LA-UR-16-22436 (Accepted Manuscript)

A hybrid incremental projection method for thermal-hydraulics applications

Christon, Mark A.
Bakosi, Jozsef
Nadiga, Balasubramanya T.
Berndt, Markus
Francois, Marianne M.
Stagg, Alan K.
Xia, Yidong
Luo, Hong

Provided by the author(s) and the Los Alamos National Laboratory (2017-01-12).

To be published in: Journal of Computational Physics

DOI to publisher's version: 10.1016/j.jcp.2016.04.061

Permalink to record: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-16-22436

Disclaimer:
Approved for public release. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

A Hybrid Incremental Projection Method for Thermal-Hydraulics Applications

Mark A. Christon[a,b], Jozsef Bakosi[c], Balasubramanya T. Nadiga[c], Markus Berndt[c,*], Marianne M. Francois[d], Alan K. Stagg[e], Yidong Xia[f], Hong Luo[g]

[a] *Computational Sciences International, Los Alamos, NM 87544, USA*
[b] *Guest Scientist – Computational Physics and Methods Group (CCS-2) Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[c] *Computational Physics and Methods Group (CCS-2) Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[d] *Methods and Algorithms Group (XCP-4) Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[e] *Modeling and Simulation Group Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA*
[f] *Department of Energy Resource Recovery & Sustainability Idaho National Laboratory, Idaho Falls, ID 83415, USA*
[g] *Department of Mechanical and Aerospace Engineering North Carolina State University, Raleigh, NC 27695, USA*

Abstract

A new second-order accurate, hybrid, incremental projection method for time-dependent incompressible viscous flow is introduced in this paper. The hybrid finite-element/finite-volume discretization circumvents the well-known Ladyzhenskaya-Babuška-Brezzi conditions for stability, and does not require special treatment to filter pressure modes by either Rhie-Chow interpolation or by using a Petrov-Galerkin finite element formulation. The use of a co-velocity with a high-resolution advection method and a linearly consistent edge-based treatment of viscous/diffusive terms yields a robust algorithm for a broad spectrum of incompressible flows. The high-resolution advec-

*Corresponding author.

Email addresses: machriston@gmail.com (Mark A. Christon), jbakosi@lanl.gov (Jozsef Bakosi), balu@lanl.gov (Balasubramanya T. Nadiga), berndt@lanl.gov (Markus Berndt), mmfran@lanl.gov (Marianne M. Francois), staggak@ornl.gov (Alan K. Stagg), Yidong.Xia@inl.gov (Yidong Xia), hluo2@ncsu.edu (Hong Luo)

tion method is shown to deliver second-order spatial convergence on mixed element topology meshes, and the implicit advective treatment significantly increases the stable time-step size. The algorithm is robust and extensible, permitting the incorporation of features such as porous media flow, RANS and LES turbulence models, and semi-/fully-implicit time stepping. A series of verification and validation problems are used to illustrate the convergence properties of the algorithm. The temporal stability properties are demonstrated on a range of problems with $2 \leq CFL \leq 100$. The new flow solver is built using the Hydra multiphysics toolkit. The Hydra toolkit is written in C++ and provides a rich suite of extensible and fully-parallel components that permit rapid application development, supports multiple discretization techniques, provides I/O interfaces, dynamic run-time load balancing and data migration, and interfaces to scalable popular linear solvers, e.g., in open-source packages such as HYPRE, PETSc, and Trilinos.

Keywords: FVM, FEM, incompressible flow, monotonicity-preserving advection, projection method, mixed-topology meshes, thermal-hydraulics

---

1. Introduction

The solution of the time-dependent incompressible Navier-Stokes equations poses multiple algorithmic challenges due to the divergence-free velocity constraint and the associated restrictions on the velocity and pressure discretization, as well as, the concomitant spatial and temporal resolution required to perform high Reynolds number flow simulations in complex geometries. Although fully-coupled solution strategies are available, the cost of such methods is prohibitive for time-dependent simulations where complex geometries and high-resolution grids are involved.

Projection methods methods have grown in popularity over the past 20 years due to their simplicity, ease of implementation, and computational performance. This is reflected by the volume of work published on the development of second-order accurate projection methods, see, for example, van Kan [1], Bell, et al. [2], Gresho, et al. [3–6], Almgren, et al. [7–9], Rider, et al. [10–13], Minion [14], Guermond and Quartapelle [15], Puckett, et al. [16], Sussman, et al. [17], and Knio, et al. [18], and Christon, et al. [19, 20]. The numerical performance of projection methods has been considered by Brown and Minion [21, 22], Wetton [23], Guermond [24, 25], Guermond and Quartapelle [26, 27], and Almgren et al. [9].

In the hybrid projection algorithm, we select a cell-centered finite-volume discretization for all transported variables, and a node-centered finite-element method for the pressure. This choice is important because it circumvents the well-known Ladyzhenskaya-Babuška-Brezzi stability conditions – see [28–31]. That is, the spatial discretization avoids special treatment of pressure modes, and does not require Rhie-Chow interpolation or the use of a Petrov-Galerkin finite element formulation with "tune-able" parameters. A high-resolution advection algorithm that is monotonicity preserving and exhibits excellent dispersive characteristics was developed for the hybrid projection algorithm. The advection and viscous/diffusive terms in the conservation equations are treated with an edge-based implementation. In addition, the viscous/diffusive terms are discretized using an edge-based least-squares method that is linearly consistent on arbitrary unstructured mixed topology meshes. The hybrid projection algorithm is flexible and extensible and can can be extended to handle conjugate heat transfer, fluid-structure interaction (FSI), volume-tracking for immiscible fluids, and multiphase flow problems.

The rest of the paper is organized as follows. §2 presents the formulation of the hybrid second-order accurate approximate incremental projection solver and the associated numerics. Details on the monotonicity preserving advection method are presented in §3. A series of prototypical verification and validation problems are discussed in §4, followed by summary and conclusions in §5.

2. Hybrid Incremental Projection Method

The Navier-Stokes equations for an incompressible fluid are

$$\nabla \cdot \mathbf{v} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = \nabla \cdot (-p\mathbf{I} + \boldsymbol{\tau}) + \rho \mathbf{f} \tag{2}$$

defined in the domain Ω with boundary Γ in the time interval $[0, T]$. Here, ρ is the mass density, $\mathbf{v} = (v_x, v_y, v_z)^T$ is the velocity, p is the fluid pressure, $\boldsymbol{\tau}$ is the deviatoric stress, and $\mathbf{f}$ is the body force per unit mass. A constitutive equation relates the deviatoric stress and the strain rate, $\boldsymbol{\tau} = 2\mu\mathbf{S}$, where $\mathbf{S} = \frac{1}{2}[\nabla\mathbf{v} + (\nabla\mathbf{v})^T]$.

The momentum equations, Eq. (2), are subject to boundary conditions that consist of prescribed velocity $\mathbf{v}(\mathbf{x}, t) = \hat{\mathbf{v}}(\mathbf{x}, t)$ on Γ_1, or prescribed traction $\{-p\mathbf{I} + 2\mu\mathbf{S}\}\cdot\mathbf{n} = \hat{\mathbf{f}}(\mathbf{x}, t)$ on Γ_2 where $\Gamma = \Gamma_1 \cup \Gamma_2$ is the domain boundary

with outward normal $\mathbf{n}$. A detailed discussion of boundary conditions for the incompressible Navier-Stokes equations may be found in [32].

Velocity initial conditions are prescribed as $\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}^0(\mathbf{x})$. For a mathematically well-posed incompressible flow problem, the prescribed initial and boundary conditions on the velocity field must satisfy

$$\nabla \cdot \mathbf{v}^0 = 0 \tag{3}$$

$$\mathbf{n} \cdot \mathbf{v}(\mathbf{x}, 0) = \mathbf{n} \cdot \mathbf{v}^o(\mathbf{x}) \tag{4}$$

If $\Gamma_2 = 0$, i.e., enclosure flows with $\mathbf{n} \cdot \mathbf{v}$ prescribed on all surfaces, then

$$\int_\Gamma \mathbf{n} \cdot \mathbf{v}^0 d\Gamma = 0 \tag{5}$$

must also be satisfied.

The design of the hybrid projection algorithm is constructed using edge-based algorithms for both advective and viscous/diffusive terms. The use of edge-based algorithms for advection (whether cell-centered or node-centered) rely on topological constructs for primal and dual grids. Each element of the primal grid consists of an ordered set of vertices connected by cell edges or faces. Figure 1 illustrates a primal grid and the associated dual grids. Each vertex of the dual grid is associated with a primal mesh cell. Edges generically connect a pair of vertices in either the primal or the dual grid. Typical finite element mesh descriptions rely on identification of elements or cells where vertices and their coordinates are associated with a unique cell. For the hybrid projection algorithm, we equip a standard finite element mesh description with a dual-edge data structure. This permits treating edges as a primary mesh entity and the implementation of algorithms that are essentially independent of the element topology and dimensionality.

Turning to the hybrid algorithm, the premise of projection algorithms is to attempt to provide a legitimate way to decouple the pressure and velocity fields to provide an efficient computational method for transient incompressible flow simulations. The projection step can be derived by decomposing the velocity field into divergence-, and curl-free components using a Helmholtz decomposition,

$$\mathbf{v}^* = \mathbf{v} + \frac{1}{\rho}\nabla\lambda, \tag{6}$$

where $\mathbf{v}^*$ is an arbitrary velocity field, $\mathbf{v}$ is the divergence-free, while $\nabla\lambda$ is curl-free, since $\nabla \times \nabla\lambda = 0$. The projection removes the dilatational part

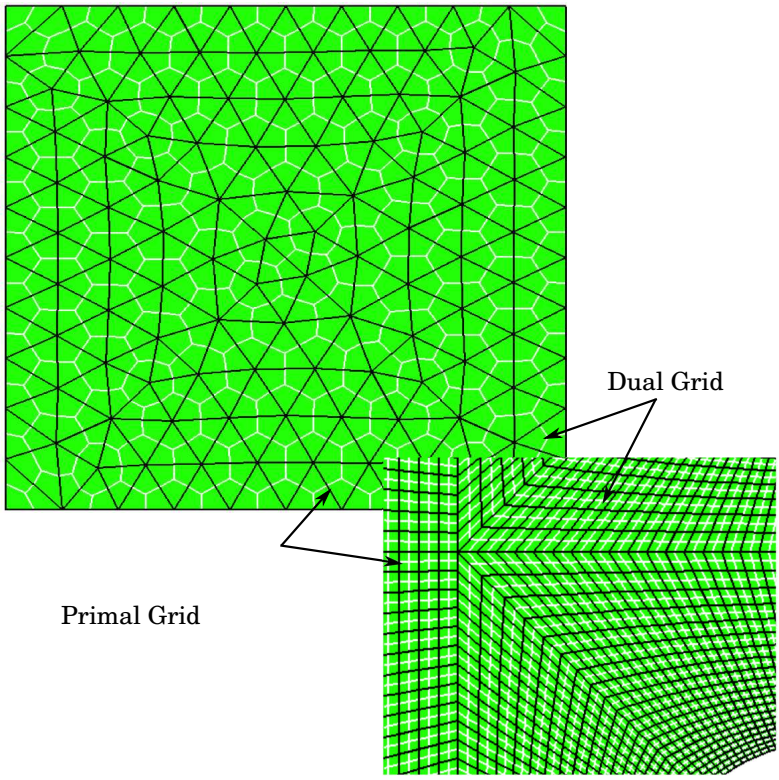


Figure 1: Primal, median dual, and centroidal dual grids.

of the velocity field $\mathbf{v}^*$. We note in passing that the projection makes use of physical boundary conditions, and that both the continuous and discrete projection operators are norm-reducing, see [3, 4].

As a starting point for the hybrid projection method, we adopted Gresho's second-order "P2" method [3, 4]. The desire for local-conservation initially drove the decision to use the finite volume method (FVM) with all variables cell-centered, i.e., collocated velocity and pressure. However, to avoid deleterious and troublesome pressure modes, a Galerkin formulation with node-centered pressure was adopted. Thus, a hybrid FVM – Galerkin FEM discretization forms the basis for the incompressible solver.

The use of a hybrid discretization for the implementation of a projection solver is not entirely new. The first use of a predominantly finite volume scheme with a continuous Galerkin pressure-Poisson operator appeared in the work of Bell, et al. [2], albeit using structured meshes with explicit advection, and a time-step strictly constrained by $CFL < 1$. More recently,

Aliabadi and co-workers have developed a method [33–35] that is similar to that presented here, although they use a more complex pressure update.

In our formulation, we begin with a discontinuous Galerkin formulation that proceeds by discretizing in space, e.g., with quadrilaterals, triangles, or some combination. At each time $t \in [0, T]$, an approximate solution $\mathbf{v}^h$ is sought in the finite element space of *discontinuous* functions $\mathbf{W}^h$. We defer, for the moment, the discussion of the pressure space. Multiplying Eq. (2) by a test function and integrating by parts yields

$$\begin{aligned} \int_{\Omega^e} \Big\{ \mathbf{w}^h \cdot \left(\frac{\partial \rho \mathbf{v}^h}{\partial t} \right) + \nabla \cdot \{ \mathbf{w}^h \cdot (\rho \mathbf{v}^h \mathbf{v}^h) \} - \nabla \mathbf{w}^h \cdot (\rho \mathbf{v}^h \mathbf{v}^h) - \\ \nabla \cdot (\mathbf{w}^h \cdot \boldsymbol{\tau}) + \nabla \mathbf{w}^h \cdot \boldsymbol{\tau} + \mathbf{w}^h \cdot \nabla p^h + \mathbf{w}^h \cdot (\rho \mathbf{f}) \Big\} \, d\Omega = \mathbf{0} \\ \forall \mathbf{w}^h \in \mathbf{W}^h \end{aligned} \tag{7}$$

where Ω^e is the element volume. Selecting piecewise-constant weight functions, and using the divergence theorem yields a system of ordinary differential equations (ODEs) for the momentum transport equations

$$\rho \Omega^e \frac{d\mathbf{v}}{dt} + \oint_{\Gamma^e} \rho \mathbf{v} (\mathbf{v} \cdot \mathbf{n}) \, d\Gamma^e - \oint_{\Gamma^e} \boldsymbol{\tau} \cdot \mathbf{n} \, d\Gamma^e + \int_{\Omega^e} \nabla p \, d\Omega^e - \int_{\Omega^e} \rho \mathbf{f} \, d\Omega^e = \mathbf{0} \tag{8}$$

with a cell-averaged velocity defined as

$$\mathbf{v} = \frac{1}{\Omega^e} \int_{\Omega^e} \mathbf{v}^h d\Omega^e \tag{9}$$

Here, the superscript h has been dropped for simplicity. Note that the discrete momentum equations in conjunction with the discrete divergence constraint form a system of differential algebraic equations (DAEs) rather than ODEs.

The discrete projection algorithm can be derived in a number of ways. We first develop the time-integrator, and identify the terms associated with the projection via a Helmholtz decomposition of the velocity. We define the

element-level mass, advective, viscous, gradient, and body-force operators as

$$M^e = \rho\Omega^e \tag{10}$$

$$A^e(\rho,\mathbf{v})\mathbf{v} = \oint_{\Gamma^e} \rho\mathbf{v}(\mathbf{v}\cdot\mathbf{n})\,d\Gamma^e \tag{11}$$

$$K^e\mathbf{v} = \oint_{\Gamma^e} \boldsymbol{\tau}\cdot\mathbf{n}\,d\Gamma^e \tag{12}$$

$$\mathbf{B}^e p\ \Omega^e = \int_{\Omega^e} \nabla p\,d\Omega^e \tag{13}$$

$$\mathbf{F}^e = \int_{\Omega^e} \rho\mathbf{f}\,d\Omega^e \tag{14}$$

Discretizing Eq. (8) in time, we first apply forward-, then backward–Euler time discretization, and take the θ-weighted sum of the fully-discrete systems, yielding

$$\begin{aligned} M\frac{\mathbf{v}^{n+1}-\mathbf{v}^n}{\Delta t} - \theta K\mathbf{v}^{n+1} &= (1-\theta)\mathbf{F}^n + \theta\mathbf{F}^{n+1} \\ &\quad - A(\rho,\mathbf{v})\mathbf{v}^n + (1-\theta)K\mathbf{v}^n - \mathbf{B}p^n - \theta\mathbf{B}(p^{n+1}-p^n) \end{aligned} \tag{15}$$

where $0 \le \theta \le 1$. For simplicity, explicit advection has been used here for the sake of demonstrating the derivation of the method. Setting $\theta = 0$ corresponds to a forward-Euler treatment of viscous and body-force terms, while $\theta = 1/2$ is the trapezoidal rule, and $\theta = 1$ provides a backward-Euler treatment.

Turning to the velocity decomposition, an approximate velocity for the time-step, $\mathbf{v}^*$, may be decomposed using Eq. (6) into div-free and curl-free components $\mathbf{v}^* = \mathbf{v}^{n+1} + \frac{1}{\rho}\nabla\lambda$. Taking the divergence of the velocity field yields

$$\nabla\cdot\frac{1}{\rho}\nabla\lambda = \nabla\cdot\mathbf{v}^* \tag{16}$$

where $\nabla\cdot\mathbf{v}^* \neq 0$ and $\nabla\cdot\mathbf{v}^{n+1} = 0$. For steady Dirichlet pressure boundary conditions, $\lambda = 0$ on Γ_p and $\dfrac{\partial\lambda}{\partial\mathbf{n}} = 0$ on $\Gamma\cap\Gamma_p$. With this choice for the velocity decomposition,

$$\lambda = \theta\Delta t(p^{n+1}-p^n) \tag{17}$$

When combined with Eq. (15), this yields the fully-discrete momentum equation to be solved for the approximate velocity $\mathbf{v}^*$

$$[M - \theta\Delta t K]\mathbf{v}^* = [M + (1-\theta)\Delta t K]\mathbf{v}^n \\ - \Delta t A(\rho, \mathbf{v})\mathbf{v}^n - \Delta t \mathbf{B} p^n + \Delta t\{(1-\theta)\mathbf{F}^n + \theta \mathbf{F}^n\} \quad (18)$$

For the, the pressure increment, or Lagrange multiplier, we seek a solution for λ^h in the finite element space of continuous functions where $S^h = V^h = \{q^h | q^h \in H^{1h}(\Omega)\}$ and $H^{1h}(\Omega)$ is the finite dimensional space of square integrable functions. Multiplying Eq. (16) by the test-function, q^h, and integrating by parts produces the discrete pressure-Poisson (PPE) problem.

$$\int_\Omega \nabla q^h \cdot \left(\frac{1}{\rho}\nabla \lambda^h\right) d\Omega = \int_\Gamma q^h \left\{\frac{1}{\rho}\frac{\partial \lambda^h}{\partial n}\right\} d\Gamma - \int_\Gamma q^h \mathbf{v}^* \cdot \mathbf{n}\, d\Gamma + \int_\Omega \mathbf{v}^* \cdot \nabla q^h \, d\Omega \quad (19)$$

The discrete PPE, written in terms of operators is $K_p \lambda = D(\mathbf{v}^*)$, where

$$K_p = \int_\Omega \nabla q^h \cdot \left(\frac{1}{\rho}\nabla q^h\right) d\Omega \quad (20)$$

and $D(\mathbf{v}^*)$ is

$$D(\mathbf{v}^*) = \int_\Gamma q^h \left\{\frac{1}{\rho}\frac{\partial \lambda^h}{\partial n}\right\} d\Gamma - \int_\Gamma q^h \mathbf{v}^* \cdot \mathbf{n}\, d\Gamma + \int_\Omega \mathbf{v}^* \cdot \nabla q^h \, d\Omega \quad (21)$$

An alternative form of the PPE problem simply uses the element-centered divergence of the intermediate velocity for the right-hand-side of the PPE, e.g., $\int_\Omega q^h \nabla \cdot \mathbf{v}^* d\Omega$. Operational experience with the hybrid formulation has shown that this form of the PPE leads to smoother pressures in general, but also to less accurate initial velocities and pressures.

Given a velocity and pressure at time n, the hybrid algorithm proceeds as follows for a single time step.

1. Solve for $\mathbf{v}^*$

$$[M - \theta\Delta t K]\mathbf{v}^* = [M + (1-\theta)\Delta t K]\mathbf{v}^n \\ - \Delta t A(\rho, \mathbf{v})\mathbf{v}^n - \Delta t \mathbf{B} p^n + \Delta t\{(1-\theta)\mathbf{F}^n + \theta \mathbf{F}^n\} \quad (22)$$

2. Form the right-hand-side of the PPE, solve for λ,

$$K_p \lambda = D(\mathbf{v}^*) \quad (23)$$

3. Update the pressure

$$p^{n+1} = p^n + \frac{1}{\theta \Delta t}\lambda \tag{24}$$

4. Project the cell-centered velocities

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{1}{\rho}\mathbf{B}\lambda \tag{25}$$

5. Compute face gradients and project the face-centered velocities

$$v_f = v_f^* - \frac{1}{\rho}(\mathbf{B}\lambda)_f \cdot \mathbf{n} \tag{26}$$

The pressure and Lagrange-multiplier gradients are computed using the uniform-strain finite element gradient, $\mathbf{B}$ defined in Eq. (13), see also [36]. The pressure gradient force contribution to the momentum equations is computed using the $\mathbf{B}$-matrix as

$$(\mathbf{B}p) = \sum_{I}^{N_{npe}} \mathbf{B}_I^e p_I^e \Omega^e \tag{27}$$

where $1 \leq I \leq N_{npe}$, and N_{npe} is the number of nodes per element.

Given the element-level gradient, face-centered gradients are computed using the elements attached to dual-edges as neighbors. The elements in the neighbor list include those attached by dual-edges to the vertices of a given dual-edge where the gradient is desired,

$$(\mathbf{B}\lambda)_f = \frac{\sum_i^{N_{nbr}} (\mathbf{B}^e \lambda\, \Omega^e)_i\, \Omega_i}{\sum_i^{N_{nbr}} \Omega_i}, \tag{28}$$

where N_{nbr} is the number of neighbor elements.

The calculation of dual-edge quantities, i.e., centered at unique element faces, occurs at a number of places in the hybrid projection algorithm. This calculation is formulated as an inverse-distance-weighted L^2 projection onto a Galerkin basis, and illustrated for a 2D mesh in Figure 2(a). The data at a face is computed as

$$\phi_f = \xi_a \phi_a + (1 - \xi_a)\phi_b \tag{29}$$

where ϕ is a cell-centered field variable, and $\xi_a = h^b/(h^a + h^b)$.

The implementation of the hybrid projection algorithm relies on the so-called "co-velocity" approach where dual-edge velocities are made divergence-free and used for advection. This enables the incompressible flow solver to be easily extended to perform volume-tracking for multi-fluid simulations. To calculate the dual-edge velocities, first a symmetric mirror of the velocities across the domain boundaries is performed as shown in Figure 2(b) where g identifies the ghost data, and d is the normal distance from the element centroid to the element boundary. Alternatively, this can be thought of as extrapolation to the boundary edges as $\mathbf{v}_f = \mathbf{v}$ as shown in Figure 2(c). Where Dirichlet velocity boundary conditions are imposed, the ghost data is specified by the prescribed velocity boundary conditions.

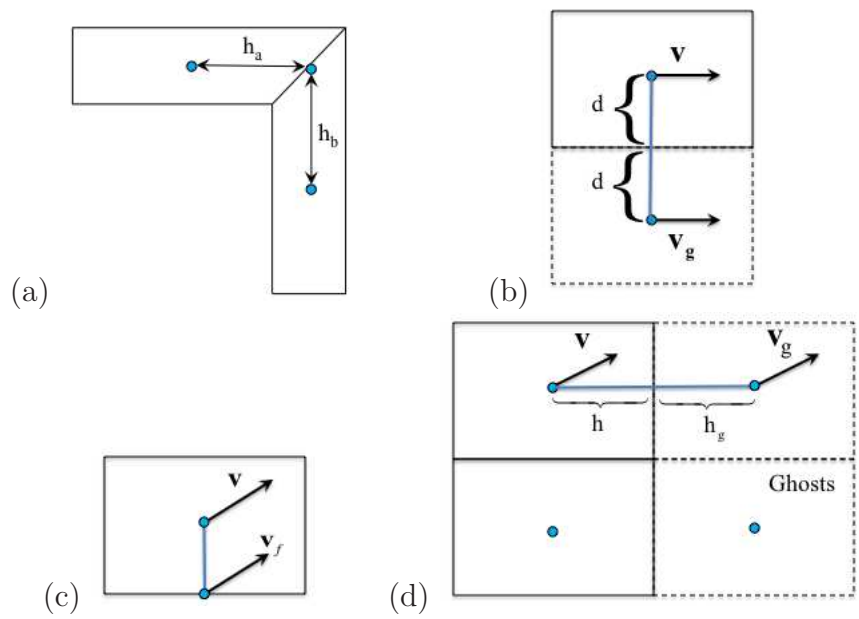


Figure 2: (a) dual-edge data for edge projection, (b) symmetric mirror for ghost data at a boundary, (c) edge velocity extrapolation to a physical boundary, (d) edge velocity projection for overlapping "ghost" elements for parallel calculations.

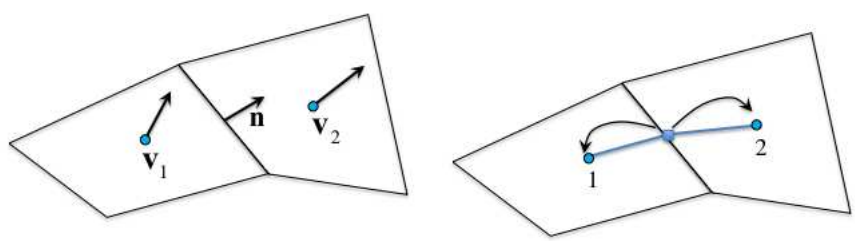
The edge velocity at the boundaries is then computed as $\mathbf{v}_f = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_g)$. The factor of 1/2 is consistent with Eq. (29) since the boundary ghosts receive data from the symmetric mirror at a location that is twice the distance from the element centroid to the boundary. For distributed-memory domain-based parallelism, it is necessary to communicate overlapping elements or ghost

elements across the decomposed computational domain as indicated in Figure 2(d). Edges with a vertex containing ghost data are populated in buffers and communicated with neighboring processors before the dual-edge velocities external to a processor domain are computed. Once all the data attached to external edges has been constructed, the dual-edge velocities are computed as $v_f = \mathbf{v}_f \cdot \mathbf{n}$ for all external edges.

After the external dual-edge velocities have been computed, taking boundary conditions into consideration, the dual-edge velocities associated with edges, internal to a processor domain, are computed using

$$v_f = \left[\xi \mathbf{v}_1 + (1-\xi)\mathbf{v}_2\right] \cdot \mathbf{n} \tag{30}$$

where $\mathbf{n}$ is the unit normal of the face where the dual-edge velocity is centered as shown in Figure 3a.



(a) Dual-Edge Normal (b) Dual-Edge Assembly

Figure 3: (a) normal velocity on internal dual-edges, and (b) dual-edge assembly from elements 1 and 2.

The dual-edge velocities are used to compute the the divergence as

$$\mathrm{div}(\mathbf{v})^e = \frac{1}{\Omega^e} \sum_i^{Nedge} v_{f_i} \Gamma_i \tag{31}$$

where $Nedge$ is the number of dual-edges connected to element e. The computation of the divergence of the velocity proceeds edge-by-edge with an assembly to vertices of the dual, that is, to elements as indicated in Figure 3b.

2.1. Startup Procedure

This section describes the startup procedure for a well-posed incompressible Navier-Stokes problem. By constructing a well-posed problem, this also

insures that the solution to index-1 DAEs via the projection method produce solutions identical to the those for the index-2 DAEs, i.e., the fully-coupled velocity-pressure system. In addition to the calculation of a suitable initial velocity field that satisfies the boundary conditions and is divergence-free, an initial pressure is required that is consistent with the initial velocity field and the prescribed boundary conditions.

Given a set of prescribed boundary and initial conditions, $\widehat{\boldsymbol{v}}$ on Γ_v, $\widehat{p}$ on Γ_p, and $\mathbf{v}^0(\mathbf{x})$ in Ω, the startup procedure is as follows. First, we define the initial edge-normal velocities v_f^0, setup ghost data, and calculate the edge-normal velocities. This typically yields dual-edge velocities, v_f^*, that are in general not divergence-free. Given the dual-edge velocities and using Eq. (31), the root-mean-square (RMS) divergence is computed as

$$\|\mathrm{div}(\mathbf{v})\|_{\mathrm{RMS}} = \frac{1}{\Omega}\sqrt{\frac{\sum_{i=1}^{N_{el}}[\mathrm{div}(\mathbf{v})^i]^2}{Nel}} \tag{32}$$

If the RMS divergence is larger than a prescribed tolerance, the initial velocity is projected onto a divergence-free subspace. The projection begins with the usual decomposition of the velocity into div-free and curl-free components $\mathbf{v}^* = \mathbf{v} + \frac{1}{\rho}\nabla\lambda$, where $\mathbf{v}^0 = \mathbf{v}^*$, and $\nabla\cdot\mathbf{v}^* \neq 0, \quad \nabla\cdot\mathbf{v}^0 = \mathbf{0}$.

Taking the divergence of the decomposed velocity field yields the same pressure-Poisson problem in Eq. 16. After solving the PPE problem for λ, we project both the element and dual-edge velocities. Using the Helmholtz decomposition, and knowing v_f^* and λ, we can compute the div-free edge velocities

$$v_f = v_f^* - \frac{1}{\rho}(\boldsymbol{B}\lambda)_f \cdot \mathbf{n}_f \tag{33}$$

where $(\boldsymbol{B}\lambda)_f$ is computed using Eq. (28). Following the projection of the dual-edge velocities, the prescribed Dirichlet velocities are injected in the dual-edge velocity to ensure that the final div-free dual-edge velocities exactly match the prescribed velocities. Finally, the element-velocity is projected onto the div-free subspace as

$$\mathbf{v} = \mathbf{v}^* - \frac{1}{\rho}(\boldsymbol{B}\lambda)^e \tag{34}$$

Once the divergence-free velocities have been obtained, an initial pressure, consistent with the initial and boundary conditions, is computed. We begin

with the momentum written in terms of the acceleration using the divergence-free velocity, $\mathbf{v}^0$,

$$\mathbf{a}^0 = \frac{1}{\rho}\left\{\rho\mathbf{f} + \nabla\cdot[\mu(\nabla\mathbf{v}^0 + (\nabla\mathbf{v}^0)^T)] - \nabla\cdot(\rho\mathbf{v}^0\mathbf{v}^0) - \nabla p^0\right\} \tag{35}$$

Since $\nabla\cdot\mathbf{v}^0 = 0$, $\nabla\cdot\mathbf{a}^0 = 0$, thus we form the pressure-Poisson equation for p^0

$$\nabla\cdot\frac{1}{\rho}\nabla p^0 = \nabla\cdot\frac{1}{\rho}\left\{\rho\mathbf{f} + \nabla\cdot[\mu(\nabla\mathbf{v}^0 + (\nabla\mathbf{v}^0)^T)] - \nabla\cdot(\rho\mathbf{v}^0\mathbf{v}^0)\right\} \tag{36}$$

Using the "partial acceleration",

$$\widetilde{\mathbf{a}} = \frac{1}{\rho}\left\{\rho\mathbf{f} + \nabla\cdot[\mu(\nabla\mathbf{v}^0 + (\nabla\mathbf{v}^0)^T)] - \nabla\cdot(\rho\mathbf{v}^0\mathbf{v}^0)\right\} \tag{37}$$

the continuous PPE is written as

$$\nabla\cdot\frac{1}{\rho}\nabla p^0 = \nabla\cdot\widetilde{\mathbf{a}}, \tag{38}$$

where $p^0 = \widehat{p}^0$ on Γ_p. The partial acceleration is computed as

$$\int_{\Omega_e}\rho\widetilde{\mathbf{a}}d\Omega^e = \int_{\Omega_e}\rho\mathbf{f}d\Omega^e + \int_{\Omega_e}\nabla\cdot[\mu(\nabla\mathbf{v}^0 + (\nabla\mathbf{v}^0)^T)]d\Omega^e - \int_{\Omega_e}\nabla\cdot(\rho\mathbf{v}^0\mathbf{v}^0)d\Omega^e \tag{39}$$

Dual-edge accelerations are computed following the same procedure outlined for the dual-edge velocities. For boundaries where a time-dependent velocity is prescribed, e.g., FSI, we impose $\widetilde{\mathbf{a}}_f = \partial\mathbf{v}/\partial t$, otherwise $\widetilde{\mathbf{a}}_f = 0$.

Using the same finite element formulation, the weak form of Eq. (38) leads to a linear system of equations, $K_p p^0 = D(\widetilde{\mathbf{a}})$. Solving for p^0, we have the initial pressure and velocities, $\mathbf{v}^0$, $\mathbf{v}_f^0$, that satisfy boundary and initial conditions, and the well-posedness requirements, and are ready to start time-marching.

2.2. Time-Step Estimation

There are three options for advection in the hybrid projection algorithm: (1) explicit, (2) the so-called "semi-implicit" method with the primary variables treated implicitly and the reconstructed terms handled explicitly, and (3) fully-implicit treatment with non-linear iteration. The details of these

methods are beyond the scope of this paper, but here, we describe the implications on the time-step.

The time-step estimation uses multiple characteristic element dimensions and a centroid velocity that is projected onto the relevant element dimension. The element dimensions, $\mathbf{h}_\xi$, $\mathbf{h}_\eta$, and $\mathbf{h}_\zeta$ are shown in Figure 4 for an arbitrary hexahedral element. The velocity is projected onto each of the characteristic length scales for the element, and for a given CFL condition, a minimum over all possible length and velocity scale choices determines the acceptable time-step size.

$$\Delta t = min\left(\frac{\|\mathbf{h}_\xi\| CFL_{max}}{\mathbf{v}\cdot\mathbf{h}_\xi}, \frac{\|\mathbf{h}_\eta\| CFL_{max}}{\mathbf{v}\cdot\mathbf{h}_\eta}, \frac{\|\mathbf{h}_\zeta\| CFL_{max}}{\mathbf{v}\cdot\mathbf{h}_\zeta}\right) \tag{40}$$

Here, CFL_{max} is prescribed based on the advection algorithm. Since we use hybrid meshes, the element dimensions used in the time-step estimation are specific to hexahedra, tetrahedra, wedge and pyramid elements.

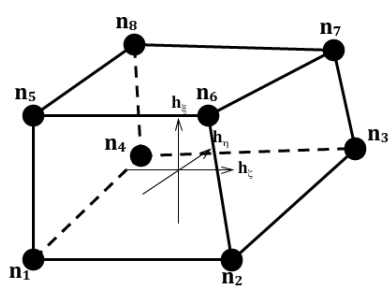


Figure 4: Hexahedral element showing local length scales and directions for time-step estimation.

For the explicit advection, a strict $CFL_{max} \leq 1$ condition is required. For the implicit advection, the algorithm is unconditionally stable for scalar linear advection. For the Navier-Stokes equations, a sharp stability estimate is not available, however, in our experience computing a wide variety of flows indicates numerical stability with $5 \leq CFL_{max} <\sim 10-20$ for time-accurate problems. The fully-implicit algorithm is unconditionally stable, but requires non-linear iteration at each time-step.

3. Monotonicity-Preserving Advection

One of the most important aspects of any CFD solution algorithm for high Reynolds number flows is the numerical discretization of the advection terms. This section outlines the principal aspects of the advection method using a prototypical scalar conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u, \mathbf{c}) = 0 \;\; \text{in} \;\; \Omega \times (0, T) \tag{41}$$

where $\mathbf{f}(u, \mathbf{c}) = \mathbf{c}u$ is the physical flux, $\mathbf{c}$ is the advective velocity, and u is the dependent variable. Eq. (41) is subject to initial conditions

$$u(\mathbf{x}, 0) = u^0(\mathbf{x}) \;\; \forall \;\; \mathbf{x} \in \Omega \tag{42}$$

and boundary conditions

$$u = \hat{u} \;\; \text{on} \;\; \Gamma_D \tag{43}$$

and

$$(\mathbf{f} \cdot \mathbf{n}) = (\hat{\mathbf{f}} \cdot \mathbf{n}) \;\; \text{on} \;\; \Gamma_N \tag{44}$$

where $\Gamma = \Gamma_D \cup \Gamma_N$. Here, $\hat{u}$ and $\hat{\mathbf{f}}$ are prescribed values of the conserved variable and the corresponding flux on the boundary, respectively.

The discontinuous Galerkin formulation proceeds by discretizing in space, e.g., with quadrilaterals, triangles, or some combination. At each time $t \in [0, T]$, an approximate solution u^h is sought in the finite element space of *discontinuous* functions W^h. In order to determine the approximate solution, the weak formulation is required, i.e., on each element

$$\int_{\Omega^e} w^h \left\{ \frac{\partial u^h}{\partial t} + \nabla \cdot \mathbf{f}(u^h, \mathbf{c}) \right\} d\Omega^e = 0 \;\; \forall w^h \in W^h \tag{45}$$

where Ω^e is the element volume. Applying the divergence theorem yields

$$\frac{d}{dt} \int_{\Omega^e} w^h u^h d\Omega^e + \oint_{\Gamma^e} w^h \mathbf{f}(u^h, \mathbf{c}) \cdot \mathbf{n} d\Gamma^e - \int_{\Omega^e} \nabla w^h \cdot \mathbf{f}(u^h, \mathbf{c}) \cdot n d\Omega^e = 0, \;\; \forall w^h \in W^h \tag{46}$$

If the test function, w^h, is chosen to be piecewise-constant over each element, a finite volume formulation is recovered. We chose w^h, such that

$$\frac{d}{dt} \int_{\Omega^e} 1 \; u^h d\Omega^e + \oint_{\Gamma^e} 1 \; \mathbf{f}(u^h, \mathbf{c}) \cdot \mathbf{n} d\Omega^e = 0 \tag{47}$$

Remark 1. *In order to provide compatibility with modern piecewise linear interface construction (PLIC) based volume-tracking algorithms that are based on a finite volume formalism, we choose piecewise-constant test functions for the hybrid projection algorithm in the Hydra toolkit. However, it should be noted that this choice does not restrict the current methods or software from extension to include higher-order (higher than linear) reconstruction techniques or the incorporation of higher-order test functions.*

Integrating Eq. (47) over a time-step,

$$\int_{t'=t}^{t'=t+\Delta t} \frac{d\overline{u}}{dt} dt' + \frac{1}{\Omega^e} \int_{t'=t}^{t'=t+\Delta t} \oint_{\Gamma^e} \mathbf{f}(u^h, \mathbf{c}) \cdot \mathbf{n} d\Gamma^e dt' = 0 \tag{48}$$

which permits the definition of a time-averaged (or numerical) flux,

$$\boldsymbol{\mathcal{F}} \cdot \mathbf{n} \approx \frac{1}{\Delta t} \int_{t'=t}^{t'=t+\Delta t} \mathbf{f}(u^h, \mathbf{c}) \cdot \mathbf{n} dt' \tag{49}$$

yielding

$$\int_{t'=t}^{t'=t+\Delta t} \frac{d\overline{u}}{dt} dt' + \frac{1}{\Omega^e} \oint_{\Gamma^e} \boldsymbol{\mathcal{F}}(u^h, \mathbf{c}) \cdot \mathbf{n} d\Gamma^e = 0 \tag{50}$$

There are many ways to formulate the numerical approximation to the physical flux. We use the local Lax-Friedrichs flux which is an approximate, two-point, monotone, Lipschitz flux. At a given dual-edge, i, the numerical flux is defined as

$$\boldsymbol{\mathcal{F}}_i \cdot \mathbf{n}_i = \frac{1}{2} \left\{ (\mathbf{f}_i(u_i^-) + \mathbf{f}_i(u_i^+)) \cdot \mathbf{n}_i - a(u_i^+ - u_i^-) \right\} \tag{51}$$

where a is the maximal eigenvalue of the flux Jacobian and the (normal) flux Jacobian is

$$\left(\frac{\partial \mathbf{f}_i}{\partial u} \right) \cdot \mathbf{n}_i = \mathbf{c}_i \cdot \mathbf{n}_i \tag{52}$$

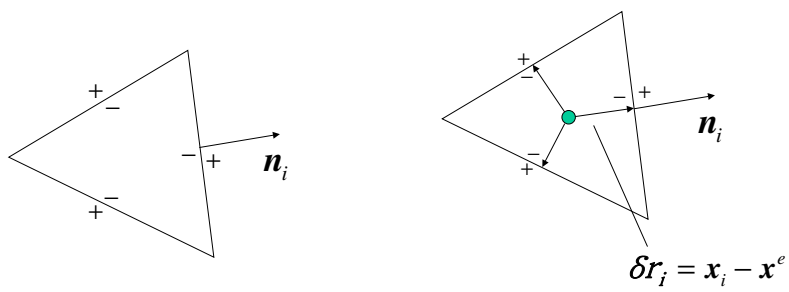
Thus, for the scalar advection problem, a is the normal velocity at a cell face.

Figure 5(a) shows the $+/-$ states used in the numerical flux. By convention, the $-$ state is always located on the inside of the cell, and the $+$ state is based on the adjacent cell and is on the positive normal side of the cell. We associate the two elements as vertices of the edge in the dual grid and assign ownership of the outward facing normal and face area in a dual-edge data structure.

The evaluation of the numerical flux relies on multi-dimensional reconstruction of the field variables at the $\pm$ locations of each cell as shown in Figure 5. The reconstruction of the field at face i in a given cell i is computed using the limited gradient as

$$u_i^- = \overline{u} + \phi \nabla_c \overline{u} \cdot \vec{\delta r}_i, \tag{53}$$

with the $+$ values reconstructed from the adjacent cell and its limited gradient and cell-averaged data. Figure 5(b) shows the reconstruction at a typical cell face. Here, ∇_c indicates a centered gradient with $\phi \in [0, 1]$ the slope limiter, discussed in the following section. A detailed study of gradient estimation techniques for unstructured grids may be found in Chapters 7 – 8 of [37].



(a) Cell $\pm$ locations (b) Reconstruction

Figure 5: Cell face locations for reconstructed $+/-$ values used in the numerical flux evaluation.

3.1. Phase-speed and Artificial Diffusivity

The performance of the numerical flux and the associated advection scheme may be understood in terms of the phase error and artificial diffusivity introduced by the discretization. A detailed discussion of the analysis techniques used to assess numerical performance may be found in [38–40].

We consider the performance of the semi-discrete method described above. In this context, the numerical phase speed is the ratio of the apparent discrete advective speed and the true speed, $\tilde{c}/c$. The diffusive behavior is captured in terms of an inverse Péclet number which, for a purely non-diffusive method, $1/Pe^{art} = 0$.

Figure 6 shows the non-dimensional phase speed and artificial diffusivity as functions of the non-dimensional wave number up to the grid Nyquist

limit, $2\Delta x/\lambda = 1.0$. The values that the associated limiters may achieve based on the input data are shown in the inset. There are four cases that correspond to various forms of data that may be present on the grid.

1. $\phi_{m-1} = \phi_m = 0$: This corresponds to $2\Delta x$ signals on the grid where the slope limiters are fully activated. In this case, the phase speed corresponds to that associated with a central-differences (CD) discretization with first-order upwinding (FOU) at the Nyquist grid limit. In other words, the method attempts to completely damp these signals to avoid oscillatory solutions.

2. $\phi_{m-1} = 0, \phi_m = 1$: In this case a change in slope occurs with smooth data to the right. The artificial diffusivity is reduced to 50% of the first-order upwind case, and the phase-speed is increased at long wavelengths. This has the effect of controlling oscillations and avoiding signal separation due to dispersive effects.

3. $\phi_{m-1} = 1, \phi_m = 0$: In this case as well a change in slope occurs with smooth-data to the left. The artificial diffusivity is reduced to 50% of the first-order upwind case, and the phase-speed is decreased at long wavelengths. Similar to the previous case, this case controls oscillations and avoids signal separation due to dispersive effects.

4. $\phi_{m-1} = 1, \phi_m = 1$: In this case smooth data is present and there is no limiting. The artificial diffusivity is zero and corresponds to the limit of pure advection, and the phase-speed is that associated with a second-order central-difference approximation of the gradient. This corresponds to the case when all signals are adequately resolved on the grid.

The advection methods used here are nonlinear, monotonicity-preserving, and follow the behavior described above. For smooth data the methods are non-dissipative and deliver second-order spatial convergence. For the limit of oscillatory data at the $2\Delta x$ grid Nyquist limit, the methods attempt to damp only the short wavelength data while preserving the resolved long wavelength data on the grid.

Various metrics are available that may be used to assess the accuracy of a solution method or, alternatively, to verify its correctness. We are interested in extracting the leading order terms in the discretization error. The

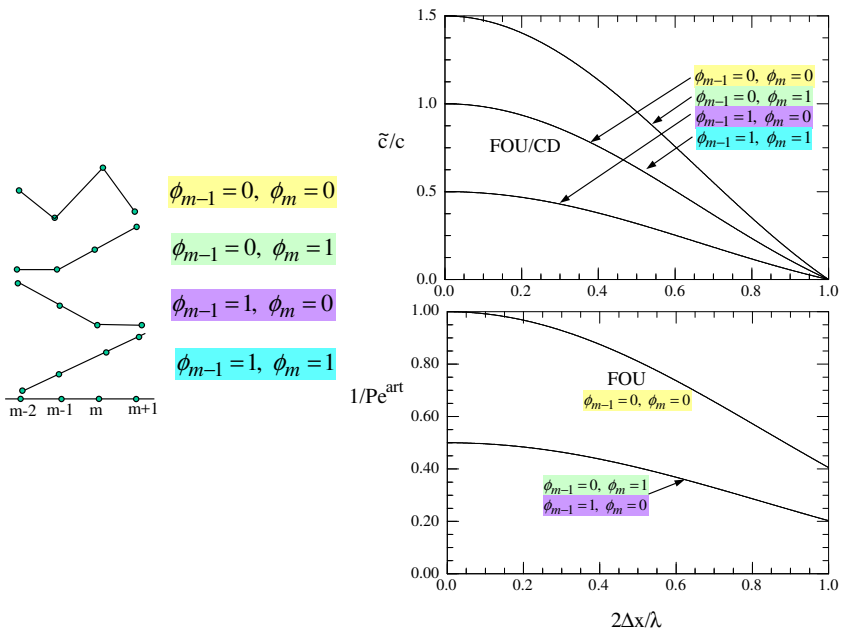


Figure 6: Phase speed and artificial diffusivity as a function of the non-dimensional wave number. The bounding phase and diffusivity behavior for first-order upwind (FOU) and central-differences (CD) bound the behavhior of the nonlinear advection method.

discretization error is comprised of both spatial and temporal errors and is defined as the difference between the computed and exact solutions,

$$\mathbf{e}_i = \overline{u}_i^e - \overline{u}_i, \tag{54}$$

where the exact grid function, $\overline{u}_i^e$, is defined in terms of the exact solution, u_i^e, as

$$\overline{u}_i^e = \frac{1}{\Omega_i} \int_{\Omega_i} u_i^e d\Omega. \tag{55}$$

It can be shown that the discretization error is proportional to the truncation error which permits the discretization error to be written as

$$\mathbf{e}_j = \alpha h^p + \beta \Delta t^q + H.O.T., \tag{56}$$

where p corresponds to the order of the spatial discretization, q to the order of the temporal discretization, and $H.O.T.$ indicates higher-order terms.

It is typical to use reconstruction methods that are matched with the time integrator, i.e., $p = q$. In addition, the time step and mesh size are related by the CFL number,

$$\Delta t = \frac{CFL\ h}{\lambda} \tag{57}$$

where λ is the magnitude of the largest eigenvalue of the flux Jacobian. This permits the discretization error to be written as

$$\mathbf{e}_j = \left(\alpha + \beta\ \frac{CFL}{\lambda}\right) h^p + H.O.T. \tag{58}$$

or, in a simplified form as, $\|\mathbf{e}\| = \overline{\alpha} h^p + H.O.T$. For all of the computations presented in this paper, errors are measured in terms of a discrete L^1 norm,

$$\|\mathbf{e}\| = \frac{\sum_i |\overline{u}_i^e - \overline{u}_i|}{\sum_i |\overline{u}_i^e|}. \tag{59}$$

Using a sequence of two grids, the global error metric may be used to extract the order of accuracy p, e.g.,

$$\frac{\|\mathbf{e}_1\|}{\|\mathbf{e}_2\|} = \left(\frac{h_1}{h_2}\right)^p, \tag{60}$$

where the subscript refers to the grid level. For all calculations presented here, $h_1/h_2 = 2$.

For smooth solutions we expect to recover an order of accuracy that is consistent with the formal accuracy of the method, e.g., $O(h^2)$ for a formally second-order spatial discretization ($p = 2$). For solutions with non-smooth data, e.g., shocks, we expect first-order or $O(h)$ convergence rates measured in the L^1 norm. In addition to the order of accuracy, the magnitude of the measured error is also useful in a comparison between different methods.

The ensuing discussion presents a series of convergence studies that illustrate the behavior of the advection algorithm. An emphasis is placed on testing the behavior on meshes with mixed element topology. To test the convergence behavior of the advection method, we first consider a translating Gaussian. Using Eq. 41 with initial conditions

$$u(x, 0) = \exp\left\{-\frac{(x - \overline{x})^2}{2\sigma_0^2}\right\}, \tag{61}$$

where $\overline{x} = x_0 + \int_0^t c(\tau)d\tau$, and $\sigma^2 = \sigma_0^2 + 2\kappa t$, the exact solution is

$$\hat{u}(x,t) = \frac{\sigma_0}{\sigma} \exp\left\{ -\frac{(x-\overline{x})^2}{2\sigma^2} \right\}. \tag{62}$$

For the numerical solution a computational domain with $0 \leq x \leq 15$ is used. The initial conditions are specified as $\mathbf{c} = (1.0, 0)$ and $\sigma_0 = 0.25$ with the Gaussian centered at $x_0 = 2.05$. To compute convergence rates, the solution error is computed as a function of time at $t = 2.5, 5.0, 10.0$ time units for a variety of grids discretized with triangular, quadrilateral elements and mixed-topology (triangles and quadrilaterals).

Three mixed-topology cases are presented here: a) a longitudinally split domain with quadrilaterals on the top, b) a mixed domain starting with a patch of quadrilaterals, and c) a mixed domain starting with a patch of triangles. Additional results may be found for all-triangle and all-quadrilateral meshes in [37]. Figure 7 shows the mesh topology and L^1 error for the three mixed-topology cases. The L^1 error exhibits a convergence rate of h^2 on all cases as the meshes are refined.

We next consider pure quadrilateral and mixed quadrilateral-triangular discretizations. The rotating cone problem has been documented in [41]. In Eq. (41), we use $\mathbf{c} = (-\omega\ y, \omega\ x)$, and $\omega = \pi/100$. The domain is defined on $-100 \leq x \leq 100$, and $-100 \leq y \leq 100$. The exact solution is

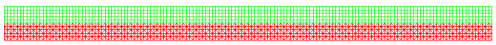
$$\hat{u}(x,y,t) = \exp\left\{ -\frac{(x-\overline{x})^2}{2\sigma_0^2} - \frac{(y-\overline{y})^2}{2\sigma_0^2} \right\}, \tag{63}$$

where $\overline{x} = x_0 + \int_0^t c_x(\tau)d\tau$, and $\overline{y} = y_0 + \int_0^t c_y(\tau)d\tau$.

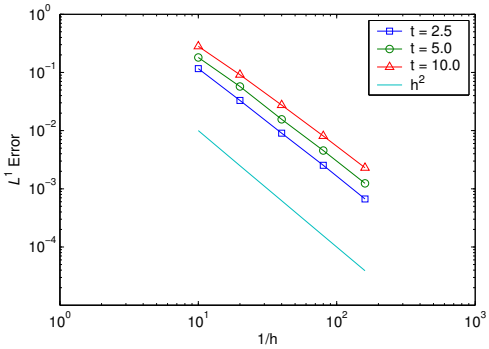
Figure 8(a) shows the pure quadrilateral mesh, and Figure 8(b) shows the L^1 error at three times. Similarly, Figure 8(c) – (d) shows a hybrid tri-quad mesh and the associated L^1 error. In both cases, an overall h^2 rate is observed.

4. Results

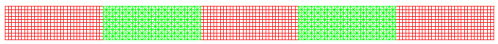
This section presents several verification problems to illustrate the convergence characteristics of the hybrid projection algorithm. A more detailed validation study using implicit large-eddy simulation maybe found in [42, 43], along with several verification and validation studies using multiple RANS models in [42].



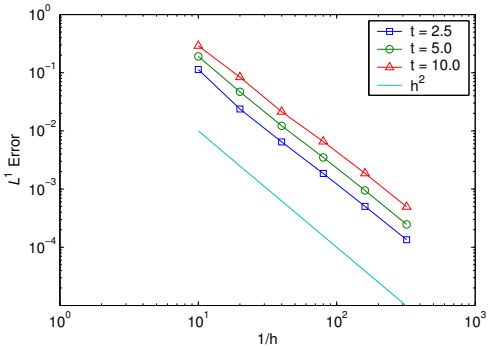
(a) Mixed Mesh – Longitudinal Split



(b) L^1 Error

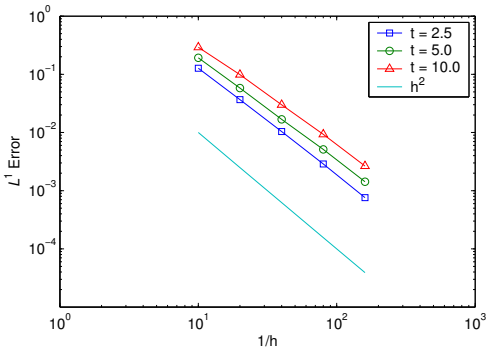


(c) Quad/Tri Patches



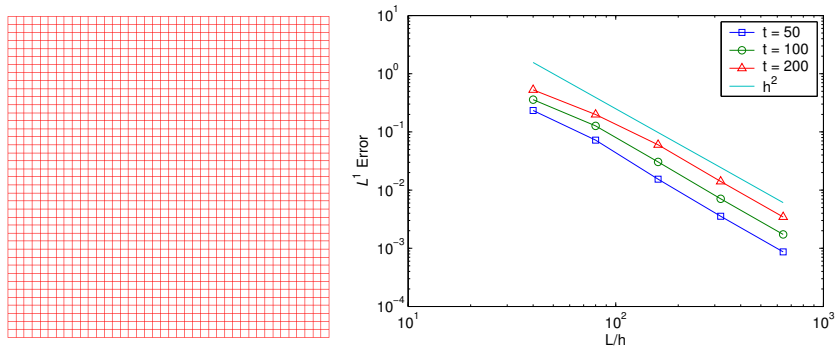
(d) L^1 Error

(e) Tri/Quad Patches

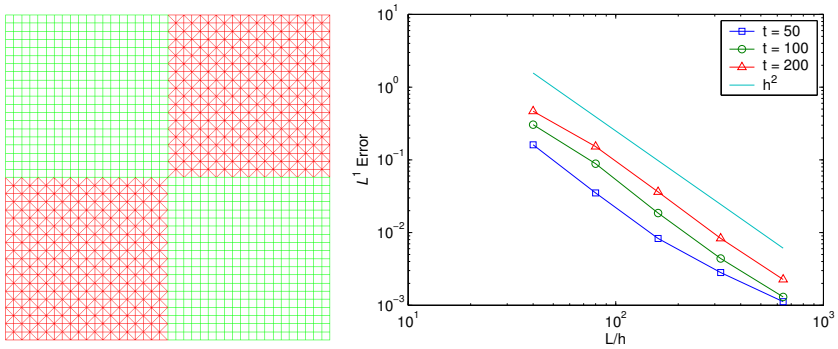


(f) L^1 Error

Figure 7: Mesh configuration for the case-a tri-quad meshes, and L^1 errors at $t = 2.5$, 5.0, and 10.0 for the case-a tri-quad meshes.



(a) All Quadrilaterial Mesh (b) L^1 Error



(c) Tri-Quad Mesh (d) L^1 Error

Figure 8: Mesh configuration for the quadrilateral meshes, and L^1 error as a function of time for the quadrilateral meshes.

4.1. Skewed Lid-Driven Cavity

The domain geometry for the skewed lid-driven cavity problem [44], is shown in Figure 9. On the bottom and side walls, no-slip and no-penetration boundary conditions are prescribed. Along the top "lid", a no-penetration boundary condition with unit velocity is prescribed. A single nodal pressure was prescribed in the bottom right-hand corner to set the hydrostatic pressure level, and make the pressure-Poisson equation non-singular.

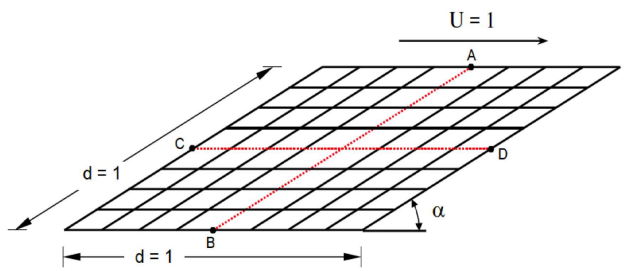
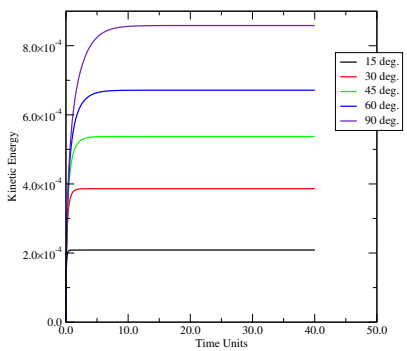


Figure 9: Skewed lid driven cavity geometry (reproduced from [44] without permission).

We computed five configurations in which the domain is skewed with degrees: $\alpha = 15°$, $30°$, $45°$, $60°$, and $90°$. Each configuration is computed using three grid resolutions with 32×32, 128×128 and 256×256 elements. The meshes use uniform mesh spacing, albeit with severely skewed elements for the $15°$ and $30°$ cases. Since only the final steady-state solution is of interest, all computations employ backward-Euler time integration with time-step control that bounds the CFL number at $CFL_{max} = 10$. Time history plots in Figure 10a of the global kinetic energy, $KE = \frac{1}{2}\int_\Omega \rho\, \mathbf{v}\cdot\mathbf{v}\, d\Omega$, indicate that a steady-state solution is reached by $t \approx 10$ time units. All problems for this verification problem are run for $t = 40$ time units.

Fig 10b shows the time-asymptotic behavior of the kinetic energy as a function of the mesh size in x direction, h. Here, for each cavity angle, a direct polynomial fit of the asymptotic kinetic energy data was performed. For each cavity angle, the results indicate $O(h^2)$ convergence in all velocity components. Velocity data is extracted along the red center lines shown in Figure 9 for direct comparison with the reference data provided in [44]. Two components of the velocity are plotted in Figure 11 along their respective center-lines.



(a) Kinetic Energy

Cavity Angle	Global Kinetic Energy
15°	$0.00020907 - 0.006877\ h^2$
30°	$0.00038731 - 0.010046\ h^2$
45°	$0.00053854 - 0.014590\ h^2$
60°	$0.00067314 - 0.019690\ h^2$
90°	$0.00086136 - 0.029630\ h^2$

(b) Convergence

Figure 10: (a) kinetic energy vs. time for the 128×128 grids for $\alpha = 15, 30, 45, 60, 90^o$ in terms of mesh size h, and (b) convergence behavior of the global kinetic energy vs. h for the lid-driven skewed cavities.

4.2. De Vahl Davis Differentially-Heated Cavity

To verify the numerical solution of buoyancy-driven flow, we computed the differentially-heated cavity case of De Vahl Davis [45, 46]. Figure 12a shows the computational domain and mesh used. The four meshes used here are in shown Figure 12b.

The non-dimensional governing equations for time-dependent thermal convection in this case are

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla P + Pr \nabla^2 \mathbf{v} + RaPr\hat{k}\Theta \tag{64}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{65}$$

$$\frac{\partial \Theta}{\partial t} + \mathbf{v} \cdot \nabla \Theta = \nabla^2 \Theta \tag{66}$$

where $\mathbf{v}$, P and Θ are the velocity, the deviation from hydrostatic pressure, and temperature respectively, and $\hat{k}$ is the unit vector in the z-direction. These non-dimensional equations were obtained using the characteristic length, L, velocity, $V = \alpha/L$, time scale, $\tau = L^2/\alpha$, and pressure, $\tilde{P} = \rho V^2$ as described in De Vahl Davis [46]. Here, ρ is the mass density, g the gravitational acceleration, $\alpha = k/\rho C_p$ is the thermal diffusivity, and ν is the kinematic viscosity. The Prandtl number is $Pr = \nu/\alpha$ and fixed at $Pr = 0.71$. The

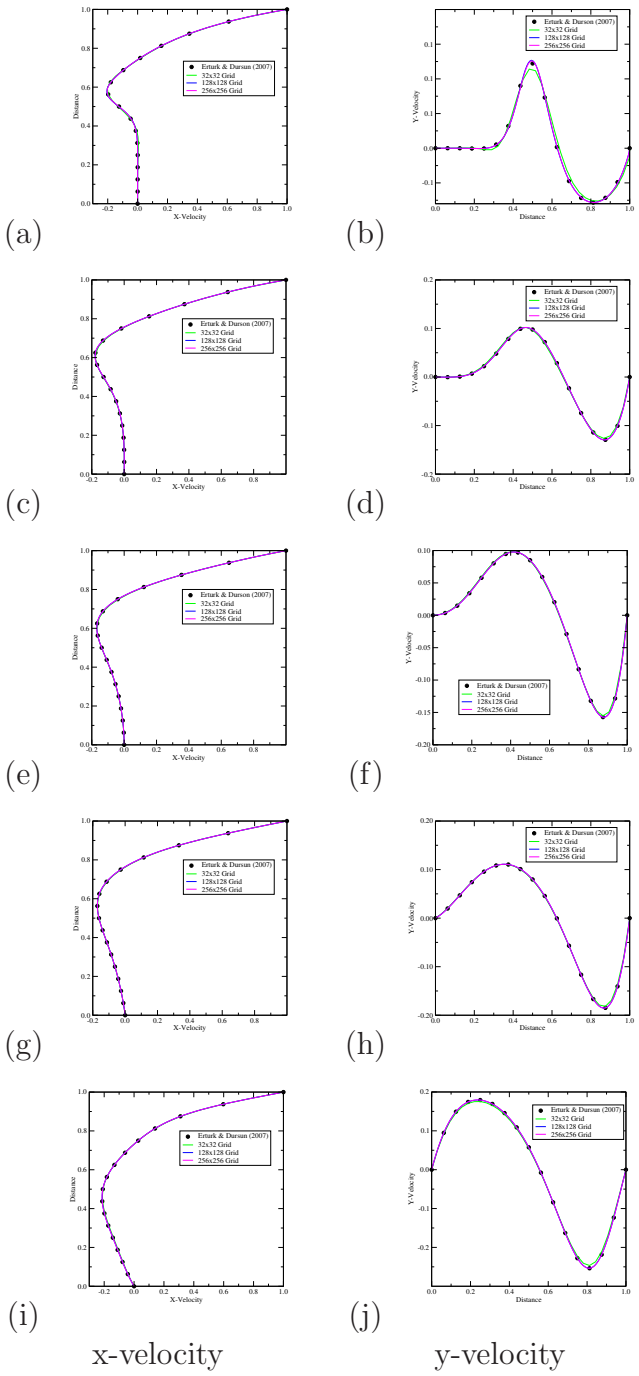
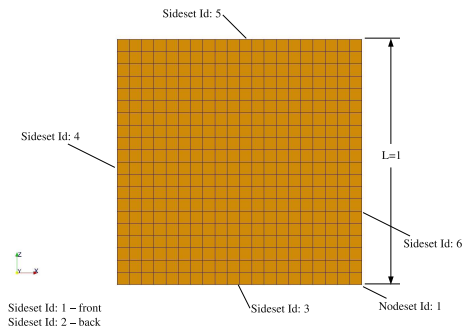


Figure 11: Velocity profiles for lid-driven cavity flow: (a) – (b) $\alpha = 15°$, (c) – (d) $\alpha = 30°$, (e) – (f) $\alpha = 45°$, (g) – (h) $\alpha = 60°$, (i) – (j) $\alpha = 90°$.



(a)

Mesh	Mesh Size	h
A	40×40	2.500E-2
B	80×80	1.250E-2
C	160×160	6.250E-3
D	320×320	3.125E-3

(b)

Figure 12: De Vahl Davis problem setup with (a) the mesh configuration, and (b) the meshes and associated sizes.

Rayleigh number is

$$Ra = \frac{g\beta(T_h - T_c)L^3}{\nu\alpha}, \tag{67}$$

where $T_h - T_c$ is the temperature difference between the hot and cold walls, and β the coefficient of thermal expansion. The non-dimensional temperature is defined in terms of the wall temperature difference

$$\Theta = \frac{T - T_c}{T_h - T_c}, \tag{68}$$

where T_h and T_c are the prescribed temperatures of the hot and cold walls, respectively.

The boundary conditions are no-slip and no-penetration walls with the top and bottom insulated. The left wall is held at a hot temperature, and the right wall is at the cold temperature corresponding to $\Theta = 1$ along $x = 0$ and $\Theta = 0$ along $x = 1$. The initial conditions are given by $\mathbf{v} = \mathbf{0}$ and $T = (T_h + T_c)/2$ which corresponds to $\Theta(\mathbf{x}, 0) = 1/2$.

Four calculations are documented below for $10^3 \leq Ra \leq 10^6$ using meshes labeled by A to D. In order to minimize the volume of data presented for this verification study, one mesh resolution at each Ra was selected. The mesh resolution was selected using the temperature error data in Figure 8 of [46], and extrapolating the error level to a nearly constant level of 1% for all Ra. The total time scale of each calculation is 2.0 time units which corresponds to twice the diffusional time-scale for the differentially-heated cavity.

This is sufficient for the flow to establish steady-state conditions. Time-step control that limits the maximum CFL to $CFL_{max} = 40$ is used with the backward-Euler time integrator. Time-integration is carried out until a steady-state condition results, monitored in terms of the time-history of the integrated wall heat transfer rate, velocity, temperature, and global kinetic energy. Based on the kinetic energy, all simulations reach an approximate steady state by 1 time unit. Surface-based time-history output is used to monitor the velocity and temperature at the elements at the mid-side of the vertical walls and to output the integrated heat transfer rate on the heated wall.

Figure 13(a) – (d) shows the temperature distribution for the four Rayleigh numbers considered. Figure 13(e) shows the variation in the Nusselt number along the vertical heated wall for different Rayleigh numbers. Figure 13(b) shows the time-history of the kinetic energy. Figure 13(c) and (d) show the x-, and y-velocity profiles along the vertical and horizontal centerlines, respectively.

A comparison of the computed results is given in Table 1 to data obtained using Richardson extrapolation by De Vahl Davis [46]. The minimum and maximum velocities are computed along the horizontal and vertical centerlines of the cavity. The mean Nusselt number is computed as

$$\overline{Nu} = \frac{1}{A} \int_{\Gamma} \nabla \Theta \cdot \mathbf{n} \; d\Gamma \tag{69}$$

where A is the surface area of the domain. For all computations, a z-dimension of $\Delta z = 0.0125$ was used with $L = 1$ resulting in an area $A = 0.0125$. In order to compute the mean Nusselt number, the surface heat transfer is scaled by $1/A$. The minimum and maximum Nusselt numbers were extracted from the non-dimensional heat flux distribution along the heated wall.

4.3. Porous Medium

This verification problem computes fluid flow at the interface of a fluid layer and a porous medium, first considered in [47]. The interface is setup by confining the porous medium to the top half of a channel shown in Figure 14. We use both a uniform and graded meshes in the convergence studies and also verify the fully-implicit time stepping algorithm in the problem. For details on porous drag and fully-implicit time stepping see [37].

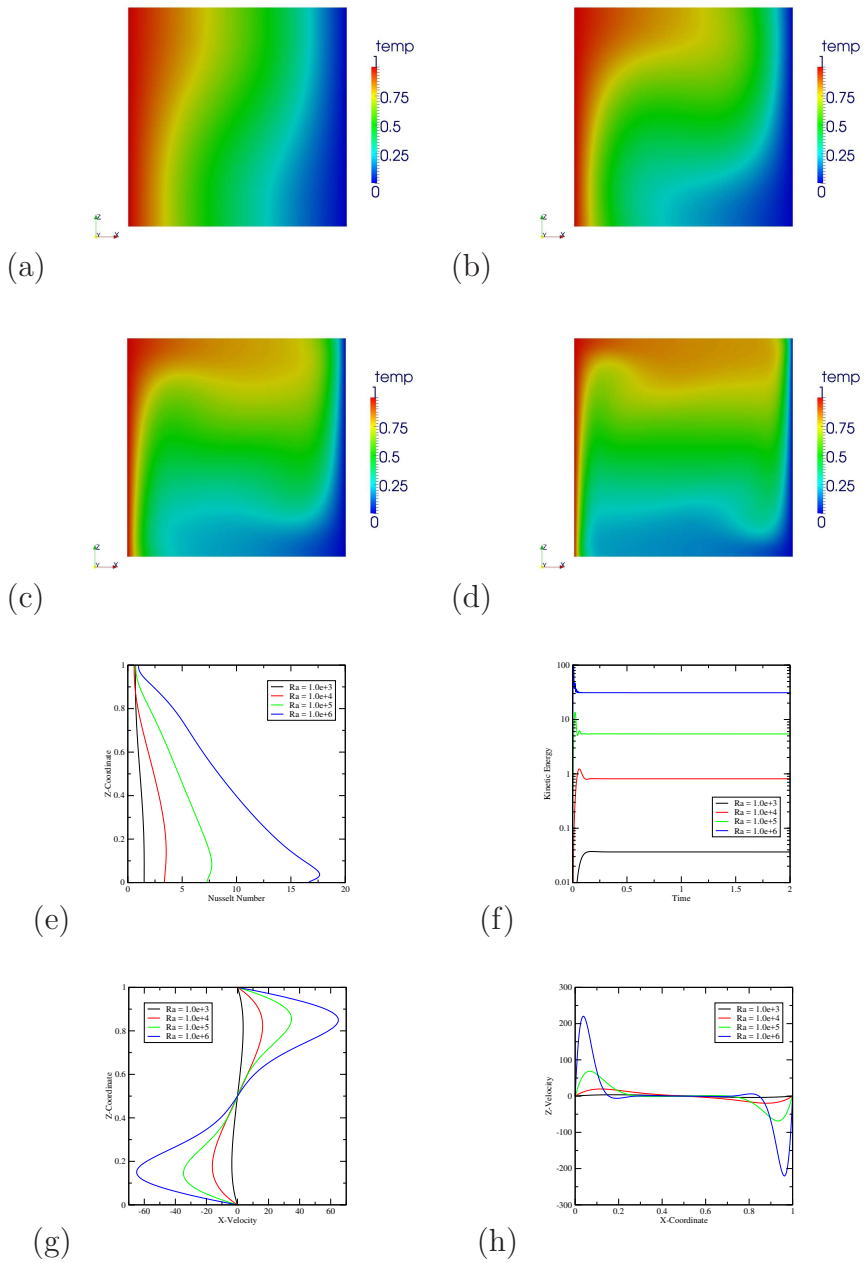


Figure 13: Temperature distribution at $t = 2$ for (a) $Ra = 10^3$ using mesh A, (b) $Ra = 10^4$ using mesh B, (c) $Ra = 10^5$ using mesh C, and (d) $Ra = 10^6$ using mesh D, and (e) Nusselt number profile along the vertical heated wall, (f) global kinetic energy time history, (g) x velocity along the vertical centerline, and (h) z-velocity along the horizontal centerline for $Ra = 10^3, 10^4, 10^5, 10^6$.

	Rayleigh Number (Ra)					
	10^3			10^4		
	Ref. [46]	80×80	% Error	Ref. [46]	160×160	% Error
$v_{x_{max}}$	3.659	3.648	0.300	16.178	16.179	0.006
$v_{z_{max}}$	3.697	3.692	0.135	19.617	19.574	0.219
$\overline{Nu}$	1.118	1.118	0.000	2.243	2.246	0.134
Nu_{min}	0.692	0.691	0.144	0.586	0.585	0.171
Nu_{max}	1.505	1.507	0.133	3.528	3.536	0.227

	10^5			10^6		
	Ref. [46]	320×320	% Error	Ref. [46]	320×320	% Error
$v_{x_{max}}$	34.73	34.79	0.173	64.63	64.91	0.433
$v_{z_{max}}$	68.59	68.62	0.044	219.36	220.33	0.442
$\overline{Nu}$	4.519	4.523	0.089	8.800	8.841	0.466
Nu_{min}	0.729	0.727	0.274	0.989	0.978	1.112
Nu_{max}	7.717	7.727	0.130	17.925	17.643	1.573

Table 1: Maximum velocities, mean, minimum and maximum Nusselt numbers compared to those reported De Vahl Davis [46] using Richardson extrapolation.

Using a volume-averaging approach, the averaged form of the mass and momentum conservation equations for a fluid-saturated "constant" porosity medium that takes into account the presence of solid boundaries (Brinkman correction [48]) and inertial effects (Forchheimer correction) is [49, 50].

$$\nabla \cdot \langle \mathbf{v} \rangle = 0. \tag{70}$$

$$\frac{\rho}{\epsilon}\left(\frac{\partial \langle \mathbf{v} \rangle}{\partial t} + \frac{1}{\epsilon}\langle \mathbf{v} \rangle \cdot \nabla \langle \mathbf{v} \rangle\right) = -\nabla \langle p \rangle^{\beta} + \nabla \cdot \mu_{\beta} \nabla \langle \mathbf{v} \rangle - \mu \mathbf{K}^{-1} \langle \mathbf{v} \rangle - \rho C_F \mathbf{K}^{-\frac{1}{2}} |\langle \mathbf{v} \rangle| \langle \mathbf{v} \rangle, \tag{71}$$

where, $\langle p \rangle^{\beta}$ is the intrinsic phase-averaged pressure, $\langle \mathbf{v} \rangle$ is the superficial phase-averaged velocity in the porous medium, μ_{β} is the Brinkman effective viscosity, C_F is the Forchheimer coefficient, and $\mathbf{K}$ is the permeability tensor. The Forchheimer coefficient C_F is expressed as

$$C_F = \frac{1.75}{\sqrt{150\epsilon^3}}. \tag{72}$$

Figure 14: Schematic of the Beavers-Joseph problem.

The flow is driven by specifying inlet and outlet pressures as

$$p = \langle p \rangle^{\beta} = p_{in} \qquad \text{at } x = 0 \tag{73}$$

$$p = \langle p \rangle^{\beta} = 0 \qquad \text{at } x = L, \tag{74}$$

and, wall boundary conditions that consist of

$$\mathbf{v} = \langle \mathbf{v} \rangle = \mathbf{0} \qquad \text{at } y = \pm H. \tag{75}$$

At the interface, both the tangential velocity and the vertical shear of the tangential velocity must match:

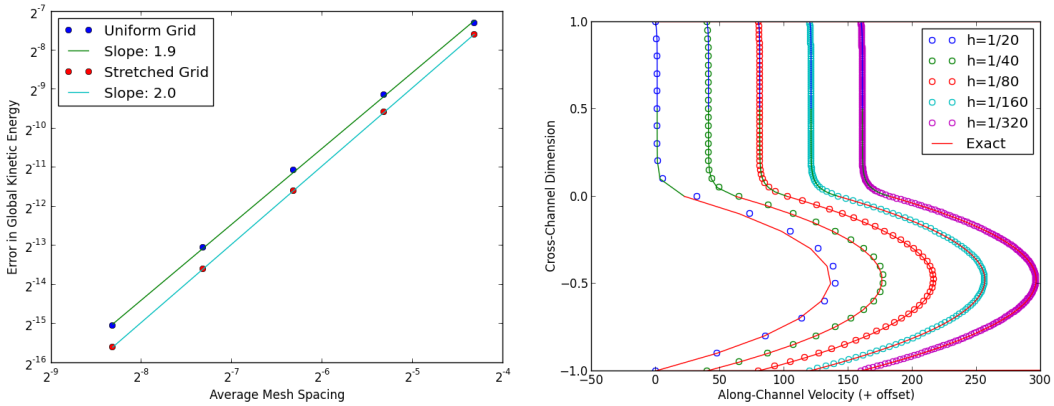
$$\langle v_x \rangle|_{0^+} = v_x|_{0^-} \tag{76}$$

$$\left.\frac{d\langle v_x \rangle}{dy}\right|_{0^+} = \left.\frac{dv_x}{dy}\right|_{0^-}. \tag{77}$$

A series of five meshes with uniform spacing is considered for the initial convergence study. As a first measure, we consider the asymptotic value of the global kinetic energy. Richardson extrapolation was used to accelerate the convergence of the series of values of steady-state kinetic energy at the five resolutions to estimate the value at infinite resolution. Subsequently, the kinetic energy error measures are computed with respect to this extrapolated value. An analytical solution is developed in Appendix §6 and used to directly assess errors in the computed velocity field.

Error on the five meshes is shown using filled blue circles in Fig. 15a. Also shown is the least-squares linear fit of the error on the five successively refined meshes; with the slope indicated in the legend. Since the resolution is successively doubled, a base-2 logarithmic scaling is used for the axes to facilitate visual evaluation of the scaling of error. A close to quadratic reduction of the error is clear in this figure for each level of mesh refinement. In

Fig. 15b, the velocity profile computed on the series of five meshes is plotted. Also plotted is the above exact solution in red. To facilitate comparison, the velocity profiles on the different meshes are offset by a constant value. Reduction in error with increasing resolution is visually evident in going from $h = 1/20$ to $h = 1/40$ to $h = 1/80$.



(a) Global Kinetic Energy

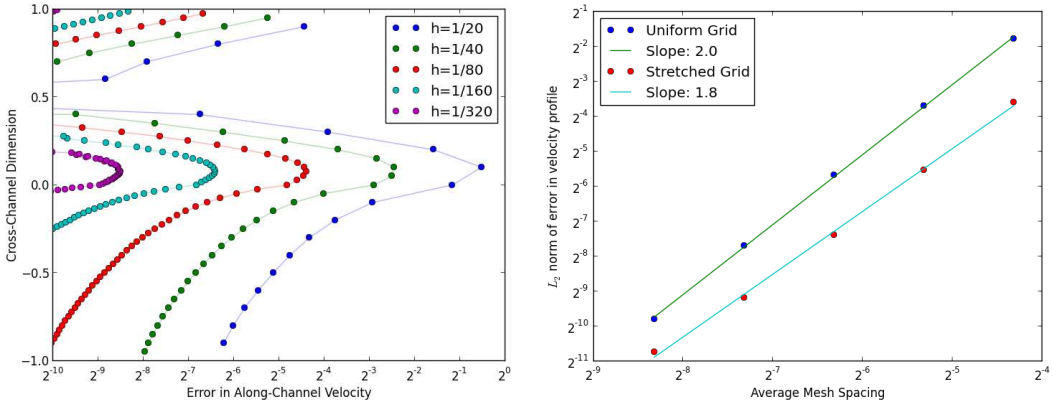
(b) Velocity Profiles

Figure 15: (a) Error in the global kinetic energy converges approximately quadratically, a series of uniform and a series of graded meshes, and (b) velocity profile in the Beavers-Joseph problem at five different resolutions of an uniform mesh.

To better quantify errors in the velocity profile, Fig. 16a shows profiles of error in velocity as a function of the cross-channel dimension. In this figure, a base-2 logarithmic scaling is used again to facilitate visual evaluation of reduction in error with increasing resolution. The peak fractional error occurs in the porous medium close to the interface with the pure fluid region. Further, this peak error can be seen to scale quadratically with resolution: a two-fold increase in resolution is seen to reduce error approximately by a factor of four.

To examine the scaling of the global error in velocity profile, the L_2 norm of the profile of errors is computed and plotted using filled blue circles in Fig. 16b. The corresponding linear-least-squares fit is shown as a green line. Its slope of two, indicated in the legend, verifies quadratic convergence.

We next consider a series of five graded meshes. In this case, the mesh spacing in the flow direction is held constant and the same as before. Only the cross-channel mesh spacing is graded to allow increased resolution in the region of boundary and interface layers. This was achieved by using



(a) Fractional Errors

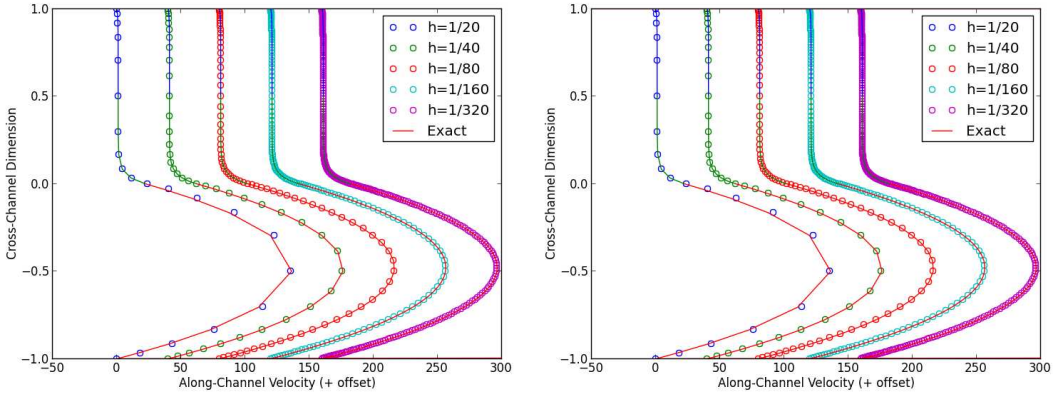
(b) L_2 Error on Graded Meshes

Figure 16: (a) fractional errors—errors normalized by exact solution are considered. Errors are largest in the porous region close to the fluid interface, (b) convergence of L_2 norm error in velocity profile using uniform and graded meshes is seen to be quadratic or close to quadratic.

geometric stretching factors that enforced the ratio of the largest to the smallest mesh size in the cross-channel direction to be ten at each of the resolutions. Figure 17a shows the velocity profile using the same format as before. A reduction in error with respect to the exact solution, and as compared to error on the uniform mesh, is visually evident at low resolutions, attesting to the utility of the graded mesh.

The reduction in error is quantified in Fig. 16b. Here the L_2 error in the velocity profile with the graded mesh is seen to be consistently lower than that with uniform mesh spacing (by a factor of between two to three). Further, a close to quadratic scaling of this error is seen on the graded mesh as well. The slightly lower convergence rate in this measure is not clearly understood. Indeed, this lower rate may not be significant, because the global kinetic energy error on the graded mesh converges slightly faster than on the uniform mesh.

Using the same set of five uniform meshes and five graded meshes considered above, time stepping is changed from the default semi-implicit projection method to the fully-implicit (Picard) method. While the time step in the semi-implicit projection method was limited to not exceed $CFL_{max} = 10$, this limit was chosen to be $CFL_{max} = 100$ in the fully-implicit calculation. The same set of diagnostics considered previously are shown for this set of



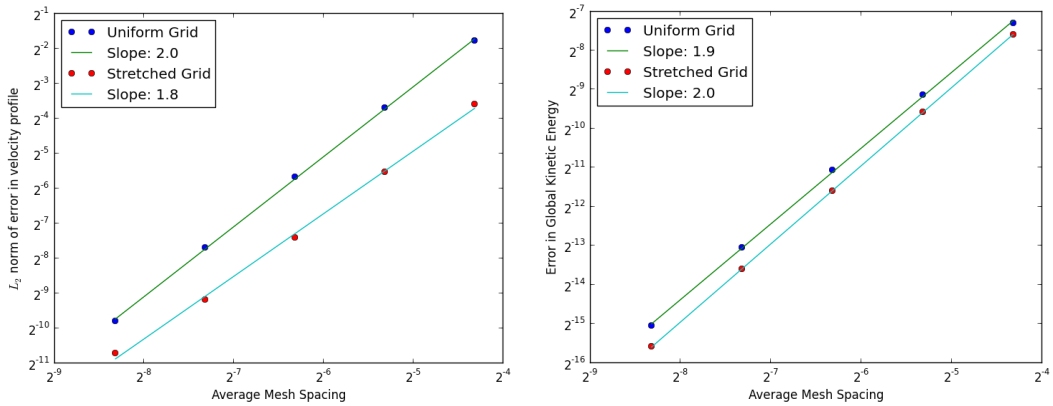
(a) Semi-Implicit Projection (b) Fully-Implicit Projection

Figure 17: Velocity profile using a graded mesh at five different resolutions using (a) semi-implicit projection, and (b) fully-implicit projection.

computations in Figures 17b, 18a, and 18b. Close correspondence in these diagnostics between the semi-implicit and fully-implicit computations verifies the accuracy of the fully-implicit discretization of the advection term. Ultimately, the choice between the semi-implicit and fully-implicit is problem dependent with a trade-off between overall computational time and time-step size. In general, when the number of non-linear iterations is smaller than the ratio of the fully- and semi-implicit time-step sizes, the fully-implicit method delivers shorter overall computational times.

5. Summary & Future Work

A new second-order, incremental projection algorithm for transient, incompressible viscous flows has been introduced. The hybrid finite-element/finite-volume discretization is locally-conservative, circumvents stability problems due to the divergence operator without requiring a stabilized Petrov-Galerkin formulation or Rhie-Chow interpolation. Second-order accurate discretization of advection has been demonstrated on mixed element topology meshes. The algorithm is extensible, permitting the incorporation of features such as porous media flow, RANS and LES turbulence models, and semi-, and fully-implicit time stepping. A series of verification problems have been used to illustrate the convergence properties of the algorithm. The temporal stability properties have been demonstrated on a range of



(a) L_2 Velocity Errors (b) Global Kinetic Energy

Figure 18: (a) convergence of L_2 norm error in velocity profile using fully-implicit time stepping on uniform and graded meshes is quadratic or close to quadratic and is seen to be approximately the same as with semi-implicit time stepping, (b) convergence of error in global kinetic energy when using fully-implicit time stepping on a series of five uniform and graded meshes is quadratic or close to quadratic, closely resembling convergence with semi-implicit time stepping.

problems with $2 \leq CFL \leq 100$. Future work will report on the development of a fully-implicit hybrid projection algorithm, extensions to arbitrary Lagrangian-Eulerian computations, conjugate heat transfer, and Eulerian multiphase flows.

Acknowledgments

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (http://www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. DE-AC52-06NA25396. Los Alamos Report LA-UR-16-22436.

6. Appendix

In order to develop an exact solution for the Beavers-Joseph verification problem, discussed in §4.3, it is assumed that the thickness of the momentum boundary layer in the porous medium is smaller than $H/2$. This assumption is equivalent to an assumption of flatness and symmetry of the velocity profile at $H/2$:

$$\frac{d^2\langle v_x\rangle}{dy^2} = 0 \qquad \text{at } y = \frac{H}{2} \tag{78}$$

$$\frac{d\langle v_x\rangle}{dy} = 0 \qquad \text{at } y = \frac{H}{2} \tag{79}$$

We follow an approach analogous to that used by Vafai and Kim, 1990 [51]. The momentum equation in the porous medium is

$$0 = -\frac{d\langle p\rangle^\beta}{dx} + \frac{\mu}{\epsilon}\frac{d^2\langle v_x\rangle}{dy^2} - \mu K^{-1}\langle v_x\rangle - \rho C_F K^{-\frac{1}{2}}\langle v_x\rangle^2. \tag{80}$$

In the (pure) fluid region, the momentum equation reduces to

$$0 = -\frac{dp}{dx} + \frac{d^2 v_x}{dy^2} \tag{81}$$

The problem therefore reduces to solving the above pair of equations for $v_x(y), \langle v_x\rangle(y)$ subject to the boundary and matching conditions in Eq. (75) – (79).

The exact solution is then a combination of three piecewise solutions that satisfy the appropriate matching conditions:

- In the porous region $\frac{H}{2} \leq y \leq H$, the solution from the previous section can be used.
- In the porous region $0 \leq y \leq \frac{H}{2}$, a solution has to be found that satisfies Eq. (79) – (77).
- In the fluid region $-H \leq y \leq 0$ a solution has to be found that satisfies Eq. (77) and the boundary condition Eq. (75) at $y = -H$.

In the porous region $0 \leq y \leq \frac{H}{2}$, we again assume that the half-width of the porous layer is greater than interface layer. Then considering Eq. (80)

independently within the interface layer and the central porous layer, the pressure gradient can be eliminated to yield

$$\frac{d^2\langle v_x\rangle}{dy^2} = B\left(\langle v_x\rangle - 1\right) + A\left(\langle v_x\rangle^2 - 1\right). \tag{82}$$

Again, the above equation is valid in the region $0 \leq y \leq \frac{H}{2}$. Here, y has been non-dimensionalized by H and $\langle v_x\rangle$ has been non-dimensionalized by $\langle v_x\rangle|_{H/2}$. In the above equation,

$$A = \frac{\epsilon H^2 C_F \, \langle v_x\rangle|_{H/2}}{\nu\sqrt{K}}, \tag{83}$$

$$B = \frac{\epsilon H^2}{K}. \tag{84}$$

Similarly, in the fluid region $-H \leq y \leq 0$, the pressure gradient can be eliminated to reduce Eq. (81) to

$$\frac{d^2 v_x}{dy^2} = -(A+B). \tag{85}$$

The above equation is readily integrated to yield

$$v_x = -\frac{A+B}{2}(y+1)^2 + \left(v_x|_i + \frac{A+B}{2}\right)(y+1) \tag{86}$$

Using $\frac{d\langle v_x\rangle}{dy}$ as the integrating factor in Eq. (82) yields

$$\frac{d\langle v_x\rangle}{dy} = -\left(\langle v_x\rangle - 1\right)\sqrt{\frac{2A}{3}(\langle v_x\rangle - u_1)} \tag{87}$$

where

$$u_1 = -\left(2 + \frac{3B}{2A}\right).$$

The interface matching condition reduces to an algebraic equation that can then be solved for the interface velocity using Newton's method:

$$v_x|_i - \frac{A+B}{2} = -\left(v_x|_i - 1\right)\sqrt{\frac{2A}{3}(v_x|_i - u_1)}. \tag{88}$$

With this, Eq. (86) is fully determined and provides the exact parabolic profile in the fluid region.

Finally, following Vafai and Kim, 1990, Eq. (82) can be integrated to yield

$$\langle v_x \rangle = u_1 + (1 - u_1) \left(\frac{1 + z}{1 - z} \right)^2 \tag{89}$$

where

$$z = \frac{\sqrt{v_x|_i - u_1} - \sqrt{1 - u_1}}{\sqrt{v_x|_i - u_1} + \sqrt{1 - u_1}} \exp \left[-y \sqrt{\frac{2A}{3}(1 - u_1)} \right]$$

References

[1] J. V. Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM Journal for Scientific and Statistical Computing 7 (1986) 870–891.

[2] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible navier-stokes equations, Journal of Computational Physics 85 (1989) 257–283.

[3] P. M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 1: Theory, International Journal for Numerical Methods in Fluids 11 (1990) 587–620.

[4] P. M. Gresho, S. T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 2: Implementation, International Journal for Numerical Methods in Fluids 11 (1990) 621–659.

[5] P. M. Gresho, S. T. Chan, M. A. Christon, A. C. Hindmarsh, A little more on stabilized $q1q1$ for transient viscous incompressible flow, International Journal for Numerical Methods in Fluids 21 (1995) 837–856.

[6] P. M. Gresho, S. T. Chan, Projection 2 goes turbulent – and fully implicit, International Journal for Computational Fluid Dynamics 9 (3-4) (1998) 249 – 272.

[7] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, An adaptive projection method for the incompressible euler equations, in: Eleventh AIAA Computational Fluid Dynamics Conference, AIAA, 1993, pp. 530–539.

[8] A. S. Almgren, J. B. Bell, W. G. Szymcyzk, A numerical method for the incompressible navier-stokes equations based on an approximate projection, SIAM Journal for Scientific Computing 17 (2) (1996) 358–369.

[9] A. S. Almgren, J. B. Bell, W. Y. Crutchfield, Approximate projection methods: Part i. inviscid analysis, SIAM Journal on Scientific Computing 22 (4) (2000) 1139–1159.

[10] W. J. Rider, The robust formulation of approximate projection methods for incompressible flows, Tech. Rep. LA-UR-3015, Los Alamos National Laboratory (1994).

[11] W. J. Rider, Filtering nonsolenoidal modes in numerical solutions of incompressible flows, Tech. Rep. LA-UR-3014, Los Alamos National Laboratory, Los Alamos, New Mexico (September 1994).

[12] W. J. Rider, D. B. Kothe, S. J. Mosso, J. H. Cerutti, J. I. Hochstein, Accurate solution algorithms for incompressible multiphase flows, Tech. Rep. AIAA-95-0699, AIAA, Reno, Nevada (January 1995).

[13] W. J. Rider, Approximate projection methods for incompressible flow: implementation, variants and robustness, Tech. Rep. LA-UR-2000, Los Alamos National Laboratory, Los Alamos, New Mexico (July 1995).

[14] M. L. Minion, A projection method for locally refined grids, Journal of Computational Physics 127 (1996) 158–178.

[15] J.-L. Guermond, L. Quartapelle, Calculation of incompressible viscous flow by an unconditionally stable projection fem, Journal of Computational Physics 132 (1997) 12–23.

[16] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, Journal of Computational Physics 130 (1997) 269–282.

[17] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for two-phase flows, Journal of Computational Physics 148 (1999) 81–124.

[18] O. M. Knio, H. N. Najm, P. S. Wyckoff, A semi-implicit numerical scheme for reacting flow. ii. stiff operator-split formulation, pre-print submitted to Journal of Computational Physics 154 (2) (1999) 428–467.

[19] M. A. Christon, R. S. Patil, A finite element projection method for low-Mach number reacting flows, in: K. J. Bathe (Ed.), Third MIT Conference on Computational Fluid and Solid Mechanics, Elsevier, New York, 2005, pp. 617–622.

[20] S. P. Schofield, M. A. Christon, V. Dyadechko, R. V. Garimella, R. B. Lowrie, B. K. Swartz, Multi-material incompressible flow simulation using the moment-of-fluid method, International Journal for Numerical Methods in Fluids 63 (2010) 931–952.

[21] D. L. Brown, M. L. Minion, Performance of under-resolved two-dimensional incompressible flow simulations, Journal of Computational Physics 122 (1995) 165–183.

[22] M. L. Minion, D. L. Brown, Performance of under-resoloved two-dimensional incompressible flow simulations, II, Journal of Computational Physics 138 (1997) 734–765.

[23] B. B. Wetton, Error analysis of pressure increment schemes, submitted to SINUM 38 (1) (1998) 160–169.

[24] J.-L. Guermond, Some implementations of projection methods for Navier-Stokes equations, Mathematical Modelling and Numerical Analysis 30 (5) (1996) 637–667.

[25] J.-L. Guermond, A convergence result for the approximation of the Navier-Stokes equations by an incremental projection method, C. R. Acad. Sci. Paris 325 (1997) 1329–1332.

[26] J.-L. Guermond, L. Quartapelle, On the approximation of the unsteady Navier-Stokes equations by finite element projection methods, Numerische Mathematik 80 (1998) 207–238.

[27] J.-L. Guermond, L. Quartapelle, On stability and convergence of projection methods based on pressure Poisson equation, International Journal for Numerical Methods in Fluids 26 (1998) 1039–1053.

[28] O. A. Ladyzhenkaya, The Mathematical Theory of Viscous Incompressible Flow, 2nd Edition, Gordon and Breach Scienc Publishers, Inc., 1969.

[29] I. Babuška, Error-bounds for the finite element method, Numerische Mathematik 16 (1971) 322–333.

[30] I. Babuška, The finite element method with Lagrangian multipliers, Numerische Mathematik 20 (1973) 179–192.

[31] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers, Revue Francais d' Automatique, Informatique, et Recherce Opérationnelle (R.A.I.R.O.) R-2 (1974) 129–151.

[32] P. M. Gresho, R. L. Sani, Incompressible flow and the finite element method, Advection-diffusion and isothermal laminar flow, John Wiley & Sons, Chicester, England, 1998.

[33] S. Tu, S. Aliabadi, Development of a hybrid finite volume/element solver for incompressible flows, International Journal for Numerical Methods in Fluids 20 (2007) 177–203.

[34] S. Tu, S. Aliabadi, R. Patel, M. Watts, An implementation of the spalart-allmaras des model in an implicit unstructured hybrid finite volume/element solver for incompressible turbulent flow, International Journal for Numerical Methods in Fluids 30 (2009) 1051–1062.

[35] T. Wan, S. Aliabadi, C. Bigler, A hybrid scheme based on finite element/volume methods for two immiscible fluid flows, International Journal for Numerical Methods in Fluids 61 (2009) 930–944.

[36] D. P. Flanagan, T. Belytschko, A uniform strain hexahedron and quadrilateral with orthogonal hourglass control, International Journal for Numerical Methods in Engineering 17 (1981) 679–706.

[37] M. A. Christon, Hydra-TH Theory Manual, Tech. Rep. LA-UR 11-05387, Los Alamos National Laboratory (September 2011).

[38] M. A. Christon, M. J. Martinez, T. E. Voth, Generalized Fourier analysis of the advection-diffusion equation – part I: One-dimensional domains, International Journal for Numerical Methods in Fluids 45 (2004) 839–887.

[39] T. E. Voth, M. J. Martinez, M. A. Christon, Generalized Fourier analysis of the advection-diffusion equation – part II: Two-dimensional domains, International Journal for Numerical Methods in Fluids 45 (2004) 889–920.

[40] M. A. Christon, D. I. Ketcheson, A. C. Robinson, An assessment of semi-discrete central schemes for hyperbolic conservation laws, SAND2003-3238, Sandia National Laboratories, Albuquerque, New Mexico (May 2003).

[41] A. M. Baptista, E. E. Adams, P. Gresho, Benchmarks for the transport equation: the convection-diffusion equation and beyond, Quantitative Skill Assessment for Coastal Ocean Models 47 (1995) 241–268.

[42] X. Yidong, C. Wang, H. Luo, M. Christon, J. Bakosi, Assessment of a hybrid finite element and finite volume code for turbulent incompressible flows, Journal of Computational Physics 307 (2016) 653 – 669.

[43] M. A. Christon, R. Lu, J. Bakosi, B. Nadiga, Z. Karoutas, M. Berndt, Large-eddy simulation, fuel rod vibration and grid-to-rod fretting, submitted to Journal of Computational Physics.

[44] E. Erturk, B. Dursun, Numerical solutions of 2-d steady incompressible flow in a driven skewed cavity, ZAMM - Journal of Applied Mathematics and Mechanics 87 (2007) 377–392.

[45] G. de Vahl Davis, I. P. Jones, Natural convection in a square cavity: a comparison exercise, International Journal for Numerical Methods in Fluids 3 (1983) 227–248.

[46] G. de Vahl Davis, Natural convection of air in a square cavity: a bench mark numerical solution, International Journal for Numerical Methods in Fluids 3 (1983) 249–264.

[47] G. S. Beavers, D. D. Joseph, Boundary conditions at a naturally permeable wall, Journal of Fluid Mechanics 1 (1967) 197 – 207.

[48] H. C. Brinkman, A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles, Applied Sci. Res 1 (1) (1949) 27–34.

[49] A. R. A. Khaled, K. Vafai, The role of porous media in modeling flow and heat transfer in biological tissues, Int. J. Heat Mass Transf. 46 (26) (2003) 4989–5003.

[50] P. Nithiarasu, K. S. Sujatha, K. Ravindran, T. Sundararajan, K. N. Seetharamu, Non-Darcy natural convection in a hydrodynamically and thermally anisotropic porous medium, Comput. Meth. Appl. Mech. Eng. 188 (1-3) (2000) 413–430.

[51] K. Vafai, S. Kim, Fluid mechanics of the interface region between a porous medium and a fluid layeran exact solution, International Journal of Heat and Fluid Flow 11 (3) (1990) 254–256.