Toward verification of formal time accuracy for a family of approximate projection methods using the method of manufactured solutions

By S. P. Domino†

This paper provides an overview of the approximate projection methods commonly used in the unstructured finite volume community that have been implemented within SIERRA/Fuego. The paper outlines splitting and stabilization errors that appear in many common approximate projection methods. Smoothing errors, or stabilization errors, can appear in both projection and fully coupled algorithms and, as will be shown, can have drastic effects on the formal time accuracy of a chosen algorithm. A time-dependent manufactured solution is presented and used to verify formal order of accuracy for a suite of approximate projection algorithms. For a commonly used set of projection scaling time scales, first-order accuracy is demonstrated regardless of time integration scheme, i.e., backward Euler or Crank-Nicholson. A new class of approximate projection algorithms is presented that circumvents the stabilization error. This method is based on the classic pressure projection method, where the pressure is computed based on solving a Pressure Poisson Equation that is derived by taking the divergence of the momentum equation. Results show, however, that this algorithm (as implemented) is not competitive based on the significant expense of solving this second Poisson equation.

1. Introduction

The role of pressure smoothing, or explicit stabilization, was first developed in the context of collocated finite volume schemes by Rhie and Chow (1983). Although this original paper did not explore the formal error introduced by this explicit stabilization, Majumdar (1988) later displayed the sensitivity of steady results on relaxation parameters and provided a methodology to circumvent this issue. In general, such early papers (cf. Perić *et al.* (1988)) as well as other more recent papers, (cf. Papageorgakopoulos *et al.* (2000)) introduced the role of stabilization almost by happenstance as it entered only through the specific choice of the convecting velocity formula, i.e., the integration point velocity that forms the mass flow rate.

Studies of Codina (2001) and Soto *et al.* (2001), each in the context of a finite element algorithm, have commented on the role of stabilization that is provided by the approximation of the derived pressure correction system, namely that $\mathbf{L} \neq \mathbf{DG}$, where $\mathbf{L}$ is the given discrete Laplacian operator and $\mathbf{D}$ and $\mathbf{G}$ are the chosen discrete divergence and gradient operators, respectively. Numerical algorithms for which the Laplacian operator does not equal the discrete divergence of gradient operator have been termed "approximate projection" algorithms (cf. Almgren *et al.* (2000), Codina and Badia (2005)) in the context of solenoidal flow; in general for non-solenoidal flow the formalism of the projection derivation results in an affine projection (see Section 7).

† Thermal Fluids Computational Engineering Sciences, Sandia National Laboratories
spdomin@sandia.gov

Recent work by Sandia National Laboratories has cast the general approximate projection algorithm within a family of smoothing and time scaling choices. The analysis of choice that has been followed is to cast the algorithm in terms of an approximate factorization (cf. Dukowicz and Dvinsky (1992)), and note the added stabilization (herein also known as *pressure smoothing terms*), and splitting errors. This analysis has been extremely useful in understanding the formal accuracy, and even consistency, of a given numerical scheme.

The analysis of a given computational fluids algorithm in the context of an approximate factorization begins with the discrete momentum and continuity equations written in matrix form. The matrix $\mathbf{A}$ contains discrete, linearized contributions to the momentum equations from the time derivative, convection, and diffusion terms,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix}. \tag{1.1}$$

The discrete nodal gradient and nodal divergence are $\mathbf{G}$ and $\mathbf{D}$ respectively (note that the operator $\mathbf{D}$ may include aspects of the algorithm due to a variable density field). The function $\mathbf{f}$ contains the additional terms for the momentum equations, e.g., body force terms, lagged stress tensor terms, etc., while the the function $\mathbf{b}$ contains the appropriate terms for a non-solenoidal velocity field, i.e., $-\int \frac{\partial \rho}{\partial t} dV$. The pressure is appropriately interpreted as the pressure at the $n + \frac{1}{2}$ step, (cf. Strikwerda and Lee (1999)). The form of the matrix operators can be found in the body of literature for control-volume finite element methods (cf. Schneider (1988)). Note that Eq. (1.1) is not really complete as the boundary condition values are omitted, however, they are not essential in describing the bulk of the splitting and stabilization analysis as noted by Perot (1993). The boundary conditions would simply enter through an additional vector on the right-hand side and modified entries in the matrix operators.

The approximate factorization of Eq. (1.1) takes the general form of

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & \mathbf{B_1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B_2 G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{A B_2 G} \\ \mathbf{D} & \mathbf{B_1} + \mathbf{D B_2 G} \end{bmatrix}. \tag{1.2}$$

The factor $\mathbf{B_2}$ determines the projection time scale. The factor $\mathbf{B_1}$ defines the linear system for pressure. Ideally, $\mathbf{B_1}$ could be selected to cancel splitting errors in the continuity equation. Practically, the form of $\mathbf{B_1}$ is governed by implementation and linear solver efficiency.

A completely generalized set of incremental pressure projection methods with potential stabilization can be written by formally defining the operators $\mathbf{B_1}$ and $\mathbf{B_2}$ above, here shown as part of the sequence of equations solved,

$$\mathbf{A}\Delta\hat{\mathbf{u}} = \mathbf{f} - \mathbf{G}p^{n-\frac{1}{2}} - \mathbf{A}\mathbf{u}^n, \tag{1.3}$$

$$-\mathbf{L_1}\Delta p^{n+\frac{1}{2}} = -\mathbf{D}\left(\hat{\mathbf{u}} + \tilde{\tau}_2 \mathbf{G} p^{n-\frac{1}{2}}\right) + \mathbf{L_2} p^{n-\frac{1}{2}} + b, \tag{1.4}$$

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \tilde{\tau}_3 \mathbf{G}\Delta p^{n+\frac{1}{2}}. \tag{1.5}$$

Laplacian operators acting on a general scalar ϕ, which define the approximate nature of the projection method, are given by,

$$\mathbf{L}_1\phi = \tau_1 \nabla\phi \cdot d\mathbf{A}, \tag{1.6}$$

$$\mathbf{L_2}\phi = \tau_2 \nabla\phi \cdot d\mathbf{A}. \tag{1.7}$$

$$\tag{1.8}$$

For an approximate projection method,

$$\mathbf{L_2} \neq \mathbf{D}\tilde{\tau}_2\mathbf{G}, \tag{1.9}$$

while for an exact projection,

$$\mathbf{L_2} = \mathbf{D}\tilde{\tau}_2\mathbf{G}. \tag{1.10}$$

Exact projections can be easily constructed on unstructured collocated meshes (cf. Chorin (1968)), although classically this results in a wide Laplacian stencil that admits pressure oscillations yet does not add discrete errors in the continuity solve. We assume that τ_i factors defined above are represented by a diagonal matrix that corresponds to a particular time scale of choice. The relationship between $\tilde{\tau}_i$ and τ_i is normalization by a density and volume,

$$\tilde{\tau}_i = \frac{\tau_i}{\rho V}. \tag{1.11}$$

The choice of these scaling factors defines the scheme in terms of both stabilization and projection scaling. For example, the ideal form for $\tilde{\tau}_3$ is the inverse of $\mathbf{A}$. The exact choice of $\tilde{\tau}_3$ in a practical sense affects the stability of the scheme. The stabilization terms are represented by operators including both $\tilde{\tau}_1$ and $\tilde{\tau}_2$ that are required to prevent velocity and pressure decoupling in schemes for which $\mathbf{L} \neq \mathbf{DG}$.

Rearrangement of Eq. (1.5), in terms of $\hat{\mathbf{u}}$, and substitution of this modified equation into Eq. (1.3) and Eq. (1.4) provides the full set of splitting and stabilization errors:

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau}_3)\,\mathbf{G}\Delta p^{n+\frac{1}{2}} \\ (\mathbf{L_1} - \mathbf{D}\tilde{\tau}_3\mathbf{G})\Delta p^{n+\frac{1}{2}} + (\mathbf{L_2} - \mathbf{D}\tilde{\tau}_2\mathbf{G})p^{n-\frac{1}{2}} \end{bmatrix}. \tag{1.12}$$

The error appearing in the momentum equation is due to splitting and generally can be repaired by non-linear iteration, although ideally single iteration methods are desired (as shown).

Again it is emphasized that for approximate projection methods, $\mathbf{L_2} \neq \mathbf{D}\tilde{\tau}_2\mathbf{G}$, whereas for exact projection methods, which are usually based on staggering velocity and pressure, $\mathbf{L_2} = \mathbf{D}\tilde{\tau}_2\mathbf{G}$ and there is no stabilization error (as there is no need to provide stabilization). Frequently, the stabilization terms within Eq. (1.4) are included in a modified provisional velocity (cf. Kim and Choi (2000)), i.e., $\tilde{\mathbf{u}} = \hat{\mathbf{u}} + \tilde{\tau}_2\mathbf{G}p^{n-\frac{1}{2}}$, that can often hide the true role of stabilization.

A similar analysis for *pressure free* projection methods (cf. Kim (1985)) can be carried out, in which case the equations solved are given by,

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{f} - \mathbf{A}\mathbf{u}^n, \tag{1.13}$$

$$-\mathbf{L_1}\Delta\phi^{n+1} = -\mathbf{D}\hat{\mathbf{u}} + \mathbf{L_1}\phi^n + b, \tag{1.14}$$

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \tilde{\tau}_1\mathbf{G}\phi^{n+1}, \tag{1.15}$$

with errors,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} -\mathbf{A}\tilde{\tau}_3\mathbf{G}\phi^{n+1} + \mathbf{G}p^{n+\frac{1}{2}} \\ (\mathbf{L_1} - \mathbf{D}\tilde{\tau}_1\mathbf{G})\,\phi^{n+1} \end{bmatrix}. \tag{1.16}$$

The error term in the continuity equation is retained to emphasize that this algorithm can

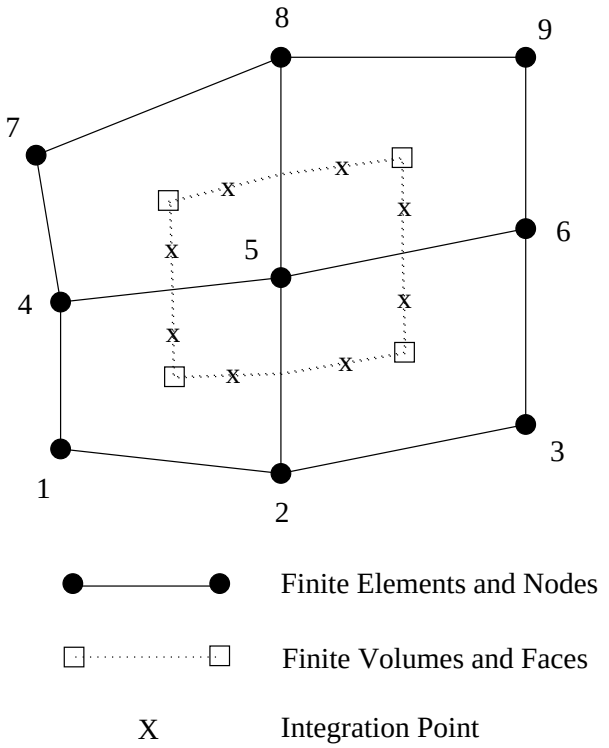

FIGURE 1. A control volume centered about a finite-element node in a collection of 2-D quadrilateral elements.

be considered in the context of an approximate projection method. Assuming that the Laplacian and gradient operators commute, it is necessary to compute $p^{n+1/2} = \mathbf{A}\tau_3\phi^{n+1}$ to obtain the second-order pressure field, while the relationship $p^{n+\frac{1}{2}} = \phi^{n+1}$ will result in a first-order pressure field with splitting error $(\mathbf{I} - \mathbf{A}\tau_3)\mathbf{G}p^{n+\frac{1}{2}}$ (Brown *et al.* (2001)).

Although the above set of algorithms have been written in terms of a two step scheme, i.e., predict $\hat{\mathbf{u}}$ and correct $\hat{\mathbf{u}}$ by the appropriately scaled scalar gradient, non-linear iterations can also be taken. In this case, the ϕ^{n+1} and $\mathbf{u}^{n+1}$ state are replaced with the $k+1$ state, whereas the $n+\frac{1}{2}$ pressure state is replaced by the $k+\frac{1}{2}$ state. For the residual form, the n^{th} state is replaced with the current iterate, k^{th} state. At convergence within the time step, $\phi^{n+1} = \phi^{k+1}$, $\mathbf{u}^{n+1} = \mathbf{u}^{k+1}$, and $p^{n+\frac{1}{2}} = p^{k+\frac{1}{2}}$.

2. CVFEM operators

SIERRA/Fuego uses the finite volume technique known as the control volume finite element method of Schneider and Raw (1987). Control volumes (the mesh dual) are constructed about the nodes, as shown in Fig. 1. Each element contains a set of subfaces that define control-volume surfaces. The subfaces consist of line segments (2-D) or surfaces (3-D). The 2-D segments are connected between the element centroid and the edge centroids. The 3-D surfaces are connected between the element centroid, the element face centroids, and the edge centroids. Integration points also exist within the subcontrol volume centroids. Such integration points are used for volume integrals such as source terms, the mass matrix, and, if chosen, gradients.

Defining ϕ_K to be the value of ϕ at node K, then the variation of ϕ within an element that contains the point location $\mathbf{x}$ is given by

$$\phi(\mathbf{x}) = \sum_{K \in \mathcal{N}} N_K(\mathbf{x}) \phi_K, \tag{2.1}$$

where $N_K(\mathbf{x})$ is the shape function associated with node K at position $\mathbf{x}$, and $\mathcal{N}$ is the set of all nodes that defines the element. For the CVFEM, either trilinear (3-D) or bilinear (2-D) shape functions are used. Currently, Fuego supports heterogeneous element topologies consisting of hex, tet, pyramid, and wedges.

The discrete nodal gradient operator for direction i can be written as a surface integral on control volume L,

$$\mathbf{G}\phi = (G\phi)_{Li} = \int_{\Gamma_L} \phi(\mathbf{x}) dn_i \approx \sum_{\alpha \in \mathcal{B}_L} \left(\sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) \phi_K \right) n_i(\mathbf{x}_\alpha) \Delta A_\alpha, \tag{2.2}$$

where $\mathcal{B}_L$ is the set of surface integration points for control volume L. Similarly, the discrete divergence operator at node L acting on vector u_i is

$$\mathbf{Du} = (Du_i)_L = \int_{\Gamma_L} \rho(\mathbf{x}) u_i(\mathbf{x}) dn_i \approx \sum_{\alpha \in \mathcal{B}_L} \rho(\mathbf{x}_\alpha) \left(\sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) u_{Ki} \right) n_i(\mathbf{x}_\alpha) \Delta A_\alpha, \tag{2.3}$$

and the Laplacian operator that includes spatially varying timescale, τ, is

$$\mathbf{L}_\tau \phi = (L_\tau \phi)_L = \int_{\Gamma_L} \tau(\mathbf{x}) \frac{\partial \phi}{\partial x_j} dn_j \approx \sum_{\alpha \in \mathcal{B}_L} \tau(\mathbf{x}_\alpha) \left(\sum_{K \in \mathcal{N}} \frac{\partial N_K(\mathbf{x}_\alpha)}{\partial x_j} \phi_K \right) n_j(\mathbf{x}_\alpha) \Delta A_\alpha. \tag{2.4}$$

Note that an alternative to the gradient operator given in Eq. (2.2), which is provided via the CVFEM is

$$\mathbf{G}\phi = (G\phi)_{Li} = \int_{\Gamma_L} \frac{\partial \phi}{\partial x_i} dV \approx \sum_{\alpha' \in \mathcal{B}_L} \left(\sum_{K \in \mathcal{N}} \frac{\partial N_K(\mathbf{x}_{\alpha'})}{\partial x_i} \phi_K \right) dV_{\alpha'}, \tag{2.5}$$

where $\mathcal{B}_L$ is now the set of all subcontrol volume integration points for control volume L (for clarity, α' denotes the subcontrol volume integration point location).

The general term $\mathbf{D}\tilde{\tau}\mathbf{G}\phi$ deserves a special note in the case of variable density flows. Specifically, the interpolation is currently provided by the following equation:

$$\mathbf{D}\tilde{\tau}_i \mathbf{G}\phi = \sum_{\alpha \in \mathcal{B}_L} \rho(\mathbf{x}_\alpha) \frac{\tilde{\tau}_i(\mathbf{x}_\alpha)}{\rho(\mathbf{x}_\alpha)} \left(\sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) \frac{G_{Ki}}{V_K} \right) n_i(\mathbf{x}_\alpha) \Delta A_\alpha, \tag{2.6}$$

$$= \sum_{\alpha \in \mathcal{B}_L} \tilde{\tau}_i(\mathbf{x}_\alpha) \left(\sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) \frac{G_{Ki}}{V_K} \right) n_i(\mathbf{x}_\alpha) \Delta A_\alpha. \tag{2.7}$$

3. Smoothing algorithms defined

Now that the smoothing and splitting errors have been formally defined, it is useful to consider three projection algorithms that have been implemented and verified within SIERRA/Fuego in the context of the classic two equation k-ϵ model, with steady method of manufactured solutions (MMS) (cf. Domino (2006)).

3.1. *Fourth-order smoothing with characteristic or time step scaling*

In this algorithm, the projection time scales are defined by either

$$\tau = \tau_1 = \tau_2 = \tau_3 = \tau_{char}, \tag{3.1}$$

or

$$\tau = \tau_1 = \tau_2 = \tau_3 = \mathbf{I}\Delta t. \tag{3.2}$$

Here, characteristic scaling, τ_{char}, is a diagonal matrix that represents a time scale based on convection and diffusion contributions, while for time step scaling, the time scale is based on the local time step. The characteristic scaling very closely follows the standard finite element method stabilization parameter.

The smoothing and splitting errors are now given by

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})p^{n+\frac{1}{2}} \end{bmatrix}. \tag{3.3}$$

Of particular interest to this research is the role of the stabilization term, $(\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})p^{n+\frac{1}{2}}$, on formal time accuracy when $\tau = \mathbf{I}\Delta t$ (a scheme that has been shown to display more appealing stability characteristics). Clearly, a scheme that uses explict pressure stabilization with time step scaling is first-order accurate. Expanding this stabilization term shows the fourth-order pressure derivative scaled by a length scale cubed. Therefore, by refining the time step *and* mesh, one might be able to show a second-order accuracy for sufficiently resolved meshes.

In practice, the stabilization terms are carried within the mass flow rate that forms part of the right-hand side of the Pressure Poisson Equation solve and the convection term for the transport of any scalar field. The mass flow rate is defined as

$$\dot{m}^k = \left(\overline{\rho}\overline{\hat{\mathbf{u}}} + \overline{\frac{\tau \mathbf{G} p^{n-\frac{1}{2}}}{V}} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} \right) d\mathbf{A}, \tag{3.4}$$

where the introduction of the over bar is noted to represent interpolation of a nodal field to an integration point. Note that in the bulk of the collocated unstructured finite volume literature, the form of the mass flow rate defines the stabilization (the difference between the nodal gradient operator $\mathbf{G}$ and the interior element operator ∇^h). Above we note the independent interpolation of the density and velocity rather than $\overline{\rho\hat{\mathbf{u}}}$, as is done in Stanford's ASC Alliance code CDP. It does seem that the full interpolation of $\rho\hat{\mathbf{u}}$ may be more consistent, although the effect of this algorithmic detail has not been explored.

3.2. *Stabilized smoothing*

The stabilized projection algorithm is based on the work of Soto *et al.* (2001), that was derived from the monolithic scheme of Codina (2001). In this algorithm, the projection time scales are defined as

$$\tau_1 = \Delta t\mathbf{I} + \tau_{char}. \tag{3.5}$$

$$\tau_2 = \tau_{char}. \tag{3.6}$$

$$\tau_3 = \tau_{char}. \tag{3.7}$$

With the above definitions, the smoothing and splitting errors are now defined as

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau}_{char})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_{\tau_{\mathbf{char}}} - \mathbf{D}\tilde{\tau}_{char}\mathbf{G})p^{n+\frac{1}{2}} + \Delta t\mathbf{L}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \end{bmatrix}. \tag{3.8}$$

The mass flow rate now includes an additional stabilization factor and is now defined as

$$\dot{m}^k = \left(\overline{\rho}\overline{\hat{\mathbf{u}}} + \overline{\frac{\tau \mathbf{G} p^{n-\frac{1}{2}}}{V}} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} - \Delta t \mathbf{L} \Delta p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{3.9}$$

Note that at full convergence, the stabilized scheme reduces to the fourth-order characteristic scaling algorithm.

3.3. *Second-order smoothing with characteristic or time step scaling*

In fact, the scaled nodal gradient need not be included in the mass flow rate equation, e.g.,

$$\dot{m}^k = \left(\overline{\rho}\overline{\hat{\mathbf{u}}} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{3.10}$$

This is equivalent to neglecting the $\tilde{\tau}_2 \mathbf{G} p^{n-\frac{1}{2}}$ term in Eq. (1.4), or by defining $\tilde{\mathbf{u}} = \hat{\mathbf{u}}$.

The smoothing for this algorithm is provided by the local Lapacian operator. The smoothing and splitting errors for this method are now given by

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})\Delta p^{n+\frac{1}{2}} + \mathbf{L}_\tau p^{n-\frac{1}{2}} \end{bmatrix}. \tag{3.11}$$

3.4. *Zeroth-order smoothing with time step or characteristic scaling*

Certainly, the pressure smoothing can be removed, i.e., $\tau_2 = 0$, that leads to the following set of errors,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \end{bmatrix}. \tag{3.12}$$

where τ is either the characteristic scale, τ_{char}, or the simulation time step, $\mathbf{I}\Delta t$ (with $\tau_1 = \tau_3$). Although the converged error is zero, this lack of smoothing can lead to a decoupled pressure field in certain flows.

Here, the mass flow rate reduces to a simple interpolation of nodal velocities within the element

$$\dot{m}^k = \left(\overline{\rho}\overline{\hat{\mathbf{u}}} - \overline{\tau}\nabla^h \Delta p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{3.13}$$

The unsmoothed algorithm is very similar to the staggered formulation of SIMPLE, (cf. Patankar (1988)), with $\tau = A_p^{-1}$ (the inverse of the diagonal matrix from operator $\mathbf{A}$). However, by design, the staggered mesh arrangement holds the property that $(\mathbf{L}_\tau - \mathbf{D}\tau\mathbf{G}) = 0$. In this method, no stabilization is added as none is required.

4. Time integration scheme

The Crank-Nicholson method discribed in Ober and Shadid (2004) will be used to obtain a second-order integration scheme (in the context of our zeroth-order smoothing algorithm). In this implementation, the generalized method is written as

$$\frac{\partial \phi}{\partial t}^{n+1} = \eta \frac{\phi^{n+1} - \phi^n}{\Delta t} + (1 - \eta)\frac{\partial \phi}{\partial t}^n, \tag{4.1}$$

where η is a blending coefficient between 1 and 2. Values of η of unity result in first-order backward Euler, while values of 2 result in second order Crank-Nicholson, i.e.,

$$\frac{\partial \phi}{\partial t}^{n+1} = 2\frac{(\phi^{n+1} - \phi^n)}{\Delta t} - \frac{\partial \phi}{\partial t}^{n}. \tag{4.2}$$

A linearization is given by

$$\frac{\partial \phi}{\partial t}^{n+1} = 2\frac{(\phi^{k} - \phi^n)}{\Delta t} - \frac{\partial \phi}{\partial t}^{n}, \tag{4.3}$$

where the old time derivative is computed based on the old solution of the partial differential equation of interest.

4.1. *Variable density*

In the case of variable density, the full time term is

$$\frac{\partial \rho\phi}{\partial t}^{n+1} = \eta\frac{\rho^{n+1}\phi^{n+1} - \rho^n\phi^n}{\Delta t} + (1-\eta)\frac{\partial \rho\phi}{\partial t}^{n}, \tag{4.4}$$

where it is noted that the full time derivative at n^{th} state is saved. The linearization is given by

$$\frac{\partial \rho\phi}{\partial t}^{n+1} = \eta\frac{\rho^{k}\phi^{k} - \rho^n\phi^n}{\Delta t} + (1-\eta)\frac{\partial \rho\phi}{\partial t}^{n}. \tag{4.5}$$

The above algorithm is especially useful in that it avoids the need to evaluate complex right-hand side source terms at the n+1 and n state, e.g., simulations that include the need to compute turbulence production, reaction rate terms, etc.

5. Discrete system of equations

The full approximate pressure projection scheme for non-uniform density is now written as

$$\eta M_L^k \Delta\hat{u}_i + C_L(\dot{m}^k)\Delta\hat{u}_i - T_{Lj}\Delta\hat{u}_i = -r_i, \tag{5.1}$$

$$-L_{\tau_1 L}\Delta p^{n+\frac{1}{2}} = -D_L(\hat{u}_i) - L_{\tau_1}p^k + (L_2 - D\tilde{\tau}_2 G)_L p^k + b, \tag{5.2}$$

$$u_{Li}^{n+1} = \hat{u}_{Li} - \tilde{\tau}G_{Li}\Delta p^{n+\frac{1}{2}}. \tag{5.3}$$

The variable $-r_i$ is the residual that includes body source terms, pressure gradient, the non-symmetric part of the viscous stress term, $T_{Li}^{ns}u_j^k$, parts of the time term and the left-hand side set of operators acting on the u_i^k state,

$$-ri = -\eta M_L^k u_i^k - C_L(\dot{m}^k)u_i^k + T_{Lj}\Delta u_i^k + T_{Li}^{ns}u_j^k + S_{Li} - (1-\eta)M_L(\rho\dot{u}_i^n) - G_{Li}p^{n-\frac{1}{2}}. \tag{5.4}$$

The mass matrix, $M_L^k\Delta\hat{u}_i$, is defined by

$$M_L^k\Delta\hat{u}_i = \sum_{\alpha'\in\mathcal{B}_L}\left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})\frac{\rho_K^k}{\Delta t}\right)\left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})\Delta\hat{u}_{Ki}\right)dV_{\alpha'}. \tag{5.5}$$

The shape function above, $N_K(\mathbf{x}_{\alpha'})$, is frequently evaluated at $\mathbf{x}_{\mathcal{N}}$, the coordinates of the vertex associated with the transport equation, i.e., the case where a lumped mass matrix is used.

For simplicity, the central difference operator is provided in $C_{Li}\Delta\hat{u}_i$ as

$$C_L\Delta\hat{u}_i = \sum_{\alpha\in\mathcal{B}_L} m_\alpha^k \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\Delta\hat{u}_{Ki} \right). \tag{5.6}$$

In the preceding equation, the mass flow rate has been linearized within the iteration step and may or may not include the explicit stabilization terms. Moreover, the shape function operator, $N_K(\mathbf{x}_\alpha)$, may be evaluated at the edge midpoints to retain the skew symmetric aspect of the operator C_L. By default, this term is evaluated at the subcontrol surface integration points, which retains the CVFEM canonical 27-point stencil.

The symmetric part of the stress tensor is given by

$$T_{Lj}\hat{u}_i = \sum_{\alpha\in\mathcal{B}_L} \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left(\sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_j}\hat{u}_{Ki} \right) n_j(\mathbf{x}_\alpha)\Delta A_\alpha, \tag{5.7}$$

while the non-symmetric stress tensor is given by

$$\begin{aligned} T_{Li}^{ns}u_j^k = &\sum_{\alpha\in\mathcal{B}_L} \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left(\sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_i}u_{Kj}^k \right) n_j(\mathbf{x}_\alpha)\Delta A_\alpha \\ &- \frac{2}{3}\sum_{\alpha\in\mathcal{B}_L} \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left(\sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_p}u_{Kp}^k \right) \delta_{ip}n_p(\mathbf{x}_\alpha)\Delta A_\alpha. \end{aligned} \tag{5.8}$$

Note that the nodal pressure gradient at node L for control volume L for direction i is defined by Eq. (2.2). The operator, S_{Li}, contains the gravitational term as well as the [potentially] subtracted out hydrostatic term,

$$S_{Li} = \sum_{\alpha'\in\mathcal{B}_L} \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})(\rho_K^k - \rho^{ref}) \right) g_i dV_{\alpha'}. \tag{5.9}$$

The old time term contribution, $M_L(\rho\dot{u}_i{}^n)$, is defined by

$$M_L(\rho\dot{u}_i{}^n) = \sum_{\alpha'\in\mathcal{B}_L} \left(\sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})\rho_K\dot{u}_{Ki}{}^n \right) dV_{\alpha'}. \tag{5.10}$$

Again, $\alpha'\in\mathcal{B}_L$ is the set of all subcontrol volume integration points for control volume L, $\alpha'\in\mathcal{B}_L$ is the set of all subcontrol surface integration points for control volume L, and $K\in\mathcal{N}$ is the set of all nodes within the element.

5.1. *Predictor*

In general, there are a number of predictors that are supported. The easiest predictor is a simple predictor in which the old value is mapped into the current iterate. Predictors that incorporate old time derivatives include the forward Euler and Adams-Bashforth methods, e.g.,

$$\phi^{k+1} = \phi^n, \tag{5.11}$$

$$= \phi^n + \Delta t\dot{\phi}^n, \tag{5.12}$$

$$= \phi^n + \frac{\Delta t^n}{2}\left((2 + \frac{\Delta t^n}{\Delta t^{n-1}})\dot{\phi}^n - \frac{\Delta t^n}{\Delta t^{n-1}}\dot{\phi}^{n-1}\right). \tag{5.13}$$

6. Analysis

Based on the above analysis of smoothing and splitting errors, it is clear that the formal time accuracy of a stabilized scheme can be limited by the leading error term of the scaling of $\mathbf{L} - \mathbf{DG}$ when time step scaling is used. Expanding the above operator in a one-dimensional case has been shown to be equivalent to injecting a fourth-order pressure dissipation error term that is scaled by Δx^3. Commonly, verification problems chosen that have a linear pressure gradient, e.g., specified pressure drop in a duct, do not show this error term.

To circumvent the order Δt leading stabilization error, either alternative scaling can be used (as previously described) or the explicit stabilization can be removed. However, numerical experiments and stability analysis on alternative scaling parameters have demonstrated these schemes as being less than optimal, i.e., near-unity CFL stability limits, slow convergence within a time step, etc. Removing explicit stabilization has also been demonstrated to slow convergence (sometimes even stagnation) with wildly varying pressure plots. Arguably, the role of pressure within an accoustically incompressible scenario is incosequential given that a spatially constant, yet potentially varying in time thermodynamic pressure is used within the equation of state. However, the convergence/stability problems noted must be addressed, and represent a current research topic.

6.1. *Manufactured solution*

The manufactured solution that has been used in this study is given by

$$u = -U_o cos(\pi x) sin(\pi y) sin(\omega t), \tag{6.1}$$

$$v = +U_o sin(\pi x) cos(\pi y) sin(\omega t), \tag{6.2}$$

$$p = -P_o \left(\frac{cos(2\pi x) + cos(2\pi y)}{4} \right) sin^2(\omega t). \tag{6.3}$$

The set of source terms to be placed within the fluids solver is obtained by allowing the analytical differential operators to act on the manufactured solutions. The set of source terms are as follows:

$$S_u^{mms} = -cos(\pi x) sin(\pi y) U_o \left(2\mu\pi^2 sin(\omega t) + \rho\omega cos(\omega t) \right), \tag{6.4}$$

$$S_v^{mms} = +sin(\pi x) cos(\pi y) U_o \left(2\mu\pi^2 sin(\omega t) + \rho\omega cos(\omega t) \right), \tag{6.5}$$

$$S_p^{mms} = 0, \tag{6.6}$$

with the set of constraints; $V_o = U_o$, and $P_o = U_o^2 \rho$.

Time accuracy plots for a square uniform, orthogonal mesh (200x200) over the domain $x = -0.5 : 0.5$ and $y = -0.5 : 0.5$ (Re = 1000) using the backward Euler and Crank-Nicholson integration algorithm ($\eta = 1$ and $\eta = 2$, respectively) are given in Fig. 2. Convergence plots were determined by running each simulation at a fixed mesh out to 10 seconds while refining the time step. As can be seen, both integration algorithms recover to a near first-order behavior.

Simulations that used some aspect of the characteristic scaling were attempted, however, convergence within the time step was extremely problematic. Regardless of the time step taken, the non-linear system was not able to be converged to an acceptable level as with the time step scaling algorithm. It is not clear why this sensitivity exists; analysis is underway to further understand this algorithmic nuance.

Due to time constraints, simulations using the zeroth-order and second-order smoothing scheme were not completed. Based on the numerical analysis provided, running a

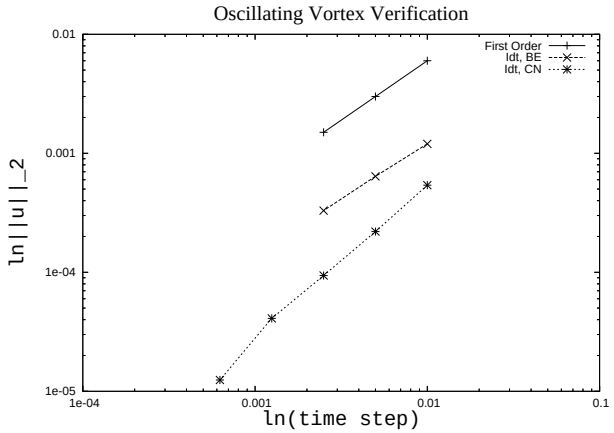

FIGURE 2. Time accuracy plots for the backward Euler and Crank-Nicholson method.

set of simulations that refined both mesh spacing and time step should provide a second order slope.

7. Approximate pressure poisson algorithm

A new class of algorithms based on a classic pressure Poisson algorithm is now presented. A classic pressure Poisson algorithm considers the equations of motion as a natural projection algorithm. In general, any vector can be written as a Hodge decomposition, or in terms of a vector of known divergence and a curl-free part,

$$\mathbf{F} = \mathbf{F^{kd}} + \nabla\phi, \tag{7.1}$$

with the known divergence given by

$$\nabla \cdot \mathbf{F^{kd}} = \mathcal{S}. \tag{7.2}$$

The Poisson system is provided by

$$\nabla \cdot \nabla\phi = \nabla \cdot \mathbf{F} - \nabla \cdot \mathbf{F^{kd}} = \nabla \cdot \mathbf{F} - \mathcal{S} \tag{7.3}$$

with solution,

$$\phi = \Delta^{-1}(\nabla \cdot \mathbf{F} - \mathcal{S}), \tag{7.4}$$

and

$$\begin{aligned}
\mathbf{F^{kd}} &= \mathbf{F} - \nabla\phi, && (7.5)\\
&= \mathbf{F} - \nabla(\Delta^{-1}(\nabla \cdot \mathbf{F} - \mathcal{S})), && (7.6)\\
&= (\mathbf{I} - \nabla(\Delta^{-1}\nabla\cdot))\mathbf{F} + \nabla\Delta^{-1}\mathcal{S}, && (7.7)\\
&= \mathcal{P}\mathbf{F} + \mathcal{B}, && (7.8)\\
&= \mathcal{P}^{af}\mathbf{F}. && (7.9)
\end{aligned}$$

For a solenoidal flow, the known divergence is zero, with $\mathcal{B} = 0$, and we define $\mathcal{P} = \mathcal{P}^{af}$. Moreover, the operator $\mathcal{P}$ is idempotent, i.e., $\mathcal{P} = \mathcal{P}^2$, which formally defines $\mathcal{P}$ as a *projection operator*. For a non-solenoidal flow, the known divergence is $\mathcal{S}$, and we now define the general operator to be an affine projection operator, i.e., $\mathcal{P}^{af}(\mathcal{P}^{af}\mathbf{F}) =$

$\mathcal{P}(\mathcal{P}\mathbf{F}+\mathcal{B})+\mathcal{B}=\mathcal{P}\mathbf{F}+\mathcal{B}$. Whether appropriate discrete operators can be defined in practice remains a challenge.

The projection analysis for the equations of motion is completed by the following definitions:

$$\mathbf{F}=-\nabla\cdot\rho\mathbf{u}\mathbf{u}+\nabla\cdot\mu(\nabla\mathbf{u}+\nabla^T\mathbf{u})+\rho\mathbf{g}+\mathbf{s}, \tag{7.10}$$

$$\mathbf{F}^{\mathbf{kd}}=\frac{\partial\rho\mathbf{u}}{\partial t}, \tag{7.11}$$

$$\nabla\cdot\mathbf{F}^{\mathbf{kd}}=-\frac{\partial^2\rho}{\partial t^2}, \tag{7.12}$$

$$\nabla\phi=\nabla p, \tag{7.13}$$

or

$$\nabla\cdot\nabla p=\nabla\cdot\left(-\nabla\cdot\rho\mathbf{u}\mathbf{u}+\nabla\cdot\mu\left(\nabla\mathbf{u}+\nabla^T\mathbf{u}\right)+\rho\mathbf{g}+\mathbf{s}\right)+\frac{\partial^2\rho}{\partial t^2}. \tag{7.14}$$

The above algorithm has been outlined many times before, (see Ferziger and Perić (1996), Guermond and Shen (2003), Codina and Badia (2005)), however, it is usually dismissed due to the complexity associated with evaluation of the stress tensor as it contains second derivatives. Ferziger also notes the need for consistency between the gradient operator that appears in the momentum equations and the Laplacian that appears in the pressure Poisson solve, although he does not cast this requirement in the context of an approximate projection.

We seek to implement a version of this algorithm in the context of an incremental pressure approximate projection. Specifically, we can consider this algorithm an extension of the unsmoothed stabilized method previously discussed. Such an algorithm is similar to the finite element work of Stevens *et al.* (2002). The equations solved are the following:

$$\mathbf{A}\hat{\mathbf{u}}=f-\mathbf{G}p^{n-\frac{1}{2}}-\mathbf{A}\mathbf{u}^n, \tag{7.15}$$

$$-\mathbf{L_1}\Delta\phi^{n+1}=-\mathbf{D}\hat{\mathbf{u}}+b, \tag{7.16}$$

$$\mathbf{u}^{n+1}=\hat{\mathbf{u}}-\tilde{\tau}_3\mathbf{G}\Delta\phi^{n+1}, \tag{7.17}$$

$$\mathbf{L}p^{n+\frac{1}{2}}=\hat{\mathbf{D}}\mathbf{F}+c, \tag{7.18}$$

where $\hat{\mathbf{D}}$ does not include a density factor, c represents the volume integral of the second derivative of density with respect to time, and $\mathbf{F}$ represents the discretized convection, diffusion, and body source terms. The scaling in this algorithm was chosen to be $\mathbf{I}\Delta t$ with stabilization errors provided by the unsmoothed algorithm. The initial guess for the pressure Poisson equation is given by $p^{n+\frac{1}{2}}|_o=p^{n-\frac{1}{2}}+\phi^{n+1}$.

In the algorithm tested, pressure gradients are obtained via Eq. (2.5). Moreover, second derivatives that are within $\mathbf{F}$, i.e., the viscous stress tensor, are constructed via a two-pass algorithm. First, velocity derivatives, which are stored at element vertices, are constructed via Eq. (2.5)). In words, we compute velocity gradients at the subcontrol-volume centers, weigh the derivatives by the subcontrol volumes, and assemble each of them to the element vertices. Normalization by the control volume provides the stored nodal gradients. Next, the computed velocity derivatives and the given velocity field are used to construct $\mathbf{F}$, again by use of Eq. (2.5)), in another assembly algorithm. This includes convection, the stress tensor and any appropriate body force terms.

Alternatively, one could imagine that the pressure gradient term is provided by the classic Gauss-Green surface integration, i.e., Eq. (2.2)) and $\mathbf{F}$ constructed based on nor-

malization of part of the momentum residual (convection, diffusion, and body forces) by the control volume. Due to time constraints, this method was not tested.

As a proof of concept, the above algorithm was implemented within Fuego and used to run a standard driven cavity problem (Ghia et al. (1983)). Preliminary results indicated that the pressure Poisson solve was approximately two times more expensive to solve than Eq. (7.16)) thus making this method prohibitively expensive. However, results compared very well to the fourth-order stabilized method. Comparing results between the unsmoothed method and the newly proposed scheme showed good agreement as well. However, the pressure field in this newly implemented algorithm was very smooth compared to the pressure obtained through the unsmoothed algorithm. Therefore, it appeared that the Poisson solve did in fact provide some sort of smoothing operator to the pressure.

8. Conclusion

A suite of approximate projection methods was analyzed using a block factorization approach to show the leading errors due to both splitting and stabilization. Through this analysis, it was shown that a suite of approximate projection methods suffer from a first-order time error regardless of the time integration scheme used. Using the method of manufactured solutions, the backward Euler and Crank-Nicholson time integration scheme were both verified to be first-order in time when using the standard time step scaling approximate projection method. Verification studies that attempted to use the characteristic scaling approach did not display satisfactory convergence within a time step, although this scheme has been shown to work successfully in a suite of steady verification problems.

Lastly, a new class of approximate projection methods based on the classic pressure Poisson system was implemented and tested with Fuego. Although the algorithm was stable, the second Poisson system proved the scheme to be prohibitively expensive.

Acknowledgments

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL850000.

The author would like to thank the CTR host, Dr. Frank Ham for his time over the summer session. The many useful conversations regarding stabilized finite volume methods were very much appreciated. The author would also like to thank Fuego team members Chris Moen and Greg Wagner for each of their displayed commitment towards understanding stabilized finite volume algorithms.

REFERENCES

Almgren, A.S., Bell, J.B. & Crutchfield, W.Y. 2000 Approximate projection methods: part I. inviscid analysis. *SIAM J. Sci. Comp.*, **22**, 1139–1159.

Brown, D.L., Cortez, R., & Minion, M. 2001 Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, **168**, 464–499.

Chorin, A.J. 1968 Numerical solutions of the Navier-Stokes equations. *Math. Compt.*, **22**, 745–762.

Codina, R. 2001 Pressure stability in fractional step finite element methods for incompressible flows. *J. Comp. Phys.*, **170**, 112–140.

Codina, R. & Badia, S. 2005 On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Comp. Methods. Appl. Mech. Engr.*, in press.

Domino, S.P. 2006 Fuego Verification Manual. *Sandia National Laboratory Internal Draft.*

Dukowicz, J.K. & Dvinsky, A.S. 1992 Approximate factorization as a high order splitting for the implicit incompressible flow equations. *J. Comp. Phys.*, **102**, 336–347.

Ferziger, J.H. & Perić, M. 1996 *Computational Methods for Fluid Dynamics*, Springer-Verlag.

Ghia, U., Ghia, K.N. & Shinn, C.T. 1983 High-RE solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comp. Phys.*, **48**, 387–411.

Guermond, J.L. & Shen, J. 2003 A new class of truely consistent splitting schemes for incompressible flows. *J. Comp. Phys.*, **192**, 262–276.

Kim, J. and Moin, P. 1985 Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, **59**, 308-323.

Kim, D. & Choi, H. 2000 A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. *J. Comp. Phys.*, **162**, 411–428.

Majumdar, S. Role of under-relaxation in momentum interpolation for calculation of flow with non-staggered grids. *Num. Heat Transfer*, **13**, 125–132.

Moen, C.D., Evans, G.H., Domino, S.P. & Burns, S.P. 2002 A multi-mechanics approach to computational heat transfer. *Paper IMECE2002-33098*, ASME International Mechanical Engineering Congress and Exposition, New Orleans, LA.

Ober, C.C. & Shadid, J.N. 2004 Studies on the accuracy of time-integration methods for the radiation-diffusion equations. *J. Comp. Phys.*, **195**, 743-772.

Papageorgakopoulos, J., Arampatzis, G., Assimacopoulos, D. & Markatos, N.C. 2000 Enhancement of the momentum interpolation method on non-structured grids. *Int. J. Numer. Meth. Fluids*, **33**, 1–22.

Patankar, S.V. 1988 Recent developments in computational heat transfer. *J. Heat Transfer*, **110**, 1037–1045.

Perot, J.B. 1993 An analysis of the fractional step method. *J. Comp. Phys.*, **108**, 51–58.

Perić, M., Kessler, R. & Scheuerer, G. 1988 Comparison of finite-volume numerical methods with staggered and colocated grids. *Comp. Fluids*, **16**, 389–403.

Rhie, C.M. & Chow, W.L. 1983 Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.*, **21**, 1525–1532.

Schneider, G.E. 1988 Elliptic systems: finite-element method I. In *Handbook of Numerical Heat Transfer*, John Wiley & Sons.

Soto, O.R., Löhner, F. & Cebral, J. An implicit monolithic time accurate finite element scheme for incompressible flow problems. *AIAA-2001-2616.*

Strikwerda, J.C. & Lee, Y.S. 1999 The accuracy of the fractional step method. *SIAM J. Numer. Anal.*, **37**, 37-48.

Schneider, G.E. & Raw, M.J. 1987 Control volume finite element method for heat

transfer and fluid flow using colocated variables: Part I. computational crocedure. *Num. Heat Trans.*, **11**, 363–390.

Stevens, D.E., Chan, S.T. & Gresho, P. 2002 An approximate projection method for incompressible flow. *Int. J. Numer. Meth. Fluids.*, **40**, 1303–1325.