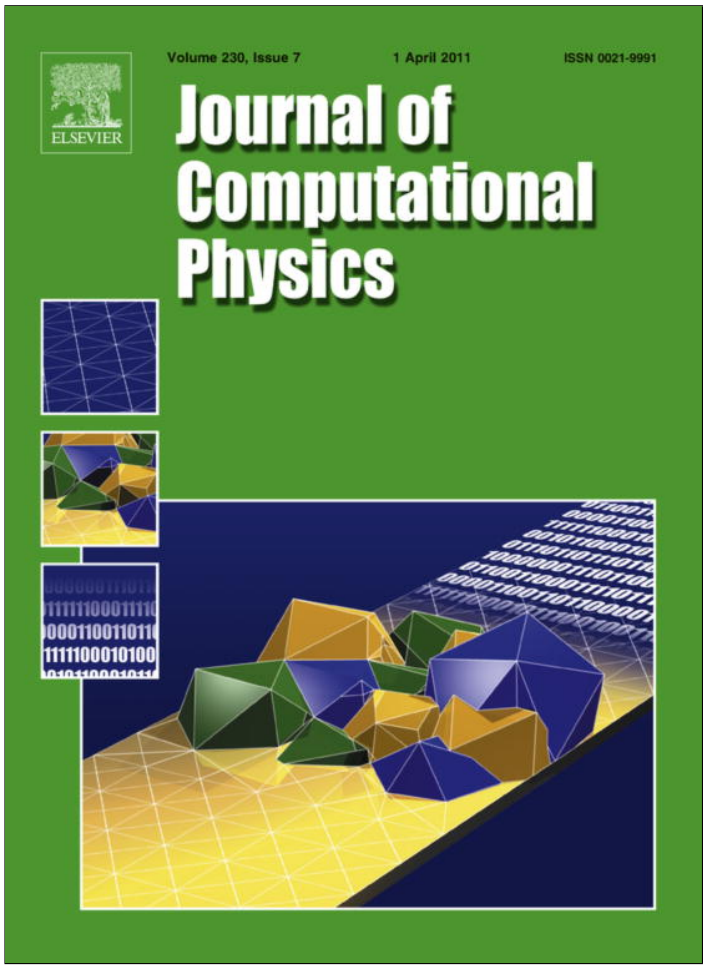

Journal of Computational Physics 230 (2011) 2451–2469

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions

R.K. Crockett *, P. Colella, D.T. Graves

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720, United States

ARTICLE INFO

Article history:
Received 23 October 2009
Received in revised form 13 December 2010
Accepted 13 December 2010
Available online 19 December 2010

Keywords:
Heat equation
Jump conditions
Discontinuous coefficient
Irregular domain
Finite volume methods
Multigrid methods

ABSTRACT

We present a method for solving Poisson and heat equations with discontinuous coefficients in two- and three-dimensions. It uses a Cartesian cut-cell/embedded boundary method to represent the interface between materials, as described in Johansen and Colella (1998). Matching conditions across the interface are enforced using an approximation to fluxes at the boundary. Overall second order accuracy is achieved, as indicated by an array of tests using non-trivial interface geometries. Both the elliptic and heat solvers are shown to remain stable and efficient for material coefficient contrasts up to 10^6, thanks in part to the use of geometric multigrid. A test of accuracy when adaptive mesh refinement capabilities are utilized is also performed. An example problem relevant to nuclear reactor core simulation is presented, demonstrating the ability of the method to solve problems with realistic physical parameters.

1. Introduction

We consider elliptic and parabolic problems in regions with two materials, each of which occupies a bounded subset Ω^p, $p = +, -$, of the overall domain $\Omega = \bigcup_p \Omega^p$. One region usually encloses the other; in that case we refer to the inclusion, or interior region, by Ω^- and the exterior region by Ω^+. At the boundary $\delta\Omega^{\pm}$ between materials, jump conditions on the solution φ and flux $\mathbf{F}^B$ are specified

$$\begin{aligned} [\varphi^B] &= \varphi^{B,+} - \varphi^{B,-} = g_D(\mathbf{x}, t), \\ \left[\mathbf{F}^B \cdot \hat{\mathbf{n}}^B\right] &= \mathbf{F}^{B,+} \cdot \hat{\mathbf{n}}^B - \mathbf{F}^{B,-} \cdot \hat{\mathbf{n}}^B = g_N(\mathbf{x}, t). \end{aligned} \tag{1}$$

Here $\hat{\mathbf{n}}^B$ is the normal to the boundary, and the functions g_D and g_N describe the magnitude of the jump at each point in time and space. For many problems $g_D = g_N = 0$. However, this method can be used when more general jump conditions are required, for instance when sources are present at the interface. The method can handle material domains Ω^p that consist of any number of spatially distinct sub-domains. It may be applied to domains consisting of more than two materials as well, so long as the interfaces between materials remain spatially distinct. For the present work, we restrict the scope to two materials.

Our method solves the heat equation in two materials,

$$\partial_t \varphi = \kappa^p \Delta \varphi + f \text{ on } \Omega^p, \quad \varphi(\mathbf{x}, 0) = \varphi_0(\mathbf{x}), \tag{2}$$

* Corresponding author. Tel.: +1 303 996 2037; fax: +1 303 448 7756.
E-mail address: RKCrockett@txcorp.com (R.K. Crockett).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.12.017

subject to the above jump conditions at the interface. In this case, as well as in the elliptic equations to which the method is applied, the flux $\mathbf{F}^B$ is proportional to the material coefficient. While this coefficient is constant within each material, it is discontinuous across the interface.

A number of schemes for handling elliptic and parabolic problems of this type exist in the literature. Finite difference schemes for fixed boundaries, of the type pioneered by Shortley and Weller [1], have been greatly improved upon in the intervening years. Most importantly in the context of this work, they have been extended to handle discontinuous jumps in the form of Eq. (1). Immersed boundary methods, modifications of the method first presented by Peskin [2], discretize a delta-function source term on the boundary, retaining the symmetric form of the linear system. Immersed interface methods [3] use analytic continuation of the solution across the interface to explicitly incorporate the jump condition into the underlying finite difference stencil coefficients. This results in a scheme that more accurately represents the jump conditions, at the expense of considerable additional complexity and the loss of symmetry in the underlying linear system for non-constant coefficients. Finally in the context of finite difference schemes, ghost fluid methods [4] also use analytic continuation of the solution. A ghost fluid, residing in the regions outside the solution domain, is used to explicitly enforce the matching conditions. As originally formulated, the method was first order accurate. It has been extended to second order accuracy for boundaries with continuous second derivatives [5]. They have the advantage of retaining a symmetric system, allowing the use of a wider range of fast linear solvers. More recently, another symmetry-preserving method developed by Chen and Strain [6] couple a polynomial reconstruction of the solution at the interface with a multigrid-preconditioned Krylov solver that improves efficiency for large material coefficient contrasts.

Outside the realm of finite difference methods, integral methods recast the elliptic PDE via potential theory as integral equations. Fast integral solvers can often be utilized against this class of problem; these generally fall into two categories. One method involves the use of a fast Poisson solver on a simple (e.g. Cartesian) enclosing domain, plus the application of a suitable correction at the boundary [7]. The second combines a fast (e.g. multipole or FFT) method and an iterative solver. In either case, conditioning issues can arise in problems with large discontinuities in the material coefficient [8], necessitating a modification of the underlying integral equation representation. Nevertheless, these methods are efficient in a wide variety of problems.

Our method is based on a finite volume approach to the spatial discretization of elliptic equations. The method is conservative, a distinct advantage in certain classes of problems. For low-Mach flows with heat transfer, for instance, conservative schemes avoid unphysical results arising in marginally resolved or under-resolved situations [9,10].

Finite volume methods for interface problems encompass a variety of approaches. In the context of conjugate heat transfer in complex geometries, the overlapping grid method of Henshaw and Chand [11] decomposes the domain into a number of sub-domains. The grid on each sub-domain is boundary fitting, an advantage that comes at the expense of the loss of conservation. Each sub-domain uses a solver specific to the relevant physics in it.

Oevermann et al. [12,13] present a hybrid finite volume method for variable and discontinuous coefficient elliptic problems in two- and three-dimensions. In 3-D, it relies on tri-linear approximations to the solution within each Cartesian control volume to discretize the integral form of the divergence theorem in a finite element fashion. Small volume cells are handled via an asymptotic approach. The method exhibits local and global second order accuracy on this class of problems.

Our work follows in the steps of the work in [14–16] in using pure finite-volume schemes for elliptic and parabolic equations with embedded boundaries (EB). The first step in using Cartesian EB methods is grid generation, which has been studied extensively using a number of different representations of the geometry. Surface triangulations [17] are widely used, particularly in engineering contexts involving extremely complex geometries. Our method uses an implicit function representation [18] that provides discretizations of complex geometries accurate to arbitrary order in a straightforward manner.

The main shortcoming in previous Cartesian EB methods, insofar as their application to multi-material problems, lies in their use of prescribed boundary conditions at the EB. A Neumann interface gave boundary fluxes directly, while Dirichlet boundary conditions at the EB necessitated defining a stencil for calculating fluxes at the boundary using data at neighboring cells. In the multi-material context boundary conditions at the interface are not directly prescribed, but instead constrained by matching conditions on the jump in the solution and flux across it. The present work extends the EB methodology to handle such jump conditions, and thereby solve multi-material Poisson and heat equations with a discontinuity in the material coefficient at the interface between the two. Like previous work, it maintains global second order accuracy. By treating special cases related to under-resolved geometries, like the presence of multiple interfaces within a single Cartesian control volume, we are able to use geometric multigrid methods for efficient solution of elliptic equations. Moreover, our use of the Chombo software infrastructure provides important capabilities from a computational efficiency standpoint. The primary one is adaptive mesh refinement, which is crucial in many problems involving widely separated spatial scales.

Our time discretization of this equation necessitates solving a set of elliptic equations during each step forward in time. Specifically, we solve the Helmholtz equation

$$(\alpha^p + \beta^p \Delta)\varphi = \rho, \tag{3}$$

where α^p and β^p are the material coefficients, subject to jump conditions across the boundary $\delta\Omega^{\pm}$. We first describe the elliptic algorithm, starting with the spatial discretization in Section 2. This is followed by a treatment of special considerations for the use of geometric multigrid, in Section 3. The stability of the spatial discretization and time integration are explored in Section 4. An outline of the overall algorithm is given in Section 5.

Descriptions of the tests used to validate the method follow the algorithmic sections. These include two and three dimensional solution error tests confirming second order accuracy for the Poisson equation in Section 6.1 and Section 6.3. Tests of the efficiency of the multigrid solver in 2-D and 3-D are described in Section 6.2. The final set of tests are of accuracy for the heat equation in 3-D, in Section 6.4. We conclude with results for a more realistic problem, showing heat conduction in a nuclear reactor fuel assembly, in Section 6.5.

2. Spatial discretization of the Laplacian

The underlying discretization of space is given by rectangular control volumes on a Cartesian grid: $\Upsilon_{\mathbf{i}} = \left[(\mathbf{i}-\frac{1}{2}\mathbf{u})h, (\mathbf{i}+\frac{1}{2}\mathbf{u})h\right],\ \mathbf{i}\in\mathbb{Z}^d$, where d is the dimensionality of the problem, h is the mesh spacing, and $\mathbf{u}$ is the vector whose entries are all ones. Each material domain Ω^p and material interface $\delta\Omega^{pp'}$ is represented by its intersection with the Cartesian grid. In general, a given Cartesian control volume $\Upsilon_{\mathbf{i}}$ may be intersected by one or more material interfaces. Consequently, there may be multiple p-material control volumes associated with each $\Upsilon_{\mathbf{i}}$; see Fig. 3. For simplicity we will delay treatment of multi-cells until Section 3. In the case that there is only one p-material control volume in each Cartesian cell, we denote them by $V^p_{\mathbf{i}}$, and each face by $A^p_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s}$. The latter is defined as the intersection of Ω^p with $\delta\Upsilon_{\mathbf{i}}$, the boundary of the control volume $\Upsilon_{\mathbf{i}}$ given by the coordinate planes $\{\mathbf{x} : x_s = (i_s \pm \frac{1}{2}\hat{\mathbf{e}}_s)h\}$, where $\hat{\mathbf{e}}_s$ is the unit vector in the s direction. Finally, associated with each Cartesian control volume through which a material boundary passes is a boundary face, or interface, $A^{B,p}_{\mathbf{i}} = dV^p_{\mathbf{i}} \cap dV^{p'}_{\mathbf{i}} \subset \delta\Omega^{pp'}$ with normal $\hat{\mathbf{n}}^{B,p}_{\mathbf{i}}$ facing out of material p. We assume that each material interface connects a single p-material control volume to a single p'-material control volume. In other words, each p-material interface must be connected in a one-to-one fashion to another face in material $p' \neq p$ having the same area fraction, but opposite normal (i.e. $\left|A^{B,p}_{\mathbf{i}}\right| = \left|A^{B,p'}_{\mathbf{i}}\right|$ and $\hat{\mathbf{n}}^{B,p}_{\mathbf{i}} = -\hat{\mathbf{n}}^{B,p'}$).

The construction of our finite-volume method follows McCorquodale et al. [15], with suitable modifications for multi-material equations. Based on the description above, we construct geometric quantities:

- The dimensionless volumes/areas of each p-material control volume/face. Volume fractions $v^p_{\mathbf{i}} = |V^p_{\mathbf{i}}|h^{-d}$, face apertures $\alpha^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s} = \left|A^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s}\right|h^{-(d-1)}$, and boundary apertures $\alpha^{B,p}_{\mathbf{i}} = |A^B_{\mathbf{i}}|h^{-(d-1)}$.
- The locations of volume, face, and boundary centroids; and average outward normal to the boundary

$$\mathbf{x}^p_{\mathbf{i}} = \frac{1}{|V^p_{\mathbf{i}}|}\int_{V^p_{\mathbf{i}}} \mathbf{x}\,dV, \tag{4}$$

$$\mathbf{x}^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s} = \frac{1}{\left|A^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s}\right|}\int_{A^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s}} \mathbf{x}\,dA, \tag{5}$$

$$\mathbf{x}^{B,pp'}_{\mathbf{i}} = \frac{1}{|A^B_{\mathbf{i}}|}\int_{A^B_{\mathbf{i}}} \mathbf{x}\,dA, \tag{6}$$

$$\hat{\mathbf{n}}^{B,p}_{\mathbf{i}} = \frac{1}{|A^B_{\mathbf{i}}|}\int_{A^B_{\mathbf{i}}} \hat{\mathbf{n}}^{B,p}\,dA, \tag{7}$$

where $\hat{\mathbf{n}}^{B,p}$ is the normal, facing outward from phase p, to $\delta\Omega^{pp'}$, defined at each point on $\delta\Omega^{pp'}$.

Finite volume methods are based on the divergence form of the underlying equation, which we now recast in terms of the above defined geometric quantities. For the Poisson (heat) equation, the divergence form is $\nabla\cdot\mathbf{F} = \rho$ ($\nabla\cdot\mathbf{F} = \partial_t\varphi - f$), with $\mathbf{F}^p = \kappa^p\nabla\varphi$. Our conservative discretization for the divergence operator in each material control volume $V^p_{\mathbf{i}}$ is

$$(\nabla\cdot\mathbf{F})^p_{\mathbf{i}} \approx \frac{1}{V^p_{\mathbf{i}}}\int_{V^p_{\mathbf{i}}}\nabla\cdot\mathbf{F}\,dV = \frac{1}{V^p_{\mathbf{i}}}\int_{\delta V^p_{\mathbf{i}}}\mathbf{F}\cdot\hat{\mathbf{n}}\,dA \simeq \frac{1}{v^p_{\mathbf{i}}h}\left[\sum_{\pm=+,-}\sum_{s=1}^{d}\pm\alpha^p_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_s}F^{s,p}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_s} + \alpha^B_{\mathbf{i}}F^{B,p}_{\mathbf{i}}\right]. \tag{8}$$

Here $F^{s,p}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_s}$ is the flux into the p-material through the face with normal $\pm\mathbf{e}_s$, and $F^{B,p}_{\mathbf{i}}$ is the corresponding flux through the material boundary. So, for example, in the material control volume at top in Fig. 1, the divergence is

$$(\nabla\cdot\mathbf{F})^-_{\mathbf{i}} = \frac{1}{v^-_{\mathbf{i}}h}\left[\alpha^-_{\mathbf{i}-\frac{1}{2}\mathbf{e}_0}F^{0,-}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_0} - \alpha^-_{\mathbf{i}+\frac{1}{2}\mathbf{e}_0}F^{0,-}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_0} - \alpha^-_{\mathbf{i}+\frac{1}{2}\mathbf{e}_1}F^{1,-}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_1} + \alpha^B_{\mathbf{i}}F^{B,-}_{\mathbf{i}}\right]. \tag{9}$$

With the discretization in place, we need to calculate fluxes at all faces and boundaries in order to update the governing equation.

2.1. Flux calculation and enforcing multi-material matching conditions

The Laplacian operator in the Poisson (heat) equation implies a flux $\mathbf{F}^p = \kappa^p\nabla\varphi$. Calculation of fluxes at faces $A^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s}$ is a relatively straightforward matter of linearly interpolating fluxes at face centers to the face centroids $\mathbf{x}^p_{\mathbf{i}+\frac{1}{2}\mathbf{e}_s}$; an example is given in [15], Eqs. (6) and (7). For interfaces A^B, the process is more involved. Let us begin with the flux equations applicable to the Poisson and Helmholtz equations,

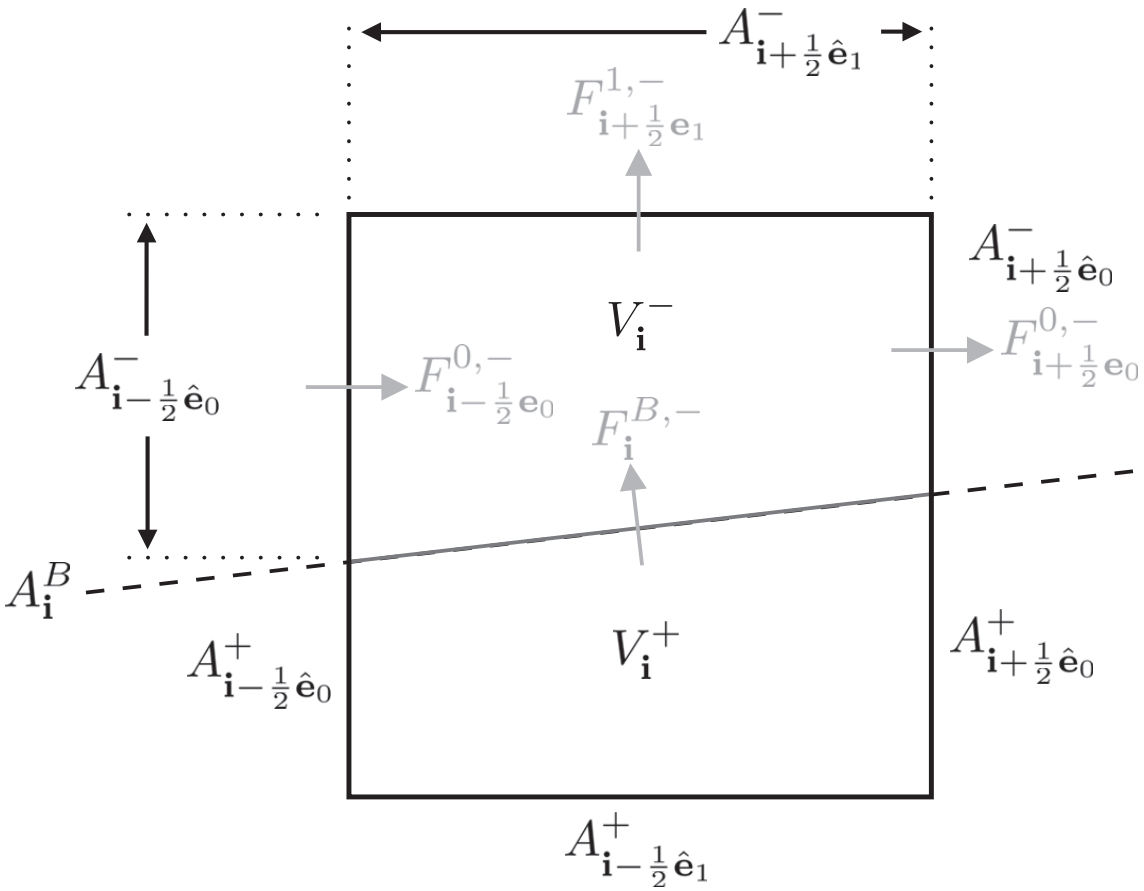

Fig. 1. A 2-D Cartesian control volume, and its associated material control volumes. In this case the volume is crossed by a single material interface. The top region is in the $p = -$ phase and the bottom region in phase $p' = +$. Each material control volume is labeled by the Cartesian volume containing it, $\mathbf{i}$, and its phase, p, as $V^p_{\mathbf{i}}$. The interface between material control volumes is shown as a solid, dark grey line whose area is labeled A^B. Non-zero faces of the material control volumes, lying on the boundary of the Cartesian control volume, are labeled $A^p_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s}$. Fluxes are indicated for the upper material volume $V^-_{\mathbf{i}}$.

$$\hat{\mathbf{n}}^{B,p}\cdot\mathbf{F}=\hat{\mathbf{n}}^{B,p}\cdot(\kappa^p\nabla\varphi)^{B,p}\equiv\kappa^p\frac{\partial\varphi^{B,p}}{\partial n}. \tag{10}$$

In the single-material case [15,16], Neumann or Dirichlet boundary conditions are prescribed at the interface. In the former case, the required interface flux $\mathbf{F}^B$ can be calculated directly. In the Dirichlet case, the boundary condition at the interface φ^B and the state at neighboring cells $\varphi_{\mathbf{i}}$ is used to approximate the normal derivative at the interface $\frac{\partial\varphi}{\partial n}$. In the multi-material case, neither Dirichlet nor Neumann boundary conditions at the interface are known a priori. Instead, we use the matching conditions, Eq. (1), along with equations approximating the normal derivative in each material, in order to calculate the boundary flux in Eq. (8).

We illustrate the flux determination for the case of the Poisson equation in two materials. The same procedure applies to the Helmholtz equation solves done in the heat equation update. First, we rewrite the second (flux) jump condition of Eq. (1) as

$$\kappa^+\frac{\partial\varphi^{B,+}}{\partial n}-\kappa^-\frac{\partial\varphi^{B,-}}{\partial n}=g_N. \tag{11}$$

The states in either material at the interface, $\varphi^{B,p}$, and normal derivatives, $\frac{\partial\varphi^{B,p}}{\partial n}$, are not known. Our method for closing this set of equations is to use an approximation to the normal derivative of the form

$$\frac{\partial\varphi^{B,p}}{\partial n}=w^{B,p}\varphi^{B,p}+\sum_{\substack{\mathbf{i}\\ \mathbf{i}\in\Omega^p}}w_{\mathbf{i}}\varphi_{\mathbf{i}}+\mathbf{O}(h^q), \tag{12}$$

where q is the order of approximation. The weight w^B associated with the unknown boundary state and the weights $w_{\mathbf{i}}$ associated with the known states in neighboring cells both depend on the geometric quantities defined in Eqs. (4)–(7). (In much of the remainder of this section we will drop the p superscript for simplicity; all quantities are understood to be calculated in both materials.) In general, a second order approximation [14] to the normal derivative is used. In cases where there are insufficient neighboring cells $\varphi_{\mathbf{i}}$ for a second order approximation (e.g. due to an under-resolved geometry, or proximity to the domain boundary), a first order approximation [16] is used.

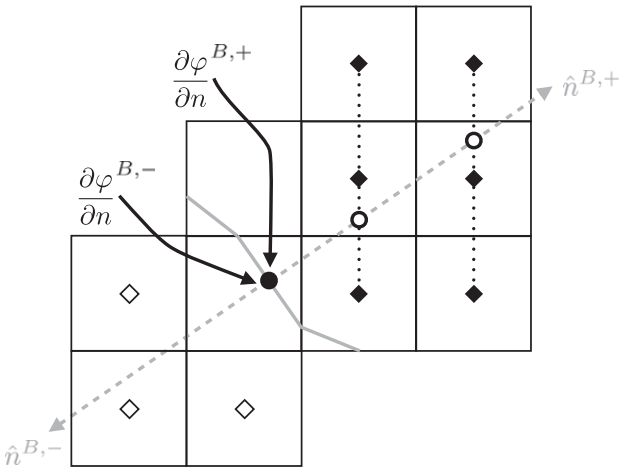

Fig. 2. Illustration of stencil calculation for multi-material case. The boundary is shown as a solid grey line, and the inward-facing normal in each material, $\hat{n}^{B,\pm}$, as a dotted grey line. In this case, in the $p = -$ material (lower-left) least squares is used to determine a stencil approximation to the normal derivative at the boundary, $\frac{\partial\varphi^{B,-}}{\partial n}$. Three neighboring points (open diamonds) are chosen, each of which is assigned a stencil weight based on the least squares algorithm. In the $p = +$ material, at upper-right, a quadratic approximation to the solution along the dotted lines gives a seven point stencil (interface state plus six filled diamonds) for the normal derivative. The stencil weights depend only on geometric quantities. By combining the two stencils on either side of the interface with the two jump conditions, Eq. (1), we are able to calculate fluxes at the interface.

Since we are using two different approximation schemes it remains to show here that, independent of which combination of approximation schemes is used on either side of the interface, the matching conditions can still be enforced. We do so by showing that both can be recast in the form Eq. (12) given above. Clearly, given its linear dependence on φ^B, this allows a simple direct solution of the four-by-four system for φ^B and $\frac{\partial\varphi^B}{\partial n}$ on both sides of the material boundary.

First, consider the second-order stencil based on the quadratic interpolation of two values, φ_1 and φ_2, at distances d_1 and d_2 along the normal to the interface,

$$\frac{\partial\varphi^B}{\partial n} = \frac{1}{d_2 - d_1}\left(\frac{d_2}{d_1}(\varphi^B - \varphi_1) - \frac{d_1}{d_2}(\varphi^B - \varphi_2)\right). \tag{13}$$

This is the situation illustrated in the material at top-right in Fig. 2. The two state values φ_1 and φ_2 depend on geometric quantities and the states in neighboring cells (the filled diamonds in Fig. 2), but not φ^B. Therefore, it is a simple matter to rewrite this equation in the form above

$$\frac{\partial\varphi^B}{\partial n} = \frac{1}{d_2 - d_1}\left(\frac{d_2}{d_1} - \frac{d_1}{d_2}\right)\varphi^B + \frac{1}{d_2 - d_1}\left(\frac{d_1}{d_2}\varphi_2 - \frac{d_2}{d_1}\varphi_1\right) \tag{14}$$

$$= w^B\varphi^B + \sum_{\mathbf{i}\in\Omega_1} w_{\mathbf{i}}\varphi_{\mathbf{i}} + \sum_{\mathbf{i}\in\Omega_2} w_{\mathbf{i}}\varphi_{\mathbf{i}}, \tag{15}$$

where Ω_1 and Ω_2 correspond to the domain of dependence used in calculating φ_1 and φ_2. This is clearly in the form of Eq. (12).

In the alternative case, when a suitable second order approximation is not available, we use the Schwartz et al. prescription for calculating the normal derivative based on least squares estimation. This is the situation illustrated in the bottom-left material of Fig. 2. The method involves choosing a suitable set of neighboring points – three in the case of 2-D, and seven in 3-D – for performing least squares estimation. A matrix of displacements of these points from the material interface is defined, $(\mathbb{A})_{s\mu} = \delta x_s^\mu$, where μ indexes the points and s indexes direction. Next define a vector of differences between the state at each point, φ^μ, and the state at the boundary, φ^B: $(\delta\varphi)^\mu = (\varphi)^\mu - \varphi^B$. The algorithm solves $\mathbb{A}\nabla\varphi = \delta\varphi$ in order to obtain an estimate for $\frac{\partial\varphi}{\partial n} = \hat{\mathbf{n}}\cdot\nabla\varphi$. The least squares estimate can be rewritten as

$$\nabla\varphi \approx (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\delta\varphi = (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T(\varphi - \varphi^B\mathbf{u}) = \left((\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\mathbf{u}\right)\varphi^B + (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\varphi. \tag{16}$$

An expression for the normal derivative can then be constructed

$$\frac{\partial\varphi}{\partial n} = \hat{\mathbf{n}}\cdot\nabla\varphi = -\left(\hat{\mathbf{n}}^T(\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\mathbf{u}\right)\varphi^B + \hat{\mathbf{n}}^T(\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\varphi = w^B\varphi^B + \sum_{\mathbf{i}} w_{\mathbf{i}}\varphi_{\mathbf{i}}. \tag{17}$$

The last step follows from the association of the components of the vector φ with the neighboring state values, $\varphi_{\mathbf{i}}$, and the matrix product $\hat{\mathbf{n}}^T(\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\mathbf{u}$ with the scalar w^B.

We therefore have two formulae for approximating the normal derivative, both of which are linear in the φ^B. We have only to choose either quadratic Johansen or least squares stencil on both sides of the interface. By combining the two stencil approximation equations with the two jump conditions, we have a system that can be directly solved for the boundary states

and normal derivatives. From these we calculate boundary fluxes $\mathbf{F}^B$, which are used in performing Gauss–Seidel relaxation in each phase. In order that the matching conditions are always satisfied, boundary fluxes are recalculated before each relaxation step of the multi-material elliptic solve.

3. Multigrid solver considerations

In order to speed the elliptic equation solver, the relaxation step is embedded in a multigrid solver. This also has the advantage of being highly compatible with adaptive mesh refinement. Use of a multigrid solver involves progressive coarsening of the problem domain, and the geometry along with it. Generally, the further one coarsens in multigrid, the larger the benefit in terms of solver efficiency. The extent to which one can coarsen is limited, however, by the need for an adequate number of cells with which to create a stencil for estimating the flux. This domain coarsening can lead to pathological cases, which we describe and outline our approach to below.

3.1. Multicells

The most important issue for multigrid solvers is the case where there is more than one interface in a Cartesian control volume. One very simple example is an inclined ellipsoid with a large axis ratio. With enough coarsening, the minor axis of the ellipsoid becomes smaller than the grid spacing, leading to a situation like that pictured in Fig. 4. Such multi-interface cells we term multi-cells. Instead of the usual single sub-volume of a given material in each Cartesian cell, multi-cells contain more than one sub-volume for at least one of the materials. In order to unambiguously identify each sub-volume, we index them by $\gamma = 0, 1, \ldots$ The volume of each is denoted by $V^p_{\mathbf{i},\gamma}$, and each face by $A^p_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s,\gamma}$. The area of the interface between sub-volume γ in material p and sub-volume γ' in material $p' \neq p$ is denoted by $A^{B,p}_{\mathbf{i}}\left(V^p_{\mathbf{i},\gamma}, V^{p'}_{\mathbf{i},\gamma'}\right)$, with associated area fraction $\alpha^{B,pp'}_{\mathbf{i},\gamma\gamma'}$. The normal facing out of material p is denoted $\hat{\mathbf{n}}^{B,pp'}_{\mathbf{i},\gamma\gamma'}$.

The divergence form of the equations in the presence of multi-cells is

$$(\nabla\cdot\mathbf{F})^p_{\mathbf{i},\gamma} \approx \frac{1}{v^p_{\mathbf{i},\gamma}h}\left[\sum_{\pm=+,-}\sum_{s=1}^{d}\pm\alpha^p_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_s,\gamma}F^{s,p}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_s,\gamma} + \sum_{\substack{p'\neq p\\ \gamma'}}\alpha^{B,pp'}_{\mathbf{i},\gamma\gamma'}F^{B,pp'}_{\mathbf{i},\gamma\gamma'}\right].$$

So, for example, in the sub-volume at upper-left in Fig. 3, the divergence is

$$(\nabla\cdot\mathbf{F})^+_{\mathbf{i},0} = \frac{1}{v^+_{\mathbf{i},0}h}\left[\alpha^+_{\mathbf{i}-\frac{1}{2}\mathbf{e}_0,0}F^{0,+}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_0,0} - \alpha^+_{\mathbf{i}+\frac{1}{2}\mathbf{e}_1,0}F^{1,+}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_1,0} + \alpha^{B,+-}_{\mathbf{i},00}F^{B,+-}_{\mathbf{i},00}\right]. \tag{18}$$

Fluxes, and therefore normal derivatives, must be calculated for each sub-volume of a multi-cell separately. These are used to calculate boundary fluxes for the VoF. For the sake of efficiency and simplicity, however, we do not store all values of the boundary flux $F^{B,pp'}_{\mathbf{i},\gamma\gamma'}$ for a given VoF. Instead, we store a single boundary flux, corresponding to the sum over γ' term in Eq. (18), from which the update can be calculated. This total boundary flux is calculated from an area-weighted average of the normal derivatives at each interface in the cell:

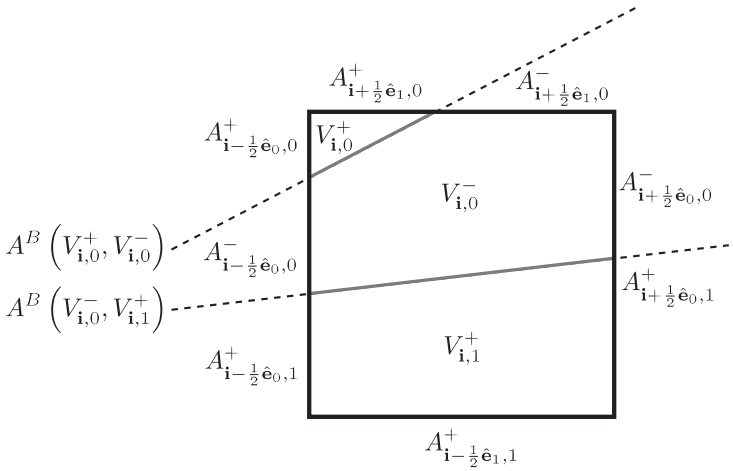

Fig. 3. A 2-D Cartesian control volume in the case where multiple interfaces cross the cell. Here, two interfaces divide the cell into three sub-volumes, or VoFs, one in the $p = -$ material and two in the $p' = +$ material. Each material control volume is labeled by the Cartesian volume containing it, $\mathbf{i}$, its phase p, and the index γ, as $V^p_{\mathbf{i},\gamma}$. Interfaces between material control volumes are shown as solid grey lines, whose areas are labeled $A^B\left(V^p_{\mathbf{i},\gamma}, V^{p'}_{\mathbf{i},\gamma'}\right)$. Non-zero faces of the material control volumes, lying on the boundary of the Cartesian control volume, are labeled $A^p_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s,\gamma}$.

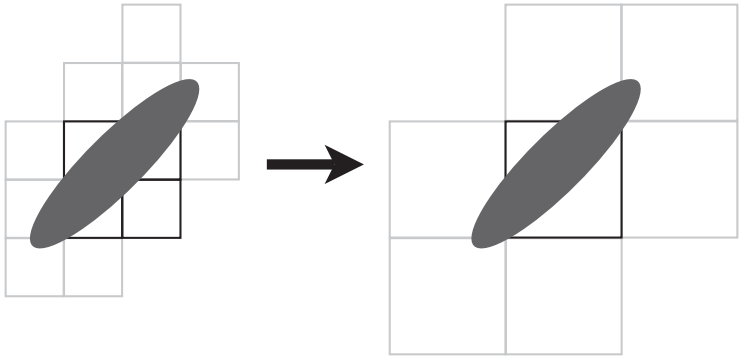

Fig. 4. Example of a coarsening resulting in multiple VoFs in a single rectangular control volume. On the left, the cells in black contain a single VoF of each material. After coarsening, the grid is as on the right, with the cell outlined in black having two VoFs of one material, and a single VoF of the other material.

$$\frac{\partial \varphi^{B,p}}{\partial n_{\mathbf{i},\gamma}} = \frac{1}{\bar{\alpha}^{B}_{\mathbf{i},\gamma}} \sum_{\substack{p' \neq p \\ \gamma'}} \alpha^{B,pp'}_{\mathbf{i},\gamma\gamma'} \frac{\partial \varphi^{B,pp'}}{\partial n_{\mathbf{i},\gamma\gamma'}}. \tag{19}$$

Here the denominator $\bar{\alpha}^{B}_{\mathbf{i},\gamma}$ is the average material face aperture. The total flux into the VoF is

$$F^{B,p}_{\mathbf{i},\gamma} = \kappa^{p} \frac{\partial \varphi^{B,p}}{\partial n_{\mathbf{i},\gamma}} \tag{20}$$

giving the correct total flux into the VoF.

3.2. Failure of the least squares stencil

In this case, we use least squares, but instead of including only those nearest neighbors within the same quadrant (octant in 3D) as the normal vector as was done in [16], we include all nearest neighbors to which there exists a monotone path from the VoF in question.

We find that this method allows coarsening to levels not possible using the least squares stencil as described previously. Moreover, it is useful in preserving symmetry in cases where the boundary normal is along a cardinal direction. It has little effect on the overall accuracy of the scheme, which remains second order in our tests.

3.3. VoFs missing one or more stencils

Occasionally, at some level in the coarsening, no stencil is available on one or both sides of the interface, and we are left with a choice. One option is to stop coarsening and back up to a finer lever at which stencils are available. Another is to fashion a suitable approximation for these under-resolved cases.

In the case that one side of an interface is without a stencil, we are unable to solve the matching conditions in the manner outlined in Section 2.1. Therefore, we approximate the gradient on the side with the stencil directly, taking a simple finite difference using the available cells in that material,

$$(\nabla \varphi)_s = \pm \frac{\varphi_{\mathbf{i} \pm \hat{\mathbf{e}}_s} - \varphi_{\mathbf{i}}}{h}, \tag{21}$$

where $\hat{\mathbf{e}}_s$ is the unit vector in the s direction, and either the backward- or forward-difference is used based on the availability of data φ. We arrive at an estimate of the flux in material p by using the normal to the interface, $\frac{\partial \varphi^{B,p}}{\partial n} = \hat{\mathbf{n}}^{B,p} \cdot \nabla \varphi$. Using this estimate of the flux in phase p, we then calculate the jump in phase p':

$$\frac{\partial \varphi^{B,p'}}{\partial n} = \frac{1}{\kappa^{p'}} \left(\kappa^{p} \frac{\partial \varphi^{B,p}}{\partial n} + g_N \right). \tag{22}$$

This avoids the need for a value of φ^{B} in calculating the normal derivative. Effectively, we are using a first-order accurate approximation to the gradient at the cell-center to approximate the gradient at the interface.

Cases where there is no available stencil on either side of a material interface are rare, even when the geometry is quite under-resolved. In practice, we avoid them by limiting the degree of coarsening the multigrid solver performs, which in turn limits the extent of under-resolution of the geometry. This has the potential to make multigrid less efficient. However, we show in Section 6.2 that it remains efficient across a range of resolutions, even for quite complex geometries.

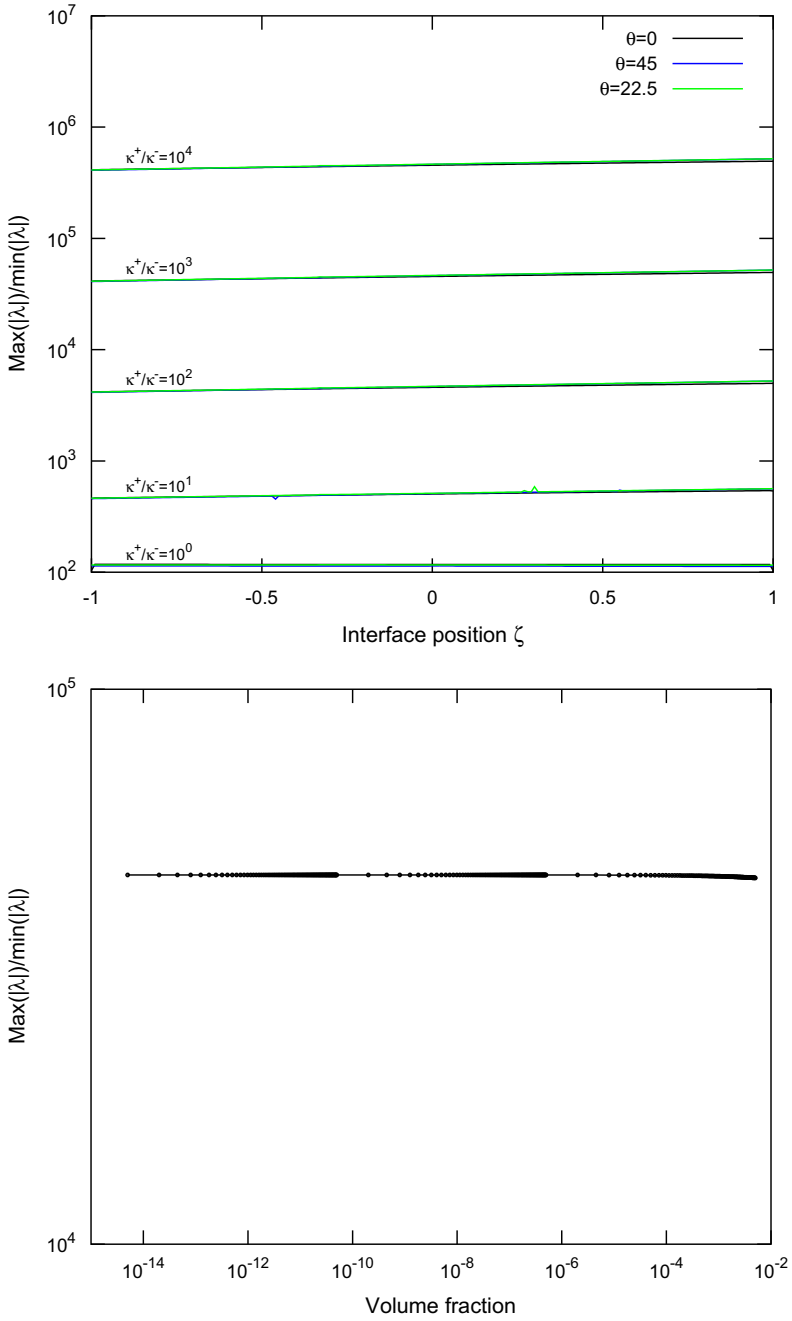

Fig. 5. Top: Condition number versus interface position for a range of coefficient ratios $r = \kappa^-/\kappa^+$. Three cases are presented: grid aligned ($\theta = 0$; black lines), and inclined planes at $\theta = 45$ (blue lines) and $\theta = 22.5$ (green lines). Each case is shown at five values of r, from 10^0 to 10^4. Bottom: Condition number versus volume fraction, planar interface at $\theta = 45$ and $r = 10^3$.

4. Stability analysis

The choice of spatial discretization above raises two questions. The first is whether it is stable. The second is the stability of the TGA time integration scheme for the heat equation under this spatial discretization. The first is difficult to prove mathematically. The approach of Li and Ito [19] was to use an optimization technique to ensure that the stencil coefficients satisfy the discrete maximum principle. Our more empirical approach relies on numerical experiments that test robustness under a variety of conditions. We use the condition number of the matrix representation of the spatial discretization as an indicator of stability. Specifically, for a series of one- and two-dimensional test problems with different geometries we constructed this matrix representation. By calculating the eigenvalues of these matrices we were able to discern how the matrix condition number (which we estimate by the ratio of largest to smallest eigenvalue magnitude) and stability function for the TGA scheme vary with volume fraction, grid resolution, and geometry.

For embedded boundary methods in a single material, previous authors [14] found that, in grid-aligned, planar geometries, the condition number of the spatial discretization is essentially constant as very small volume fractions are reached.

The good conditioning of the linear problem is owed to the volume-weighting of the matrix, which prevents a v^{-1} scaling of the largest eigenvalues. We find that grid-aligned planar geometries show similar conditioning in the multi-material case. Our test places the planar interface at two hundred positions within a single cell, calculating the condition number at each. Fig. 5 shows, at top, a plot of condition number versus position of the interface within the cell; black lines represent the grid-aligned planar geometries for a variety of coefficient ratios $r = \kappa^-/\kappa^+$. Note that for the case where the plane intersects a cell face, producing a zero volume cell in one material, there is no associated increase in the condition number.

A reasonable next step is to use non-aligned planar geometries, following Oevermann and Klein [12]. A plane whose normal is inclined at an angle of either 45° or 22.5° to the horizontal is moved across a single cell. As above, condition numbers are calculated for 200 positions of the interface. We found that a simple volume fraction weighting of the equations does not suffice to produce good conditioning. Rather, the smallest eigenvalues scale with the surface area of the cell and can cause unbounded condition numbers. In order to address this issue, we use a modified scaling, which we give here for the case of the Poisson equation:

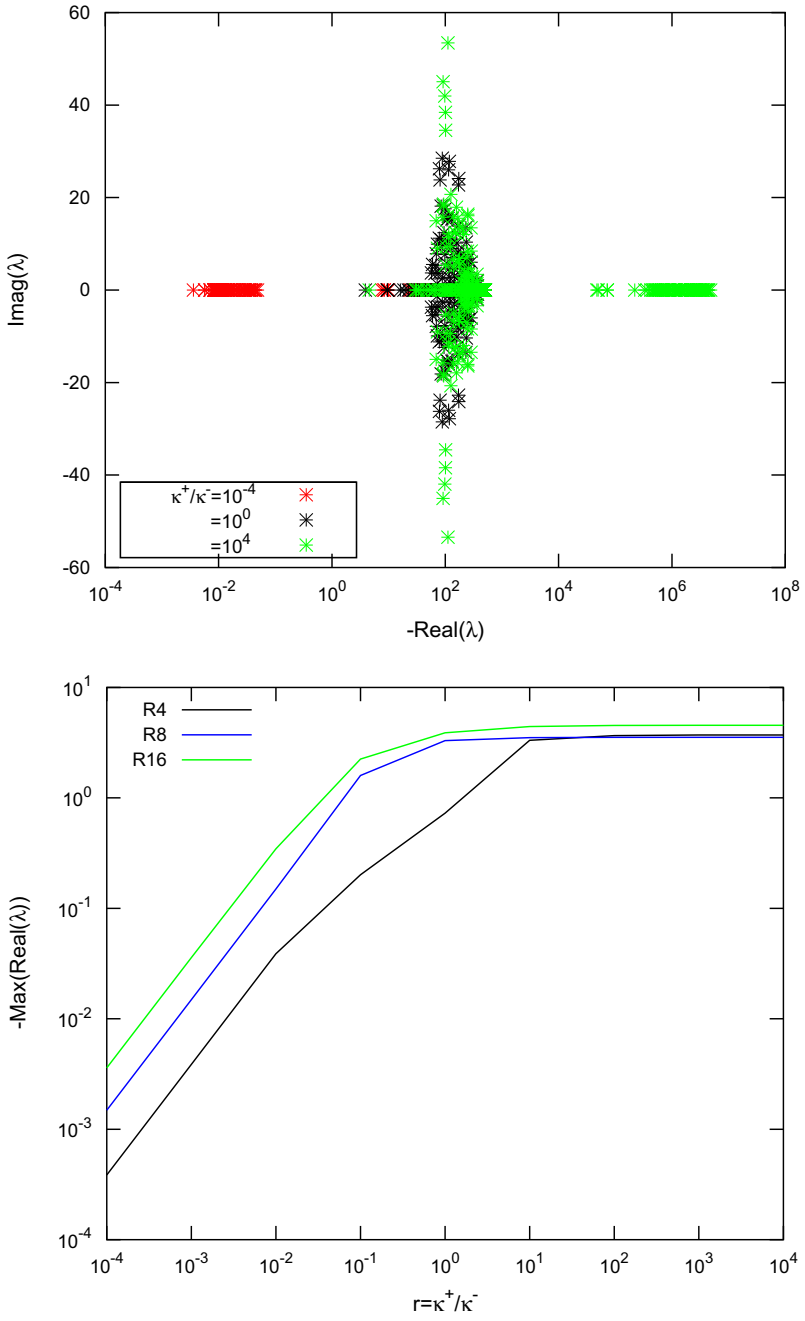

Fig. 6. Eigenvalues of the Laplacian for rhodonea geometry. Top: Scatter plot of the real and imaginary parts of the eigenvalues for rhodonea geometry at R_{16}. Coefficient ratios of $r = 10^{-4}$ (red), 10^0 (black), and 10^4 (green) are shown. All eigenvalues are in the left half-plane, indicating stability of the TGA algorithm. Bottom: Largest real component of the eigenvalue of the Laplacian plotted versus coefficient ratio. Results for three different resolutions – R_4 (black line), R_8 (blue line) and R_{16} (green line) – indicate a weak dependence of the largest eigenvalue on the resolution.

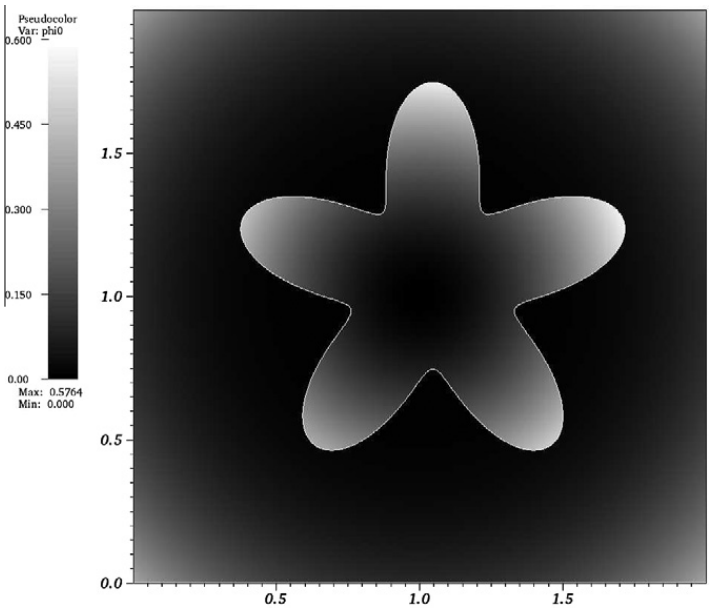

Fig. 7. Rhodonea geometry, outlined in white, showing the solution to Poisson's equation inside and outside, with $\kappa^- = 10^0$ and $\kappa^+ = 10^1$.

$$\frac{v_{\mathbf{i}}}{\sum_{s,\pm=+,-}\alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s}+\alpha^B_{\mathbf{i}}}(\mathbb{L}\varphi)_{\mathbf{i}} = \frac{v_{\mathbf{i}}}{\sum_{s,\pm=+,-}\alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_s}+\alpha^B_{\mathbf{i}}}\rho_{\mathbf{i}}. \tag{23}$$

The original volume fraction scaling is now divided by the total surface area fraction of the sub-volume. In practice, we find that this scaling has little effect on the either the asymptotic convergence rate of point relaxation, or the overall convergence rate of geometric multigrid. This is due to the diagonal scaling performed in relaxation schemes. We also note that for solving the Helmholtz equation, Eq. (3), this scaling will generally be unnecessary due to the effect of the α^p term in that equation on the diagonal entries.

Using the modified scaling, we calculated condition numbers for coefficient ratios in the range $10^0 - 10^4$, plotted versus interface position as blue and green lines in the top plot of Fig. 5. Note that, once again, pathological cases are handled correctly. When the material boundary intersects a cell corner, producing a cell with volume fraction $v = 0$, there is little effect on the condition number. We find the dependence of condition number on volume fraction is very similar to that of Oevermann et al. Also note that, in using two hundred positions for the interface, we cover a range in volume fraction down to $\sim 10^{-5}$. Through further tests using the non-aligned geometry, we show that the condition number exhibits identical behavior down to volume fractions of 10^{-14}, as shown in the bottom plot of Fig. 5.

The stability of our scheme for solution of the heat equation depends on the eigenvalues of the Laplacian residing in the left half-plane, owing to the L_0-stability of the TGA time integrator. In all planar geometry cases above this was observed to be the case. We have also extended the eigenvalue analysis to a more complex geometry. The rhodonea geometry, pictured in Fig. 7, is also used for solution error testing of the scheme in Section 6.1. Here, we use it to show how the eigenvalue spectrum varies with coefficient ratio. In the top plot of Fig. 6, the spectrum is plotted for three values of the ratio, $r = 10^{-4}$, 10^0, and 10^4, at the very coarse resolution of four cells per linear dimension. The real part of the eigenvalues remain negative, scaling linearly with the coefficient ratio. Furthermore, by observing the dependence of the real part of the eigenvalues on coefficient ratio at multiple resolutions, as in the bottom plot of Fig. 6, we show that the increase of $\max(\mathrm{Re}(\lambda))$ with resolution is relatively weak. Together, these results indicate that the multigrid solver used here will, even in the presence of large coefficient ratios, produce well conditioned problems despite under-resolving complex geometries.

5. Time discretization and algorithm outline

We follow [15] in using a second order in time Runge–Kutta solver [20] in solving the heat equation. The time discretization is

$$T^{n+1} = (\mathbb{I} - \mu_1\mathbb{L})^{-1}(\mathbb{I} - \mu_2\mathbb{L})^{-1}\left[(\mathbb{I} + \mu_3\mathbb{L})T^n + (\mathbb{I} + \mu_4\mathbb{L})f^{n+\frac{1}{2}}\right] \tag{24}$$

with the μ parameters chosen so as to simultaneously achieve second order accuracy and L_0 stability.

The heat equation algorithm proceeds as follows, omitting details of the multigrid operations:

1. Grid generation
   (a) Calculation of geometric quantities
   (b) Calculation of stencil weights

2. Heat equation update step solve
   (a) Calculate source term at half-timestep, $f^{n+\frac{1}{2}}$
   (b) Apply μ_4 operator
   (c) Apply μ_3 operator
   (d) Solve elliptic problem for μ_2 operator
      i. Calculate boundary fluxes
      ii. Single relaxation step in each material
      iii. Repeat (i) and (ii) to convergence
   (e) Solve elliptic problem for μ_1 operator (repeating steps i–iii above)

Elliptic solver operations form the bulk of the computational work. In order to speed convergence, we plan to investigate the need for performing step (i) before every relaxation step. We leave this investigation for future work, however.

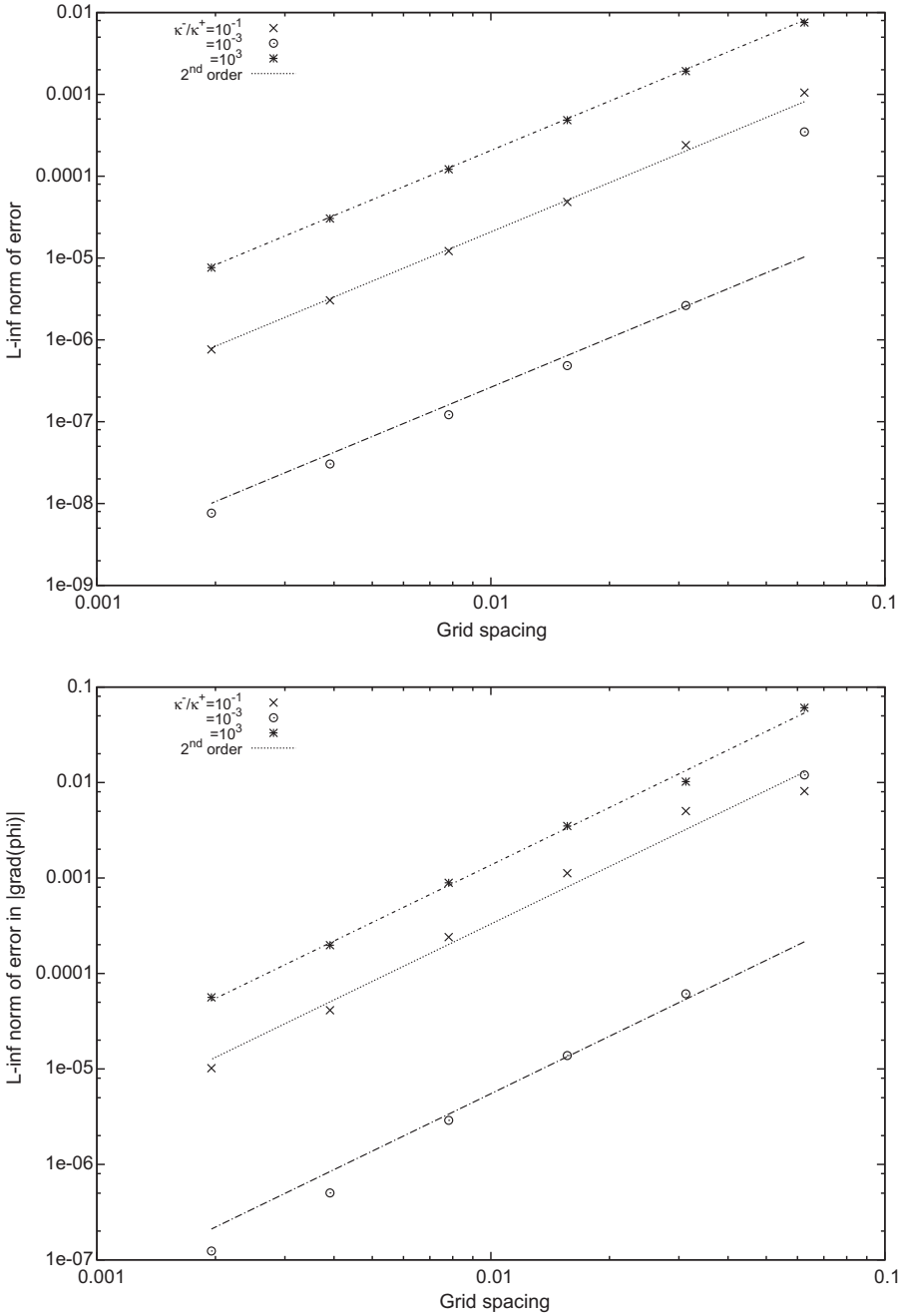

Fig. 8. Solution error convergence for the rhodonea example, following Oevermann et al. Top: L_∞ norm of the error in the solution. Note that L_1 and L_2 norms also converge at second order. Bottom: L_∞ norm of the error in the magnitude of the gradient of the solution. Here again, the L_1 and L_2 norms also converges at second order.

6. Numerical results

6.1. 2-D tests of solution error for Poisson's equation on fixed and adaptive meshes

As a 2-D test of the elliptic solver with a non-trivial geometry, we solve the Poisson equation $\kappa\Delta\varphi = \rho$ on a single grid, with two materials whose boundary is a rhodonea curve, originally due to Li [21] and also used by Oevermann and Klein [12]. The equation for this curve in polar coordinates is

$$r = r_0 + r_1 \sin(\omega\theta), \tag{25}$$

where r_0 and r_1 are the inner- and outer-radii, respectively, and ω the number of lobes of the rhodonea (see Fig. 7). Our tests use $r_0 = 0.5$, $r_1 = 0.1$, and $\omega = 5$, as in previous work. Note that the rhodonea is slightly offset from the origin of the domain,

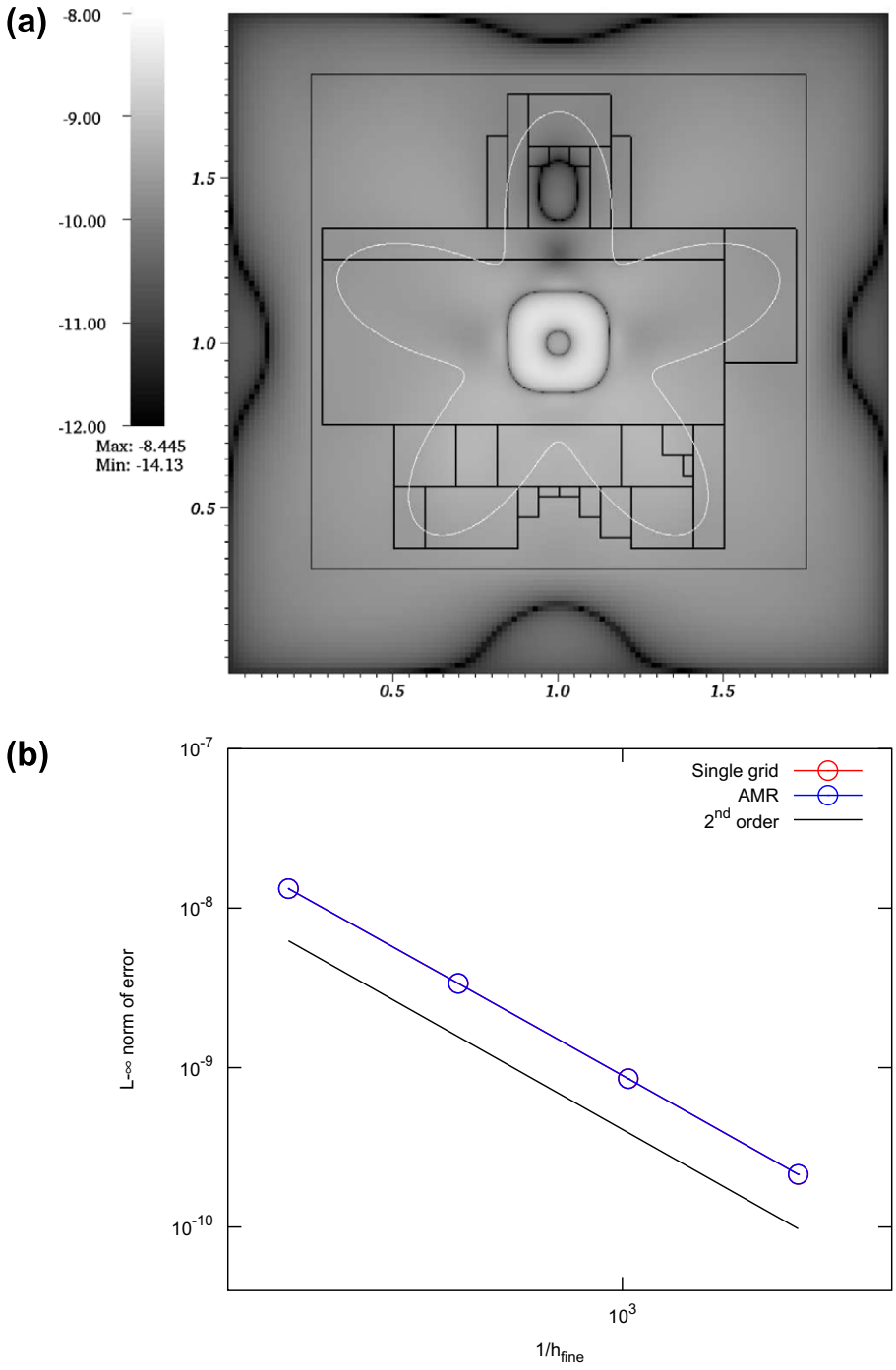

Fig. 9. Top: Plot of the log of the magnitude of the error for the AMR test with rhodonea geometry. The rectangle with a thin bounding line shows the extent of the single level two grid. The rectangles with thick bounding lines show the extent of the level three grids. Bottom: L_∞ norm of the error for three level AMR (red), and single level (blue) calculations versus the inverse of the finest grid spacing. The black line is a reference showing second order convergence.

being centered at $x = y = 0.2/\sqrt{20}$. We label quantities on the interior of the rhodonea, the inclusion region, with a minus, such as κ^-. Similarly for the exterior, using κ^+.

This example is a good test of the EB methodology due to the high curvature near the inner radius. For the purposes of comparison, we reproduce Example 2 of Oevermann et al. Here, as there, a range of coefficient ratios, $\kappa^-/\kappa^+ = 10^{-3}$, 10^{-1}, and 10^3 was run. The exact solution is given by

$$\varphi^{ex,+} = \frac{r^4 + c_0 \log(2r)}{\kappa^+}, \tag{26}$$

$$\varphi^{ex,-} = \frac{r^2}{\kappa^-}, \tag{27}$$

from which we calculate the solution error $e_{\mathbf{i}} = \kappa_{\mathbf{i}}(\varphi_{\mathbf{i}} - \varphi^{ex}_{\mathbf{i}})$. Plots of the norm of the solution error and the error in the magnitude of the gradient of the solution (again weighted by volume fraction) are shown in Fig. 8. Both are second order accurate in 1-, 2-, and max-norm. Especially noteworthy is the second order convergence of the gradient error, which compares favorably with [12].

A test with even more challenging geometries have also been run, though we omit details for the sake of brevity. A cardioid, which is difficult due to a lack of Lipschitz continuity, was used in a previous investigation [5]. We find second-order convergence for this geometry, as well.

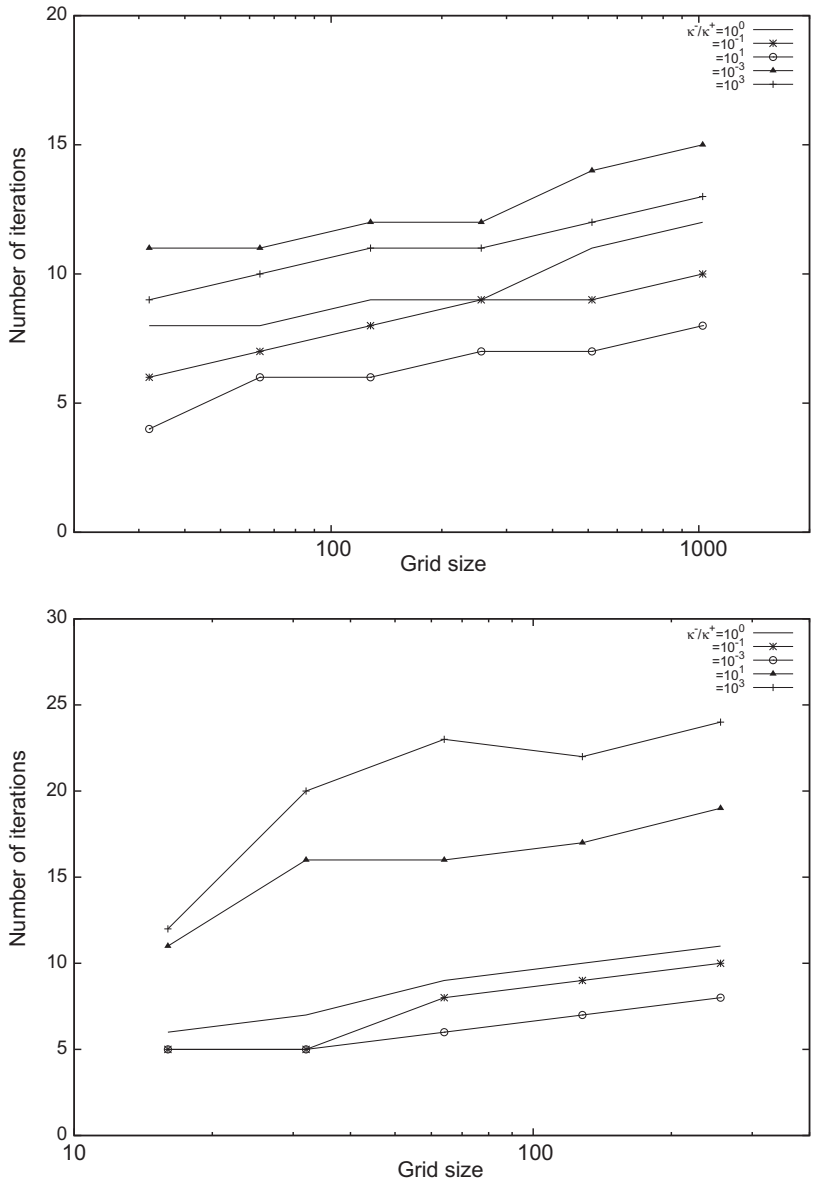

Fig. 10. Number of iterations required for convergence versus grid size for an ellipsoid geometry. Coefficient ratios in the range 10^{-3} to 10^3 are plotted. Top is the 2-D result, and the bottom 3-D. In general, efficiency shows a weak dependence on grid size. At higher coefficient ratios, however, the dependence is stronger.

The second test uses the same rhodonea geometry, and targets the adaptive mesh refinement (AMR) capabilities of Chombo in the multi-material context. It involves a quadratic source in the center of the rhodonea

$$\rho = \kappa^p \left(a - a^2\right)^4, \tag{28}$$

$$\varphi^{ex} = \begin{cases} r_0^2 \left[a^6 \left(c_4 a^4 + c_3 a^3 + c_2 a^2 + c_1 a + c_0\right) \right. \\ \quad -(c_4 + c_3 + c_2 + c_1 + c_0) \\ \quad \left. +c \log(r_0)\right] & \text{if } a < 1, \\ r_0^2 (c \log(r)) & \text{otherwise}, \end{cases} \tag{29}$$

a problem adapted from [15]. Here $a = r/r_0$ and we use a value of $r_0 = 0.2$. The constants are $c = \frac{1}{10} - \frac{4}{9} + \frac{6}{8} - \frac{4}{7} + \frac{1}{6}$, $c_0 = \frac{1}{36}$, $c_1 = -\frac{4}{49}$, $c_2 = \frac{6}{64}$, $c_3 = -\frac{4}{81}$, and $c_4 = \frac{1}{100}$. We use a modest ratio of $\kappa^-/\kappa^+ = 1/10$. The matching conditions at the boundary are homogeneous.

Our strategy is to use a single level run to verify the results from an AMR run, in this case with two levels of refinement. With effective gridding, normed errors in the AMR case should be very similar to those of the single grid case with equivalent resolution. We found it simplest to cover the inner ($r < r_0$) region of the rhodonea entirely with the level two and three grids in order to ensure that the region with the largest solution gradient is at the highest resolution. Level two and level three grids also cover the boundary, which is the other large error region. (An alternate strategy for controlling grid placement, Richardson error extrapolation, is more algorithmically complex, and is left for future work.)

Successive AMR runs increase the base level resolution while keeping the number of levels fixed. By comparing the error from these runs with that of single grid runs with equivalent effective resolution, we are able to verify that AMR is not introducing spurious errors. A plot of the error for a R_{128} base grid (effective resolution R_{512}) is shown in Fig. 9, at top. At bottom is shown the L_∞ norm of the error versus the effective (finest) grid size. Both single grid and AMR results are second order. Just as importantly, the magnitude of the normed AMR errors are only marginally higher than those of the single grid calculations.

6.2. 2-D and 3-D tests of multigrid solver efficiency

We have tested the performance of the multigrid solver for a range of grid sizes and coefficient ratios. An ellipsoid geometry in two and three dimensions is used. There is no refinement of the domain using AMR. We measured the number of

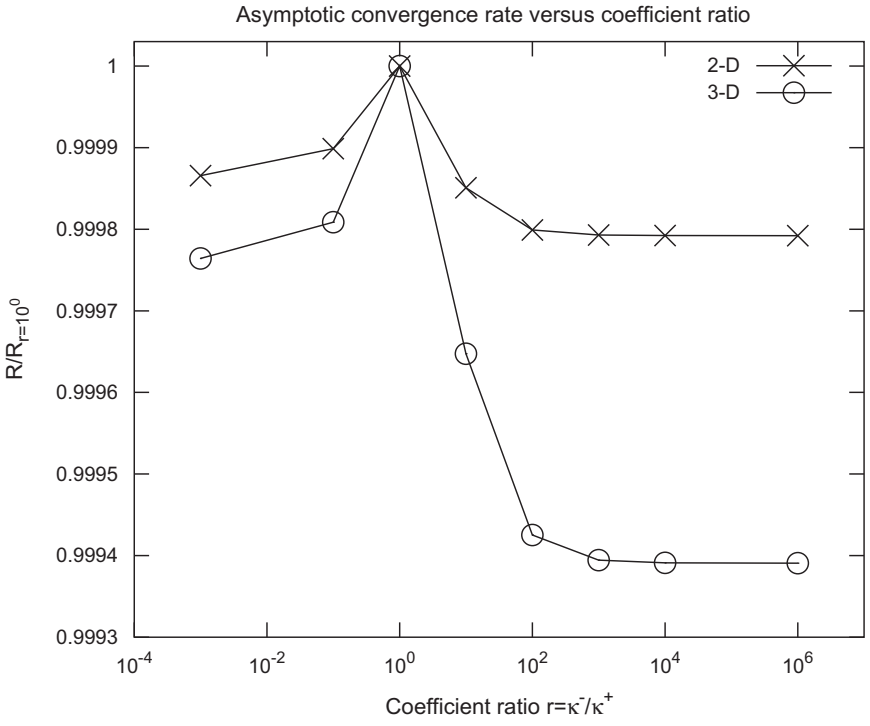

Fig. 11. Asymptotic convergence rate of point relaxation versus coefficient ratio for an ellipsoid geometry. Convergence rates are plotted scaled to the rate at a coefficient ratio of unity. Higher ratios correspond to a more nearly homogeneous Neumann boundary condition on the interior of the interface. While convergence initially slows as the ratio increases above unity, the effect saturates at ratios above 10^3.

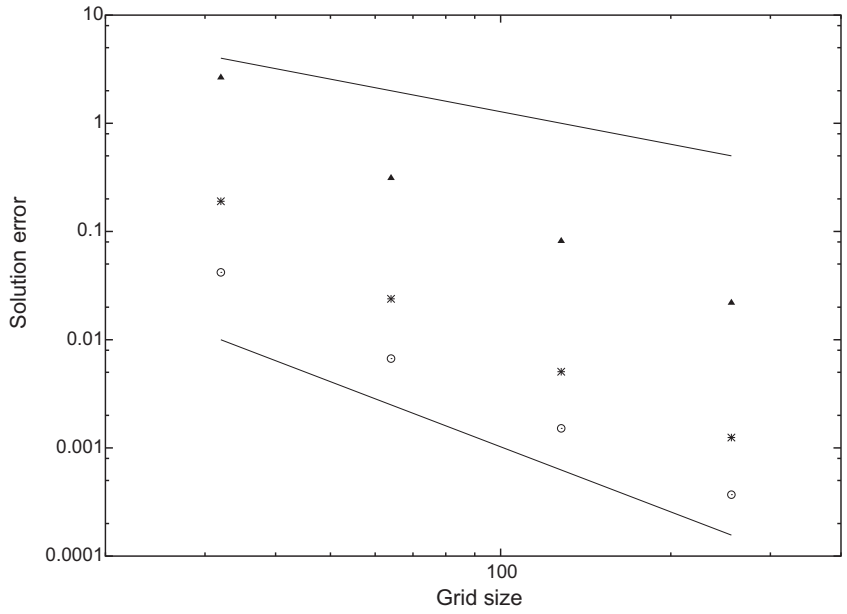

Fig. 12. Convergence of the solution error for the Poisson equation in three dimensions with a spherical boundary between materials. The coefficient in the inclusion is $\kappa^- = 1$, and on the exterior it is $\kappa^+ = 10$. The source and domain boundary conditions are chosen such that the jump conditions are homogeneous. Norms of the solution error are plotted, with triangles denoting the L_∞ norm, stars L_1, and circles L_2 norm. The top line is a reference showing first-order convergence, and the bottom line showing second-order convergence.

multigrid iterations required to reduce the L_2 norm of the error by ten orders of magnitude. Optimally, the iteration count is independent of the grid size. In Fig. 10 we plot the iteration count versus grid size for resolutions in the range R_{32} to R_{1024} in the 2-D case, and R_{16} to R_{256} in the 3-D case. The figure shows results for coefficient ratios from $r = \kappa^-/\kappa^+ = 10^{-3}$ to 10^3. Though the iteration count increases somewhat with increasing grid size, there is a stronger dependence on coefficient ratio, particularly when the latter is greater than one. Solvability issues associated with large κ inclusions is well documented in the literature; see [12,8]. We have explored this issue further by measuring the asymptotic convergence rate for point relaxation over a larger range in coefficient ratios, at a single resolution – R_{128} in 2-D and R_{64} in 3-D. As expected, Fig. 11 shows that the asymptotic rate decreases as r increases above unity. However, the rate reaches an asymptotic value at a ratio of 10^2 (10^3 in 3-D). We also observe this behavior in the multigrid solver, which performs better than might naively be expected as the coefficient ratio increases above 10^2.

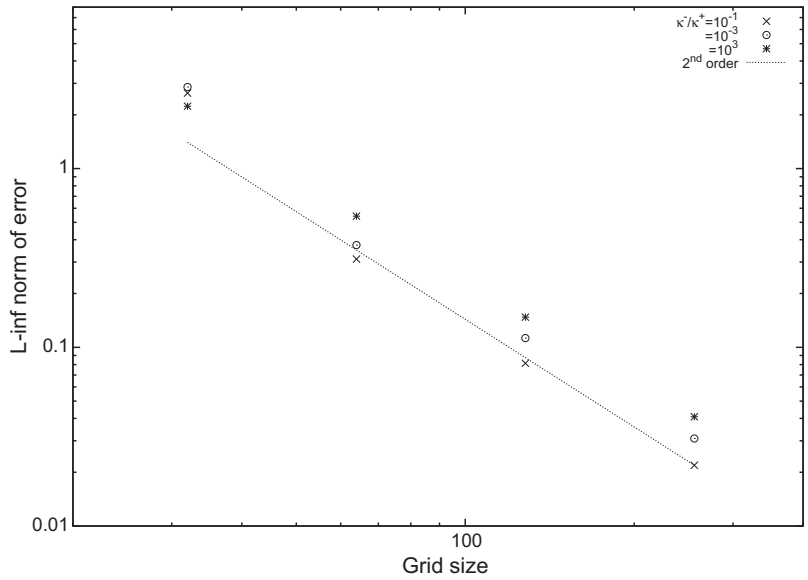

Fig. 13. Convergence of the L_∞ norm of the solution error for the Poisson equation in three dimensions, across a range in coefficient ratios. Pluses show a ratio κ^-/κ^+ of one. Open circles show the case with high κ on the outside, while filled circles show high κ on the inside. The line is a reference showing second-order convergence.

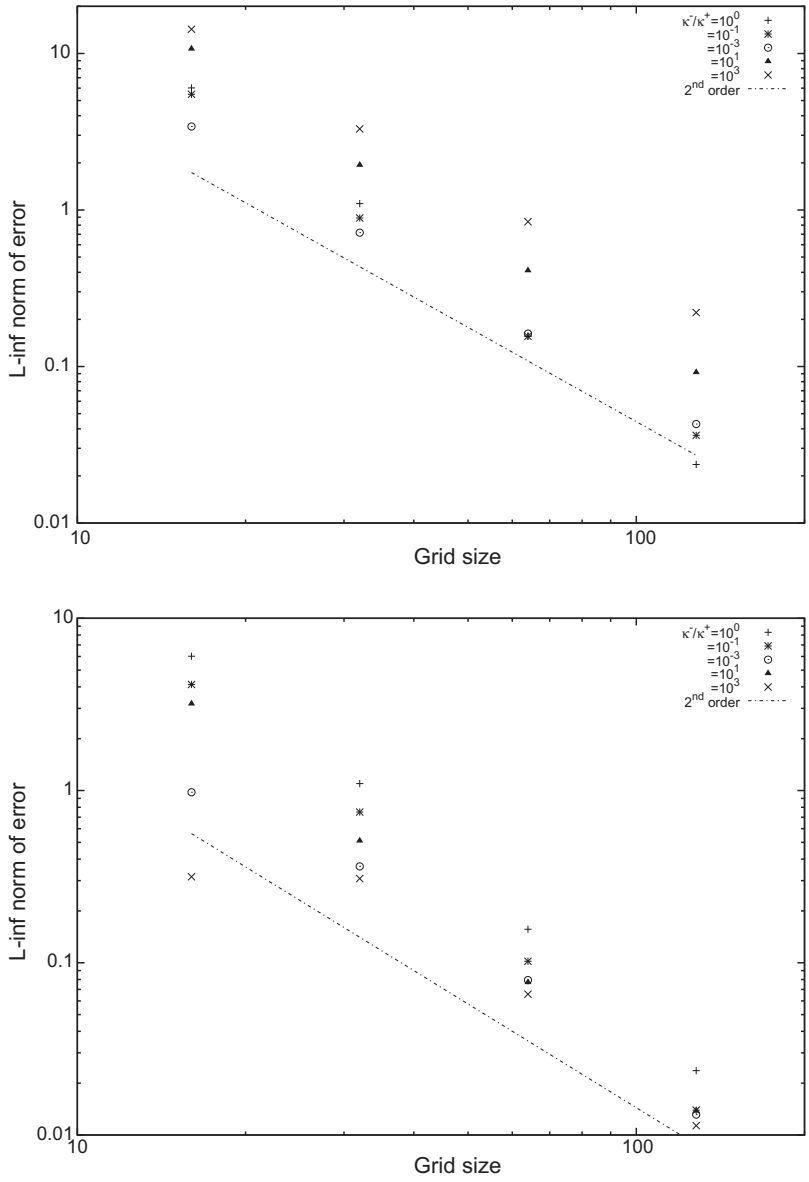

Fig. 14. Convergence of the L_∞ norm of the solution error for the heat equation. Top: homogeneous Dirichlet embedded boundary. Bottom: Homogeneous Neumann embedded boundary. Results span a range in thermal diffusivity ratios. Black pluses denote a ratio of one. Two cases have larger coefficients outside the sphere, 10^{-1} (∗) and 10^{-3} (⊙). Two cases have larger coefficients inside the sphere, 10^{1} (▲) and 10^{3} (×). The line is a reference indicating second-order convergence.

6.3. 3-D test of solution error for Poisson's equation

Our test of solution error convergence in three dimensions uses a sphere of radius $R = 0.392$, and a source and exact solution proportional to an eigenfunction of the Laplacian in spherical coordinates:

$$\rho(r) = -k^2 \frac{\sin(kr)}{r}, \tag{30}$$

$$\varphi^{ex,p}(r) = \frac{1}{\kappa^p} \frac{\sin(kr)}{r} + c^p. \tag{31}$$

The constant c^p is chosen such that $[\varphi^{ex}] = 0$, giving homogeneous matching conditions at the material boundary. Results for solution error are shown in Fig. 12 and Fig. 13. Convergence is second order, independent of the norm and coefficient ratio chosen.

6.4. 3-D tests of solution error for the heat equation

In order to test the accuracy of our multi-material method for the heat equation, we once again use the method of manufactured solutions. For the desired exact solution φ^{ex} of Eq. (2) in material p we again choose for the spatial component the

Table 1
Table of parameters used in fuel bundle test.

Name	Value	Source
Horizontal domain size	12	None
Pin radius	0.800	[22]
Wire radius	0.103	[22]
Pin-wire separation	0.478	None
Pin length	3.000	None
Wire separation	20.0	[22]
Pin + wire thermal diffusivity	0.9007	[11]
Moderator density	0.852	[23]
Moderator specific heat	1.2768×10^7	[23]
Moderator thermal diffusivity	0.6251	[23]

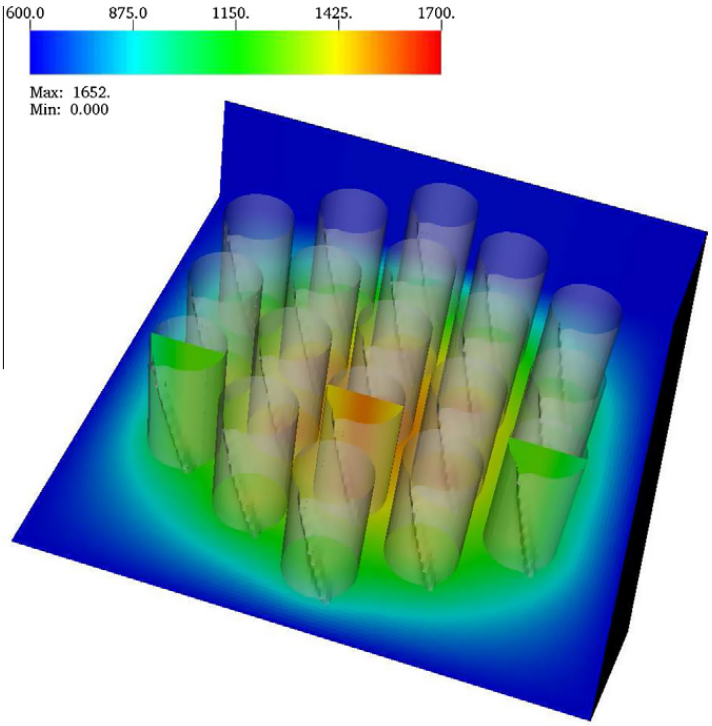

Fig. 15. Temperature distribution at steady state for a 19-pin fuel bundle. The interfaces between fuel pin and moderator are represented by semi-transparent surfaces. A slice of the solution inside the third row of pins is shown. The solution on the exterior is shown in a slice at the bottom of the domain.

trigonometric eigenfunction of the spherical Laplacian from Section 6.3. Time dependence is added via an exponential decay term. The source term f^p is chosen such the heat equation is satisfied. The exact solution and source are of the form

$$\varphi^{ex,p} = c_0^p \frac{\sin(kr)}{r} e^{-\gamma t} + c_1^p, \tag{32}$$

$$f^p = c_0^p(\kappa^p k^2 - \gamma)\frac{\sin(kr)}{r} e^{-\gamma t}. \tag{33}$$

In all cases we choose $k = 2\pi$. The interface between materials is also a sphere of radius $R = 0.392$.

Due to the difficulty of choosing solutions such that the matching conditions Eq. (1) are simultaneously zero, we test each in turn. In other words, we perform one test with a homogeneous Dirichlet boundary, where the solution in each phase is the same while the jump in the flux varies in both time and space. The other test uses a homogeneous Neumann boundary, where $\kappa^+\varphi^+ = \kappa^-\varphi^-$ so that the flux is continuous across the boundary.

For the homogeneous Dirichlet case we use $c_0^- = c_0^+ = 10$, $c_1^- = c_1^+ = 100$, and $\gamma = \max(\kappa^+, \kappa^-)$. All simulations were run for the same number of timesteps, to a final time such that $\gamma t_{final} = 0.2$. For the homogeneous Neumann case we use $c_0^\pm = 10/\kappa^\pm$, with all other parameters the same as in the homogeneous Dirichlet case. Results for solution error convergence are shown in Fig. 14, and confirm second order accuracy for both cases.

6.5. Conjugate heat transfer in a nuclear reactor fuel bundle

Finally, we test the multi-material method on a more realistic example, inspired by nuclear reactor core simulation. The geometry consists of a set of nineteen coaxial reactor fuel pins. Each pin is wrapped in a helical wire. Outside of the fuel pin is a moderator, in this case a stationary material with thermal properties equivalent to liquid sodium. The geometric parameters used closely follow [22], and are listed in Table 1.[1] All parameters are in CGS units. The source was a Gaussian that falls off with distance from the pin axis,

$$\tilde{f} = \frac{f}{\rho c_P} = \tilde{F} \exp\left(-r_{pin}^2/s^2\right). \tag{34}$$

The source strength $\tilde{F} = 2.5 \times 10^3$, and the width parameter $s = 1/4$. This choice provides a steady-state power density of approximately 10^2 W/m^2. Initially, the temperature is uniform at 600 K. The top and bottom domain boundaries were adiabatic ($\frac{\partial T}{\partial z} = 0$), while the others were kept constant at 600 K. The simulation was run using timesteps of $dt = 0.1$ to a final time of $t = 51.0$, at which point the heat flux through the boundary had reached a steady state. Fig. 15 shows the temperature distribution at this final time.

7. Conclusions

We have presented an approach to solving elliptic and parabolic equations using Cartesian grid embedded boundary methods which is second order accurate and computationally efficient. The former was achieved using the usual five-point stencil in cells not intersected by the boundary, and a quadratic approximation to the state at the interface otherwise. The discretization results in a non-symmetric matrix whose eigenvalues satisfy the stability criterion for the time integration scheme for all tested geometries. Stability of the spatial discretization, as measured by the condition number of the operator, was verified using numerical experiments using both simple and complex geometries, small (10^{-14}) volume fraction cells, and a range of resolutions.

In order to efficiently solve elliptic equations, we use geometric multigrid. This necessitated treatment of a number of special cases that arise when the geometry is under-resolved. The solver was found to be efficient for all geometries tested. Moreover, the method was also shown not to suffer from condition-based solver convergence issues. It remains stable for ratios up to 10^6. Moreover, we find that the rate of convergence improved at very high material contrast ratios, a good indicator that it will remain stable and efficient beyond those tested.

The method was tested on a complex nuclear reactor fuel bundle geometry that underscored the flexibility of the grid generation approach. This, along with the parallel computing and AMR capabilities leveraged from Chombo, are essential to approaching large scale problems. In the future, higher order extensions of the underlying method will be explored using both the existing capabilities for high order geometric representations [18] and by extending the stencil approximations to higher order.

Parallel scaling of the algorithm will also be explored in future work. Preliminary results suggest that the algorithm will be scalable to thousands of processors. This is due to the similarities with the single material algorithm, whose scaling was explored by Trebotich et al. [24].

Acknowledgments

We thank Brian Van Straalen for help with the Chombo software library. We also thank the reviewers, whose thoughtful comments helped greatly improve upon the initial version of this paper. This work was supported by the Department of Energy under contract number DE-AC02-05-CH11231.

References

[1] G.H. Shortley, R. Weller, The numerical solution of Laplace's equation, Journal of Applied Physics 9 (5) (1938) 334–348.
[2] Charles S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.
[3] Randall J. LeVeque, Zhilin Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM Journal on Numerical Analysis 31 (4) (1994) 1019–1044.
[4] Xu-Dong Liu, Ronald P. Fedkiw, Myungjoo Kang, A boundary condition capturing method for Poisson's equation on irregular domains, Journal of Computational Physics 160 (1) (2000) 151–178.
[5] Songming Hou, Xu-Dong Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, Journal of Computational Physics 202 (2) (2005) 411–445.
[6] Tianbing Chen, John Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, Journal of Computational Physics 227 (16) (2008) 7503–7542.
[7] Anita Mayo, The rapid evaluation of volume integrals of potential theory on general regions, Journal of Computational Physics 100 (2) (1992) 236–245.
[8] Leslie Greengard, June-Yub Lee, Electrostatics and heat conduction in high contrast composite materials, Journal of Computational Physics 211 (1) (2006) 64–76.

[1] The value of the thermal diffusivity in the pins was inferred. Specifically, the sodium (moderator) thermal conductivity listed in [23] was multiplied by the ratio of pin to moderator conductivities used in [11].

[9] Mark S. Day, John B. Bell, Numerical simulation of laminar reacting flows with complex chemistry, Combustion Theory and Modelling 4 (4) (2000) 535–556.
[10] F. Nicoud, Conservative high-order finite-difference schemes for low-Mach number flows, Journal of Computational Physics 158 (1) (2000) 71–97.
[11] William D. Henshaw, Kyle K. Chand, A composite grid solver for conjugate heat transfer in fluid–structure systems, Journal of Computational Physics 228 (10) (2009) 3708–3741.
[12] M. Oevermann, R. Klein, A cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces, Journal of Computational Physics 219 (2) (2006) 749–769.
[13] M. Oevermann, C. Scharfenberg, R. Klein, A sharp interface finite volume method for elliptic equations on Cartesian grids, Journal of Computational Physics 228 (14) (2009) 5184–5206.
[14] Hans Johansen, Phillip Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1) (1998) 60–85.
[15] Peter McCorquodale, Phillip Colella, Hans Johansen, A Cartesian grid embedded boundary method for the heat equation on irregular domains, Journal of Computational Physics 173 (2) (2001) 620–635.
[16] Peter Schwartz, Michael Barad, Phillip Colella, Terry Ligocki, A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions, Journal of Computational Physics 211 (2) (2006) 531–550.
[17] Michael J. Aftosmis, Marsha J. Berger, John E. Melton, Robust and efficient Cartesian mesh generation for component-based geometry, AIAA Journal 36 (6) (1998) 952–960.
[18] T.J. Ligocki, P.O. Schwartz, J. Percelay, P. Colella, Embedded boundary grid generation using the divergence theorem, implicit functions, and constructive solid geometry, Journal of Physics: Conference Series (125) (2008) 012080 (5 pp).
[19] Zhilin Li, Kazufumi Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM Journal on Scientific Computing 23 (1) (2001) 339–361.
[20] E.H. Twizell, A.B. Gumel, M.A. Arigu, Second-order, L0-stable methods for the heat equation with time-dependent boundary conditions, Advances in Computational Mathematics 6 (1) (1996) 333–352.
[21] Zhilin Li, A fast iterative algorithm for elliptic interface problems, SIAM Journal on Numerical Analysis 35 (1) (1998) 230–254.
[22] Kurt D. Hamman, Ray A. Berry, A CFD process for fast reactor fuel assemblies, in: Experiments and CFD Code Applications to Nuclear Reactor Safety (XCFD4NRS 2008), 2008.
[23] J.K. Fink, L. Leibowitz, Thermodynamic and transport properties of sodium liquid and vapor, Technical Report ANL/RE-95/2, Argonne National Laboratory, 1995.
[24] D. Trebotich, B.V. Straalen, D. Graves, P. Colella, Performance of embedded boundary methods for cfd with complex geometry, Journal of Physics: Conference Series 125 (1) (2008) 012083.