

A FOURTH-ORDER ACCURATE FINITE-VOLUME METHOD WITH STRUCTURED ADAPTIVE MESH REFINEMENT FOR SOLVING THE ADVECTION-DIFFUSION EQUATION*

QINGHAI ZHANG[†], HANS JOHANSEN[‡], AND PHILLIP COLELLA[‡]

Abstract. We present a fourth-order accurate algorithm for solving Poisson’s equation, the heat equation, and the advection-diffusion equation on a hierarchy of block-structured, adaptively refined grids. For spatial discretization, finite-volume stencils are derived for the divergence operator and Laplacian operator in the context of structured adaptive mesh refinement and a variety of boundary conditions; the resulting linear system is solved with a multigrid algorithm. For time integration, we couple the elliptic solver to a fourth-order accurate Runge–Kutta method, introduced by Kennedy and Carpenter [*Appl. Numer. Math.*, 44 (2003), pp. 139–181], which enables us to treat the nonstiff advection term explicitly and the stiff diffusion term implicitly. We demonstrate the spatial and temporal accuracy by comparing results with analytical solutions. Because of the general formulation of the approach, the algorithm is easily extensible to more complex physical systems.

Key words. Poisson’s equation, the heat equation, the advection-diffusion equation, adaptive mesh refinement, additive Runge–Kutta method, finite volume, conservation form

AMS subject classifications. 76R99, 65M08, 65L06

DOI. 10.1137/110820105

1. Introduction. The advection-diffusion equation governs numerous physical processes. Morton [17] lists ten sample applications ranging from semiconductor simulation to financial modelling. He also observes that “Accurate modelling of the interaction between convective and diffusive processes is the most ubiquitous and challenging task in the numerical approximation of partial differential equations.” This observation is partly due to the fact that algorithms and analysis tend to be very different in the two limiting cases of elliptic and hyperbolic equations. Also, even for very simple initial and boundary conditions, the true solution may contain multiple length-scales that vary drastically across the spatial domain; see (3.1) in [18] for such an example.

A finite-volume (FV) formulation is often preferred for applications where conservation is a primary concern. In its simplest form, the FV formulation is derived by applying the divergence theorem over the cells of a regular computational grid:

$$(1.1) \quad \frac{1}{h^D} \int_{V_i} \nabla \cdot \vec{F} dx = \frac{1}{h} \sum_{d=1}^D \left(\langle F_d \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle F_d \rangle_{i-\frac{1}{2}\mathbf{e}^d} \right),$$

where a *face-averaged* quantity is defined as

$$(1.2) \quad \langle F_d \rangle_{i+\frac{1}{2}\mathbf{e}^d} \equiv \frac{1}{h^{D-1}} \int_{A_{i+\frac{1}{2}\mathbf{e}^d}} F_d dA.$$

*Submitted to the journal’s Computational Methods in Science and Engineering section January 5, 2011; accepted for publication (in revised form) January 17, 2012; published electronically April 12, 2012.

<http://www.siam.org/journals/sisc/34-2/82010.html>

[†]Applied Numerical Algorithms Group, High Performance Computing Research Department, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, and Department of Mathematics, University of California, Davis, CA 95616 (qinghai@math.ucdavis.edu).

[‡]Applied Numerical Algorithms Group, High Performance Computing Research Department, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (hjohansen@lbl.gov, colella@hpcrdm.lbl.gov).

Here h denotes the uniform mesh spacing, $\mathbf{i} \in \mathbb{Z}^D$ a cell multi-index, and $\mathbf{e}^d \in \mathbb{Z}^D$ the vector with its d th component equal to one and all other components zero. Let $\mathbb{1} \in \mathbb{Z}^D$ be the vector whose elements are all equal to one; then $V_{\mathbf{i}} = [\mathbf{i}h, (\mathbf{i} + \mathbb{1})h]$ denotes a grid cell, and $A_{\mathbf{i}-\frac{1}{2}\mathbf{e}^d} = [\mathbf{i}h, (\mathbf{i} + \mathbb{1} - \mathbf{e}^d)h]$ and $A_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = [(\mathbf{i} + \mathbf{e}^d)h, (\mathbf{i} + \mathbb{1})h]$ denote the two faces of cell $V_{\mathbf{i}}$ along the d th dimension.

The relationship (1.1) is exact; approximations are obtained from replacing the integrals over faces (1.2) by quadratures. Traditionally, FV methods have been based on using the midpoint rule for the face integrals, which leads to a second-order accurate discretization. More recently, there has been increasing interest in using methods based on higher-order accurate FV methods of this form, with quadratures using the midpoint rule plus corrections computed using the transverse derivatives of the fluxes, or, for the case of fluxes that are linear functions of the unknowns, using deconvolution of face-averages from cell averages, following the ideas in [8, 21]. This has been done for Poisson’s equation [1], for hyperbolic problems on mapped grids [7], and for non-linear hyperbolic conservation laws on locally refined grids [14]. For time-dependent problems, a method-of-lines approach has been employed, using the high-order discretization methods in space and the classical explicit fourth-order accurate Runge–Kutta method in time. In this paper, we demonstrate that the approach described above can be used for advection-diffusion problems based on a semi-implicit time discretization. In particular, we use a fourth-order accurate additive Runge–Kutta method described in [12], treating the advection terms explicitly and the diffusion terms implicitly, with the spatial discretization performed on locally refined grids. The resulting method is fourth-order accurate, with a time-step stability constraint required only for the explicitly treated advection term.

2. FV formulation. The advection-diffusion equation is defined as

$$(2.1) \quad \frac{\partial \phi}{\partial t} = -\nabla \cdot (\mathbf{u}\phi) + \nu \Delta \phi + f,$$

where the constant diffusivity ν , the velocity field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, and the forcing term $f = f(\mathbf{x}, t)$ are known a priori. To generate an FV discretization, we average (2.1) over each control volume, $V_{\mathbf{i}}$, and apply the divergence theorem as in (1.1) to obtain a system of ordinary differential equations (ODEs) on \mathbb{Z}^D :

$$(2.2a) \quad \frac{d \langle \phi \rangle_{\mathbf{i}}}{dt} = L_{\text{adv}}(\langle \phi \rangle, t)_{\mathbf{i}} + L_{\text{diff}}(\langle \phi \rangle)_{\mathbf{i}} + \langle f \rangle_{\mathbf{i}},$$

$$(2.2b) \quad L_{\text{adv}}(\langle \phi \rangle, t)_{\mathbf{i}} = -\frac{1}{h} \sum_{d=1}^D \left(\langle u_d \phi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} - \langle u_d \phi \rangle_{\mathbf{i}-\frac{1}{2}\mathbf{e}^d} \right),$$

$$(2.2c) \quad L_{\text{diff}}(\langle \phi \rangle)_{\mathbf{i}} = \nu \langle \Delta \phi \rangle_{\mathbf{i}} = \frac{\nu}{h} \sum_{d=1}^D \left(\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} - \left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\mathbf{i}-\frac{1}{2}\mathbf{e}^d} \right),$$

where a *cell-averaged* quantity is defined as

$$(2.3) \quad \langle q \rangle_{\mathbf{i}} \equiv \frac{1}{h^D} \int_{V_{\mathbf{i}}} q(\mathbf{x}, t) d\mathbf{x}.$$

Note that (2.2a)–(2.2c) are still exact relationships with no discretization error.

In our notation, a subscript of the form $\langle \cdot \rangle_{\mathbf{i}}$ indicates a cell-averaged quantity defined by (2.3), while $\langle \cdot \rangle_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d}$ indicates a face-averaged quantity defined by (1.2). The differences between the three quantities account for much of the complexity of fourth-order FV methods that is unnecessary in second-order methods. Typically, $O(h^4)$ FV stencils (acting on *averages*) are different from corresponding finite differences (of *point values*), because of higher-order derivatives that must be included. Appendix A lists the commonly used formulas and derives specific ones used here.

2.1. Spatial discretization. The FV formulations of the two operators, (2.2b) and (2.2c), requires face-averaged quantities to be approximated from cell-averaged ones. The connection between cell-averaged quantities and face-averaged quantities for higher-order FV hyperbolic discretizations was established in [8]. For fourth-order accuracy, we use simple differences to approximate face-averaged values from cell-averaged values (derived in Appendix A):

$$(2.4) \quad \langle \phi \rangle_{\mathbf{i} + \frac{1}{2} \mathbf{e}^d} = \frac{7}{12} \left(\langle \phi \rangle_{\mathbf{i}} + \langle \phi \rangle_{\mathbf{i} + \mathbf{e}^d} \right) - \frac{1}{12} \left(\langle \phi \rangle_{\mathbf{i} - \mathbf{e}^d} + \langle \phi \rangle_{\mathbf{i} + 2\mathbf{e}^d} \right) + O(h^4),$$

$$(2.5) \quad \left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\mathbf{i} + \frac{1}{2} \mathbf{e}^d} = \frac{1}{12h} \left(15 \langle \phi \rangle_{\mathbf{i} + \mathbf{e}^d} - 15 \langle \phi \rangle_{\mathbf{i}} - \langle \phi \rangle_{\mathbf{i} + 2\mathbf{e}^d} + \langle \phi \rangle_{\mathbf{i} - \mathbf{e}^d} \right) + O(h^4).$$

The discrete FV approximations of the operators, (2.2b) and (2.2c), are derived by substituting (2.4) and (2.5) and adding additional terms for face averages of the product, $\langle \phi u_d \rangle$. See Appendix B for the complete derivation. The resulting discrete FV approximations act only on cell-averaged quantities:

$$(2.6a) \quad L_{\text{adv}}(\langle \phi \rangle, t)_{\mathbf{i}} = -\frac{1}{h} \sum_{d=1}^D \sum_{\pm=+,-} \left(\pm \langle \phi \rangle_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d} \langle u_d \rangle_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d} \pm \frac{h^2}{12} \sum_{d' \neq d} (G_{d'}^{\perp} \phi)_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d} (G_{d'}^{\perp} u_d)_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d} \right) + O(h^4),$$

$$(2.6b) \quad (G_{d'}^{\perp} q)_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d} = \frac{1}{2h} \left(\langle q \rangle_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d + \mathbf{e}^{d'}} - \langle q \rangle_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}^d - \mathbf{e}^{d'}} \right),$$

$$(2.7) \quad \langle \Delta \phi \rangle_{\mathbf{i}} = \frac{1}{12h^2} \left(-30D \langle \phi \rangle_{\mathbf{i}} + \sum_{d=1}^D \sum_{\pm=+,-} (16 \langle \phi \rangle_{\mathbf{i} \pm \mathbf{e}^d} - \langle \phi \rangle_{\mathbf{i} \pm 2\mathbf{e}^d}) \right) + O(h^4).$$

Note that (2.6) is the same formula as was used in [7, 14], while (2.7) is different from the fourth-order ‘‘Mehrstellen’’ stencil derived in [1], which requires a correction to the leading error term that is not generally applicable to variable coefficient problems.

Since the discrete divergence is calculated from face averages as in (1.1), a face average over any interior edge participates in the calculation exactly twice: once for the cell at the lower side and once for the cell at the higher side. Hence the sum of discrete divergence over the whole computational domain vanishes if the corresponding fluxes (normal velocity or gradient) sum up to zero over the domain boundary. This defines the *conservation property* of the FV formulation.

2.2. Boundary conditions. At nonperiodic domain boundaries, “ghost cells” are used with the standard interior stencils to evaluate the advection and Laplacian operators. For example, let $\langle \phi \rangle_{\mathbf{i}}$ denote the cell-averaged scalar of the interior cell abutting a high-side boundary, and let $\langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d}$ & $\langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d}$ denote those of the two ghost cells to be calculated. In the case of Dirichlet boundary conditions, we specify $\langle \phi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}$ and use an extrapolation based on the *integrals* of interpolating polynomials; this leads to the following ghost cell formulas for the fourth-order case:

$$(2.8a) \quad \langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} = \frac{1}{3} \left(-13 \langle \phi \rangle_{\mathbf{i}} + 5 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 12 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^4),$$

$$(2.8b) \quad \langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} = \frac{1}{3} \left(-70 \langle \phi \rangle_{\mathbf{i}} + 32 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 7 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 48 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^4).$$

A fifth-order interpolation leads to

(2.9a)

$$\langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} = \frac{1}{12} \left(-77 \langle \phi \rangle_{\mathbf{i}} + 43 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 17 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 3 \langle \phi \rangle_{\mathbf{i}-3\mathbf{e}^d} + 60 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^5)$$

(2.9b)

$$\langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} = \frac{1}{12} \left(-505 \langle \phi \rangle_{\mathbf{i}} + 335 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 145 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 27 \langle \phi \rangle_{\mathbf{i}-3\mathbf{e}^d} + 300 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^5).$$

Similarly, for Neumann-type boundary conditions we specify $\langle \frac{\partial \phi}{\partial x_d} \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}$, and a fourth-order interpolation yields

$$(2.10a) \quad \langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} = \frac{1}{11} \left(9 \langle \phi \rangle_{\mathbf{i}} + 3 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 12 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^4),$$

$$(2.10b) \quad \langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} = \frac{1}{11} \left(-30 \langle \phi \rangle_{\mathbf{i}} + 56 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 15 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 48 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^4),$$

while a fifth-order interpolation yields

(2.11a)

$$\langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} = \frac{1}{10} \left(5 \langle \phi \rangle_{\mathbf{i}} + 9 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 5 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + \langle \phi \rangle_{\mathbf{i}-3\mathbf{e}^d} + 12 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^5),$$

(2.11b)

$$\langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} = \frac{1}{10} \left(-75 \langle \phi \rangle_{\mathbf{i}} + 145 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - 75 \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} + 15 \langle \phi \rangle_{\mathbf{i}-3\mathbf{e}^d} + 60 \langle g \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \right) + O(h^5).$$

We use fourth-order formulas (2.8) and (2.10) for the advection operator, and fifth-order formulas (2.9) and (2.11) for the Laplacian operator.

2.3. Nested refinement. The uniform grid discretization above can also be extended to structured adaptive mesh refinement (AMR), i.e., a locally refined, nested hierarchy of rectangular grids. Our notation is based on previous $O(h^2)$ structured AMR work (see [13]), but we will reiterate parts of the notation for the purpose of explaining the present algorithm.

On a family of nested discretizations of a rectangular domain $\{\Gamma^l\}_{l=0}^{l_{\max}}$, $\Gamma^l \subset \mathbb{Z}^D$, control volumes $V_{\mathbf{i}}^l = [\mathbf{i}h^l, (\mathbf{i} + \mathbb{1})h^l]$ are represented with multi-indices $\mathbf{i} \in \Gamma^l$, with h^l

denoting the uniform mesh spacing and $n_{\text{ref}}^l = h^{l-1}/h^l$ the refinement ratio between levels Γ^l and Γ^{l-1} . To relate geometric regions and variables on different levels of the hierarchy to one another, we define a coarsening operator for indices,

$$\mathcal{C}_{n_{\text{ref}}^l}(\mathbf{i}) = \left(\left\lfloor \frac{i_1}{n_{\text{ref}}^l} \right\rfloor, \dots, \left\lfloor \frac{i_D}{n_{\text{ref}}^l} \right\rfloor \right).$$

Similarly, we define the refining operator as $\mathcal{C}_{n_{\text{ref}}^l}^{-1}(\Gamma^{l-1}) = \Gamma^l$.

At any given point in time, our computed solution will be defined on the computational domain $\{\Omega^l\}_{l=0}^{l_{\text{max}}}$, $\Omega^l \subset \Gamma^l$, $\mathcal{C}_{n_{\text{ref}}^l}(\Omega^l) \subset \Omega^{l-1}$, $\Omega^0 = \Gamma^0$. The Ω^l 's are assumed to satisfy ‘‘proper nesting’’ conditions: that $\mathcal{C}_{n_{\text{ref}}^l}^{-1}(\mathcal{C}_{n_{\text{ref}}^l}(\Omega^l)) = \Omega^l$ (coarsening or refining levels does not change the region covered), and that there are at least s_n points in any direction in Ω^l separating $\mathcal{C}_{n_{\text{ref}}^{l+1}}(\Omega^{l+1})$ (the finer level coarsened to l) from $\mathcal{C}_{n_{\text{ref}}^l}^{-1}(\Omega^{l-1}) - \Omega^l$ (the coarser level refined to l). In the case of periodic domains, this condition is assumed to hold with respect to the periodic extensions of the grids. In the present work, we assume $s_n = 3$ to support the interpolation stencils below.

The primary dependent variables on each level are defined only on the part of Ω^l not covered by finer grids:

$$(2.12a) \quad \phi^{\text{comp}} = \{\phi^l\}_{l=0}^{l_{\text{max}}}, \quad \phi^l : \Omega_{\text{valid}}^l \rightarrow \mathbb{R},$$

$$(2.12b) \quad \Omega_{\text{valid}}^l = \Omega^l - \mathcal{C}_{n_{\text{ref}}^{l+1}}^{-1}(\Omega^{l+1}), \quad \Omega^{\text{comp}} = \bigcup_{l=0}^{l_{\text{max}}} \Omega_{\text{valid}}^l.$$

We generalize the definitions of the operators L_{diff} , L_{adv} to operate on ϕ^{comp} , but with the flux calculations modified to account for the changes in grid resolution at refinement boundaries. First, we extend the ϕ^l 's to all of Ω^l by averaging down from the finer levels:

$$(2.13) \quad \langle \phi \rangle_{\mathbf{i}}^l = \frac{1}{(n_{\text{ref}}^l)^D} \sum_{\mathbf{j} \in \mathcal{C}_{n_{\text{ref}}^l}^{-1}(\{\mathbf{i}\})} \langle \phi \rangle_{\mathbf{j}}^{l+1} \text{ on } \mathcal{C}_{n_{\text{ref}}^{l+1}}(\Omega^{l+1}), \quad l = l_{\text{max}} - 1, \dots, 0.$$

Then, for each level, we compute approximate values of $\langle \phi \rangle$ over control volumes outside of Ω^l to facilitate the evaluation of L_{adv} and L_{diff} on all of Ω_{valid}^l . The constrained least-squares approach described in [14] is adopted to derive these ‘‘coarse-fine’’ interpolation stencils.

Specifically, we interpolate $O(h^{P+1})$ -order accurate values for $\mathbf{j} \in \mathcal{C}_{n_{\text{ref}}^l}^{-1}(\{\mathbf{i}\})$ by computing a polynomial interpolant of the form

$$(2.14) \quad \mathcal{P}_{\mathbf{i}}(\mathbf{x}) = \sum_{\mathbf{p}: p_d \geq 0, p_1 + \dots + p_D \leq P} a_{\mathbf{p}} \mathbf{x}^{\mathbf{p}}, \quad \mathbf{x}^{\mathbf{p}} = \prod_{d=1}^D x_d^{p_d},$$

subject to the constraint that the coarse $\langle \phi \rangle^l$ are the averages of fine values $\langle \phi \rangle_{\mathbf{j}}^{l+1}$ as required by (2.13). The coefficients $a_{\mathbf{p}}$ are computed as the least-squares approximation to an overdetermined system of linear equations $\langle \phi \rangle_{\mathbf{j}} = \langle \mathcal{P}_{\mathbf{i}} \rangle_{\mathbf{j}}$, $\mathbf{j} \in \mathcal{N}(\mathbf{i})$, where $\mathcal{N}(\mathbf{i})$ is a collection of nearby points in Ω^l . The number of the chosen points exceeds the number of terms in the sum in (2.14); the points must also be chosen so

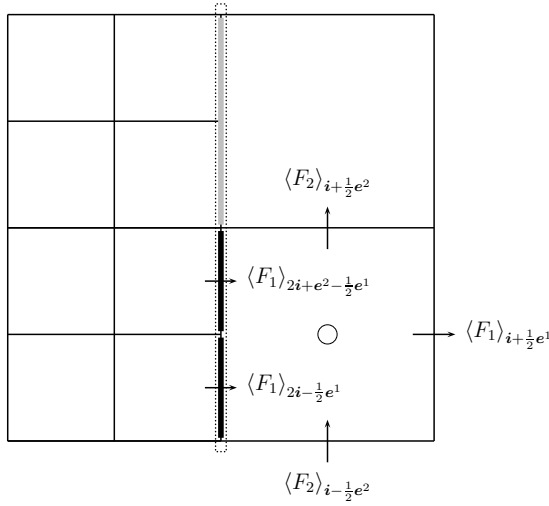


FIG. 2.1. Two-dimensional (2D) refluxing example for the two-level divergence operator. Cell \mathbf{i} is marked by \circ . Its coarse-fine interface $\chi^-(\mathbf{i} - \frac{1}{2}\mathbf{e}^1) = \{2\mathbf{i} - \frac{1}{2}\mathbf{e}^1, 2\mathbf{i} + \mathbf{1} - \frac{3}{2}\mathbf{e}^1\}$ is marked by thick line segments.

that the resulting linear system is of maximal rank and, when appropriately scaled, well-conditioned.

For computing the fluxes in L_{adv} , we use the fourth-order accurate stencil ($P = 3$) to fill ghost cell values, as used in [14]. For computing the fluxes in L_{diff} , we use a generalization of the same stencil that is fifth-order accurate ($P = 4$). For example, the stencil away from the domain boundary is

$$(2.15) \quad \mathcal{N}(\mathbf{i}) = \mathbf{i} + \left([-\mathbf{1}, \mathbf{1}] \cup \left(\bigcup_{\pm=+,-} \bigcup_{d=1}^D \{ \pm 2\mathbf{e}^d + s\mathbf{e}^{d'} : d' \neq d, s = 0, \pm 1 \} \right) \right).$$

See [22] for a more general approach.

As in [13], after the fluxes on all the levels are computed in this fashion, we replace the fluxes on faces of level l control volumes that are adjacent to Ω^{l+1} by the average of the fluxes at level $l + 1$. Figure 2.1 illustrates an example with $D = 2$, $d = 1$, $\mathbf{i} \in \Omega_{\text{valid}}^l$, and $\mathbf{i} - \mathbf{e}^d \in \mathcal{C}_{n_{\text{ref}}^l}(\Omega_{\text{valid}}^{l+1})$: the flux computed with the uniform grid stencils in (2.2b) and (2.2c) is replaced with

$$(2.16) \quad \langle F_d \rangle_{\mathbf{i} - \frac{1}{2}\mathbf{e}^d}^l = \frac{1}{(n_{\text{ref}}^l)^{D-1}} \sum_{\mathbf{i}^{l+1} - \frac{1}{2}\mathbf{e}^d \in \chi^-} \langle F_d \rangle_{\mathbf{i}^{l+1} - \frac{1}{2}\mathbf{e}^d}^{l+1},$$

where $\chi^- = [\mathbf{i}n_{\text{ref}}^l - \frac{1}{2}\mathbf{e}^d, (\mathbf{i} + \mathbf{1} - \mathbf{e}^d)n_{\text{ref}}^l - \mathbf{1} + \frac{1}{2}\mathbf{e}^d]$.

3. Time integration. Using the nested refinement spatial discretization described above, we obtain a system of ODEs,

$$(3.1) \quad \frac{d \langle \phi \rangle^{\text{comp}}}{dt} = L_{\text{adv}}(\langle \phi \rangle^{\text{comp}}, t) + L_{\text{diff}}(\langle \phi \rangle^{\text{comp}}) + \langle f(t) \rangle,$$

and then use the “method of lines” to advance all of the discrete variables $\langle \phi \rangle^{\text{comp}}$ in the AMR hierarchy.

Our approach is to use an additive, implicit-explicit Runge–Kutta method introduced by Kennedy and Carpenter [12] for convection-diffusion-reaction equations. In particular, we select ARK4(3)6L[2]SA, a six-stage, fourth-order accurate, L -stable method, with an explicit treatment for the advection operator L_{adv} and source term f and an implicit treatment for the diffusion term L_{diff} . Although this method has been shown to suffer from the well-known problem of order reduction in the stiff limit [12, 15], it does have a good region of stability for the explicit terms, as shown in the next section.

We integrate (3.1) by setting $\langle \phi \rangle^{(1)} = \langle \phi \rangle^n$, and then we calculate the five subsequent stage values $\langle \phi \rangle^{(s)}$, $s = 2, 3, 4, 5, 6$, by solving

$$(3.2) \quad (I - \Delta t \gamma L_{\text{diff}}) \langle \phi \rangle^{\text{comp},(s)} = \langle \phi \rangle^n + \Delta t \tilde{L},$$

where

$$\tilde{L} = \sum_{j=1}^{s-1} a_{s,j}^{[E]} \left(L_{\text{adv}} \left(\langle \phi \rangle^{\text{comp},(j)}, t^{(j)} \right) + \langle f \rangle (t^{(j)}) \right) + \sum_{j=1}^{s-1} a_{s,j}^{[I]} L_{\text{diff}} \left(\langle \phi \rangle^{\text{comp},(j)} \right),$$

$t^{(s)} = t^n + c_s \Delta t$, and $\gamma = a_{s,s}^{[I]}$ is a constant for all stages. The coefficients $\{a_{i,j}^{[E]}\}$, $\{a_{i,j}^{[I]}\}$, $\{b_j\}$, $\{c_j\}$ are defined in [12]; for completeness they are also provided in Appendix C as decimal values [20].

At each intermediate stage $s = 2, 3, 4, 5, 6$, we must solve the Helmholtz-type linear system (3.2), with its right-hand side explicitly calculated from values of previous stages. As discussed in [19, p. 441], the negation of the fourth-order discrete operator $\langle \Delta \phi \rangle$ defined in (2.7) is of *essentially positive type* [3], which carries the essential properties of M-matrices such as positive definiteness. Consequently, the Helmholtz-type operator $(I - \Delta t \gamma L_{\text{diff}})$ also has eigenvalues of positive real parts; this can be verified by Fourier analysis. A standard multigrid method with Gauss–Seidel red-black relaxation [4] is employed in solving (3.2), with a V-cycle performed on each level of the AMR hierarchy. In the fourth-order case, writing the results of the red relaxation directly to the black points is incorrect, since the stencil as defined by (2.7) involves both red and black points. Instead we use an auxiliary storage to hold the results of the red relaxation in preparation for the ensuing black relaxation. As confirmed by the numerical tests in section 5, this works well for both the Laplacian operator and the Helmholtz-type operator.

Once all six stage values are known, the final calculation,

$$(3.3) \quad \langle \phi \rangle^{n+1} = \langle \phi \rangle^{(6)} + \Delta t \sum_{j=1}^6 \left(b_j - a_{6,j}^{[E]} \right) \left(L_{\text{adv}} \left(\langle \phi \rangle^{(j)}, t^{(6)} \right) + \langle f \rangle (t^{(6)}) \right),$$

provides the next time step value of $\langle \phi \rangle^{n+1}$, and the time integration procedure is repeated.

As the solution evolves in time, we allow the adaptive mesh grid hierarchy to evolve as well. At the end of each time step, the grid hierarchy can be refined to track emerging new features and/or coarsened to reduce computational expense. Typically the cells to be changed are “tagged” according to the evaluation of certain criteria; these criteria, specified by the user, are usually based on physical quantities or estimated errors. This dynamic grid generation is done in the standard fashion described in [2, 13], by averaging down from finer grids to coarser grids as the former disappear,

and interpolating conservatively from coarser grids to newly refined regions. The principal difference in the present AMR algorithm is the higher-order, conservative coarse-fine interpolation and its supporting multigrid-based solver, as described above and in section 2.3.

4. Analysis. In this work, *solution error* refers to the difference between the true and the computed solutions. In contrast, *truncation error* relates to approximating the advection and Laplacian operators, and it is caused by replacing continuous operators with discrete finite-difference stencils in forming the ODE system (3.1).

4.1. Error analysis. Away from the coarse-fine interface and nonperiodic physical domain boundaries, the truncation errors for these two operators are both $O(h^4)$; cf. (2.6) and (2.7). However, as discussed in the previous section, fourth- and fifth-order coarse-fine interpolations of the solution $\langle \phi \rangle$ are used, respectively, for evaluating the advection and diffusion operators near the coarse-fine interface. Hence the truncation error of both operators is $O(h^3)$ for control volumes near the coarse-fine interface. A similar statement holds for a nonperiodic physical domain boundary, due to the extrapolation formulas in section 2.2. These observations are consistent with the discussion in Appendix B that the truncation error is $O(h^3)$ at the coarse-fine interface due to refluxing. Nonetheless, we expect the 1-norm of the truncation error to be $O(h^4)$, since the truncation error is $O(h^3)$ only on a set of codimension 1.

For Poisson's equation, it is well known [10] that the accuracy of the solution error with respect to the ∞ -norm is one order higher than that of the truncation error, so long as the lower-order truncation error is restricted to a set of codimension 1. As for diffusion processes with large higher-order derivatives *and* high Reynolds number ($\text{Re} \geq 10^5$), the lower-order truncation errors on a set of codimension 1 might lead to solution errors of the same order [24, sect. 2]. However, since this work aims to address *moderately stiff* problems (with moderate to large ν), we expect the solution errors to be fourth-order accurate for all norm types.

4.2. Stability analysis. Using discrete Fourier analysis, we can convert the ODE system (3.1), with constant velocity \mathbf{u} and $\langle f \rangle = 0$ on a periodic domain, to a system of decoupled ODEs of the form

$$(4.1) \quad \frac{dy}{dt} = \lambda y = (\lambda^d + i\lambda^a) y,$$

where $\lambda^d, i\lambda^a$ are the eigenvalues of the diffusion and advection operators:

$$(4.2a) \quad \lambda^d = -4\frac{\nu}{h^2} \sum_{d=1}^D \sin^2 \frac{\theta_d}{2} \left(1 + \frac{1}{3} \sin^2 \frac{\theta_d}{2} \right),$$

$$(4.2b) \quad \lambda^a = \frac{|u_{d,\max}|}{h} \sum_{d=1}^D \sin \theta_d \left(1 + \frac{2}{3} \sin^2 \frac{\theta_d}{2} \right),$$

with $\theta_d \in (0, \pi)$. Note that in deriving (4.2) we have repeatedly applied the trigonometric identities $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and $\sin \theta = 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2}$. The maximum value of the scaled advection eigenvalue is then estimated as

$$(4.3) \quad (\lambda^a \Delta t)_{\max} = C_r D \left(\sin \theta_0 + \frac{2}{3} \sin \theta_0 \sin^2 \frac{\theta_0}{2} \right)_{\max} \approx 1.37222 DC_r,$$

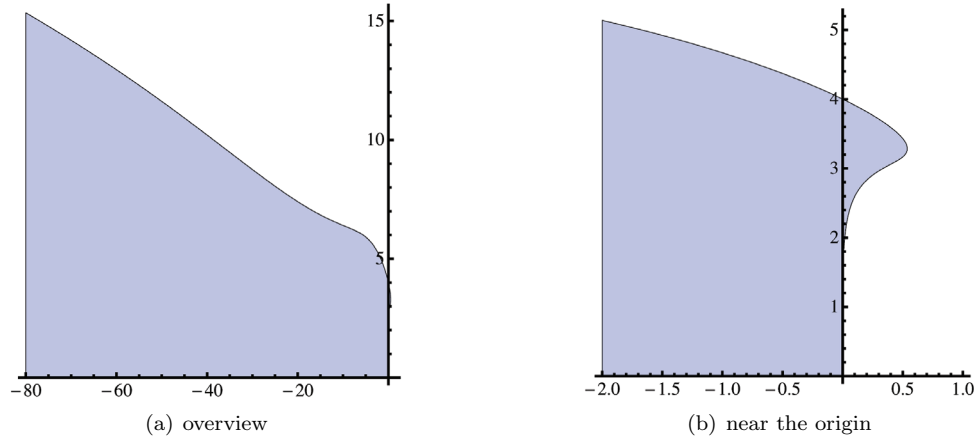


FIG. 4.1. Stability region of the proposed advection-diffusion solver on the complex plane: (a) for $\bar{\lambda}^a \in [0, 15]$, $\bar{\lambda}^d \in [-80, 0]$, and (b) detailed view of $\bar{\lambda}^a \in [0, 5]$, $\bar{\lambda}^d \in [-2, 1]$.

where the Courant number is defined as

$$(4.4) \quad C_r = \frac{|u_{d,\max}| \Delta t}{h}.$$

Generally, the *stability function* of the group of Runge–Kutta methods (see [6]) in section 3 is

$$(4.5) \quad R(\bar{\lambda}^d + i\bar{\lambda}^a) = \frac{\det(I - \bar{\lambda}^d A^{[I]} - i\bar{\lambda}^a A^{[E]} + (\bar{\lambda}^d + i\bar{\lambda}^a) \mathbb{1} \otimes \mathbf{b}^T)}{\det(I - \bar{\lambda}^d A^{[I]} - i\bar{\lambda}^a A^{[E]})},$$

where $\bar{\lambda}^a = \lambda^a \Delta t$, $\bar{\lambda}^d = \lambda^d \Delta t$, the vector \mathbf{b} and the matrices $A^{[E]}$, $A^{[I]}$ are the ARK coefficients in Appendix C.

The stability region $|R(z)| < 1$ of ARK4(3)6L[2]SA is plotted in Figure 4.1. Subplot (a) shows that the maximum stable Courant number increases as diffusion becomes stronger; subplot (b) shows that in the absence of diffusion, the scaled advection eigenvalue should be less than 4. Together with (4.3), the range of stable Courant numbers for our method is thus estimated as

$$(4.6) \quad C_r \leq \frac{2.91}{D}.$$

The above condition is the stability condition for pure advection and is verified by numerical experiments. In the case of strong diffusion, the stability condition might be much more lenient. Furthermore, when time-step size is changed adaptively to control error, which is commonly done with Runge–Kutta methods, (4.6) might be removed as superfluous.

Strictly speaking, the stability analysis in this section applies only to single level grids since Fourier analysis does not generalize to multiple levels; however, no instabilities are observed in any of our AMR tests so long as (4.6) is satisfied.

5. Results. In this section we demonstrate fourth-order convergence on test problems for Poisson’s equation and the advection-diffusion equation with forcing terms, and for an adaptive mesh and nontrivial advection velocity. The first problem is for Poisson’s equation, while all others are for the advection-diffusion equation.

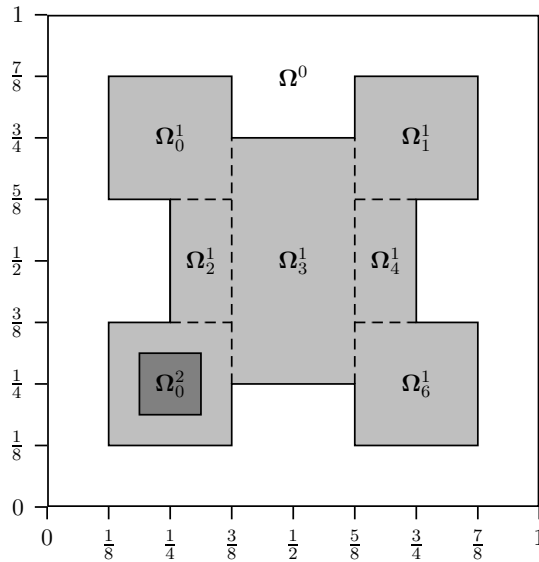


FIG. 5.1. A static locally refined grid hierarchy for Problem 1. The coarsest level ($\ell = 0$) covers the problem domain $[0, 1]^D$. The light gray area represents the intermediate level ($\ell = 1$), which is decomposed by the dashed lines into seven rectangles, all of which range from $3/8$ to $5/8$ in the third dimension. The finest level ($\ell = 2$), represented by the dark gray square, is obtained by shrinking Ω_5^1 to half of its length in each dimension.

5.1. Problem 1: Composite sinusoidal waves. We first address a test problem similar to one in [1] for Poisson's equation on the problem domain $[0, 1]^D$ with the exact solution

$$(5.1) \quad \phi(\mathbf{x}) = \sum_k \prod_{d=1}^D \sin(k\pi x_d),$$

where $\{k\}$ is a set of even integers representing different frequencies. Since the AMR Laplacian operator may depend on data from both coarser and finer levels, we use a static, three-level grid layout with a refinement ratio of $n_{\text{ref}} = 2$, as shown in Figure 5.1. The right-hand side of Poisson's equation is initialized exactly by using an analytical expression for $\langle \Delta \phi \rangle_{\mathbf{i}}$ derived from (5.1). The values of Dirichlet and Neumann boundary conditions in section 2.2 are calculated from sixth-order quadratures using the exact solution.

The truncation error and solution error are shown in Tables 5.1 and 5.2. Across different types of boundary conditions, there are no differences in errors or convergence rates. As discussed in section 4.1, although the truncation error of Poisson's equation is of third order, the solution can still be fourth-order accurate, even in the ∞ -norm sense. In addition, Figure 5.2 demonstrates satisfactory multigrid convergence, using only two relaxation pre- and postsweeps during the multigrid V-cycle across the locally refined grid hierarchy.

5.2. Problem 2: Traveling sinusoidal waves. For this test, we use an expression of the form $\phi = \prod_d \sin(k_d x_d - u_d t)$ as an exact solution to the advection-diffusion

TABLE 5.1

Truncation and solution errors of the AMR elliptic solver applied to Problem 1 in two and three dimensions, with periodic boundary conditions and wave numbers $\{k\} = \{2, 4\}$. The static locally refined hierarchy is shown in Figure 5.1.

Base grid h	1/64	Rate	1/128	Rate	1/256	Rate	1/512
2D truncation L_∞	3.57e-01	2.97	4.56e-02	2.99	5.73e-03	3.00	7.17e-04
2D solution L_∞	1.59e-05	4.14	9.00e-07	3.97	5.74e-08	3.98	3.63e-09
3D truncation L_∞	6.84e-01	3.00	8.56e-02	3.00	1.07e-02	3.00	1.34e-03
3D solution L_∞	1.68e-05	3.98	1.06e-06	4.00	6.66e-08	4.00	4.16e-09

TABLE 5.2

Truncation and solution errors of the AMR elliptic solver applied to Problem 1 in two dimensions, with Dirichlet- or Neumann-type boundary conditions and wave numbers $\{k\} = \{2, 4\}$. The static locally refined hierarchy is shown in Figure 5.1.

Base grid h	1/64	Rate	1/128	Rate	1/256	Rate	1/512
Dirichlet							
Truncation L_∞	3.80e-01	2.97	4.86e-02	2.99	6.10e-03	3.00	7.64e-04
Truncation L_1	1.46e-02	3.99	9.22e-04	4.00	5.76e-05	3.97	3.68e-06
Solution L_∞	1.50e-05	4.05	9.11e-07	3.98	5.78e-08	3.99	3.64e-09
Solution L_1	4.40e-06	4.00	2.74e-07	3.98	1.73e-08	3.99	1.09e-09
Neumann							
Truncation L_∞	3.80e-01	2.97	4.86e-02	2.99	6.10e-03	3.00	7.64e-04
Truncation L_1	1.14e-02	4.00	7.10e-04	4.00	4.42e-05	3.96	2.84e-06
Solution L_∞	1.72e-05	4.09	1.01e-06	3.94	6.61e-08	3.97	4.21e-09
Solution L_1	4.83e-06	3.93	3.17e-07	3.97	2.02e-08	3.99	1.28e-09

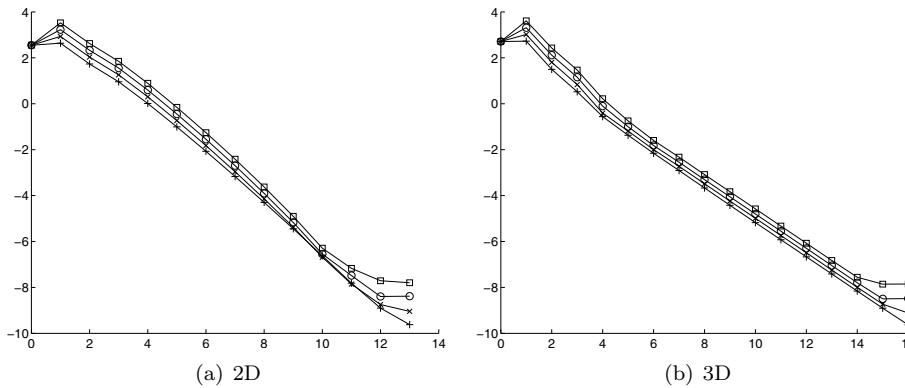


FIG. 5.2. Multigrid convergence for Problem 1. The horizontal and vertical axes are the number of iterations and base-10 logarithm of max-norm of the residual, respectively. +, x, o, □ represent the four grids from coarsest to finest.

equation with constant velocity, with corresponding forcing term in (2.1) as

$$\begin{aligned}
 (5.2) \quad f(\mathbf{x}, t) = & \left(\sum_d \nu k_d^2 \right) \prod_d \sin(k_d x_d - u_d t) \\
 & + \sum_d \left\{ u_d (k_d - 1) \cos(k_d x_d - u_d t) \prod_{d' \neq d} \sin(k_{d'} x_{d'} - u_{d'} t) \right\}.
 \end{aligned}$$

TABLE 5.3

Solution error and convergence rates for traveling waves test, Problem 2, on fixed two-level grids with periodic boundary conditions.

Base grid h	1/32	Rate	1/64	Rate	1/128
2D L_∞	1.20e-03	3.97	7.71e-05	3.99	4.87e-06
2D L_1	7.27e-06	3.95	4.72e-07	3.98	2.99e-08
2D L_2	8.59e-06	3.95	5.56e-07	3.98	3.52e-08
3D L_∞	2.36e-03	3.93	1.55e-04	3.98	9.80e-06
3D L_1	1.10e-06	3.94	7.21e-08	3.98	4.56e-09
3D L_2	1.43e-06	3.94	9.29e-08	3.98	5.87e-09

TABLE 5.4

Solution error and convergence rates for traveling waves test, Problem 2, on fixed two-level grids, with Dirichlet and Neumann boundary conditions applied to the low-side and high-side boundaries, respectively.

Base grid h	1/32	Rate	1/64	Rate	1/128
2D L_∞	1.00e-03	3.96	6.44e-05	3.99	4.06e-06
2D L_1	2.16e-04	3.95	1.39e-05	3.96	8.93e-07
2D L_2	3.07e-04	3.96	1.97e-05	3.97	1.25e-06
3D L_∞	2.53e-03	4.00	1.58e-04	4.02	9.75e-06
3D L_1	4.55e-04	3.97	2.90e-05	4.00	1.82e-06
3D L_2	6.43e-04	3.96	4.12e-05	3.99	2.59e-06

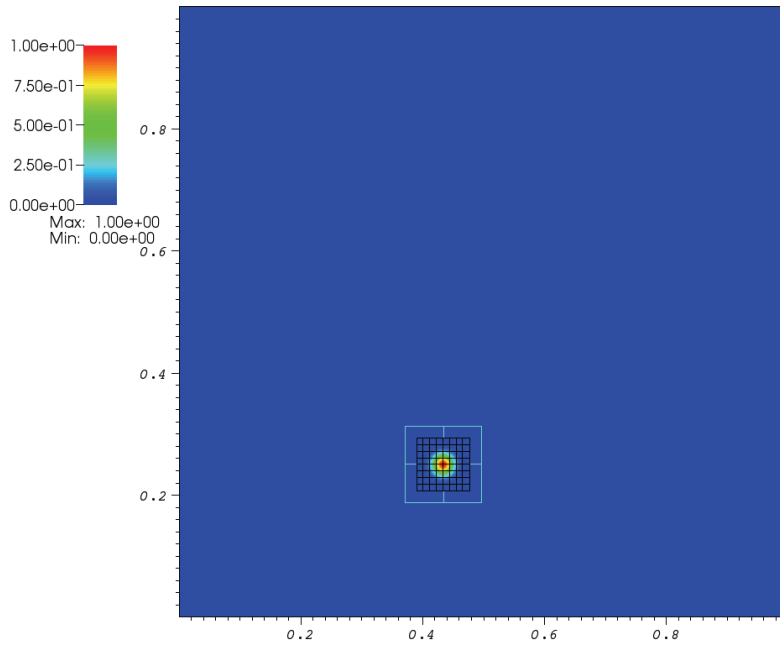
On a unit domain $[0, 1]^D$, we use a static locally refined hierarchy consisting of two levels, with the fine level covering $[\frac{1}{4}, \frac{3}{4}]^D$, and $n_{\text{ref}} = 4$. In both 2D and three-dimensional (3D) domains, the initial condition is calculated as the exact average $\langle \phi \rangle_{\mathbf{i}}$ evaluated at $t_0 = 0$, which is then advanced to $t_e = 1$ with the time step chosen such that $C_r = 1.0$. The other parameters are $\nu = 0.01$, $\mathbf{u} = (1.0, 0.5, 0.25)$, and $\mathbf{k} = (2\pi, 4\pi, 6\pi)$. Errors are calculated between the computed and analytic solutions; Tables 5.3 and 5.4 indicate fourth-order convergence of the solution in all norms and for all types of boundary conditions.

5.3. Problem 3: Gaussian patch in solid body rotation. Given the exact solution to the heat equation,

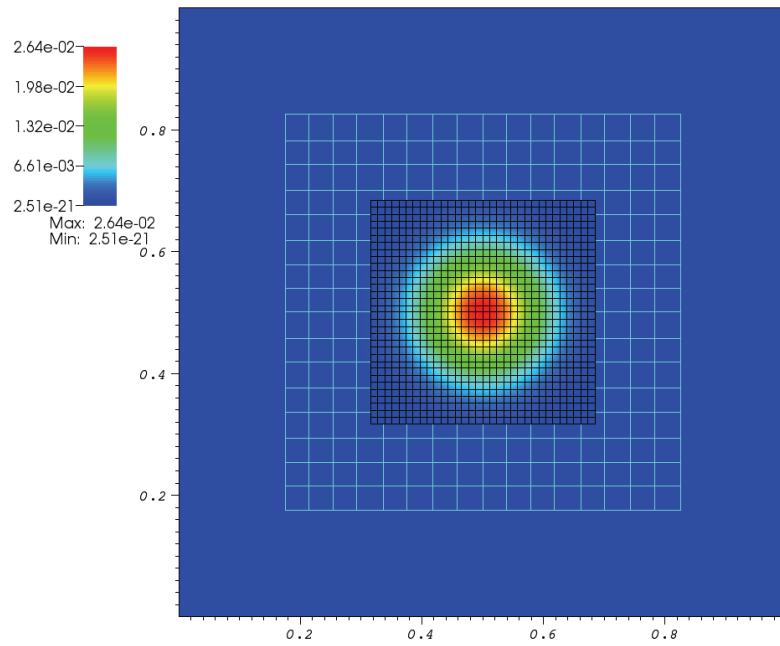
$$(5.3) \quad \phi(\mathbf{x}, t) = \left(\frac{t}{t_0} + 1 \right)^{-\frac{D}{2}} \exp \left(-\frac{|\mathbf{r} - \mathbf{r}_c|^2}{4\nu(t + t_0)} \right),$$

where \mathbf{r}_c is center of the patch, we can construct a solution to the advection-diffusion equation with the velocity field defined by solid body rotation. Although solid body rotation does not satisfy periodic boundary conditions, for short times the solution is near zero at the velocity discontinuity, and we can consider it an approximate solution. Choosing $t_0 = -r_0^2/(4\nu \ln(\epsilon))$ for $\epsilon \approx 10^{-16}$ guarantees that ϕ is less than 10^{-16} beyond $r_0 = 0.10$, and that $\phi = 1$ at the center of the patch at $t = 0$.

With the unit square $[0, 1]^2$ as the periodic domain, we adaptively refine the mesh by $n_{\text{ref}} = 4$ in the cells satisfying $|\langle \phi \rangle_{\mathbf{i}}| \geq 10^{-6}$ on level 0 to create level 1, and with $|\langle \phi \rangle_{\mathbf{i}}| \geq 10^{-3}$ on level 1 to create level 2. At the end of each time step, these cells are organized into finer-level patches, allowing them to move and grow in time as the solution advects and diffuses. The initial setup is shown in Figure 5.3(a); the center of the solid body rotation is at $(0, 1/2)$ with angular velocity $|\omega| = 2\pi$. The initial patch is centered at $(r, \theta) = (1/2, -\pi/6)$ relative to the center of rotation, with



(a) initial setup



(b) final result

FIG. 5.3. Initial setup and final results for Problem 3, a Gaussian patch in solid body rotation. On the three-level adaptive hierarchy, light blue and black boxes represent level 1 and level 2, respectively.

TABLE 5.5

Solution errors and convergence rates for Problem 3, a Gaussian profile in solid body rotation, with a dynamic three-level adaptive hierarchy using $n_{\text{ref}} = 4$.

Base grid h	1/64	Rate	1/128	Rate	1/256
Solution L_∞	9.38e-07	4.15	5.27e-08	4.41	2.48e-09
Solution L_1	7.99e-10	4.15	4.49e-11	4.03	2.76e-12
Solution L_2	2.18e-09	4.16	1.23e-10	4.11	7.09e-12

TABLE 5.6

AMR performance for the solid body rotation test. The single-level case has a uniform grid size of $h = \frac{1}{4096}$. The AMR hierarchy contains three levels with $n_{\text{ref}} = 4$, and the grid size on the coarsest level is $\frac{1}{256}$. CPU time is from 64 cores on Hopper Cray XE6 (<http://www.nersc.gov/systems/hopper-cray-xe6/>), a supercomputer at NERSC (National Energy Research Scientific Computing Center).

Case	CPU time (seconds)	Peak memory usage (MB)
Single-level grids	166818	4757.4
Three-level AMR	12486	222.7

initial radius $r_0 = 0.10$, beyond which the solution is initialized to zero. The Courant number based on the grid size of the finest level is 1.0 for all tests, and the numerical Reynolds number, Re_N , is $\|u_d\|_{\max} h_{\min} / \nu \approx 0.1$ for $\nu = 0.03$, based on the finest level $h_{\min} = 1/4096$. Figure 5.3(b) shows the resulting solution for $h = 1/256$ at time $t_e = 1/12$, when the patch is centered at $(1/2, 1/2)$. The error between the discrete and exact solutions is shown in the convergence study in Table 5.5; again, the results are fourth-order accurate in all norms.

We also compare the performance of AMR to that of a single level in Table 5.6, where the three-level AMR uses far fewer computational resources than the single-level grids; for example, the CPU time consumed by AMR is only 7.48% of that consumed by the single-level grids. As for the overhead incurred by message passing in parallel computing, our results show that it does influence the ratio of CPU time of AMR to that of the single grids, but not significantly, as the aforementioned ratio drops to 7.06% for four-core runs. Indeed, AMR tends to benefit less from a large number of cores than single-level grids do. Generally, the relative savings of AMR on computational resources depend drastically on problem-specific aspects such as the refinement ratio, the refinement criteria, and the number of levels. Hence the reader should not regard the results in Table 5.6 as universal estimates but rather just one particular example of the economy of AMR.

5.4. Problem 4: Gaussian patch in vortex shear. For this problem, we use a steady, divergence-free velocity field

$$(5.4) \quad \mathbf{u}(x, y) = a_V (\sin^2(\pi x) \sin(2\pi y), -\sin(2\pi x) \sin^2(\pi y)),$$

where $a_V = 0.1$ is a scaling parameter. See Figure 16 in [23] for a rendering of this velocity field.

On a periodic domain of the unit square $[0, 1]^2$, the advection-diffusion equation is advanced from $t_0 = 0$ to $t_e = 1/a_V$ on three successively refined adaptive hierarchies with three levels and $n_{\text{ref}} = 4$, so that finest-level resolutions are $1/512$, $1/1024$, and $1/2048$. The tagging criterion is the same as that used in section 5.3 and the Courant number is 1.0 based on the maximum velocity $\|u_d\|_{\max}$. Note that this ‘‘absolute-value’’ tagging criterion is not intrinsic to our algorithm but is merely an example of

various possible criteria. In our implementation, a user can customize this criterion according to the specific application.

The test is also carried out on a single-level grid with uniform grid size of $1/4096$; the results on this fine grid are then used as the “best” solution for calculating errors on the adaptive hierarchies. Based on this single-level grid, the numerical Reynolds number is $\text{Re}_N = \|u_d\|_{\max} h_{\min} / \nu \approx 2.44 \times 10^{-2}$ for $\nu = 0.001$. This parameter is chosen on purpose to test the *stiff* stability of the ARK4(3)6L[2]SA scheme. Two snapshots of the solution on the finest adaptive hierarchy are shown in Figure 5.4, at $t = 2$ and $t_e = 10$, and the corrected convergence results are shown in Table 5.7.

Let h_i ($i = 1, 2, \dots, N$) denote the grid size of the finest level in the i th hierarchy and N the total number of successively refined AMR hierarchies. Let $r = h_i/h_{i+1}$ be the *hierarchy* refinement ratio. To calculate the convergence rate p from numerical results of these hierarchies, we first write

$$\phi_h = \phi_{\text{Exact}} + a_p h^p + O(h^{p+1}),$$

where the coefficient a_p is independent of h . Standard Richardson extrapolation estimates the convergence rate via differences between successive hierarchies:

$$(5.5) \quad p \approx \log_r \frac{\|\phi_{h_i} - \phi_{h_{i+1}}\|}{\|\phi_{h_{i+1}} - \phi_{h_{i+2}}\|}.$$

Alternatively, a variant of Richardson extrapolation estimates the convergence rates by treating the finest solution as the “exact” solution. Define

$$(5.6a) \quad E_i = \|\phi_{h_i} - \phi_{h_{\min}}\|, \quad i = 1, 2, \dots, N,$$

$$(5.6b) \quad e_i = \frac{E_{i-1}}{E_i} = \frac{r^{p(N-i+2)} - 1}{r^{p(N-i+1)} - 1}, \quad i = 2, \dots, N,$$

where the grid size of the exact solution is $h_{\min} = h_N/r$, and e_i is the computed error ratio of two successive hierarchies. Equation (5.6b) can be solved for r^p , and then p . For $i = N$ and $i = N - 1$, we have

$$(5.7) \quad p \approx \log_r(e_N - 1), \quad p \approx \log_r \left(\sqrt{e_{N-1}^2 + 2e_{N-1} - 3} + e_{N-1} - 1 \right) - 1.$$

The difference between the estimated convergence rates by (5.5) and (5.7) is negligible in the asymptotic range, but might be substantial if the grid sizes are not small enough.

In this test, $N = 3$, $r = 2$, and we first compute the solution error by (5.6a) and then the convergence rates by (5.6b) and (5.7). As shown in Table 5.7, the resulting convergence rates are 4 or greater. Note that the error estimate at the finest grid is eight orders of magnitude smaller (10^{-8}) than the maximum solution value, which with the convergence rate would indicate that the convergence is in the asymptotic range. For the coarsest grid, $h = 1/512$, the rates are greater than the asymptotic fourth order, which might be an indication of ARK4(3)6L[2]SA losing accuracy due to stiffness of the system at the largest time step of the coarsest grid, as discussed in [12, 15]. At the coarser grid resolution and time step, the solution is slightly underresolved due to the rapid diffusion of the Gaussian peak.

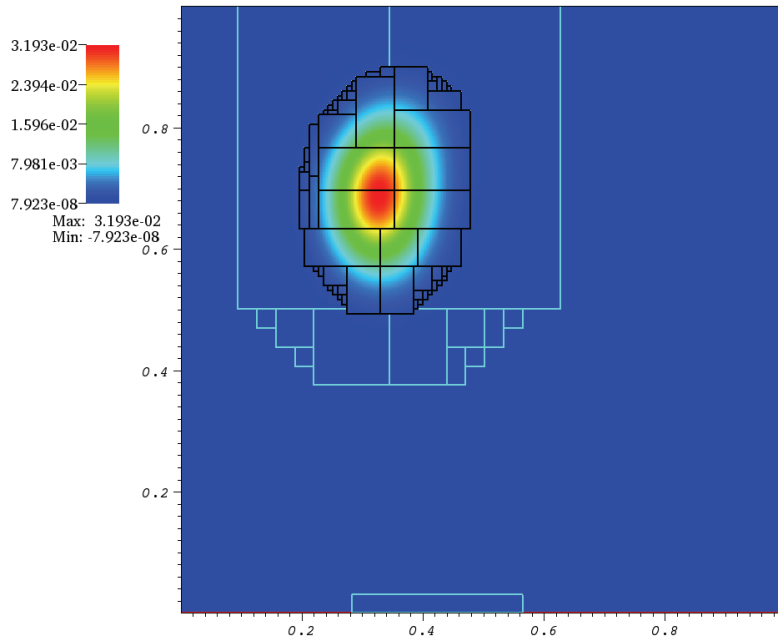
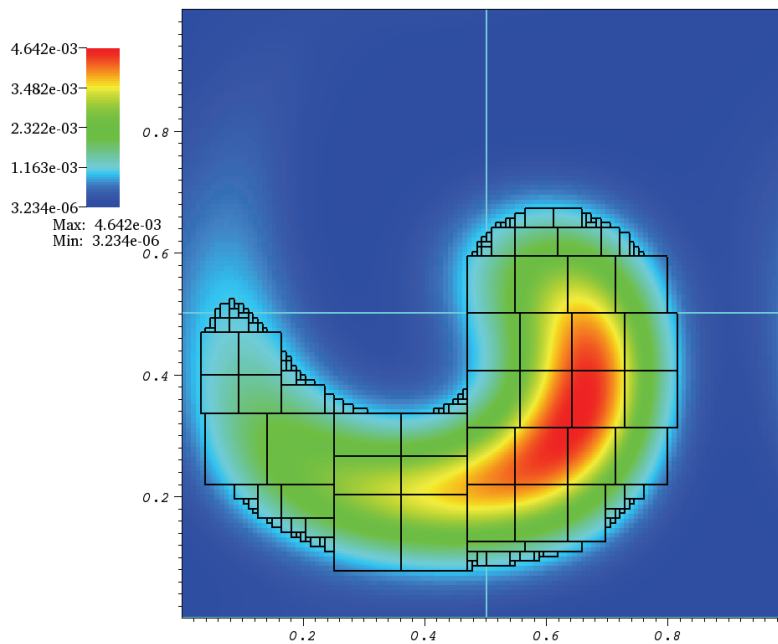
(a) Problem 4 solution at $t = 2$ (b) Problem 4 solution at $t = t_e = 10$

FIG. 5.4. Solution for Problem 4 at two time instances on a three-level adaptive hierarchy. The initial condition is a Gaussian blob, the same as that shown in Figure 5.3(a) except that its center is located at $(1/2, 3/4)$. As it diffuses, the advection velocity stretches and shears the solution at (a) $t = 2$ and (b) $t = 10$. Light blue and black boxes represent level 1 and level 2, respectively.

TABLE 5.7

Solution error (5.6a) and estimated convergence rates (5.7) of Problem 4 at $t_e = 10$ with $h_{\min} = 1/4096$.

Finest grid h	1/512	Rate	1/1024	Rate	1/2048
Solution L_∞	5.75e-06	5.19	1.58e-07	4.06	8.89e-09
Solution L_1	1.51e-08	5.01	4.71e-10	4.04	2.71e-11
Solution L_2	2.36e-08	5.05	7.11e-10	4.04	4.06e-11

6. Conclusions and future research. We have presented a fourth-order accurate algorithm for solving the advection-diffusion equation with AMR. The algorithm's AMR-enabled multigrid solver is general and can be applied to Poisson's equation and the heat equation in two and three dimensions, with a variety of boundary conditions, without modification. Although the spatial discretization was chosen to be fourth-order accurate, the algorithm—based on approximating fluxes using quadratures—is general and can be extended to higher orders in space and applied across a variety of generalized mixed hyperbolic, elliptic, and parabolic equations. The uniqueness of our approach lies in the combination of fourth-order FV stencils, AMR, the ARK time integrator, and multigrid solvers, all of which are general and capable of handling variable-coefficient problems and are fundamental components of solvers for more complicated PDEs.

Our immediate research concern is a fourth-order accurate adaptive algorithm for solving the incompressible Navier–Stokes equations, using a projection algorithm such as a generalization of the one in [13]. For periodic domains, we expect this to be a straightforward task; see [11, 16] for related work. However, in the presence of solid-wall boundaries, the lack of commutativity between the Laplacian operator and the Hodge projection operator is a nontrivial technical barrier [5] to be overcome in extending the present method to that case. In addition, we will need to adapt the method in [14] to support refinement in time and robust limiters near discontinuities.

Appendix A. From cell-averaged to face-averaged quantities.

Face-averaged and cell-averaged quantities can be expressed in terms of point values with second-order correction terms:

$$(A.1) \quad \langle \phi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} = \phi_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} + \frac{h^2}{24} \sum_{d' \neq d} \frac{\partial^2 \phi}{\partial x_{d'}^2} \Big|_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} + O(h^4)$$

and

$$(A.2) \quad \langle \phi \rangle_{\mathbf{i}} = \phi_{\mathbf{i}} + \frac{h^2}{24} \sum_{d=1}^D \frac{\partial^2 \phi}{\partial x_d^2} \Big|_{\mathbf{i}} + O(h^4),$$

where $\phi_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}$ and $\phi_{\mathbf{i}}$ denote the *point values* at the center of $A_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}$ and $V_{\mathbf{i}}$, respectively.

The derivation is as follows. Let

$$(A.3) \quad \Phi(x) = \int_{\xi}^x \phi(x') \, dx'$$

denote an indefinite integral with its lower limit ξ fixed. The average of ϕ over the interval $[i - \frac{1}{2}, i + \frac{1}{2}]h$ can be obtained by

$$(A.4) \quad h \langle \phi \rangle_i = \delta \Phi_i(y) = \Phi_{i + \frac{1}{2}} - \Phi_{i - \frac{1}{2}} = \Phi\left(\left(i + \frac{1}{2}\right)h\right) - \Phi\left(\left(i - \frac{1}{2}\right)h\right).$$

The first fundamental theorem of calculus yields

$$(A.5) \quad \phi(x) = \frac{\partial\Phi}{\partial x}.$$

Taylor expansions of $\Phi_{i+\frac{3}{2}}, \Phi_{i-\frac{1}{2}}, \Phi_{i+\frac{5}{2}}, \Phi_{i-\frac{3}{2}}$ at $(i + \frac{1}{2})h$ yield

$$(A.6) \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 2 & 4 & 8 & 16 \\ -2 & 4 & -8 & 16 \end{bmatrix} \begin{bmatrix} h \\ \frac{h^2}{2} \\ \frac{h^3}{6} \\ \frac{h^4}{24} \end{bmatrix} \begin{bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial^2\Phi}{\partial x^2} \\ \frac{\partial^3\Phi}{\partial x^3} \\ \frac{\partial^4\Phi}{\partial x^4} \end{bmatrix}_{i+\frac{1}{2}} = \begin{bmatrix} \delta\Phi_{i+1} \\ -\delta\Phi_i \\ \delta\Phi_{i+1} + \delta\Phi_{i+2} \\ -\delta\Phi_{i-1} - \delta\Phi_i \end{bmatrix} + O(h^5).$$

Hence

$$(A.7) \quad \begin{bmatrix} h\phi \\ h^2 \frac{\partial\phi}{\partial x} \\ h^3 \frac{\partial^2\phi}{\partial x^2} \\ h^4 \frac{\partial^3\phi}{\partial x^3} \end{bmatrix}_{i+\frac{1}{2}} = \begin{bmatrix} \frac{2}{3} & -\frac{2}{3} & -\frac{1}{12} & \frac{1}{12} \\ \frac{4}{3} & \frac{4}{3} & -\frac{1}{12} & -\frac{1}{12} \\ -1 & 1 & \frac{1}{2} & -\frac{1}{2} \\ -4 & -4 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta\Phi_{i+1} \\ -\delta\Phi_i \\ \delta\Phi_{i+1} + \delta\Phi_{i+2} \\ -\delta\Phi_{i-1} - \delta\Phi_i \end{bmatrix} + O(h^5).$$

Constructing an auxiliary matrix

$$(A.8) \quad \mathbf{M} = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & -1 & 1 & \\ & & 1 & -1 \end{bmatrix}$$

and adding $\mathbf{M}^{-1}\mathbf{M}$ into the middle of the right-hand side of (A.7), we have

$$(A.9) \quad \begin{bmatrix} \phi \\ h \frac{\partial\phi}{\partial x} \\ h^2 \frac{\partial^2\phi}{\partial x^2} \\ h^3 \frac{\partial^3\phi}{\partial x^3} \end{bmatrix}_{i+\frac{1}{2}} = \mathbf{P}^{(4)} \begin{bmatrix} \langle\phi\rangle_{i+1} \\ \langle\phi\rangle_i \\ \langle\phi\rangle_{i+2} \\ \langle\phi\rangle_{i-1} \end{bmatrix} + O(h^4),$$

where the fourth-order interpolation matrix $\mathbf{P}^{(4)}$ is

$$(A.10) \quad \mathbf{P}^{(4)} = \begin{bmatrix} \frac{7}{12} & \frac{7}{12} & -\frac{1}{12} & -\frac{1}{12} \\ \frac{5}{4} & -\frac{5}{4} & -\frac{1}{12} & \frac{1}{12} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -3 & 3 & 1 & -1 \end{bmatrix}.$$

The above procedures can be easily generated to higher-order accuracies; e.g., the

fifth-order interpolation matrix is

$$(A.11) \quad \mathbf{P}^{(5)} = \begin{bmatrix} \frac{49}{60} & \frac{9}{20} & -\frac{13}{60} & -\frac{1}{20} & \frac{1}{30} \\ \frac{5}{4} & -\frac{5}{4} & -\frac{1}{12} & \frac{1}{12} & 0 \\ -\frac{9}{4} & \frac{1}{2} & \frac{3}{2} & \frac{1}{4} & -\frac{1}{4} \\ -3 & 3 & 1 & -1 & 0 \\ 7 & -4 & -4 & 1 & 1 \end{bmatrix},$$

where the additional column is associated with $\langle \phi \rangle_{i+3}$. Note that the formulas of fourth order and fifth order coincide for $\frac{\partial \phi}{\partial x}$.

In a multidimensional space, averaging the first row of (A.9) over all other dimensions yields (2.4). Using the second row of (A.11) and averaging an equation similar to (A.9) yield (2.5).

Appendix B. Discrete advection and Laplacian operators.

We denote the cell center of V_i by $\mathbf{x}_i = (\mathbf{i} + \frac{1}{2}\mathbf{1})h$, and the face centers by $\mathbf{x}_{i \pm \frac{1}{2}\mathbf{e}^d} = \mathbf{x}_i \pm \frac{h}{2}\mathbf{e}^d$. Let $\mathbf{x}_c = \mathbf{x}_{i + \frac{1}{2}\mathbf{e}^d}$ be the center of $A_{i + \frac{1}{2}\mathbf{e}^d}$. Then the Taylor series of a function ϕ about \mathbf{x}_c can be expressed using standard multi-index notation [9]:

$$(B.1) \quad \phi(\mathbf{x}) = \sum_{|\mathbf{j}| \leq 3} \frac{1}{\mathbf{j}!} (\mathbf{x} - \mathbf{x}_c)^{\mathbf{j}} \phi^{(\mathbf{j})}(\mathbf{x}_c) + O(h^4) = \sum_{|\mathbf{j}| \leq 3} \frac{1}{\mathbf{j}!} \boldsymbol{\eta}^{\mathbf{j}} \phi^{(\mathbf{j})}(\mathbf{x}_c) + O(h^4),$$

where $\boldsymbol{\eta} = \mathbf{x} - \mathbf{x}_c$, so that $\eta_d = 0$ and $|\boldsymbol{\eta}| \approx O(h)$ on $A_{i + \frac{1}{2}\mathbf{e}^d}$.

Then the convolution of two functions $\phi, \psi : \mathbb{R}^D \rightarrow \mathbb{R}$ is

$$\begin{aligned} \phi(\mathbf{x})\psi(\mathbf{x}) &= \left(\sum_{|\mathbf{j}| \leq 3} \frac{1}{\mathbf{j}!} \boldsymbol{\eta}^{\mathbf{j}} \phi^{(\mathbf{j})}(\mathbf{x}_c) \right) \left(\sum_{|\mathbf{k}| \leq 3} \frac{1}{\mathbf{k}!} \boldsymbol{\eta}^{\mathbf{k}} \psi^{(\mathbf{k})}(\mathbf{x}_c) \right) + O(h^4) \\ &= \sum_{\mathbf{k}: |\mathbf{k}| \leq 3} \frac{1}{\mathbf{k}!} \boldsymbol{\eta}^{\mathbf{k}} \sum_{\mathbf{j}: \mathbf{j} \leq \mathbf{k}} \binom{\mathbf{k}}{\mathbf{j}} \phi^{(\mathbf{j})}(\mathbf{x}_c) \psi^{(\mathbf{k}-\mathbf{j})}(\mathbf{x}_c) + O(h^4), \end{aligned}$$

and the average over $A_{i + \frac{1}{2}\mathbf{e}^d}$ (dropping indices on A and evaluation at \mathbf{x}_c) is

$$\begin{aligned} \frac{1}{h^{D-1}} \int_A \phi \psi \, d\mathbf{x} &= \frac{1}{h^{D-1}} \int_A \sum_{\mathbf{k}: |\mathbf{k}| \leq 3} \frac{1}{\mathbf{k}!} \boldsymbol{\eta}^{\mathbf{k}} \sum_{\mathbf{j}: \mathbf{j} \leq \mathbf{k}} \binom{\mathbf{k}}{\mathbf{j}} \phi^{(\mathbf{j})} \psi^{(\mathbf{k}-\mathbf{j})} \, d\mathbf{x} + O(h^4) \\ &= \sum_{\mathbf{k}: |\mathbf{k}| \leq 3} \frac{1}{\mathbf{k}!} \left(\frac{1}{h^{D-1}} \int_A \boldsymbol{\eta}^{\mathbf{k}} \, d\mathbf{x} \right) \sum_{\mathbf{j}: \mathbf{j} \leq \mathbf{k}} \binom{\mathbf{k}}{\mathbf{j}} \phi^{(\mathbf{j})} \psi^{(\mathbf{k}-\mathbf{j})} + O(h^4). \end{aligned}$$

Note that if \mathbf{k} is odd in any component or $k_d \neq 0$, then the contribution of the integral of $\boldsymbol{\eta}^{\mathbf{k}}$ is 0. Hence, the only nonzero terms come from the choices of $\mathbf{k} = \mathbf{0}$, $\mathbf{j} = \mathbf{0}$ and

$\mathbf{k} = 2\mathbf{e}^{d'}, \mathbf{j} = \mathbf{0}, 2\mathbf{e}^{d'}$ with $d' \neq d$. Thus,

$$\begin{aligned} & \frac{1}{h^{D-1}} \int_A \phi \psi \, d\mathbf{x} \\ &= \phi \psi + \frac{h^2}{24} \sum_{d' \neq d} \left(\phi^{(2\mathbf{e}^{d'})} \psi + \psi^{(2\mathbf{e}^{d'})} \phi \right) + \frac{h^2}{12} \sum_{d' \neq d} \left(\phi^{(\mathbf{e}^{d'})} \psi^{(\mathbf{e}^{d'})} \right) + O(h^4) \\ &= \left(\phi + \frac{h^2}{24} \sum_{d' \neq d} \phi^{(2\mathbf{e}^{d'})} \right) \left(\psi + \frac{h^2}{24} \sum_{d' \neq d} \psi^{(2\mathbf{e}^{d'})} \right) + \frac{h^2}{12} \sum_{d' \neq d} \left(\phi^{(\mathbf{e}^{d'})} \psi^{(\mathbf{e}^{d'})} \right) + O(h^4) \\ &= \langle \phi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} \langle \psi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \left(\phi^{(\mathbf{e}^{d'})} \psi^{(\mathbf{e}^{d'})} \right) + O(h^4), \end{aligned}$$

where we have used (A.1) to convert the first two terms in parentheses to face-averaged quantities.

The last term, representing the product of “transverse gradients,” can be approximated with

$$G_{d'}^{\perp} \phi \Big|_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} = \frac{1}{2h} \left(\langle \phi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d + \mathbf{e}^{d'}} - \langle \phi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d - \mathbf{e}^{d'}} \right) = \frac{\partial \phi}{\partial x_{d'}} \Big|_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} + O(h^2),$$

leading to $O(h^4)$ overall for the average flux formula:

(B.2)

$$\langle \phi \psi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} = \langle \phi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} \langle \psi \rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \left(G_{d'}^{\perp} \phi \Big|_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} G_{d'}^{\perp} \psi \Big|_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} \right) + C_3(\mathbf{x}_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}) h^4 + O(h^5).$$

Substituting u_d for ψ yields the discrete advection operator in (2.6a); the $O(h^4)$ accuracy in (2.6a) is due to the cancellation caused by the symmetry of the difference stencils, i.e., $C_3(\mathbf{x}_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}) - C_3(\mathbf{x}_{\mathbf{i} - \frac{1}{2}\mathbf{e}^d}) = O(h)$.

As for the discrete Laplacian operator, we identify $\vec{F} = \nabla \phi$ in (1.1) to obtain

$$(B.3) \quad \langle \Delta \phi \rangle_{\mathbf{i}} = \frac{1}{h} \sum_{d=1}^D \left(\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} - \left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\mathbf{i} - \frac{1}{2}\mathbf{e}^d} \right).$$

From the second row of the fifth-order interpolation matrix (A.11), we have

$$\left\langle \frac{\partial \phi}{\partial x} \right\rangle_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d} = \frac{1}{12h} \left(15 \langle \phi \rangle_{\mathbf{i} + \mathbf{e}^d} - 15 \langle \phi \rangle_{\mathbf{i}} - \langle \phi \rangle_{\mathbf{i} + 2\mathbf{e}^d} + \langle \phi \rangle_{\mathbf{i} - \mathbf{e}^d} \right) + C_4(\mathbf{x}_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}) h^4 + O(h^5),$$

which leads to an equation identical to (2.7):

$$\langle \Delta \phi \rangle_{\mathbf{i}} = \frac{1}{12h^2} \sum_{d=1}^D \left(16 \langle \phi \rangle_{\mathbf{i} + \mathbf{e}^d} + 16 \langle \phi \rangle_{\mathbf{i} - \mathbf{e}^d} - 30 \langle \phi \rangle_{\mathbf{i}} - \langle \phi \rangle_{\mathbf{i} + 2\mathbf{e}^d} - \langle \phi \rangle_{\mathbf{i} - 2\mathbf{e}^d} \right) + O(h^4),$$

where we have used the fact that

$$C_4(\mathbf{x}_{\mathbf{i} + \frac{1}{2}\mathbf{e}^d}) - C_4(\mathbf{x}_{\mathbf{i} - \frac{1}{2}\mathbf{e}^d}) = h \frac{\partial C_4}{\partial x_d} \Big|_{\mathbf{x}_{\mathbf{i}}} + O(h^3).$$

From the derivation, it is clear that the fourth-order truncation error of (2.6) and (2.7) depends on the cancellation of the leading error functions C_3 and C_4 . However,

for a coarse cell close to the coarse-fine interface where the coarse flux is replaced by the average of the fine fluxes, the truncation error is only third-order accurate due to the lack of this cancellation.

Appendix C. ARK4 coefficients. Kennedy and Carpenter [12] studied a group of implicit-explicit Runge–Kutta schemes from third- to fifth-order accurate with the following form:

$$(C.1) \quad \begin{array}{c|c} \mathbf{c}^{[E]} & A^{[E]} \\ \hline & (\mathbf{b}^{[E]})^T \\ \hline & (\hat{\mathbf{b}}^{[E]})^T \end{array} = \begin{array}{c|ccccccc} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 2\gamma & 2\gamma & 0 & 0 & \cdots & 0 & 0 \\ c_3 & a_{31}^{[E]} & a_{32}^{[E]} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1}^{[E]} & a_{s-1,2}^{[E]} & a_{s-1,3}^{[E]} & \cdots & 0 & 0 \\ 1 & a_{s,1}^{[E]} & a_{s,2}^{[E]} & a_{s,3}^{[E]} & \cdots & a_{s,s-1}^{[E]} & 0 \\ \hline & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\ \hline & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \cdots & \hat{b}_{s-1} & \hat{b}_s \end{array},$$

$$(C.2) \quad \begin{array}{c|c} \mathbf{c}^{[I]} & A^{[I]} \\ \hline & (\mathbf{b}^{[I]})^T \\ \hline & (\hat{\mathbf{b}}^{[I]})^T \end{array} = \begin{array}{c|ccccccc} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 2\gamma & \gamma & \gamma & 0 & \cdots & 0 & 0 \\ c_3 & a_{31}^{[I]} & a_{32}^{[I]} & \gamma & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1}^{[I]} & a_{s-1,2}^{[I]} & a_{s-1,3}^{[I]} & \cdots & \gamma & 0 \\ 1 & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\ \hline & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\ \hline & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \cdots & \hat{b}_{s-1} & \hat{b}_s \end{array}.$$

The coefficients of the particular method used in this work, ARK4(3)6L[2]SA, are, in decimal form [20], $\gamma = 0.25$, $\mathbf{c}^{[E]} = \mathbf{c}^{[I]} = \mathbf{c}$, $\mathbf{b}^{[E]} = \mathbf{b}^{[I]} = \mathbf{b}$,

$$\mathbf{c} = (0.0, 0.5, 0.332, 0.62, 0.85, 1.0)^T,$$

$$b_1 = 0.15791629516167136,$$

$$b_2 = 0,$$

$$b_3 = 0.18675894052400077,$$

$$b_4 = 0.6805652953093346,$$

$$b_5 = -0.27524053099500667,$$

$$\begin{aligned}
a_{31}^{[E]} &= 0.221776, \\
a_{32}^{[E]} &= 0.110224, \\
a_{41}^{[E]} &= -0.04884659515311857, \\
a_{42}^{[E]} &= -0.17772065232640102, \\
a_{43}^{[E]} &= 0.8465672474795197, \\
a_{51}^{[E]} &= -0.15541685842491548, \\
a_{52}^{[E]} &= -0.3567050098221991, \\
a_{53}^{[E]} &= 1.0587258798684427, \\
a_{54}^{[E]} &= 0.30339598837867193, \\
a_{61}^{[E]} &= 0.2014243506726763, \\
a_{62}^{[E]} &= 0.008742057842904185, \\
a_{63}^{[E]} &= 0.15993995707168115, \\
a_{64}^{[E]} &= 0.4038290605220775, \\
a_{65}^{[E]} &= 0.22606457389066084,
\end{aligned}$$

$$\begin{aligned}
a_{31}^{[I]} &= 0.137776, \\
a_{32}^{[I]} &= -0.055776, \\
a_{41}^{[I]} &= 0.14463686602698217, \\
a_{42}^{[I]} &= -0.22393190761334475, \\
a_{43}^{[I]} &= 0.4492950415863626, \\
a_{51}^{[I]} &= 0.09825878328356477, \\
a_{52}^{[I]} &= -0.5915442428196704, \\
a_{53}^{[I]} &= 0.8101210538282996, \\
a_{54}^{[I]} &= 0.283164405707806.
\end{aligned}$$

Acknowledgments. The first author, Qinghai Zhang, would like to thank Dan Graves, Terry Ligoeki, Dan Martin, Peter McCorquodale, Peter Schwartz, David Trebotich, and Brian Van Straalen for helpful discussions during the process of this work.

REFERENCES

- [1] M. BARAD AND P. COLELLA, *A fourth-order accurate local refinement method for Poisson's equation*, J. Comput. Phys., 209 (2005), pp. 1–18.
- [2] M. J. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.
- [3] A. BRANDT, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.
- [4] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.
- [5] D. L. BROWN, R. CORTEZ, AND M. L. MINION, *Accurate projection methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 168 (2001), pp. 464–499.

- [6] M. P. CALVO, J. DE FRUTOS, AND J. NOVO, *Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations*, Appl. Numer. Math., 37 (2001), pp. 535–549.
- [7] P. COLELLA, M. R. DORR, J. A. F. HITTINGER, AND D. F. MARTIN, *High-order, finite-volume methods in mapped coordinates*, J. Comput. Phys., 230 (2011), pp. 2952–2976.
- [8] P. COLELLA AND P. R. WOODWARD, *The piecewise parabolic method (PPM) for gas-dynamical simulations*, J. Comput. Phys., 54 (1984), pp. 174–201.
- [9] L. C. EVANS, *Partial Differential Equations*, American Mathematical Society, Providence, RI, 1998.
- [10] H. JOHANSEN AND P. COLELLA, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys., 147 (1998), pp. 60–85.
- [11] S. Y. KADIOGLU, R. KLEIN, AND M. L. MINION, *A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics*, J. Comput. Phys., 227 (2009), pp. 2012–2043.
- [12] C. A. KENNEDY AND M. H. CARPENTER, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
- [13] D. F. MARTIN, P. COLELLA, AND D. GRAVES, *A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions*, J. Comput. Phys., 227 (2008), pp. 1863–1886.
- [14] P. MCCORQUODALE AND P. COLELLA, *A high-order finite-volume method for hyperbolic conservation laws on locally refined grids*, Commun. Appl. Math. Comput. Sci., 6 (2011), pp. 1–25.
- [15] M. L. MINION, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci., 1 (2003), pp. 471–500.
- [16] M. L. MINION, *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math., 48 (2004), pp. 369–387.
- [17] K. W. MORTON, *Numerical Solution of Convection-Diffusion Problems*, Appl. Math. Math. Comput. 12, Chapman & Hall, London, 1996.
- [18] M. STYNES, *Steady-state convection-diffusion problems*, Acta Numer., 14 (2005), pp. 445–508.
- [19] U. TROTTEBERG, C. W. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, San Diego, CA, 2001.
- [20] J. WILKENING, *Math 228A: Numerical Solutions of Differential Equations*, lecture notes, <http://math.berkeley.edu/~wilken/228A.F07/> (2007).
- [21] S. T. ZALESAK, *A physical interpretation of the Richtmyer two-step Lax-Wendroff scheme, and its generalization to higher spatial order*, in Advances in Computer Methods for Partial Differential Equations V, R. Vichnevetsky and R. S. Stepleman, eds., Proceedings of the Fifth IMACS International Symposium on Computer Methods for Partial Differential Equations, 1984, pp. 491–496.
- [22] Q. ZHANG, *High-order, multidimensional, and conservative coarse-fine interpolation for adaptive mesh refinement*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 3159–3168.
- [23] Q. ZHANG AND P. L.-F. LIU, *A new interface tracking method: The polygonal area mapping method*, J. Comput. Phys., 227 (2008), pp. 4063–4088.
- [24] Q. ZHANG AND P. L.-F. LIU, *Handling solid-fluid interfaces for viscous flows: Explicit jump approximation vs. ghost cell approaches*, J. Comput. Phys., 229 (2010), pp. 4225–4246.