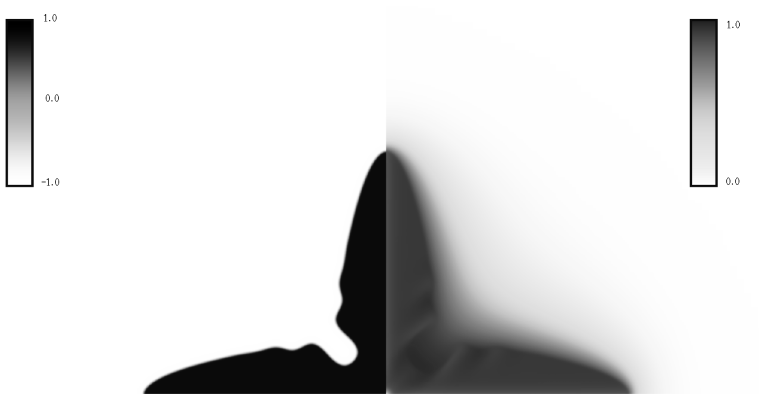


Figure 2: The 2-d interface shape at time $t = 1800$, showing the phase solution on the left, and the solute profile on the right.

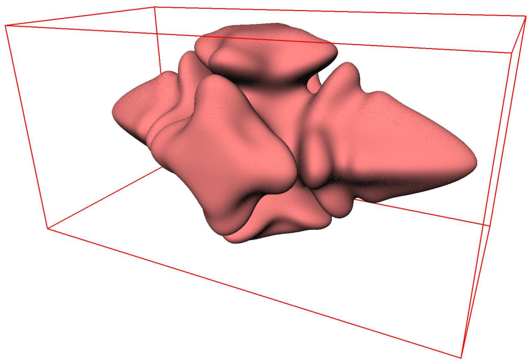


Figure 3: The 3-d interface shape (the isosurface of $\phi = 0$) at scaled time $t = 5000$, showing the phase boundary that has a single dendrite grown along the x-axis, in a contracted domain.

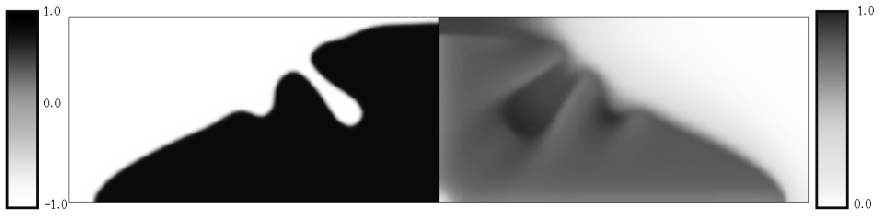


Figure 4: The interface shape at $z = 0$ for a 3-d simulation at scaled time $t = 5000$, showing the phase solution on the left, and the solute profile on the right.

In Table 1 we compare the results from the two codes for four test cases of varying diffusivity D. The scaled versions of tip velocity (Vd_0/D) and radius (R/D) are given at a scaled time $(tD/{a_1}^3)$ of 2000. In order to test our own 3-d software we have additionally solved this case on a domain of $(0,400) \times (0,400) \times (0,400)$ and using a grid level finer than for the comparison with the explicit code. It can be seen that the calculated tip velocities using the two approaches match very well and that the tip radii have only very small variations from each other. Furthermore, these radius variations reduce as D increases and, as discussed below, it is only for these larger

Table 1: Comparison of the tip radius and velocity at scaled time $t = 2000$ using our 3-d implicit software compared against an explicit 3-d code.

Method	Δx	Domain	Velocity			
			$D = 0.8$	$D = 1.0$	$D = 1.5$	$D = 2.0$
Explicit	0.8	204.8×102.4×102.4	0.057	0.055	0.051	0.047
Implicit	0.8	204.8×102.4×102.4	0.057	0.056	0.051	0.046
Implicit	0.39	400×400×400	0.059	0.058	0.053	0.048
			Radius			
			$D = 0.8$	$D = 1.0$	$D = 1.5$	$D = 2.0$
Explicit	0.8	204.8×102.4×102.4	16.61	16.95	18.77	21.73
Implicit	0.8	204.8×102.4×102.4	17.70	17.79	19.52	21.63
Implicit	0.39	400×400×400	17.30	17.64	19.76	23.05

values of D that the solutions have reached their steady state values by the scaled time of 2000. We also note that the use of the finer mesh and the larger computational domain provides some additional accuracy beyond that obtained using the explicit code or our implicit code with $\Delta x = 0.8$.

The transient behaviour of these quantities, computed using our implicit solver on the more accurate finer mesh, is shown in Fig. 5 which shows the tip velocity and radius evolution for these four different D values. This study has been performed on our larger domain, growing all of the dendrites in the eighth domain without any edge effects. As remarked above, it should be noted that at scaled time 2000, where the comparison is made in Table 1, the cases with $D = 0.8$ and $D = 1.0$ are not yet at their steady-state values. It is useful to note that taking smaller timesteps $(\Delta t < \Delta x)$ through the initial transients, for example up to scaled time 500 in this picture, can provide extra accuracy in this period, but the steady-state tip velocities are unaffected. Consequently we have allowed our solver to take larger (less accurate) timesteps in the early stages of the simulations that follow. This leads to no loss of accuracy in the reported steady state values.

The mesh convergence studies shown for 2-d cases in [47] have been reproduced here for our new 3-d solver. In Fig. 6 the convergence of the results for Δx taking the values 0.78, 0.39 and 0.19 is shown. Whilst it is clear that in the early stages the larger timesteps make a noticeable difference, the steady-state tip velocity and radius are both converged sufficiently well by Δx=0.39. This result, and the many similar cases not reproduced here, give us confidence in the results obtained with this mesh spacing in the interface region.

Convergence of the multigrid solver is shown in Fig. 7 at scaled time 1200 where the root mean square residual for the phase equation is plotted against multigrid V-cycle number. Note that since this is a transient simulation and different time step sizes and initial guesses are being used it is difficult to make precise comparisons however the rate of convergence appears to be almost independent of the finest mesh level. This suggests that optimal multigrid convergence is being achieved.

Having satisfied ourselves that our choice of Δx at the finest refinement level is sufficiently small to yield accurate steady-state results for the parameter ranges of

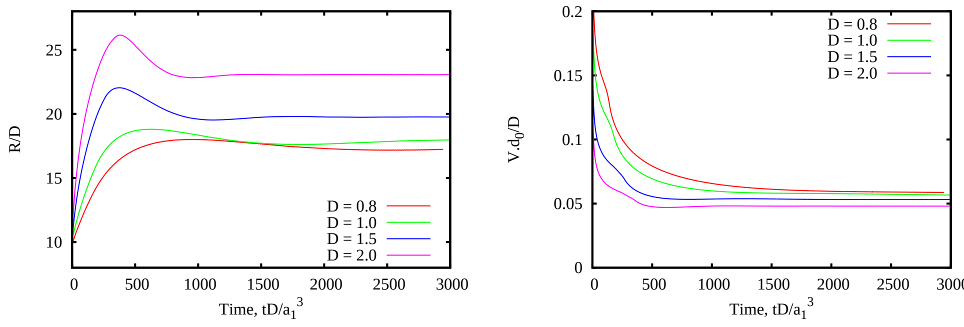


Figure 5: Results for a 3-d simulation with $\Delta x = 0.39$, $R_0 = 10$, $\Omega = 0.55$, varying the diffusivity D.

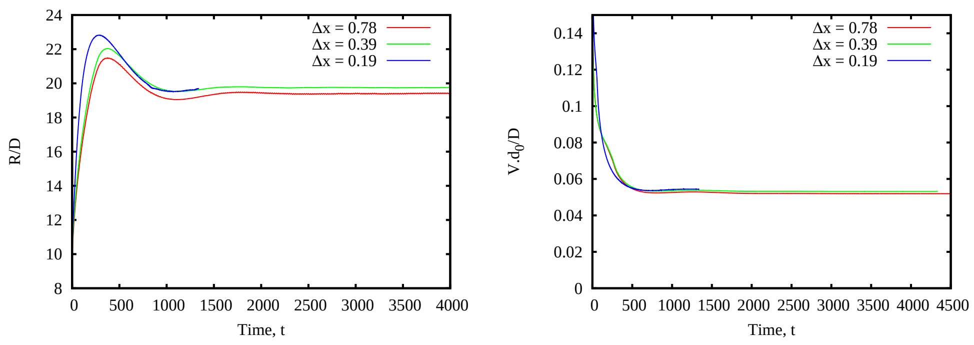


Figure 6: Results for a 3-d simulation with $D = 1.5$, $R_0 = 10$, $\Omega = 0.55$, varying the finest mesh spacing used.

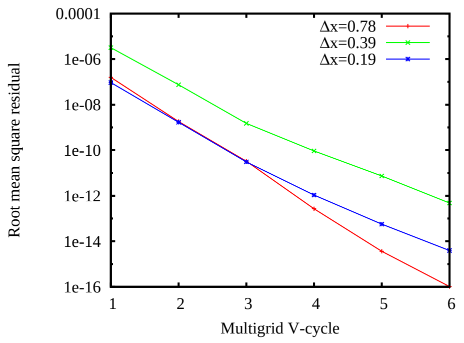


Figure 7: Multigrid convergence for 3-d simulations at scaled time 1200 with varying mesh resolutions.

interest, it is now possible to apply our code to investigate the effects of varying some of these parameters. For example, in Fig. 8 we present the effect of changing the initial data by altering the radius, R_0, of the solid seed. It is clear from these runs that, for all three sizes considered here, the dendrite geometry and speed converge to the same steady-state values respectively after (the expected) different initial transients.

In a similar manner, the final set of test cases that we present here illustrates how the variation of the undercooling affects the dendrites that develop. This is achieved through varying the parameter Ω, defined in (2.3). In Fig. 9 three different parameters are tested, namely $\Omega = 0.45$, 0.55 and 0.65, and snapshots of the resulting 3-d profiles

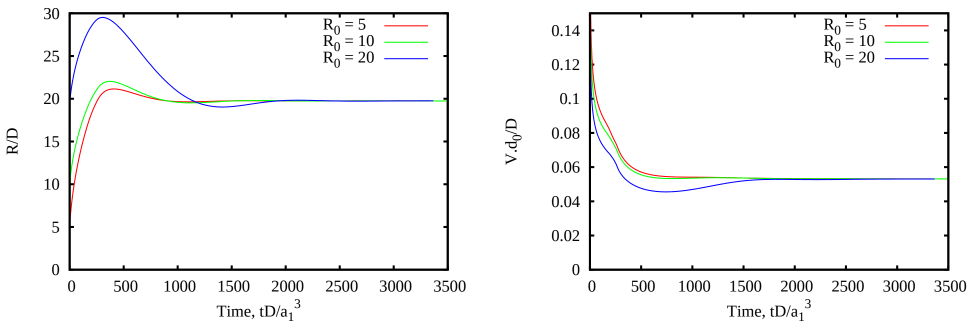


Figure 8: Results for a 3-d simulation with $D = 1.5$, $\Delta x = 0.39$, $\Omega = 0.55$, varying the size of the initial seed, R_0.

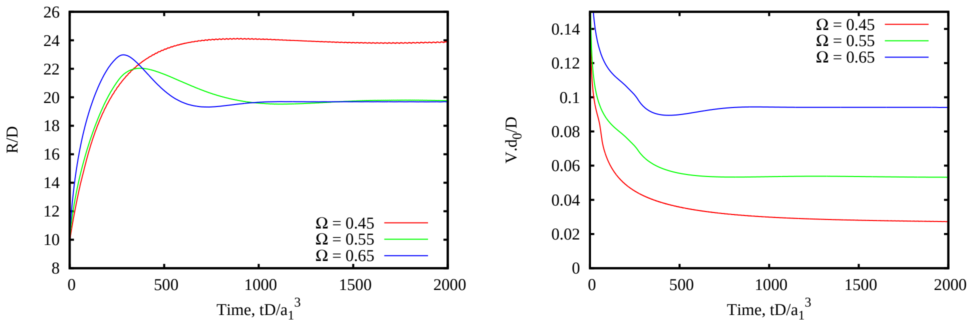


Figure 9: Results for a 3-d simulation with $D = 1.5$, $\Delta x = 0.39$, $R_0 = 10$, varying Ω.

are shown in Fig. 10. Here we have also plotted the evolution of tip location against time, which clearly shows a slower speed of evolution with smaller undercooling. By plotting the interface for each undercooling at time 2000 (images A, B and C) it can be seen that the dendritic shape formation is correspondingly much slower with smaller Ω. Also by comparing images C and D it can be seen that with the higher Ω, even when the dendrites are nominally the same size (or at least at the same location), there are many differences in the phase interfaces that have been simulated.

5 Conclusions

In this paper we have presented the first use of an implicit solver for phase-field modelling of binary alloy solidification in three dimensions. In order to validate the development of our software we have successfully compared the results to those achieved by a 2-d implicit code given in [47] and the 3-d results obtained using an explicit solver [21,42].

We have demonstrated that using a BDF2 scheme enables us to take large timesteps with small mesh spacing around the phase interface, and still obtain accurate results for the key quantities being measured as the dendrites formed reach steady-state growth. Combining the implicit timestepping with the adaptive mesh refinement strategy used within the nonlinear multigrid framework has enabled us to solve at

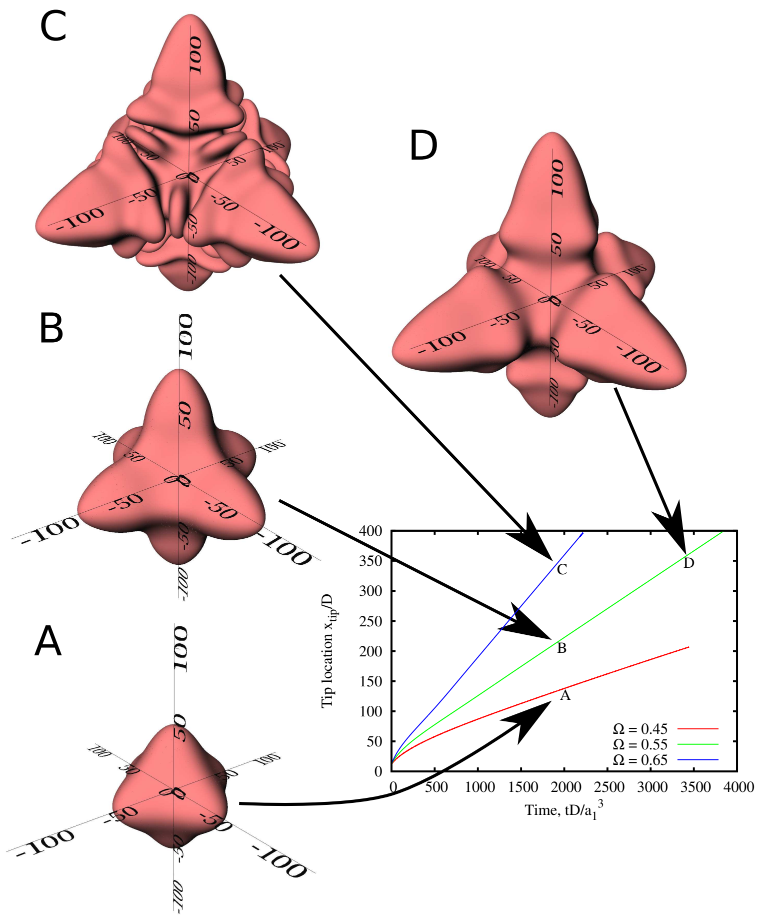


Figure 10: The interface shape for 3-d simulations with $D = 1.5$, $\Delta x = 0.39$, $R_0 = 10$, varying Ω. Four profiles are shown: A, B and C at scaled time $t = 2000$, at $\Omega = 0.45$, 0.55 and 0.65 respectively, and also D for $\Omega = 0.55$ at scaled time 3500 where the tip location is the same as in case C.

a finer mesh on a much larger domain than was possible with the explicit software.

The benefits of using the implicit solver will be even more evident when we extend this work to consider non-isothermal alloy solidification in 3-d. In such cases there is an additional governing equation for the evolution of the thermal field and an additional parameter, known as the Lewis number, which describes the ratio of the thermal to the chemical diffusivity. In order to compute cases that simulate typical metallic alloys this number needs to reach values in the range 1000 to 10000. Unfortunately, for values of the Lewis number in this range, the governing equations are extremely stiff and so even greater restrictions are placed on the maximum time step size for conditionally stable schemes than for the isothermal case that has been considered in this paper. Our on-going research is therefore focused on the extension of our fully implicit approach, which is ideally suited to very stiff problems, to tackle these non-isothermal cases. The goal will be to demonstrate the same computational capabilities in 3-d as we have previously been able to deliver in 2-d [48].

This paper does not consider the issues of the parallel scalability or efficiency of our software. Our current implementation using PARAMESH has enabled us to obtain all of the results shown using, typically, 64 computational cores. This is sufficient to provide the necessary memory to undertake the required runs (up to 32GB) however for faster simulations, or to solve even more challenging problems, such as non-isothermal models, the parallel performance of the solver will need to be improved. This is the second main strand of our current research and will be reported elsewhere.

Acknowledgments

The authors would like to thank Jan Rosam and James Green for their contributions to the development of the implicit software used in this work. The PARAMESH software used in this work was developed at the NASA Goddard Space Flight Center and Drexel University under NASA's HPCC and ESTO/CT projects and under grant NNG04GP79G from the NASA/AISR project. In addition we wish to thank Jonathan Dantzig for providing the explicit software used for comparison. The parallel simulations have been run on HECToR, the UK's National Supercomputing Service, for which we gratefully acknowledge the compute time provided to us.

References

[1] M. Ben Amar and E. A. Brener, *Theory of pattern selection in 3-dimensional nonaxisymmetric dendritic growth*, Phys. Rev. Lett., 71 (1993), pp. 589–592.

[2] B. P. Athreya and J. A. Dantzig, in Solidification Process and Microstructures: A Symposium in Honour of Wilfred Kurz, edited by M. Rappaz, C. Beckermann and R. Trivedi, TMS, Warrendale PA, (2004), pp. 357–368.

[3] M. J. Baines, M. E. Hubbard, P. K. Jimack and R. Mahmood, *A moving-mesh finite element method and its application to the numerical solution of phase-change problems*, Commun. Comput. Phys., 6 (2009), pp. 595–624.

[4] A. Barbieri and J. S. Langer, *Predictions of dendritic growth-rates in the linearized solvability theory*, Phys. Rev. A, 39 (1989), pp. 5314–5325.

[5] A. Brandt, *Multi-level adaptive solutions to boundary value problems*, Math. Comput., 31 (1977), pp. 333–390.

[6] W. L. Briggs, A Multigrid Tutorial, second edition, SIAM, 2000.

[7] D. J. Browne and J. D. Hunt, *A fixed grid front-tracking model of the growth of a columnar front and an equiaxed grain during solidification of an alloy*, Numer. Heat Trans. B, 45 (2004), pp. 395–419.

[8] G. Caginalp, *Stefan and Hele-Shaw type models as asymptotic limits of the phase-field equations*, Phys. Rev. A, 39 (1989), pp. 5887–5896.

[9] S. Chen, B. Merriman, S. Osher and P. Smereka, *A simple level set method for solving Stefan problems*, J. Comput. Phys., 135 (1997), pp. 8–29.

[10] A. I. Ciobanas, Y. Fautrelle and F. Batraretu, A. M. Bianchi and A. Noppel, in Modelling of Casting Welding and Advanced Solidification Processing XI, edited by Ch. A. Gandin and M. Bellet, TMS, Warrendale PA, (2006), pp. 299–306..

[11] M. R. Dorr, J. L. Fattebert, M. E. Wickett, J. F. Belak and P. E. A. Turchi, *A numerical algorithm for the solution of a phase-field model of polycrystalline materials*, J. Comput. Phys., 229 (2010), pp. 626–641.

[12] H. Emmerich, *Advances of and by phase-field modelling in condensed-matter physics*, Adv. Phys., 57 (2008), pp. 1–87.

[13] Ch.-A. Gandin and M. Rappaz, *A coupled finite-element cellular-automaton model for the prediction of dendritic grain structures in solidification processes*, Acta Mater., 42 (1994), pp. 2233–2246.

[14] W. L. George and J. A. Warren, *A parallel 3D dendritic growth simulator using the phase-field method*, J. Comput. Phys., 177 (2002), pp. 264–283.

[15] J. W. Gibbs, *A method of geometrical representation of the thermodynamic properties of substances by means of surfaces*, T. Conn. Acad., 2 (1873), pp. 382–404.

[16] C. E. Goodyer and M. Berzins, *Adaptive timestepping for elastohydrodynamic lubrication solvers*, SIAM J. Sci. Comput., 28 (2006), pp. 626–650.

[17] J. R. Green, P. K. Jimack, A. M. Mullis and J. Rosam, *An adaptive, multilevel scheme for the implicit solution of three-dimensional phase-field equations*, Numer. Meth. Partial Differential Eq., 27 (2011), pp. 106–120.

[18] T. Y. Hou, Z. Li, S. Osher and H. Zhao, *A hybrid method for moving interface problems with application to the Hele-Shaw flow*, J. Comput. Phys., 134 (1997), pp. 236–252.

[19] W. Hundsdorfer and J. G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer-Verlag, 2003.

[20] J. Rosam, A fully Implicit, Fully Adaptive Multigrid Method for Multiscale Phase-Field Modelling, PhD Thesis, University of Leeds, (2007).

[21] J. H. Jeong, N. Goldenfeld and J. Dantzig, *Phase field model for three-dimensional dendritic growth with fluid flow*, Phys. Rev. E, 64 (2001), 041602.

[22] A. Karma and W.-J. Rappel, *Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics*, Phys. Rev. E, 53 (1996), pp. R3017–R3020.

[23] A. Karma and W.-J. Rappel, *Numerical simulation of three-dimensional dendritic growth*, Phys. Rev. Lett., 77 (1996), pp. 4050–4053.

[24] A. Karma and W.-J. Rappel, *Quantitative phase-field modeling of dendritic growth in two and three dimensions*, Phys. Rev. E, 57 (1998), pp. 4323–4349.

[25] A. Karma, *Phase-field formulation for quantitative modeling of alloy solidification*, Phys. Rev. Lett., 87 (2001), 115701.

[26] D. A. Kessler, J. Koplik and H. Levine, *Pattern selection in fingered growth phenomena*, Adv. Phys., 37 (1988), pp. 255–339.

[27] Y. T. Kim, N. Goldenfeld and J. Dantzig, *Computation of dendritic microstructures using a level set method*, Phys. Rev. E, 62 (2000), pp. 2471–2474.

[28] R. Kobayashi, *Modeling and numerical simulations of dendritic crystal-growth*, Phys. D, 63 (1993), pp. 410–423.

[29] J. S. Langer, Directions in Condensed Matter Physics (eds. G. Grinstein and G. Mazenko), World Scientific Publishing: Singapore, 1986, pp. 164–186.

[30] H. K. Lin, C. C. Chen and C. W. Lan, *Adaptive three-dimensional phase-field modeling of dendritic crystal growth with high anisotropy*, J. Cryst. Growth, 318 (2011), pp. 51–54.

[31] P. MacNeice, K. M. Olson, C. Mobarry, R. deFainchtein and C. Packer, *PARAMESH: A parallel adaptive mesh refinement community toolkit*, Comput. Phys. Commun., 126 (2000), pp.330–354.

[32] M. A. Martorano, C. Beckermann and Ch.-A. Gandin, *A solutal interaction mechanism for the columnar-to-equiaxed transition in alloy solidification*, Metall. Mater. Trans. A,

34 (2003), pp. 1657–1674.

[33] M. A. Martorano and V. B. Biscuola, *Columnar front tracking algorithm for prediction of the columnar-to-equiaxed transition in two-dimensional solidification*, Modell. Simul. Mater. Sci. Eng., 14 (2006), pp. 1225–1243.

[34] E. Meca and M. Plapp, *Phase-field study of the cellular bifurcation in dilute binary alloys*, Metall. Mater. Trans. A, 38 (2007), pp. 1407–1416.

[35] B. Merriman, J. K. Bence and S. J. Osher, *Motion of multiple junctions-a level set approach*, J. Comput. Phys., 112 (1994), pp. 334–363.

[36] A. M. Mullis, J. Rosam and P. K. Jimack, *Solute trapping and the effects of anti-trapping currents on phase-field models of coupled thermo-solutal solidification*, J. Cryst. Growth, 312 (2010), pp. 1891–1897.

[37] K. Olson and P. MacNeice, *An over of the PARAMESH AMR software and some of its applications*, in Adaptive Mesh Refinement-Theory and Applications, Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Series: Lecture Notes in Computational Science and Engineering, eds. T. Plewa, T. Linde and G. Weirs (Berlin: Springer), (2005).

[38] K. Olson, PARAMESH: A Parallel Adaptive Grid Tool, In Parallel Computational Fluid Dynamics 2005: Theory and Applications, ed. A. Deane et al. (Elsevier), 2006.

[39] O. Penrose and P. C. Fife, *Thermodynamically consistent models of phase-field type for the kinetics of phase-transitions*, Phys. D, 43 (1990), pp. 44–62.

[40] Y. Pomeau and M. Ben-Amar, Solids far from equilibrium, Cambridge University Press, 1992.

[41] N. Provatas, N. Goldenfeld and J. Dantzig, *Adaptive mesh refinement computation of solidification microstructures using dynamic data structures*, J. Comput. Phys., 148 (1999), pp. 265–290.

[42] N. Provatas, M. Greenwood, B. Athreya, N. Goldenfeld and J. Dantzig, *Multiscale modelling of solidification: phase-field methods to adaptive mesh refinement*, Int. J. Mod. Phys. B, 19 (2005), pp. 4525–4565.

[43] T. Pusztai, G. Tegze, G. I. Toth, L. Kornyei, G. Bansel, Z. Fan and L. Granasy, *Phase-field approach to polycrystalline solidification including heterogeneous and homogeneous nucleation*, J. Phys. Condens. Matter, 20 (2008), 404205.

[44] Z. Qiao, Z. Zhang and T. Tang, *An adaptive time-stepping strategy for the molecular beam epitaxy models*, SIAM J. Sci. Comput., 33 (2011), pp. 1395–1414.

[45] J. C. Ramirez and C. Beckermann, *Examination of binary alloy free dendritic growth theories with a phase-field model*, Acta Mater., 53 (2005), pp. 1721–1736.

[46] J. C. Ramirez, C. Beckermann, A. Karma and H.-J. Diepers, *Phase-field modeling of binary alloy solidification with coupled heat and solute diffusion*, Phys. Rev. E, 69 (2004), 051607.

[47] J. Rosam, P. K. Jimack and A. M. Mullis, *A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification*, J. Comput. Phys., 225 (2007), pp. 1271–1287.

[48] J. Rosam, P. K. Jimack and A. M. Mullis, *An adaptive, fully implicit multigrid phase-field model for the quantitative simulation of non-isothermal binary alloy solidification*, Acta Mater., 56 (2008), pp. 4559–4569.

[49] L. F. Shampine, *Implementation of implicit formulas for the solution of ODEs*, SIAM J. Sci. Stat. Comput., 1 (1980), pp. 103–118.

[50] Z. Tan, K. M. Lim and B. C. Khoo, *An adaptive mesh redistribution method for the incompressible mixture flows using phase-field model*, J. Comput. Phys., 225 (2007), pp. 1137–1158.

[51] N. Touheed, P. Selwood, P. K. Jimack and M. Berzins, *A comparison of some dynamic load-balancing algorithms for a parallel adaptive flow solver,* Para. Comput., 26 (2000), pp. 1535–1554.

[52] U. Trottenberg, C. Oosterlee and A. Schuller, Multigrid, Academic Press, 2001.

[53] Y. L. Tsai, C. C. Chen and C. W. Lan, *Three-dimensional adaptive phase field modeling of directional solidification of a binary alloy: 2D-3D transitions*, Int. J. Heat Mass Trasf., 53 (2010), pp. 2272–2283.

[54] L. Vanherpe, F. Wendler, B. Nestler and S. Vandewalle, *A multigrid solver for phase field simulation of microstructure evolution*, Math. Comput. Simul., 80 (2010), pp. 1438–1448.

[55] C. Y. Wang and C. Beckermann, *Prediction of columnar to equiaxed transition during diffusion-controlled dendritic alloy solidification*, Metall. Mater. Trans. A, 25 (1994), pp. 1081–1093.

[56] H. Wang, R. Li and T. Tang, *Efficient computation of dendritic growth with r-adaptive finite element methods*, J. Comput. Phys., 227 (2008), pp. 5984–6000.