Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 209 (2005) 1–18
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fourth-order accurate local refinement method for Poisson's equation ☆,☆☆

Michael Barad [a,*], Phillip Colella [b]

[a] Department of Civil and Environmental Engineering, Lawrence Berkeley National Laboratory, University of California, Davis, Cyclotron Road Mail Stop 50A1148, Berkeley, CA 95616, United States
[b] Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

Received 23 August 2004; received in revised form 14 February 2005; accepted 15 February 2005
Available online 13 May 2005

Abstract

We present a block-structured local refinement method for computing solutions to Poisson's equation in two and three dimensions. It is based on a conservative, finite-volume formulation of the classical Mehrstellen methods. This is combined with finite volume local refinement discretizations to obtain a method that is fourth-order accurate in solution error, and with easily verifiable solvability conditions for Neumann and periodic boundary conditions.

PACS: 02.60.Lj; 02.70.Bf; 41.05.+e; 41.20.Cv

Keywords: Poisson equation; Mesh refinement; Multigrid methods

1. Introduction

In this paper, we present a fourth-order accurate numerical method for solving Poisson's equation

$$\Delta\phi = \rho \tag{1}$$

☆ This work was supported by the Computational Science Graduate Fellowship program of the Department of Energy, under Grant No. DE-FG02-97ER25308.
☆☆ Research supported by the Applied Mathematics Program of the DOE Office of Mathematics, Information, and Computational Sciences under the US Department of Energy under Contract No. DE-AC03-76SF00098.
* Corresponding author.
E-mail address: mfbarad@ucdavis.edu (M. Barad).
URL: edl.engr.ucdavis.edu (M. Barad).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.02.027

in two or three space dimensions in a rectangular domain with either Dirichlet, Neumann, or periodic boundary conditions. Our approach uses a conservative, finite-volume, block-structured local refinement discretization that generalizes the classical Mehrstellen methods [1]. Previously, nodal-point Mehrstellen discretizations of Poisson's equation have been combined with local refinement [2]. Unlike those algorithms, the present algorithm preserves discrete conservation form, thus making it compatible to solve coupled hyperbolic-elliptic problems using finite volume approximations for hyperbolic problems on locally refined grids, along the lines of those in [3]. In addition, the present method has an obvious and easily verified discretization of the solvability condition for Neumann or periodic boundary conditions for the case of a general hierarchy of locally refined grids. Such a condition is not known for the nodal-point method cited above.

Our approach proceeds along the lines of that described in [4–6]. The single-level operator is expressed in terms of a difference of fluxes on faces, with ghost-cell data interpolated using a combination of coarse and fine grid values. Neumann matching conditions at the coarse–fine interface are enforced by refluxing [3], i.e. using the average of the fluxes at the next finer level to compute the flux into coarse cells adjacent to a coarse–fine interface. In one important respect, we depart from the approach taken in the second-order algorithm, in the way we compute ghost-cell values. We compute higher-order coarsenings of the fine grid values onto the coarse grid so as to interpolate using values from a fixed stencil. In contrast, the methods cited above use stencils that depend on the local distribution of coarse and fine grids, so as to maintain the required level of accuracy while not using coarse-cell values that are covered by finer cells. In order to reduce the complexity of our coarsening process, we require the ratio between mesh spacings at successive refinement levels to be an even number no less than four. The resulting method has a truncation error that is fourth-order in the mesh spacing at all cells except those adjacent to the boundary between refinement levels. In the latter case, the truncation error is third-order in the mesh spacing. In numerical experiments, we observe solution errors that are fourth-order in the mesh spacing, uniformly in space, consistent with the modified equation analysis in [6,7]. We also observe some superconvergence in this method. In particular, we can use a fourth-order accurate interpolant for the ghost cell values, and still obtain fourth-order accuracy in the solution, even though the truncation error is second-order on a set of codimension one.

2. Notation and calculus identities

The underlying discretization of $\boldsymbol{D}$-dimensional space is given as points $(i_0, \ldots, i_{\mathbf{D}-1}) = \boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$. The problem domain is discretized using a grid $\Gamma \subset \mathbb{Z}^{\mathbf{D}}$ that is a bounded subset of the lattice. Γ is used to represent a cell-centered discretization of the continuous spatial domain into a collection of control volumes: $\boldsymbol{i} \in \Gamma$ represents a region of space,

$$V_{\boldsymbol{i}} = [\boldsymbol{i}h, (\boldsymbol{i} + \boldsymbol{u})h], \tag{2}$$

where h is the mesh spacing, and $\boldsymbol{u} \in \mathbb{Z}^{\mathbf{D}}$ is the vector whose components are all equal to one. We can also define face-centered discretizations of space based on those control volumes: $\Gamma^{e^d} = \{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d : \boldsymbol{i} \in \Gamma\}$, where $\boldsymbol{e}^d$ is the unit vector in the d direction. Γ^{e^d} is the discrete set that indexes the faces of the cells in Γ whose normals are $\boldsymbol{e}^d$

$$A_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = [(\boldsymbol{i} + \boldsymbol{e}^d)h, (\boldsymbol{i} + \boldsymbol{u})h], \boldsymbol{i} + \tfrac{1}{2}\boldsymbol{e}^d \in \Gamma^{e^d}. \tag{3}$$

We define cell-centered discrete variables on Γ

$$\phi : \Gamma \to \mathbb{R}^m.$$

We denote by $\phi_{\boldsymbol{i}} \in \mathbb{R}^m$ the value of ϕ at cell $\boldsymbol{i} \in \Gamma$. We can also define face-centered vector fields on Γ

$$\vec{F} = (F_0, \ldots, F_{\mathbf{D}-1}), \quad F_d : \Gamma^{e^d} \to \mathbb{R}^m.$$

We can define a discretized divergence operator on such a vector field.

$$(D \cdot \vec{F})_{\boldsymbol{i}} = \frac{1}{h} \sum_{d=0}^{\mathbf{D}-1} (F_{d,\boldsymbol{i}+\frac{1}{2}e^d} - F_{d,\boldsymbol{i}-\frac{1}{2}e^d}), \quad \boldsymbol{i} \in \Gamma. \tag{4}$$

In order to obtain fourth-order accurate finite volume methods, it is necessary to distinguish between point values at cell and face centers, and the averages over cells and faces. If $\psi = \psi(\vec{x})$, then we denote the point values at cell centers by $\psi_{\boldsymbol{i}} = \psi((\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h)$ and at face centers by $\psi_{\boldsymbol{i}+\frac{1}{2}e^d} = \psi((\boldsymbol{i} + \frac{1}{2}(\boldsymbol{u} + e^d))h)$, and the corresponding averages by

$$\text{cell average: } \langle \psi \rangle_{\boldsymbol{i}} = \frac{1}{h^D} \int_{V_i} \psi \, \mathrm{d}\mathbf{V}, \tag{5}$$

$$\text{face average: } \langle \psi \rangle_{\boldsymbol{i}+\frac{1}{2}e^d} = \frac{1}{h^{D-1}} \int_{A_{\boldsymbol{i}+\frac{1}{2}e^d}} \psi \, \mathrm{d}\mathbf{A}. \tag{6}$$

The point values and averages are related to each other as follows:

$$\langle \psi \rangle_{\boldsymbol{i}} = \psi_{\boldsymbol{i}} + \tfrac{h^2}{24}\Delta\psi + \mathrm{O}(h^4), \tag{7}$$

$$\langle \psi \rangle_{\boldsymbol{i}+\frac{1}{2}e^d} = \psi_{\boldsymbol{i}+\frac{1}{2}e^d} + \tfrac{h^2}{24}\Delta^{\perp,d}\psi + \mathrm{O}(h^4), \tag{8}$$

where $\Delta^{\perp,d}\psi = \sum_{d' \neq d} \frac{\partial^2 \psi}{\partial x_{d'}^2}$, and the derivatives are evaluated at the cell and face center, respectively. We also denote by D_d^2 the centered second-difference operator in the d coordinate direction.

$$(D_d^2\psi)_{\boldsymbol{i}} \equiv \frac{1}{h^2}\left(\psi_{\boldsymbol{i}+e^d} - 2\psi_{\boldsymbol{i}} + \psi_{\boldsymbol{i}-e^d}\right) = \frac{\partial^2 \psi}{\partial x_d^2} + \mathrm{O}(h^2). \tag{9}$$

3. Finite-volume formulation of Mehrstellen discretizations

It follows from (1) and the divergence theorem that

$$\langle \rho \rangle_{\boldsymbol{i}} = \frac{1}{h} \sum_{d=0}^{D-1} \left(\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}+\frac{1}{2}e^d} - \left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}-\frac{1}{2}e^d} \right). \tag{10}$$

This exact relationship is the starting point for finite-volume discretizations of (1). We approximate the average of the fluxes to fourth-order accuracy by approximating the derivative by a finite difference then using Eq. (8) to approximate the average over the face,

$$\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}+\frac{1}{2}e^d} = \frac{1}{h}(\phi_{\boldsymbol{i}+e^d} - \phi_{\boldsymbol{i}}) - \frac{h^2}{24}\frac{\partial^3 \phi}{\partial x_d^3} + \frac{h^2}{24}\Delta^{\perp,d}\left(\frac{\partial \phi}{\partial x_d}\right) + \mathrm{O}(h^4). \tag{11}$$

In order to evaluate the third derivative in a compact fashion to $\mathrm{O}(h^2)$, we use Eq. (1) and substitute

$$\frac{\partial^2 \phi}{\partial x_d^2} = \rho - \Delta^{\perp,d}\phi \tag{12}$$

into Eq. (11). This substitution yields

$$\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle = \frac{1}{h}(\phi_{i+e^d} - \phi_i) + \frac{h^2}{24}\left(\Delta^{\perp,d}\left(\frac{\partial \phi}{\partial x_d}\right) - \frac{\partial}{\partial x_d}\left(\rho - \Delta^{\perp,d}\phi\right)\right) + \mathrm{O}(h^4). \tag{13}$$

This leads to the following fourth-order accurate finite-volume discretization:

$$\langle \rho \rangle_i = (D \cdot \vec{F}^{\phi})_i - \frac{h^2}{24}(D \cdot \vec{F}^{\rho})_i, \tag{14}$$

$$= L^4(\phi)_i - \frac{h^2}{24} L^2(\rho)_i, \tag{15}$$

where away from boundaries, we have

$$F^{\phi}_{i+\frac{1}{2}e^d} = \frac{1}{h}\left(\left(\phi_{i+e^d} + \frac{h^2}{12}(\Delta_2^{\perp,d}\phi)_{i+e^d} \right) - \left(\phi_i + \frac{h^2}{12}(\Delta_2^{\perp,d}\phi)_i \right) \right), \tag{16}$$

$$F^{\rho}_{i+\frac{1}{2}e^d} = \frac{1}{h}(\rho_{i+e^d} - \rho_i), \tag{17}$$

with $\Delta_2^{\perp,d} = \sum_{d' \neq d} D^2_{d'}$.

It is convenient to express the operators L^2 and L^4 in terms of finite difference stencils away from domain boundaries. In both two and three dimensions, $L^2\rho = \sum_d D_d^2 \rho$. For L^4, we have in two dimensions,

$$L^4\phi = \frac{1}{6h^2} \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array} \phi \tag{18}$$

and in three dimensions,

$$L^4\phi = \frac{1}{6h^2} \left[\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & -24 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \right] \phi \tag{19}$$

If we utilize Eq. (7) in combination with Eq. (15) we obtain the classical Mehrstellen method,

$$(L^4\phi)_i = \rho_i + \frac{h^2}{12}(L^2\rho)_i. \tag{20}$$

3.1. Domain boundary conditions

For periodic boundary conditions, we extend the solution to the required ghost cells using periodic images of the solution. For Dirichlet boundary conditions, we use cell centered ghost values for computing fluxes. For the value of ϕ in a ghost cell that shares a face with a cell in the computational domain, we extrapolate using a quartic polynomial that matches the first four cells in the normal direction, plus the value on the Dirichlet boundary (Fig. 1).

$$\phi_G = \frac{1}{35}[128\phi_B - 140\phi_0 + 70\phi_1 - 28\phi_2 + 5\phi_3]. \tag{21}$$

To fill in the remaining values of ϕ needed at corners (2D) or edges (3D), we extrapolate from the two coordinate directions using a quartic polynomial that matches the five adjacent ghost values in each of two coordinate directions, and average the results.

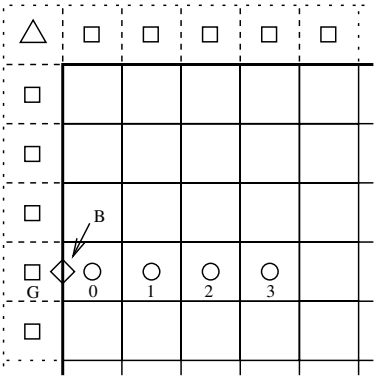

Fig. 1. Domain boundary conditions.

$$\phi_0 = 5\phi_1 - 10\phi_2 + 10\phi_3 - 5\phi_4 + \phi_5. \tag{22}$$

In the case of Neumann boundary conditions, the boundary data is assumed to be specified directly as F^B, an $O(h^4)$ value for the average of the flux over the face.

$$F^{\phi}_{\boldsymbol{i}_0+\frac{1}{2}\boldsymbol{e}^d} - \frac{h^2}{24}F^{\rho}_{\boldsymbol{i}_0+\frac{1}{2}\boldsymbol{e}^d} = F^B_{\boldsymbol{i}_0+\frac{1}{2}\boldsymbol{e}^d} = \left\langle \frac{\partial\phi}{\partial x_d} \right\rangle_{\boldsymbol{i}_0+\frac{1}{2}\boldsymbol{e}^d} + O(h^4),$$

where $\boldsymbol{i}_0 + \frac{1}{2}\boldsymbol{e}^d$ is a face on the boundary. In that case it is arbitrary to which component of the boundary flux, F^{ϕ} or $-\frac{h^2}{24}F^{\rho}$, that F^B is assigned to. For example, one can set F^{ϕ} on the boundary to zero, and set $-\frac{h^2}{24}F^{\rho} = F^B$. We also need to set ghost values for ϕ that are required to compute fluxes on interior faces. For a value of ϕ in a ghost cell that shares a face with a cell in the computational domain, we extrapolate using a quartic polynomial that matches the first four cells in the normal direction plus a $O(h^4)$ value for the normal derivative at the center of the face.

$$\phi_G = \tfrac{1}{22}\left[-24h\phi_{x_d,B} + 17\phi_0 + 9\phi_1 - 5\phi_2 + \phi_3\right]. \tag{23}$$

The remaining values at the corners are filled in exactly as in the Dirichlet case.

4. Local refinement discretization

In this section we describe the extension of the Mehrstellen algorithm given above to the case of a locally refined grid. Our approach will be to express the locally refined discretizations in terms of the corresponding uniform grid discretizations at each level. An appropriate interpolation operator provides ghost cell values for points in the stencil extending outside of the grids at that level. We will also define a conservative discretization of the divergence operator on multilevel data.

We define a coarsening operator by $\mathscr{C}_r : \mathbb{Z}^{\mathbf{D}} \to \mathbb{Z}^{\mathbf{D}}$,

$$\mathscr{C}_r(\boldsymbol{i}) = \left(\left\lfloor \frac{i_0}{r} \right\rfloor, \ldots, \left\lfloor \frac{i_{d-1}}{r} \right\rfloor \right),$$

where r is a positive integer. These operators acting on subsets of $\mathbb{Z}^{\mathbf{D}}$ can be extended in a natural way to the face-centered sets: $\mathscr{C}_r(\Gamma^{e^d}) \equiv (\mathscr{C}_r(\Gamma))^{e^d}$. We use a finite-volume discretization of space to represent a nested hierarchy of grids that discretize the same continuous spatial domain. We assume that our problem domain can be discretized by a nested hierarchy of grids $\Gamma^0 \ldots \Gamma^{l\max}$, with $\Gamma^{l+1} = \mathscr{C}^{-1}_{n^l_{\text{ref}}}(\Gamma^l)$. and that the mesh

spacings h^l associated with Γ^l satisfy $h^l/h^{l+1} = n^l_{\text{ref}}$. The integer n^l_{ref} is the *refinement ratio* between level l and $l+1$. These conditions imply that the underlying continuous spatial domains defined by the control volumes are all identical. In this paper we will further assume n^l_{ref} is even and no less than 4. For any set $\Upsilon \subseteq \Gamma^l$, we define $\mathcal{G}(\Upsilon, r)$, $r > 0$, to the set of all points within a $|\cdot|$-distance r of Υ that are still contained in Γ^l

$$\mathcal{G}(\Upsilon, r) = \Gamma^l \cap \bigcup_{|\boldsymbol{i}| \leqslant r} \Upsilon + \boldsymbol{i},$$

where $|\boldsymbol{i}| = \max_{d=0\ldots\mathbf{D}-1}(|i_d|)$, We can extend the definition to the case $r < 0$

$$\mathcal{G}(\Upsilon, r) = \Gamma^l - \mathcal{G}(\Gamma^l - \Upsilon, -r).$$

Thus $\mathcal{G}(\Upsilon, r)$ consists of all of the points in Υ that are within a distance $-r$ from points in the complement of Υ in Γ^l. In the case that there are periodic boundary conditions in one or more of the coordinate directions, we think of the various sets appearing here and in what follows as consisting of the set combined with all of its periodic images for the purpose of defining set operations and computing ghost cell values. For example, $\mathcal{G}(\Upsilon, r)$ is obtained by growing the union of Υ with its periodic images, and performing the intersections and differences with the union of Γ^l with its periodic images.

We make two assumptions about the nesting of grids at successive levels. We require the control volume corresponding to a cell in Ω^{l-1} is either completely contained in the control volumes defined by Ω^l or its intersection has zero volume. We also assume that there is at least $2n^l_{\text{ref}}$ level l cells separating level $l+1$ cells from level $l-1$ cells: $\mathcal{G}(\mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1}), 2n^l_{\text{ref}}) \subseteq \Omega^l$. We will refer to grid hierarchies that meet these two conditions as being *properly nested*. This is a much more restrictive notion of proper nesting than is typically used for second-order finite-volume methods, but is imposed to simplify the coarse–fine interpolation process.

From a formal numerical analysis standpoint, a solution on a locally refined mesh hierarchy $\{\Omega^l\}_{l=0}^{l_{\max}}$ approximates the exact solution to the PDE only on those cells that are not covered by a grid at a finer level. We define the valid region of Ω^l as,

$$\Omega^l_{\text{valid}} = \Omega^l - \mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1}).$$

A composite array ψ^{comp} is a collection of discrete values defined on the valid regions at each of the levels of refinement.

$$\psi^{\text{comp}} = \{\psi^{l,\text{valid}}\}_{l=0}^{l_{\max}}, \quad \psi^{l,\text{valid}} : \Omega^l_{\text{valid}} \to \mathbb{R}^m.$$

We can also define valid regions and composite arrays for face-centered variables. $\Omega^{l,\boldsymbol{e}^d}_{\text{valid}} = \Omega^{l,\boldsymbol{e}^d} - \mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1,\boldsymbol{e}^d})$. Thus, $\Omega^{l,\boldsymbol{e}^d}_{\text{valid}}$ consists of d-faces that are not covered by the d-faces at the next finer level. A composite vector field $\vec{F}^{\text{comp}} = \{\vec{F}^{l,\text{valid}}\}_{l=0}^{l_{\max}}$ is defined as follows:

$$\vec{F}^{l,\text{valid}} = (F_0^{l,\text{valid}} \ldots F_{\mathbf{D}-1}^{l,\text{valid}}), \quad F_d^{l,\text{valid}} : \Omega^{l,\boldsymbol{e}^d}_{\text{valid}} \to \mathbb{R}.$$

Thus a composite vector field has values at level l on all of the faces not covered by faces at the next finer level.

We want to define a composite divergence $D^{\text{comp}}(\vec{F}^{l+1,\text{valid}}, \vec{F}^{l,\text{valid}})_{\boldsymbol{i}}$ for $\boldsymbol{i} \in \Omega^l_{\text{valid}}$. To do this, we construct an extension of $\vec{F}^{l,\text{valid}}$ to the edges adjacent to Ω^l_{valid} that are covered by fine level faces. On the valid coarse-level d-faces, $F^l_d = F^{l,\text{valid}}_d$. On the faces adjacent to cells in Ω^l_{valid}, but not in $\Omega^{l,\boldsymbol{e}^d}_{\text{valid}}$, we set $F^l_d = \langle F^{l+1,\text{valid}}_d \rangle$, the average of F^{l+1}_d onto the next coarser level.

$$\langle F^{l+1}_d \rangle_{\boldsymbol{i}_c + \frac{1}{2}\boldsymbol{e}^d} = \frac{1}{(n_{\text{ref}})^{\mathbf{D}-1}} \sum_{\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d \in \mathcal{F}^d} F^{l+1}_{d,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \quad \boldsymbol{i}_c + \tfrac{1}{2}\boldsymbol{e}^d \in \zeta^{l+1}_{d,+} \cup \zeta^{l+1}_{d,-}.$$

Here, $\mathscr{F}^d$ is the set of all fine level d-faces that are covered by $A_{\boldsymbol{i}_c+\frac{1}{2}\boldsymbol{e}^d}$. $\zeta^{l+1}_{d,\pm}$ consists of all the d-faces in Ω^l on the boundary of Ω^{l+1}, with valid cells on the low ($\pm = -$) or high ($\pm = +$) side.

$$\zeta^{l+1}_{d,\pm} = \{\boldsymbol{i} \pm \tfrac{1}{2}\boldsymbol{e}^d : \boldsymbol{i} \pm \boldsymbol{e}^d \in \Omega^l_{\text{valid}},\ \boldsymbol{i} \in \mathscr{C}_{n_{\text{ref}}}(\Omega^{l+1})\}.$$

Given that extension, our composite divergence is defined

$$D^{\text{comp}}(F^{l+1,\text{valid}}, F^{l,\text{valid}})_{\boldsymbol{i}} = D \cdot \vec{F}^l_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \Omega^l_{\text{valid}}. \tag{24}$$

It is useful to express D^{comp} as an application of the level divergence operator D applied to extensions of $\vec{F}^{l,\text{valid}}$ to the entire level, followed by a step that corrects the cells in Ω^l_{valid} that are adjacent to Ω^{l+1}. We define a *flux register* $\delta\vec{F}^{l+1}$ associated with the fine level:

$$\delta\vec{F}^{l+1} = (\delta F^{l+1}_0, \ldots, \delta F^{l+1}_{\mathbf{D}-1}),$$

$$\delta F^{l+1}_d : \zeta^{l+1}_{d,+} \cup \zeta^{l+1}_{d,-} \to \mathbb{R}^m.$$

Let $\vec{F}^l$ be *any* coarse level vector field that extends $\vec{F}^{l,\text{valid}}$, i.e.

$$F^l_d = F^{l,\text{valid}}_d \quad \text{on } \Omega^{c,\boldsymbol{e}^d}_{\text{valid}}.$$

Then for $\boldsymbol{i} \in \Omega^l_{\text{valid}}$,

$$D^{\text{comp}}(\vec{F}^{l+1,\text{valid}}, \vec{F}^{l,\text{valid}})_{\boldsymbol{i}} = (D\vec{F}^l)_{\boldsymbol{i}} + D_{\text{R}}(\delta\vec{F}^{l+1})_{\boldsymbol{i}}. \tag{25}$$

Here $\delta\vec{F}^{l+1}$ is a flux register, set to be

$$\delta F^{l+1}_d = \langle F^{l+1}_d \rangle - F^l_d \quad \text{on } \zeta^l_{d,+} \cup \zeta^l_{d,-}.$$

D_{R} is the reflux divergence operator, given by the following for valid coarse level cells adjacent to Ω^{l+1}:

$$D_{\text{R}}(\delta\vec{F}^{l+1})_{\boldsymbol{i}} = \frac{1}{h^l}\sum_{d=0}^{\mathbf{D}-1} \sum_{\substack{\pm=+,-:\\ \boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d \in \zeta^{l+1}_{d,\mp}}} \pm\delta F^{l+1}_{d,\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}.$$

For the remaining cells in Ω^l_{valid}, $D_{\text{R}}(\delta\vec{F}^{l+1})$ is defined to be identically zero.

We can now define our Mehrstellen discretization of (1) on a locally refined grid as follows. On each level, we compute $\phi^{l,\text{ext}}$ on $\mathscr{G}(\Omega^l, 1) - \mathscr{G}(\mathscr{C}_{n^l_{\text{ref}}}(\Omega^{l+1}), -2)$ such that $\phi^{l,\text{ext}} = \phi^{l,\text{valid}}$ on Ω^l_{valid}. For $\boldsymbol{i} \in \Omega^l_{\text{valid}}$, we define:

$$L^{\text{comp},4}(\phi^{\text{comp}}, \rho^{\text{comp}})_{\boldsymbol{i}} \equiv (D \cdot \vec{F}^{\phi,l})_{\boldsymbol{i}} + D_{\text{R}}(\delta\vec{F}^{\phi,l+1})_{\boldsymbol{i}}, \tag{26}$$

$$L^{\text{comp},2}(\rho^{\text{comp}})_{\boldsymbol{i}} \equiv (D \cdot \vec{F}^{\rho,l})_{\boldsymbol{i}} + D_{\text{R}}(\delta\vec{F}^{\rho,l+1})_{\boldsymbol{i}} \tag{27}$$

Here $F^{\phi,l}$, $F^{\phi,l+1}$ are computed by applying (16) to $\phi^{l,\text{ext}}$, $\phi^{l+1,\text{ext}}$, combined with the problem domain boundary conditions. As we will see, the extensions of ϕ^l, ϕ^{l+1} depend linearly on ρ^l, ρ^{l+1}, a dependence which is explicitly denoted in (26). $F^{\rho,l}$, $F^{\rho,l+1}$ are computed similarly, using the second-order extensions of ρ^l, ρ^{l+1} described in [6].

We define Mehrstellen discretization of (1) as follows:

$$L^{\text{comp},4}(\phi^{\text{comp}}, \rho^{\text{comp}})_{\boldsymbol{i}} = \langle\rho^l\rangle_{\boldsymbol{i}} + \frac{(h^l)^2}{24} L^{\text{comp},2}(\rho^{\text{comp}})_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \Omega^l_{\text{valid}}, \tag{28}$$

where $\langle\rho\rangle_{\boldsymbol{i}}$ is some $O(h^4)$ accurate estimate of the average of the density. In computations presented here, we have taken $\langle\rho\rangle_{\boldsymbol{i}} = \rho_{\boldsymbol{i}} + \frac{h^2}{24} L^{\text{comp},2}(\rho)_{\boldsymbol{i}}$, where $\rho_{\boldsymbol{i}}$ is the value of the density at the cell center.

We note that the form of the equations given here leads to the following necessary conditions for solvability for the case of all periodic or Neumann boundary conditions:

$$\sum_l \sum_{\boldsymbol{i}\in\Omega^l_{\text{valid}}} (h^l)^D \langle\rho\rangle^l_{\boldsymbol{i}} = \sum_l \sum_d \sum_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d\in\mathscr{B}^l_d} (h^l)^{D-1} F^B_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \text{ (Neumann)},$$
$$= 0 \text{ (periodic)}.$$

For the Neumann case, $\mathscr{B}^l_d$ are the faces in the d direction on the intersection of the level valid cells with the domain boundary, and F^B are the specified face averages of the fluxes on the boundary. And while we have no proof of this, numerical experiments seem to indicate that these conditions are also sufficient.

4.1. Coarse–fine interpolation

To complete the definition of the discretization, we need to specify how we compute the extended values of ϕ at each level. We do this in two steps. First, we specify the calculation of the extended values on cells covered by the next finer level. Then, given that extension of the data on Ω^{l-1}, we can compute the ghost cell values on $\mathscr{G}(\Omega^l, 1) - \Omega^l$.

4.1.1. Coarsening

We need to compute $\phi^{l-1,\text{ext}}$ at coarse cell centers in areas where we do not have valid coarse data. In Fig. 2, the large open circles in the fine-grid region indicate where we need to coarsen from the fine cell data onto the coarse grid. These are needed either to evaluate the valid coarse-grid fluxes, or to perform interpolation from the coarse grid to obtain fine grid ghost cell values. To do this we employ a sixth-order accurate coarsening procedure to the fine data. For this coarsening procedure we use Taylor expansions based on valid fine data and substitute (1) to maintain a compact stencil.

For $\psi : \Omega^l \to \mathbb{R}$, we define $\langle\psi\rangle^{(2)} : \mathscr{C}_{n_{\text{ref}}}(\Omega^l) \to \mathbb{R}$ to be the coarsening of ψ from a 2^D-sized block of fine cells centered at $(\boldsymbol{i}_c + \frac{1}{2}\boldsymbol{u})h^c$

$$\langle\psi\rangle^{(2)}_{\boldsymbol{i}_c} \equiv \frac{1}{2^D} \sum_{\boldsymbol{s}:s_d=0,1} \psi_{n_{\text{ref}}(\boldsymbol{i}_c+\frac{1}{2}\boldsymbol{u})-\boldsymbol{s}}. \tag{29}$$

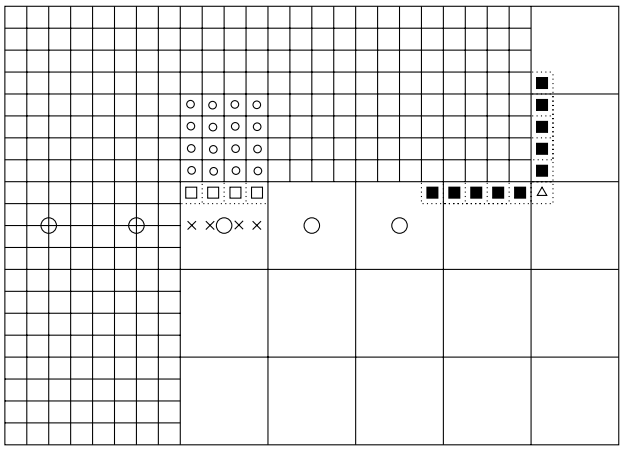

Fig. 2. Coarse–fine interpolation.

If $\psi_{\boldsymbol{i}} = \psi((\boldsymbol{i}+\frac{1}{2}\boldsymbol{u})h)$, based on Taylor expansions we have the following approximation result:

$$\langle\psi\rangle^{(2)}_{\boldsymbol{i}_c} = \psi((\boldsymbol{i}_c+\tfrac{1}{2}\boldsymbol{u})h^c) + \frac{h^2}{8}\Delta\psi + \frac{h^4}{384}\left(\Delta\Delta\psi + 4\sum_{0\leqslant d_1<d_2<D}\psi_{x_{d_1}x_{d_1}x_{d_2}x_{d_2}}\right) + \mathrm{O}(h^6), \tag{30}$$

where all of the derivatives are evaluated at $(\boldsymbol{i}_c+\frac{1}{2}\boldsymbol{u})h^c$. Applying (30) to ϕ and using (1), we obtain the following formula:

$$\begin{aligned}\langle\phi^l\rangle^{(6)}_{\boldsymbol{i}_c} &\equiv \langle\phi^l\rangle^{(2)}_{\boldsymbol{i}_c} - \frac{h^2}{8}\left(\langle\rho^l\rangle^{(2)}_{\boldsymbol{i}_c} - \frac{h^2}{8}\langle\Delta\rho^l\rangle^{(2)}_{\boldsymbol{i}_c}\right) - \frac{h^4}{384}\left(\langle\Delta\rho^l\rangle^{(2)}_{\boldsymbol{i}_c} + 4\sum_{0\leqslant d_1<d_2<D}\langle\phi^l_{x_{d_1}x_{d_1}x_{d_2}x_{d_2}}\rangle^{(2)}\right)\\ &= \phi\big((\boldsymbol{i}_c+\tfrac{1}{2}\boldsymbol{u})h\big) + \mathrm{O}(h^6),\end{aligned} \tag{31}$$

where the derivatives on the fine grid are replaced by second-order accurate finite differences: $\Delta \to \sum_d D_d^2, \partial^2_{x_{d_1}}\partial^2_{x_{d_2}} \to D^2_{d_1}D^2_{d_2}$.

Given ϕ defined on valid fine cells, we define on extended values on the region covered by the fine grid as follows:

$$\phi^{l-1,\mathrm{ext}}_{\boldsymbol{i}} = \langle\phi^l\rangle^{(6)}_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \mathscr{C}_{n^{l-1}_{\mathrm{ref}}}(\Omega^l) - \mathscr{G}(\mathscr{C}_{n^{l-1}_{\mathrm{ref}}}(\Omega^l), -2). \tag{32}$$

We note that the stencils for the extended values at a coarse cell are contained entirely in the fine cells covered by that coarse cell, provided that $n_{\mathrm{ref}} \geqslant 4$.

4.1.2. Interpolation

We use an $\mathrm{O}(h^5)$ interpolation procedure to compute $\phi^{l,\mathrm{ext}}$ on $\mathscr{G}(\Omega^l, 1) - \Omega^l$. This is done in two steps, as seen in Fig. 2. First, we compute an $\mathrm{O}(h^5)$ interpolant in the direction tangent to the coarse–fine boundary, at the locations indicated by the ×'s. This uses all the values within a $|\cdot|$-distance two in the plane parallel to the boundary. In Fig. 2, this corresponds to five of the coarse cell centers indicated by large open circles: one at the center, and two on either side. If any of these cells are covered by the next-finer grid, the coarse-grid values (32) are used. The tangentially interpolated values are used with four points on the fine grid (indicated by the small open circles) to interpolate the values at the ghost cell locations (indicated by the open boxes).

In two dimensions, the tangential interpolation is done using a quartic polynomial. For a refinement ratio of four, the interpolation formulae are given as follows (see Fig. 3):

$$\phi^X_C = \frac{1}{32768}\left[315\phi^{l-1}_1 - 2380\phi^{l-1}_2 + 32130\phi^{l-1}_3 + 3060\phi^{l-1}_4 - 357\phi^{l-1}_5\right], \tag{33}$$

$$\phi^X_D = \frac{1}{32768}\left[715\phi^{l-1}_1 - 4940\phi^{l-1}_2 + 27170\phi^{l-1}_3 + 10868\phi^{l-1}_4 - 1045\phi^{l-1}_5\right]. \tag{34}$$

The other two points (A and B) use the same formula, but with the order of the input coarse-grid values reversed. In three dimensions, the interpolation in the plane is done as a tensor product of the quartic interpolation above. The interpolation formula (33) is used in one coordinate direction to compute five coarse-grid values in each of n^l_{ref} 1D stencils in the other coordinate direction. We then apply (33) in the second coordinate direction to obtain the required values.

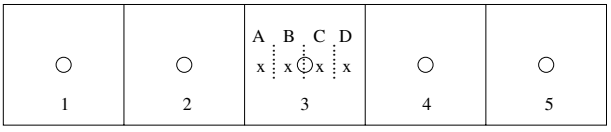

Fig. 3. Quartic interpolation parallel to the coarse–fine interface.

The values at the ghost cells are then obtained using a 1D quartic interpolant (see Fig. 4). For a refinement ratio of four, the formula is given as follows:

$$\phi_i^{l,\text{ext}} = \frac{1}{1155}\left[128\phi_1^X + 2772\phi_2^l - 2970\phi_3^l + 1540\phi_4^l - 315\phi_5^l\right]. \tag{35}$$

For coarse–fine interface ghost cell corners where we can interpolate coming from different directions (see the far left open square in Fig. 2), we average the different interpolations. To obtain the remaining corner/edge ghost cells (see the triangle in Fig. 2), we interpolate using our new ghost values (see the solid squares in Fig. 2). For this interpolation we simply extend quartic polynomials through 5 ghost points (per direction), and average results coming from different directions. The resulting corner/edge ghost points (the triangles) depend on both fine and coarse values, as the ghost points used (the solid squares) are derived from surrounding coarse and fine values.

5. Locally refined multigrid algorithm description

In order to solve the system (28) using a geometric multigrid algorithm for linear systems, we need to eliminate the dependence of L^4 on ρ^{comp}. We define $M^{\text{comp},4}(\phi^{\text{comp}}) \equiv L^{\text{comp},4}(\phi^{\text{comp}},\rho^{\text{comp}} \equiv 0)$:

$$\begin{aligned} M^{\text{comp},4}(\phi^{\text{comp}}) &= g^l, \\ &= \langle \rho^l \rangle_i + \frac{(h^l)^2}{24} L^{\text{comp},2}(\rho^{\text{comp}})_i - L^{\text{comp},4}(\phi^{\text{comp}} \equiv 0, \rho^{\text{comp}}). \end{aligned} \tag{36}$$

The system (36) is a linear system for ϕ^{comp}.

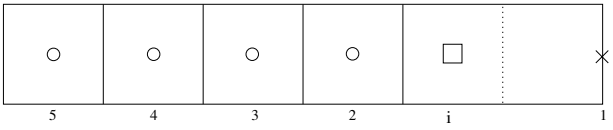

Fig. 4. Quartic interpolation normal to the coarse–fine interface.

```
procedure mgRelax(φ^f, R^f, r)
{
    for i = 1, ..., NumSmoothDown
        φ^f := φ^f + λ(M^nf(φ^f, φ^c ≡ 0) − R^f)
    end for
    if (r > 2) then
        δ^c := 0
        R^c := Average(R^f − M^nf(φ^f, φ^c ≡ 0))
        mgRelax(δ^c, R^c, r/2)
        φ^f := φ^f + I_pwc(δ^c)
        for i = 1, ..., NumSmoothUp
            φ^f := φ^f + λ(M^nf(φ^f, φ^c ≡ 0) − R^f)
        end for
    end if
}
```

Fig. 5. Recursive relaxation procedure.

We also define the operator M^{nf} is a two-level discretization of the Mehrstellen Laplacian:

$$M^{nf}(\phi^l, \phi^{l-1,\text{valid}}) = L^4(\phi^{l,\text{ext},0}), \tag{37}$$

$$\phi^l : \Omega^l \to \mathbb{R}, \tag{38}$$

where $\phi^{l,\text{ext},0} = \phi^l$ on Ω^l, and is otherwise given by the interpolation procedure in Section 4.1, but with ρ^l, $\rho^{l-1} \equiv 0$.

We solve (36) using the approach in [5,8] It is similar to the algorithm used in [9] to compute steady incompressible flow, and has been used in a variety of settings [2,10–12].

A pseudo-code description of the algorithm is given in Figs. 5 and 6. In multigrid we use standard conservative averaging and piecewise constant interpolation operators, Average and I_{pwc}, as in [6]: $\text{Average}(\varphi)_{\boldsymbol{i}^c} = \frac{1}{2^d}\sum_{\boldsymbol{i}\in\mathscr{C}_2^{-1}(\boldsymbol{i}^c)}\varphi_{\boldsymbol{i}}$ and $I_{\text{pwc}}(\varphi)_{\boldsymbol{i}^f} = \varphi_{\boldsymbol{i}}$, where $\boldsymbol{i} = \mathscr{C}_2(\boldsymbol{i}^f)$.

The smoothing operator $\text{mgRelax}(\phi^f, R^f, r)$ performs a multigrid V-cycle iteration on ϕ^f for the operator M^{nf}, assuming the coarse-grid values required for the boundary conditions are identically zero. Within mgRelax() our smoother is weighted Jacobi, which we apply twice per smooth (in Fig. 5 we set NumSmoothDown = NumSmoothUp = 2).

$\bar{\rho} := \rho + \frac{h^2}{24} L^{comp,2}(\rho)$
$g := \bar{\rho} - L^{comp,4}(0, \rho)$
$R_{initial} := g - M(\phi_{initial})$
while $(||R|| > \epsilon ||R_{initial}||)$
    VCycleMG(l^{max})
    $R := g - M(\phi)$
end while

Procedure VCycleMG(level l):
if $(l = l^{max})$ then $R^l := g^l - M^{nf}(\phi^l, \phi^{l-1})$
if $(l > 0)$ then
    $\phi^{l,save} := \phi^l$ on Ω^l
    $e^l := 0$ on Ω^l
    $\text{mgRelax}(e^l, R^l, n_{ref}^{l-1})$
    $\phi^l := \phi^l + e^l$
    $e^{l-1} := 0$ on Ω^{l-1}
    $R^{l-1} := Average(R^l - M^{nf}(e^l, e^{l-1}))$ on $\mathcal{C}_{n_{ref}^{l-1}}(\Omega^l)$
    $R^{l-1} := g^{l-1} - M^{comp,l-1}(\phi)$ on $\Omega^{l-1} - \mathcal{C}_{n_{ref}^{l-1}}(\Omega^l)$
    VCycleMG($l-1$)
    $e^l := e^l + I_{pwc}(e^{l-1})$
    $R^l := R^l - M^{nf,l}(e^l, e^{l-1})$
    $\delta e^l := 0$ on Ω^l
    $\text{mgRelax}(\delta e^l, R^l, n_{ref}^{l-1})$
    $e^l := e^l + \delta e^l$
    $\phi^l := \phi^{l,save} + e^l$
else
    solve $M^{nf}(e^0) = R^0$ on Ω^0.
    $\phi^0 := \phi^0 + e^0$
end if

Fig. 6. Pseudo-code description of the locally refined multigrid algorithm.

6. Convergence of the algorithm

We have selected four test problems to both demonstrate the algorithm, and illustrate fourth-order convergence. The first and third problems are in two dimensions, while the other two are three dimensional problems. For all problems we use local refinement, and for some we compare with single grid versions and/or second-order methods. We chose three problems with analytic solutions, and one classic problem from fluid dynamics. For the first three problems we compute errors. We compute the truncation error by,

$$\tau^{\mathrm{comp}} = g^{\mathrm{comp}} - M^{\mathrm{comp},4}(\phi^{\mathrm{comp,exact}}) \tag{39}$$

and solution error as,

$$e^{\mathrm{comp}} = \phi^{\mathrm{comp,exact}} - \phi^{\mathrm{comp}}. \tag{40}$$

We can then compute p-norms as follows:

$$\|e\|_p = \left(\sum_l \sum_{\boldsymbol{i} \in \Omega^l_{\mathrm{valid}}} |e_{\boldsymbol{i}}|^p (h^l)^{\mathbf{D}} \right)^{1/p}. \tag{41}$$

Details of the test problems are given in the following sub-sections.

6.1. Problem 1

This first test problem has a doubly periodic, unit square solution domain. Our exact solution is,

$$\phi = \sin(2\pi x)\sin(2\pi y), \tag{42}$$

$$\rho = -8\pi^2 \sin(2\pi x)\sin(2\pi y). \tag{43}$$

In Fig. 7 we specify the disjoint boxes that define the refinement region for this problem. We solved (1) for ϕ, given (43) using a number of different methods. First we computed the solution using our second-order accurate local refinement solver, and then with our fourth-order accurate local refinement solver. Results from these runs are shown in Table 1. The expected convergence rates are apparent for these tests. We achieved second-order solution error (first-order accurate truncation error not shown) for the second-order method. For the fourth-order method we achieved fourth-order accuracy for solution error and third-order for truncation error. Note the tremendous difference in magnitude of the error (fourth compared to second), even for the coarsest grids.

Subsequently, we tested the effect of lowering the accuracy of the coarse–fine interpolation on the fourth-order accurate method. For this test we first used a quadratic interpolant tangential to the interface with a quartic interpolant normal to the interface. Results for this are shown in Table 2. This first alteration lowered the truncation error to first-order accurate due to the quadratic interpolation, which yields third-order accurate solution error. Our second test was to use a quartic interpolant tangential to the interface and a cubic interpolant normal to the interface. Results are shown in Table 2. This second alteration lowered the truncation error to second-order accurate due to the cubic interpolation, which yielded a fourth-order accurate solution error. While this second alteration yielded a fully fourth-order accurate solution error, the norms of the solution error were degraded in magnitude.

6.2. Problem 2

For this second test problem we illustrate our capability to achieve fourth-order accuracy for three dimensional problems. The following expression is what we used for this doubly periodic, unit domain, test problem:

$$\phi = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z). \tag{44}$$

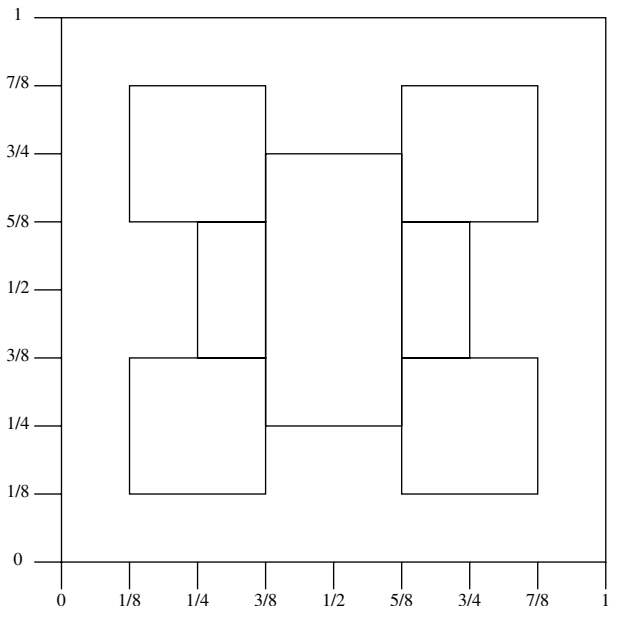

Fig. 7. Grid layout for Problem 1.

Table 1
Test Problem 1: $\phi = \sin(2\pi x)\sin(2\pi y)$, solution and truncation errors with convergence rates

Base grid	$h = 1/64$	Rate	1/128	Rate	1/256	Rate	1/512
(a) Second-order method, 2 refinement levels (2D), quadratic coarse–fine interpolation							
L_1 Solution	1.075e−04	**2.024**	2.644e−05	**2.011**	6.562e−06	**2.005**	1.635e−06
L_2 Solution	1.278e−04	**2.031**	3.129e−05	**2.013**	7.751e−06	**2.006**	1.929e−06
L_∞ Solution	2.306e−04	**2.079**	5.457e−05	**2.036**	1.330e−05	**2.017**	3.286e−06
(b) Fourth-order method, 2 refinement levels (2D), quartic coarse–fine interpolation							
L_1 Solution	1.361e−07	**4.003**	8.490e−09	**4.001**	5.302e−10	**4.001**	3.312e−11
L_2 Solution	1.672e−07	**4.001**	1.044e−08	**4.001**	6.523e−10	**4.001**	4.075e−11
L_∞ Solution	3.182e−07	**4.014**	1.970e−08	**4.004**	1.228e−09	**4.004**	7.652e−11
L_∞ Truncation	4.303e−04	**2.994**	5.402e−05	**2.997**	6.768e−06	**2.997**	8.476e−07

Again this results in an analytic form for ρ (given Eq. (1)). For this problem we use the same box layout as in problem 1, except in the third dimension the boxes range from 3/8 to 5/8. Results from this test problem are presented in Table 3.

The results are still clearly fourth-order accurate.

6.3. Problem 3

For this test problem we present two dimensional solutions with inhomogeneous Dirichlet boundary conditions. The following expressions are what we used for this unit domain test problem:

Table 2
Test Problem 1: $\phi = \sin(2\pi x)\sin(2\pi y)$, solution and truncation errors with convergence rates

Base grid	$h = 1/64$	Rate	1/128	Rate	1/256
(a) 2 Refinement levels (2D), coarse–fine interpolation with quadratic tangential and quartic normal to interface					
L_1 Solution	7.641e−07	**3.053**	9.204e−08	**3.012**	1.141e−08
L_2 Solution	1.060e−06	**3.040**	1.289e−07	**3.010**	1.600e−08
L_∞ Solution	1.245e−05	**2.857**	1.719e−06	**2.905**	2.295e−07
L_∞ Truncation	3.718e−01	**0.996**	1.864e−01	**0.999**	9.324e−02
(b) 2 Refinement levels (2D), coarse–fine interpolation with quartic tangential and cubic normal to interface					
L_1 Solution	1.403e−07	**4.024**	8.623e−09	**4.012**	5.346e−10
L_2 Solution	1.727e−07	**4.025**	1.061e−08	**4.013**	6.574e−10
L_∞ Solution	4.656e−07	**3.962**	2.987e−08	**3.982**	1.891e−09
L_∞ Truncation	9.258e−03	**2.083**	2.185e−03	**2.046**	5.290e−04

Lower order coarse–fine interpolation.

Table 3
Test Problem 2: $\phi = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$, solution and truncation errors with convergence rates

Base grid	$h = 1/8$	Rate	1/16	Rate	1/32
(a) 2 Refinement levels (3D)					
L_1 Solution	5.803e−04	**3.525**	5.042e−05	**4.027**	3.093e−06
L_2 Solution	8.320e−04	**3.610**	6.815e−05	**3.997**	4.268e−06
L_∞ Solution	2.375e−03	**3.595**	1.966e−04	**3.964**	1.260e−05
L_∞ Truncation	3.180e−01	**2.256**	6.657e−02	**2.904**	8.896e−03

$$\phi = \begin{cases} \frac{r^{10}}{100a^8} - \frac{4r^9}{81a^7} + \frac{3r^8}{32a^6} - \frac{4r^7}{49a^5} + \frac{r^6}{36a^4}, & r < a, \\ \frac{a^2}{1260}(\ln(r) - \ln(a) + \frac{1627}{2520}), & r \geqslant a, \end{cases} \tag{45}$$

$$\rho = \begin{cases} (\frac{r}{a} - (\frac{r}{a})^2)^4, & r < a, \\ 0, & r \geqslant a, \end{cases} \tag{46}$$

where r is the radius and we set $a = 0.06$. The grid layout for this problem was a simple square refinement patch nested within the coarse level. For example, on a 2 level run we used a 256×256 cell base grid with $h = 1/256$, and a centered refined patch of 256×256 cells with $h = 1/1024$. Results from this test problem are presented in Table 4. We also present solution errors for a locally refined run and the equivalent single grid run in Figs. 8 and 9. Inspection of Table 4 reveals fourth-order accuracy. Table 4 also shows approximately equivalent error magnitude for locally refined vs. single grid (with equivalent finest resolution), even though the locally refined computations used eight times fewer computational cells.

6.4. Problem 4

For this last problem we evaluate the performance of the algorithm for a complex three dimensional problem. This test problem is based on a co-rotating vortex ring problem from fluid dynamics. The right-hand side is specified as two rings modulated by a sin() function, in a periodic, unit cube domain. Each modulated ring is specified by a location of the center of the ring (x_0, y_0, z_0), the radius of

Table 4
Test Problem 3: solution errors and convergence rates

Base grid	$h = 1/128$	Rate	1/256	Rate	1/512
(a) 2 Refinement levels (2D)					
L_1 Solution	3.53e−014	**4.185**	1.94e−015	**4.290**	9.92e−017
L_2 Solution	9.02e−014	**3.868**	6.18e−015	**4.130**	3.53e−016
L_∞ Solution	1.48e−012	**3.817**	1.05e−013	**4.075**	6.23e−015
Base grid	$h = 1/512$	Rate	1/1024	Rate	1/2048
(b) Single level (2D)					
L_1 Solution	3.24e−014	**4.705**	1.24e−015	**4.775**	4.54e−017
L_2 Solution	8.55e−014	**4.126**	4.90e−015	**4.040**	2.98e−016
L_∞ Solution	1.64e−012	**4.111**	9.49e−014	**3.989**	5.98e−015

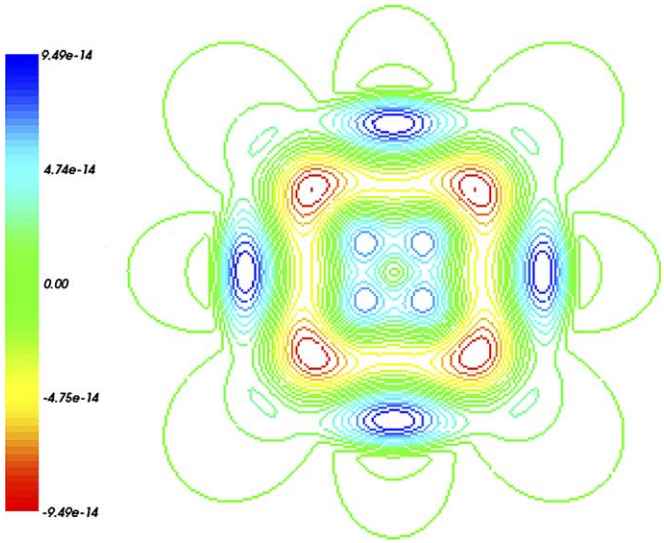

Fig. 8. Problem 3: contour plot of the solution error for a $h = 1/1024$ single grid run.

the center of the local cross-section of the ring from the center of the ring r, and the strength of the ring Γ.

The cross-sectional charge distribution in each modulated ring is given by

$$\rho(\beta,\theta) = \sin(\theta)\frac{\Gamma}{a\sigma^2}e^{-\left(\frac{\beta}{\sigma}\right)^3}, \tag{47}$$

where β is the local distance from the center of the ring cross-section, θ is the angle around the ring, $a = 2268.85$, and $\sigma = 0.0275$.

For this problem, the first modulated ring is centered at (0.5,0.5,0.4), with a radius of 0.2, and strength Γ of 1.5. The second modulated ring is centered at (0.5,0.5,0.65), with a radius of 0.25 and a strength $\Gamma = 1.0$.

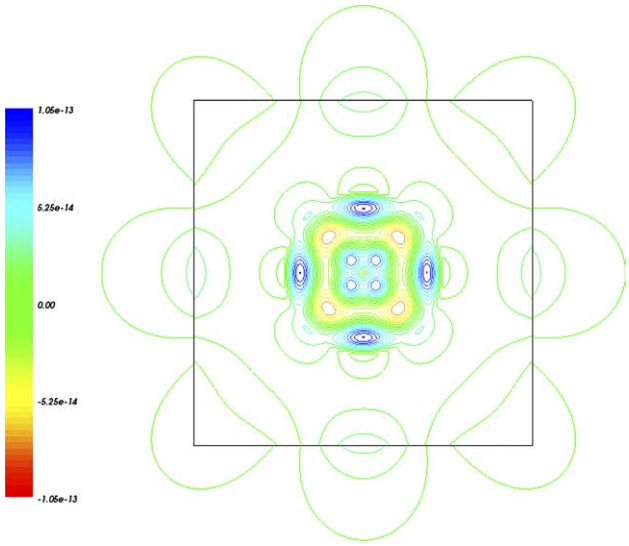

Fig. 9. Problem 3: contour plot of the solution error for a 2 level run ($h = 1/256$, $h = 1/1024$). The black box indicates the finer level.

Results from running this test problem at different resolutions, see Fig. 10, show that the method obtains residual reductions typical of multigrid. Also note that the method exhibits desired solvability characteristics, and reduces the residual by a constant factor until the stopping threshold is met. In Fig. 11 we have isolated the charge distribution ρ, by isosurfacing two equal valued regions. We slice through the solution with a plane that is colored by the solution, ϕ. The slicing plane also illustrates the individual computational cells from the 3 level refinement hierarchy used in the solve. We hope that this illustrates our capability to solve Poisson's equation for real problems.

7. Conclusions and future research

We have presented a new block-structured local refinement algorithm for Poisson's based on a finite-volume formulation of classical Mehrstellen discretization of the Laplacian, and extended to locally refined meshes using the ideas in [4,6]. The truncation error of the method is $O(h^4)$ except near boundaries between refinement levels, where it is $O(h^3)$. Modified-equation analysis suggests that the solution error is $O(h^4)$ uniformly, a result that is consistent with observed convergence of the method. We also observed a somewhat surprising superconvergence phenomenon. Even if we use an $O(h^4)$ interpolant at the boundary, we still obtain $O(h^4)$ solution error, even though the truncation error near the boundary is $O(h^2)$.

The approach described here suggests a broader program for higher-order locally refined methods. The quadrature formulas (7) and (8) provide a systematic mechanism for distinguishing between averages over cells, averages over faces, and point values, to fourth-order accuracy. This can be combined with the ideas in [13] to obtain fourth-order in space finite-volume discretizations for nonlinear hyperbolic problems on a locally refined grid. It is not obvious how to extend the Mehrstellen discretizations to the case where the right-hand side includes a time derivative, particularly in the case where implicit differencing in time is required. We will be considering a variety of possible approaches here, including fully implicit methods and

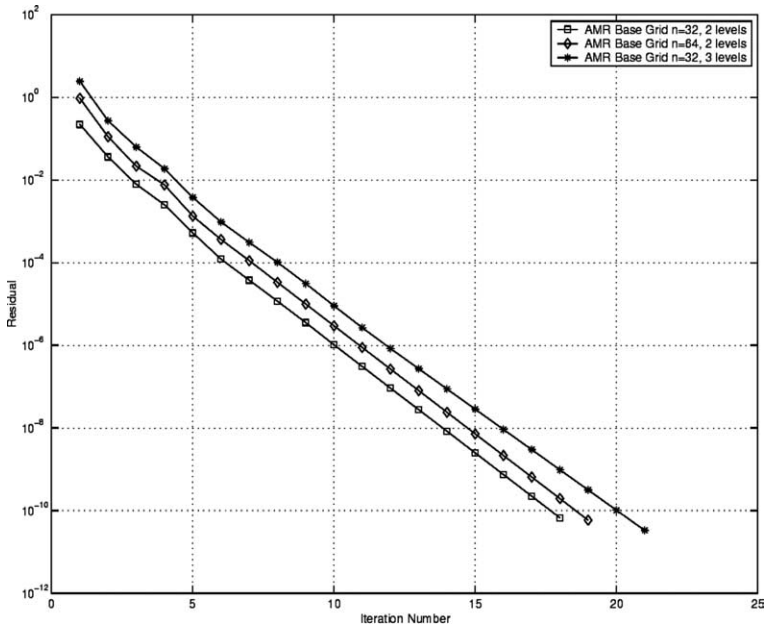

Fig. 10. Problem 4: plot of the ∞-norm of the residual versus multigrid iteration.

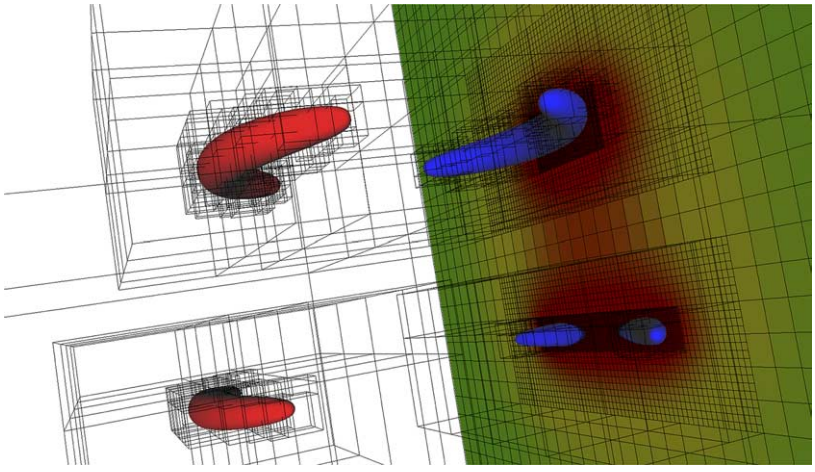

Fig. 11. Problem 4: 3D locally refined solution with isosurface of the right-hand side ρ and a slice colored by the solution ϕ. We show individual computational cells on the slice. The 3D boxes represent the disjoint union of rectangles (Ω_k^l's).

predictor-corrector approximations to such methods in which the Mehrstellen correction is treated explicitly. The approach outlined here is straightforward to pursue in conjunction with second-order accurate temporal discretizations. However, the extension to higher order in time is still an active research issue [14]. Finally, there is a possibility of extending this approach to complex geometries using embedded boundary methods [7]. In this case, it would be necessary to compute higher moments of the intersections between the irregular domain and the Cartesian grid.

Acknowledgments

The authors thank Dan Graves, Terry Ligocki, Dan Martin, and Peter Schwartz for many useful discussions during the preparation of this paper. The first author thank his Ph.D. adviser, Professor Geoffrey Schladow, for allowing him to take the additional time required to complete this project.

References

[1] L. Collatz, The Numerical Treatment of Differential Equations, Springer-Verlag, New York, NY, 1966.
[2] A.S. Almgren, T. Buttke, P. Colella, A fast adaptive vortex method in three dimensions, J. Comput. Phys. 113 (2) (1994) 177–200.
[3] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1) (1989) 64–84.
[4] M.L. Minion, A projection method for locally refined grids, J. Comput. Phys. 127 (1) (1996) 158–178.
[5] D.F. Martin, K.L. Cartwright, Solving Poisson's equation using adaptive mesh refinement, Technical Report UCB/ERL M96/66 UC, Berkeley.
[6] P. Colella, D.T. Graves, T.J. Ligocki, D.F. Martin, D. Modiano, D.B. Serafini, B.V. Straalen, Chombo Software Package for AMR Applications – Design Document, 2000 (Unpublished).
[7] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (2) (1998) 60–85.
[8] D.F. Martin, An adaptive cell-centered projection method for the incompressible Euler equations, Ph.D. Thesis, University of California, Berkeley, 1998.
[9] M.C. Thompson, J.H. Ferziger, An adaptive multigrid technique for the incompressible Navier–Stokes equations, J. Comput. Phys. 82 (1) (1989) 94–121.
[10] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.J. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1) (1998) 1–46.
[11] M.T. Bettencourt, A block-structured adaptive steady–state solver for the drift–diffusion equations, Ph.D. Thesis, Department of Mechanical Engineering, University of California, Berkeley, May 1998.
[12] P. Colella, M. Dorr, D. Wake, Numerical solution of plasma-fluid equations using locally refined grids, J. Comput. Phys. 152 (1999) 550–583.
[13] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.
[14] A. Bourlioux, A.T. Layton, M.L. Minion, High-order multi-implicit spectral deferred correction methods for problems of reacting fluid flow, J. Comput. Phys. 189 (2003) 651–675.