Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6404--05-8858

A Robust Solver for Incompressible Flow on Cartesian Grids with Colocated Variables

DAVID R. MOTT
ELAINE S. ORAN

Reactive Flow Physics
Laboratory for Computational Physics and Fluid Dynamics

CAROLYN R. KAPLAN

Center for Reactive Flow and Dynamical Systems
Laboratory for Computational Physics and Fluid Dynamics

July 13, 2005

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
13-07-2005	Memorandum	

4. TITLE AND SUBTITLE

A Robust Solver for Incompressible Flow on Cartesian Grids with Colocated Variables

5a. CONTRACT NUMBER
64-6027-05

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

David R. Mott, Carolyn R. Kaplan,* and Elaine S. Oran

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory, Code 6404
4555 Overlook Avenue, SW
Washington, DC 20375-5344

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6404--05-8858

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*Center for Reactive Flow and Dynamical Systems, Code 6410

14. ABSTRACT

We describe a second-order finite-volume formulation for solving incompressible flow problems on Cartesian grids using a colocated arrangement of variables. A projection method is used, in which a provisional cell-centered velocity is obtained by integrating the effects of advection and viscous forces over the numerical time step. Interface velocities are then interpolated from the provisional cell-centered velocities and corrected using the solution of a Poission equation for the pressure distribution. The cell-centered velocities are then interpolated from the corrected interface velocities. This pair of interpolation steps adds numerical diffusion that mimics a normal viscous stress, and the form of this damping term is similar to that which comes from stabilizing the Poisson equation for pressure by adding a term that is proportional to the fourth-derivative of pressure. The method requires no extra damping terms in order to maintain stability.

15. SUBJECT TERMS

Incompressible flow; Projection method; CFD; Backward-facing step

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UL	34	David R. Mott
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER *(include area code)* (202) 767-1974

CONTENTS

A Robust Solver for Incompressible Flow on Cartesian Grids with Colocated Variables

David R. Mott, * Carolyn R. Kaplan, and Elaine S. Oran

Laboratory for Computational Physics and Fluid Dynamics, US Naval Research Laboratory, 4555 Overlook Ave SW, Washington, DC 20375

Abstract

We describe a second-order finite-volume formulation for solving incompressible flow problems on Cartesian grids using a colocated arrangement of variables. A projection method is used, in which a provisional cell-centered velocity is obtained by integrating the effects of advection and viscous forces over the numerical time step. Interface velocities are then interpolated from the provisional cell-centered velocities and corrected using the solution of a Poission equation for the pressure distribution. The cell-center velocities are then interpolated from the corrected interface velocities. This pair of interpolation steps adds numerical diffusion that mimics a normal viscous stress, and the form of this damping term is similar to that which comes from stabilizing the Poisson equation for pressure by adding a term that is proportional to the fourth-derivative of pressure. The method requires no extra damping terms in order to maintain stability.

Key words: incompressible flow, projection method, CFD, backward-facing step

Manuscript approved December 10, 2004.

1 Introduction

Many engineering applications involve incompressible flows, from pipe-flow problems to microfluidics applications. A wide range of finite-element, finite-difference, and finite-volume methods for solving these problems numerically have been developed over the years. [1–9]. Many of the more popular finite-volume schemes are clasified as pressure-correction methods. In these methods, a provisional velocity field for the new time level or iteration in a steady state calculation is calculated that includes the effects of viscous and advective transport but not the full effect of the pressure gradient. A Poisson equation for the new pressure distribution (or for a correction to the pressure distribution) is then solved, and this solution enforces mass conservation once the velocity distibution is updated using the new pressures. The well-known SIMPLE[3], SIMPLEC[4], QUICK[5], and PISO[6] methods fall into this category. Fractional step methods can be grouped with these methods, but distinctions between various fractional step methods are often made depending on if and how the old pressure field is used to determine the provisional velocity field. The unifying theme in these projection methods is the solution of a Poisson equation for the pressure or a pressure-like variable that provides a correction to the velocity distribution that enforces conservation of mass.

The application of SIMPLE-type methods was first successful on staggered

* Corresponding Author; voice: (202) 767-1974; fax: (202) 767-4798
Email address: mott@lcp.nrl.navy.mil (David R. Mott,).

meshes in which the control volumes for the pressure and various velocity components were offset. This placed velocity components on the faces of the pressure control volume such that central-differencing of the pressure Poisson equation led to compact, consistent stencils. Staggering the meshes, however, leads to added complexity in the grid, particularly when modeling flows around complex geometries.

If pressure and velocity variables are colocated, one set of control volumes can be used for all variables. Now, however, a different problem emerges. The pressure Poission equation used in projection methods is derived by taking the divergence of the momentum equation. Therefore the Laplacian of pressure, $\nabla^2 p$, in the pressure equation is the divergence of the pressure gradient term from the momentum equation, $\nabla \cdot (\nabla p)$. Once the discretizations for the divergence and gradient are chosen, only one discretization for the Laplacian guarantees that $\nabla^2 p = \nabla \cdot (\nabla p)$. Choosing this discretization ensures that the resulting pressure equation accurately imposes $\nabla \cdot \mathbf{v} = \mathbf{0}$. If standard central-difference stencils are used to discretize the divergence and gradient operators, the resulting stencil for $\nabla \cdot \nabla = \nabla^2$ produces odd-even uncoupling. That is, $\nabla^2 \phi$ at cell (i, j) uses the values of ϕ two cells away from (i, j) (such as $(i+2, j)$ and $(i, j-2)$) but not the values of immediate neighbors (such as $(i+1, j)$ or $(i, j+1)$). Ignoring the requirement that the discrete operators satisfy $\nabla^2 \phi = \nabla \cdot (\nabla \phi)$ and using compact stencils for all three operators can produce conservation errors and instabilities. Standard ways to overcome this instability include adding a damping term proportional to $\nabla^4 p$ to the Pois-

son equation for the pressure or by correcting the interface velocities used to calculate $\nabla \cdot \mathbf{v}$ [1, 2, 9].

This paper describes a compact finite-volume method for solving incompressible flow problems on a colocated grid. Rather than use the pressure distribution to correct the cell-centered values directly, the normal velocity component at each cell interface (interpolated from the provisional cell-centered values) is corrected, and the cell-centered values are interpolated from the interface values. This process introduces numerical diffusion that has the same form as a normal viscous stress and is similar to the popular $\nabla^4 p$ damping approach for stabilizing the pressure equation. Therefore, no extra damping terms are included in the Poisson equation for the pressure in order to ensure stability.

2 Governing Equations and Basic Discretization

We begin with the incompressible Navier Stokes equations,

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) + \nabla p - \mu \nabla^2 \mathbf{v} = 0, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

in which ρ is the fluid density and μ is the viscosity. The constraint $\nabla \cdot \mathbf{v} = 0$ on the velocity $\mathbf{v}$ only requires that the substantial derivative of ρ be zero:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho = 0. \tag{3}$$

We make the additional assumption that μ and ρ are homogeneous throughout the computational domain. The component of the convective term $\nabla \cdot (\rho \mathbf{v}\mathbf{v})$ appearing in the momentum equation for direction i is $\nabla \cdot (\rho u_i \mathbf{v})$.

Our focus is the spatial discretization of the flow variables and differential operators, so we will consider steady-state calculations obtained by marching a simple time-accurate scheme to equilibrium. This basic approach allows for two paths of development in the future: improved steady-state convergence rates by implementing a local-time-stepping algorithm, and improved time resolution for unsteady calculations by increasing the order of the time discretization. A forward Euler time discretization between times t_n and t_{n+1} separated by time step Δt yields

$$\rho \mathbf{v}^{n+1} - \rho \mathbf{v}^n = \Delta t \Big(- \nabla \cdot (\rho \mathbf{v}\mathbf{v})^n - \nabla p^{n+1} + \mu \nabla^2 \mathbf{v}^n \Big) \tag{4}$$

In Eq. (4), we have treated the pressure term implicitly and the advective and viscous terms explicitly. A provisional velocity $\tilde{\mathbf{v}}$ can be calculated using

$$\rho \tilde{\mathbf{v}} = \rho \mathbf{v}^n + \Delta t \Big(- \nabla \cdot (\rho \mathbf{v}\mathbf{v})^n + \mu \nabla^2 \mathbf{v}^n \Big) \tag{5}$$

Now replace the time-level n terms in Eq. (4) using $\tilde{\mathbf{v}}$ from Eq. (5) and take the divergence of the result. Requiring $\nabla \cdot \mathbf{v} = 0$ then gives the differential form of the pressure equation:

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot (\rho \tilde{\mathbf{v}}). \tag{6}$$

Finally, $\mathbf{v}^{n+1}$ is calculated using

$$(\rho\mathbf{v})^{n+1} = \rho\tilde{\mathbf{v}} - \Delta t \nabla p^{n+1}. \tag{7}$$

From the derivation of Eq. (6), the Laplacian term arises from composition of the divergence operator with the gradient operator. The methods that compensate for the odd-even instability either add a term proportional to $\nabla^4 p$ into Eq. (6), or correct the interpolated interface velocity using centered and interpolated pressure gradients [1]. They therefore solve an altered form of Eq. (6) to find the pressure distribution and then use Eq. (7) to update the velocity values.

3 Current Method

The algortihm is presented for two-dimensional flows, but the extention to three-dimensions is straighforward. Three-dimentional simulations of microchannel flows using this algorithm have been presented elsewhere [12, 13]. The spatial discretization is done on a Cartesian grid, as illustrated in Figure 1. Integer subscripts indicate cell centered locations, and fractional subscripts indicate interface locations. For example, $p_{i,j}$ is the pressure at point (i, j), and $u_{i+1/2}$ is the x-velocity at the interface between cells (i, j) and $(i+1, j)$. In the current work, the variables are assumed to vary linearly within each cell, so the point value at the cell center and the cell-average value are identical. The cell acts as the control volume for all variables. The viscous term in Eq. (5) is discretized

using a flux formulation equivalent to second-order centered finite-differencing on cell-centered values. The effect of the advection term in Eq. (5) is calculated using flux-corrected transport (FCT) as implemented in the LCPFCT library [10, 11]. LCPFCT solves generalized conservation laws, but Eq. (6) specifies the pressure field using an algebraic constraint rather than the integration of a time rate-of-change. LCPFCT is used to calculate the contribution of the advection term, $\nabla \cdot (\rho \mathbf{v}\mathbf{v})^n$, to $\tilde{\mathbf{v}}$ so that Eq. (6) can be solved for the updated pressure field.

Rather than write Eq. (6) using the cell-centered provisional velocities, we calculate provisional interface normal velocities using a linear interpolation. For unifiorm Δx and Δy, this gives

$$\begin{aligned} \rho\tilde{u}_{i+1/2} &= \frac{1}{2}\Big(\rho\tilde{u}_{i,j} + \rho\tilde{u}_{i+1,j}\Big) \\ \rho\tilde{v}_{j+1/2} &= \frac{1}{2}\Big(\rho\tilde{v}_{i,j} + \rho\tilde{u}_{i,j+1}\Big). \end{aligned} \tag{8}$$

Then we use these interpolated normal velocities to calculate $\nabla \cdot (\rho\tilde{\mathbf{v}})$:

$$\begin{aligned} \nabla \cdot (\rho\tilde{\mathbf{v}})_{i,j} &= \frac{-\sum \text{Flux} \times \text{area}}{\text{cell volume}} \\ &= \frac{1}{\Delta x \Delta y}\Big(\Delta y(\rho\tilde{u}_{i+1/2} - \rho\tilde{u}_{i-1/2}) + \Delta x(\rho\tilde{v}_{j+1/2} - \rho\tilde{v}_{j-1/2})\Big), \end{aligned} \tag{9}$$

in which the sum is taken over the interfaces of the cell.

We then solve Eq. (6) using a flux-based approach that is equvalent on a regularly-spaced grid to using a compact central-difference stencil for the Laplacian term, and we do not add a stabilizing term to the equation. Rather than update the cell-centered values directly using the new pressure distribu-

tion, we correct the interface normal velocities using

$$(\rho u)_{i+1/2}^{n+1} = (\rho \tilde{u})_{i+1/2} - \Delta t \frac{p_{i+1,j} - p_{i,j}}{\Delta x},$$
$$(\rho v)_{j+1/2}^{n+1} = (\rho \tilde{v})_{j+1/2} - \Delta t \frac{p_{i,j+1} - p_{i,j}}{\Delta y}. \tag{10}$$

The cell centered values are then linearly interpolated from the interface values. For constant Δx and Δy, this reduces to

$$\rho u_{i,j}^{n+1} = \frac{1}{2}\left(\rho u_{i+1/2}^{n+1} + \rho u_{i-1/2}^{n+1}\right),$$
$$\rho v_{i,j}^{n+1} = \frac{1}{2}\left(\rho v_{j+1/2}^{n+1} + \rho v_{j-1/2}^{n+1}\right). \tag{11}$$

Therefore, the solution procedure is

(1) Calculate cell-centered $\rho\tilde{\mathbf{v}}$ using Eq. (5).

(2) Calculate $\tilde{u}_{i+1/2}$, $\tilde{v}_{j+1/2}$ by interpolating the cell-centered values.

(3) Use $\tilde{u}_{i+1/2}$, $\tilde{v}_{j+1/2}$ to calculate $\nabla \cdot (\rho\tilde{\mathbf{v}})$ as in Eq. (9).

(4) Solve Eq. (6) for the cell-centered pressure values, using a compact stencil for ∇^2.

(5) Update the interface normal velocity values using Eq. (10).

(6) Update cell-centered velocities using Eq. (11).

The difference between this method and others in the literature is the use of two interpolation steps: first to calculate provisional interface normal velocities from cell-centered values, and the second to calculate the cell-centered values from the corrected interface velocities. The second interpolation step is needed because the interface velocities, and not the cell-centered velocities, are

corrected using the pressure distribution. This interpolation produces a stabilizing effect, so that there is no need to add stabilizing terms to the pressure equation.

4 Numerical Diffusion

We now show that interpolating the normal velocity onto interfaces and then back to cell centers produces a numerical damping term that has the same form as a normal viscous diffusion. Consider a function $u(x, y, t)$ for which values on cell centers u_i, u_{i+1} and interfaces $u_{i+1/2}$ on an equally spaced grid are known. The interpolated value $\overline{u}_{i+1/2}$ is given by

$$\overline{u}_{i+1/2} = \frac{u_i + u_{i+1}}{2}. \tag{12}$$

Then interpolating cell-centered values to interfaces, and then back to cell centers, gives (using a double overline to denote the two interpolation steps)

$$\overline{\overline{u}}_i = \frac{\overline{u}_{i-1/2} + \overline{u}_{i+1/2}}{2} \tag{13}$$

$$= \frac{1}{2}\left(\frac{1}{2}(u_{i+1} + u_i) + \frac{1}{2}(u_i + u_{i-1})\right) \tag{14}$$

$$= u_i + \frac{\Delta x^2}{4}\left(\frac{\delta^2 u}{\delta x^2}\right)_i, \tag{15}$$

with $\delta^2 u/\delta x^2$ indicating a centered finite-difference approximation for the second derivative. This pair of interpolations occurs once per timestep, so we can

view this diffusion as an unsteady term by rearranging Eq. (15) as

$$\frac{\overline{\overline{\rho u_i}} - \rho u_i}{\Delta t} = \frac{\rho \Delta x^2}{4\Delta t} \left(\frac{\delta^2 u}{\delta x^2} \right)_i . \tag{16}$$

The discretized viscous term in the x-momentum equation is

$$(\mu \nabla^2 u) = \mu \frac{\delta^2 u}{\delta x^2} + \mu \frac{\delta^2 u}{\delta y^2} + \mu \frac{\delta^2 u}{\delta z^2}, \tag{17}$$

so the interpolation adds numerical diffusion to the normal component only:

$$\begin{aligned} &(\mu \nabla^2 u) + \text{numerical diffusion} \\ &= (\mu + \mu_n) \frac{\delta^2 u}{\delta x^2} + \mu \frac{\delta^2 u}{\delta y^2} + \mu \frac{\delta^2 u}{\delta z^2}. \end{aligned} \tag{18}$$

Defining a viscous cfl number, cfl_v as

$$\text{cfl}_v = \frac{\Delta t \mu}{\Delta x^2 \rho}, \tag{19}$$

the numerical diffusion coefficient μ_n can be written as

$$\mu_n = \frac{\mu}{4\text{cfl}_v}. \tag{20}$$

This is an approximate analysis; the solution to Eq. (6) is used to update the interface velocities bewteen the two interpolation steps, whereas the derivation of Eq. (20) lumps the two interpolation steps together and neglects the changes that the pressure update imposes. The pressure update imposes a constraint rather than calculating an evolution step based on a source term, so the pressure calculation does not necessarily preserve the diffusion that results from the initial interpolation. Equation (20), however, does show that

increasing the timestep relative to the viscous timescale will reduce the effect of this numerical diffusion.

The impact of the numerical diffusion on solution acuracy can be assessed in two ways: the magnitude of μ_n relative to μ, which is given by the viscous cfl number, or the magnitude of $\mu_n \partial^2 u/\partial x^2$ compared to the (physical) viscous, pressure, and advection terms for each of the coordinate directions. If $\mu_n \ll \mu$, then the numerical diffusion will not degrade the accuracy of the calculation. The inverse is not necessarily true: a large value of μ_n compared to μ does not guarantee that the simulation results are tainted by excess diffusion. In many flows, the normal viscous stress is of secondary importance compared to the effects of shear stresses, pressure, and advection. Then this diffusion would not be problematic using an explicit solver with a modest cfl constraint for such cases. If the global time-step constraint imposed by advection is stricter than that for viscous transport, it could be advantageous to use a local time-step algorithm. This would allow each cell to march toward steady-state based on the local value of the velocity and would increase the local time step and reduce the effect of this numerical diffusion. On the other hand, if the time-step constraint for advection is less stringent than for the viscous transport, an implicit discretization for the viscous terms would ensure that this numerical diffusion is insignificant.

We can also compare the effects of the interpolation to the effects of adding explicit damping terms to the pressure Poisson equation. Denoting $\widetilde{u}_{i+1/2}$ as the true provisional interface velocity, the interpolated provisional velocity

$\bar{\bar{\tilde{u}}}_{i+1/2}$ satisfies

$$\bar{\bar{\tilde{u}}}_{i+1/2} = \tilde{u}_{i+1/2} + \frac{\Delta x^2}{8}\left(\frac{\partial^2 \tilde{u}}{\partial x^2}\right)_i + O(\Delta x^4). \tag{21}$$

Writing a similar expression for interface $i - 1$ allows us to write

$$\begin{aligned}\frac{\rho\bar{\bar{\tilde{u}}}_{i+1/2} - \rho\bar{\bar{\tilde{u}}}_{i-1/2}}{\Delta x} &= \frac{\rho\tilde{u}_{i+1/2} - \rho\tilde{u}_{i-1/2}}{\Delta x} \\ &+ \frac{\rho\Delta x^2}{8}\frac{\partial^3 \tilde{u}}{\partial x^3}_i + O(\Delta x^4).\end{aligned} \tag{22}$$

Differentiating the x component of Eq. (7) three times with respect to x gives

$$\frac{\partial^3 u^{n+1}}{\partial x^3} = \frac{\partial^3 \tilde{u}^{n+1}}{\partial x^3} - \frac{\Delta t}{\rho}\frac{\partial^4 p}{\partial x^4}. \tag{23}$$

The solution procedure replaces $\nabla \cdot (\rho\tilde{\mathbf{v}})$ in Eq. (6) with $\nabla \cdot (\rho\bar{\bar{\tilde{\mathbf{v}}}})$, solving

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t}\nabla \cdot (\rho\bar{\bar{\tilde{\mathbf{v}}}}). \tag{24}$$

Using Eqs. (22) and (23) (and similar expressions for the y velocity terms), Eq. (24) becomes

$$\begin{aligned}\nabla^2 p^{n+1} &= \frac{1}{\Delta t}\nabla \cdot (\rho\tilde{\mathbf{v}}) \\ &+ \frac{\rho\Delta x^2}{8\Delta t}\frac{\partial^3 u^{n+1}}{\partial x^3} + \frac{\rho\Delta y^2}{8\Delta t}\frac{\partial^3 v^{n+1}}{\partial y^3} \\ &+ \frac{\Delta x^2}{8}\frac{\partial^4 p}{\partial x^4} + \frac{\Delta y^2}{8}\frac{\partial^4 p}{\partial y^4} + O(\Delta x^4) + O(\Delta y^4).\end{aligned} \tag{25}$$

Therefore, using the finite-difference approach to calculate the divergence in Eq. (6) using interpolated interface velocities is equivalent to adding two sets of terms to Eq. (6). The first set is a third-derivative of each velocity component (as one might expect since we took the divergence of a second-order error term in the provisional velocity), and the second is a fourth-derivative of p similar

to the damping term often added to enhance stability. Therefore, the process of interpolating the cell-centered velocities to the interfaces mimics in part the effect of including a fourth-derivative of pressure as a damping term in the pressure equation.

Finally, an alternate way of viewing the interpolation is to consider each interpolation step as a mapping: one mapping, B, from cell-averaged values to point values at the interfaces, and a second, D, from point values to cell averages. If these mappings were perfectly matched, then $DB = I$, that is, mapping to the interfaces and then back again would produce the original data set. As a minimal requirement, the mappings must be conservative: if $\phi' = DB\phi$, then $\|\phi'\| = \|\phi\|$ for conserved quantity ϕ and the 1-norm $\|\cdot\|$ that measures the amount of ϕ in the domain. The current method satisfies this requirement: the interpolation steps are diffusive, but they are also conservative. Additional methods can be pursued by replacing the linear interpolation used in the current work with other mappings between cell-centered and interface values.

5 Numerical Results and Discussion

The current algorithm has been implemented in the code TINY3D, and two-dimensional examples are shown here to demonstrate the method. Kaplan et al. [12, 13] present three-dimensional calculations for a channel micromixer including a herringbone-patterned surface geometry.

Figure 2 shows the flowfield in a two-dimensional channel 100 μm high and 400 μm long. The flow on the left enters at a uniform speed of 1 cm/s, and $\mu = 0.014$ gm/(cm·s). We enforce no-slip conditions on the top and bottom walls, and the outflow (right) boundary is held at $p = 1$ atm. Due to the uniform inflow, a short development region is seen during which the equilibrium parabolic profile is formed. After this developing region, the pressure drop becomes linear. As is seen in Table 1, the calculated values for the equilibirum pressure drop are converging to the theoretical value as $\Delta x \rightarrow 0$. Figure 3 illustrates that the code demonstrates second-order convergence for this calculation. The simulation using 40 $\times$ 160 cells gives $\partial p/\partial x$ =1678 dyne/cm^3. This is within 0.1% of the theoretical value of 1680 dyne/cm^3. For this case, the damping term introduced by the interpolation is of no consequence since no normal velocity gradients exist in the equilibrium profile. The curve in Fig. 3 is the parabola that crosses the y-axis at the theoretical value of 1680 dyne/cm^3 and passes through the computational data point at $\Delta x = 2.5\mu m$. The fact that this same parabola passes through the other two computational data points at $\Delta x = 5\mu m$ and $\Delta x = 10\mu m$ indicates that the error is proportional to Δx^2 and the method exhibits Δx^2 convergence.

A second set of calculations focused on the two-dimensional flow over a backward-facing step. The geometry, x-velocity distribution, and streamline traces (pathlines) are shown in Figure 4. The step consists of a 1 mm channel opening into a 1.94 mm channel, and the computational domain extends 10 mm upstream (left) of the step and 20 mm downstream. Water at 13 cm/s enters from the

left with $\mu = 0.0089$ cm^2/s. Following Armaly et al. in their seminal work on the backward-facing step problem [14], we calculate a Reynolds number of 292 using this mean velocity as the characteristic speed and twice the 1 mm entrance width as the characteristic length. At this Reynold number, a single vortex forms downstream from the step. The size of this vortex, as characterized by the reattachment point on the bottom wall, provides a test for validating the order and the accuracy of the code. In order to match the exact geometry using moderate resolution, variable grid-spacing is used. The top of the step corresponds to $y = 0.094$ cm. Cells have uniform Δx throughout the grid, $\Delta y = \Delta x$ for those cells that lie above $y = 0.094$ cm, and $\Delta y = 0.94\Delta x$ for cells that lie below $y = 0.094$ cm. Therefore, if N cells span the entrance channel, then $2N$ cells span the exit channel, and the dimension of the two channels match the 1:1.94 ratio from the experiment.

This geometry and set of flow conditions provides a severe test of the numerical diffusion in the current explicit, time-accurate implementation, as characterized by Eqs. (18) and (20). The time step for the calculation is limited by the advection, so cfl_v is low, and there is significant normal viscous stress in the region just downstream from the step. Figure 5 indicates the relative magnitudes of the normal and transverse x-velocity derivatives that contribute to the viscous term for the x-momentum equation. The ratio shown is

$$r = \frac{\|\partial^2 u/\partial x^2\|}{\|\partial^2 u/\partial x^2\| + \|\partial^2 u/\partial y^2\|}. \tag{26}$$

For much of the computational domain, this ratio is less than 0.1, indicating

that the effects of shear are at least nine times larger than the normal stress. There are, however, some regions in which the normal viscous stress is greater than the tangential stress (indicated by $r > 0.5$), most noticeably just downstream of the step. A close-up of this region is shown in Fig. (6). At steady state, the net source term for each cell is zero, so that the contributions due to viscous flux, advective flux, and pressure are balanced. Figure (6) shows the individual cells for the medium-resolution simulation. The value of r for each cell indicates the relative magnitude of the (physical) normal stress and shear stress acting on the x-velocity component at equilibrium. High values of r indicate where the shear stresses on the x-velocity do not dominate normal stresses for this same velocity component. If r is low throughout the domain and cfl_v is high, we know that the numerical diffusion introduced by the pair of interpolation steps will be insignificant. In this region with high values of r, the y-velocity is dominant, and Figure (6) does not indicate whether the normal stresses on the x-velocity component play a significant role in the overall flow solution. Given the high values of r and the the small values of cfl_v, the calculation must be compared with experimental data to demonstrate the impact that the interpolation steps have on the accuracy. One is tempted to compare $\mu_n \|\partial^2 u/\partial x^2\|$ to the pressure and advective terms, but as noted in the previous section, such a comparison would be inconclusive. The test of accuracy for this calculation must be a study of the convergence behavior of the method and a comparison with experimental data.

To test the convergence rate and accuracy, a series of calculations on various

grids were performed. Table 2 gives the reattachment lengths for several grid sizes, and these values are plotted in Figs. 7 and 8. At $Re = 292$, Armaly et al. reports a reattachment length normalized by the height of the step of 6.55 [14], and this value is marked in Fig. 7. The agreement between the computational results and the experiment is very good, even though the values of cfl_v were very low because the time step for these cases is limited by the advective and not the viscous time scale. The standard cfl number for advection for a characteristic speed w is

$$\mathrm{cfl} = \frac{w\Delta t}{\Delta x}. \tag{27}$$

To guarantee positivity, FCT requires $\mathrm{cfl} \leq 0.5$, so the limit for the time step is set to $\mathrm{cfl} = 0.45$, which results in a more stringent constraint than limiting the timestep based on cfl_v. The resulting values of cfl_v are also listed in Table 2.

The coarsest grid included in Fig. (7) used only five cells to span the inlet channel and five to span the height of the step. At this resolution a single vortex is predicted downstream of the step, but the computation does not correctly capture the basic shape of this vortex. The other four calculations correctly capture the shape of the vortex as illustrated in Fig. (4), so these computations are used to study the convergence characteristics of the algorithm. The results for these four cases are shown in Fig. (8). Figure (8) also includes an ideal curve for second-order convergence similar to the theoretical curve in Fig. (3). For the channel-flow calculations, the theoretical value of the steady-state pressure gradient is known, and the pressure gradient predicted by the code converged

to the theoretical value as $\Delta x \rightarrow 0$. A theoretical value for the reattachment length is not known, and the experimental value cannot serve as a substitute for this value due to experimental limitations and uncertanties. Therefore, the curve in Fig. (8) indicating ideal second-order convergence behavior was generated using the data points for the two smallest grid sizes. Using these two data points and extrapolating to $\Delta x = 0$, second-order convergence predicts a reattachment length of 6.563. If we define $E(\Delta x)$ as the difference between the reattachment length calculated using grid size Δx and the asymptotic value as $\Delta x \rightarrow 0$, then a second-order method will give $E(\Delta x/2) = 1/4\ E(\Delta x)$. As the graph illustrates, the result on the grid with $\Delta x = 0.1$ mm is well below the theoretical curve. This indicates that the code is converging at a rate faster than second-order: $E(0.05\text{mm}) \approx 1/40\, E(0.1\text{mm})$. Since the viscous and advection transport routines, interpolations, and pressure distribution and gradient calculations are second-order, this apparent increase in the convergence rate is due to the reduced diffusion in the interpolation of cell-centered to interface quantities as μ_n decreases. Not only are the operator discretizations and flux calculations converging as Δx^2, but these operators also are working with less-diffused data.

6 Summary

A method for interpolating between cell-centered and interface values leads to a straightforward application of a projection method for incompressible

flows on colocated, Cartesian meshes. The interpolation procedure mimics, in part, a standard stabilizing term that is proportional to the fourth-derivative of pressure, so the current method does not include an explicit stabilizing term in the pressure equation. Analysis indicates that the numerical diffusion introduced by the interpolation steps acts as a normal stress and increases as the ratio of the numerical time step to the viscous time scale decreases. Comparisons with experimental results for two-dimensional backward-facing step flow demonstrate that accuracy is maintained even when the time step is limited by advection and is much greater than the times step allowed by the viscous transport.

7 Acknowledgements

This work was funded by the Office of Naval Research. The authors would also like to thank Fran Ligler, Joel Golden, Peter Howell, and Stephanie Fertig of NRL's Center for Biomolecular Science and Engineering for their valuable experimental work on the microfluidics program that supports these computational efforts.

References

[1] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamcis, 3rd. Edition, Springer, Berlin, 2002.

[2] Z. Liliek, M. Peric, A Fourth-Order Finite Volume Method with Colocated Variable Arrangement, *Computers & Fluids* 24 (3) (1995), 239-252.

[3] L.S. Caretto, A.D. Gosman, S.V. Patankar, D.B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation. Proceedinging of the Third International conference on Numerical Methods in Fluid Dynamics, Paris, 1972.

[4] J.P. van Doormal, G.D. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, *Numerical Heat Transfer*, **7**, (1987) 147-163.

[5] B.P. Leonard, A Stable and Accurate Convective Modeling Procedure Based on Quadratic Upstream Interpolation, *Computer Methods in Applied Mechanics and Engineering*, Vol. 19 (1979) 59-98.

[6] R.I. Issa, "Solution of implicitly discretized fluid flow equations by operator splitting," *Journal of Computational Physics* **62**, (1986) 40-65.

[7] P.S.B. Zdanski, M.A. Ortega, and N.G.C.R. Fico, Jr. "A Novel Algorithm for the Incompressible Navier-Stokes Equations," AIAA Paper 2003-434 (2003).

[8] K. Kuwahara, S. Komurasaki, Simulation of High Reynolds Number Flows Using Multidirectional Upwind Scheme, AIAA paper 2002-0133, 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2002.

[9] J. Waltz, An Implicit Finit Element Fractional Step Scheme for Incompressible Flows, AIAA Paper 2002-0431, 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (2003).

[10] J. P. Boris, Alexandra M. Landsberg, Elaine S. Oran, John H. Gardner, LCPFCT — A Flux-Corrected Transport Algorithm for Solving Generalized Continuity Equations, Memorandum Report 6410-93-7192, Naval Research Laboratory, 1993.

[11] J. P. Boris, D. L. Book, Flux-Corrected Transport I: SHASTA — A Fluid Transport Algorithm That Works, *Journal of Computational Physics* 11 (1973) 38-69.

[12] C.R. Kaplan, D.R. Mott, E.S. Oran, Towards the Design of Efficient Micromixers, AIAA paper 2004-0931, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2004.

[13] C.R. Kaplan, D.R. Mott, E.S. Oran, J. Liu. "Towards the Design of Efficient Micromixers," in preparation for *Physics of Fluids.*

[14] B.F. Armaly, F. Durst, J.C.F. Periera, B. Schonung, Experimental and theoretical investigation of backward-facing step flow, *Journal of Fluid Mechanics* 127 (1983) 473-476.

Table 1

Pressure gradient results for the channel flow using three different grids. The theoretical value is 1680 dyne/cm^3.

Number of Cells (height × length)	$\partial p/\partial x$ dyne/cm^3	% error
10 × 40	1647	1.96%
20 × 80	1672	0.48%
40 × 160	1678	0.12%

Table 2

Normalized reattachment length downstream of the step for 1:1.94 step at Re = 292, for various grid sizes. Experimental value for the 1:1.94 channel is 6.55 [14]. Simulation time step is specified by the requirement that cfl = 0.45 for the advection routine. The error is relative to the experimental value.

Cells spanning inlet	cfl_v	Ratio of reattachment length to step height	% error
5	0.012	2.553	61.1%
10	0.021	5.620	14.2%
15	0.029	6.470	1.3%
20	0.039	6.539	0.2%
40	0.080	6.557	0.1%

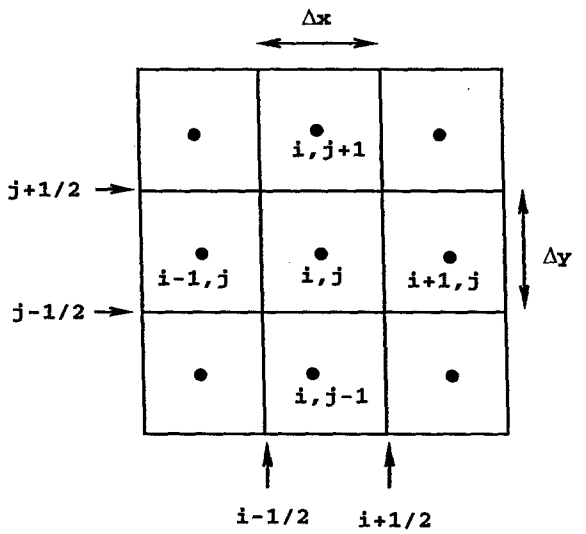

Fig. 1. Grid arrangement for the current method. All variables are stored at cell centers, and the cell acts as the control volume for all variables.

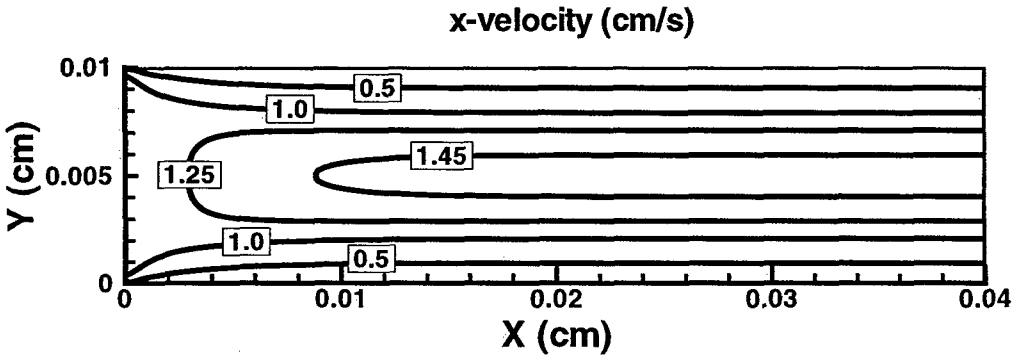

Fig. 2. X-velocity distribution for the 2D channel test problem.

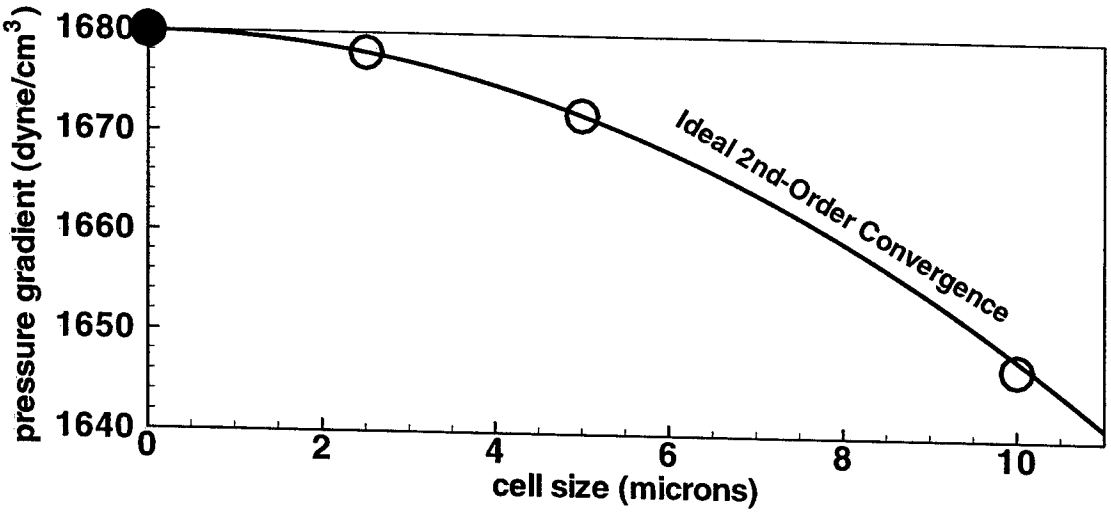

Fig. 3. Steady-state pressure gradient in the $100\mu m$ microchannel as a function of grid size. Open circles are computational results; filled circle is the exact solution.

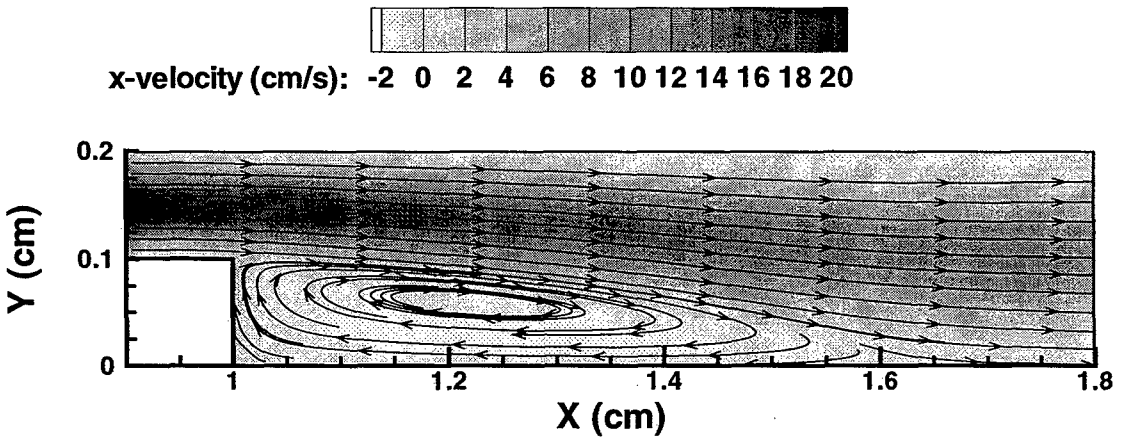

Fig. 4. x-velocity contours and stream traces for the backward-facing step problem.

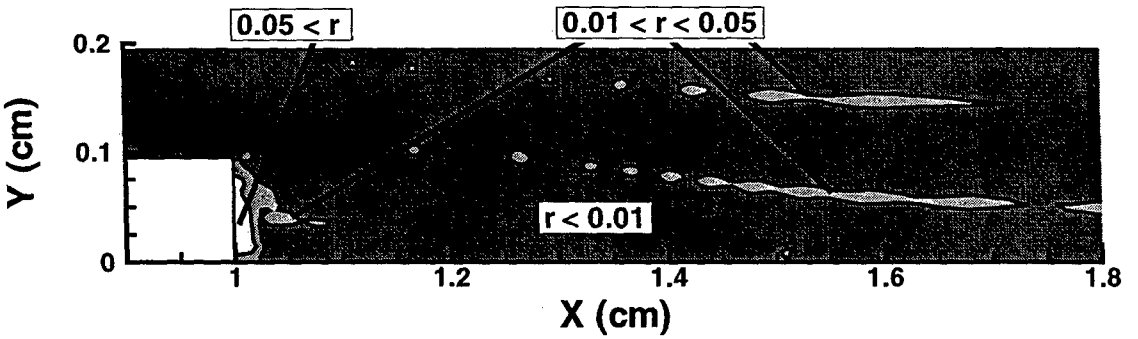

Fig. 5. The ratio $\|\partial^2 u/\partial x^2\|/(\|\partial^2 u/\partial x^2\| + \|\partial^2 u/\partial y^2\|)$, which indicates the relative importance of the normal viscous stress compared to the tangential viscous stress at steady state. Contours generated by treating the cell-centered values as point values.

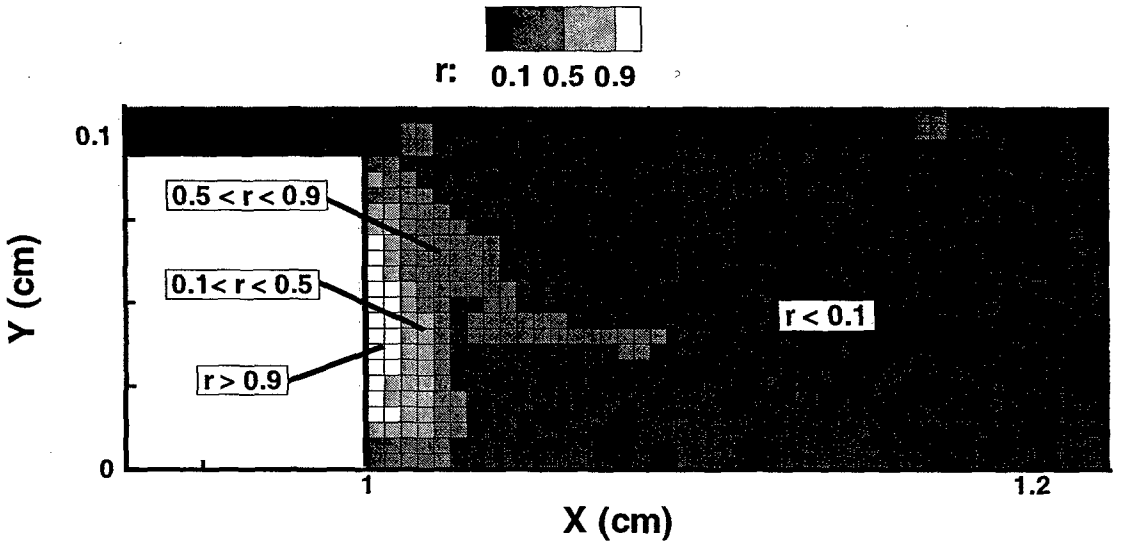

Fig. 6. Detail behind the step of the ratio $\|\partial^2 u/\partial x^2\|/(\|\partial^2 u/\partial x^2\| + \|\partial^2 u/\partial y^2\|)$ at steady state, plotted as cell values.

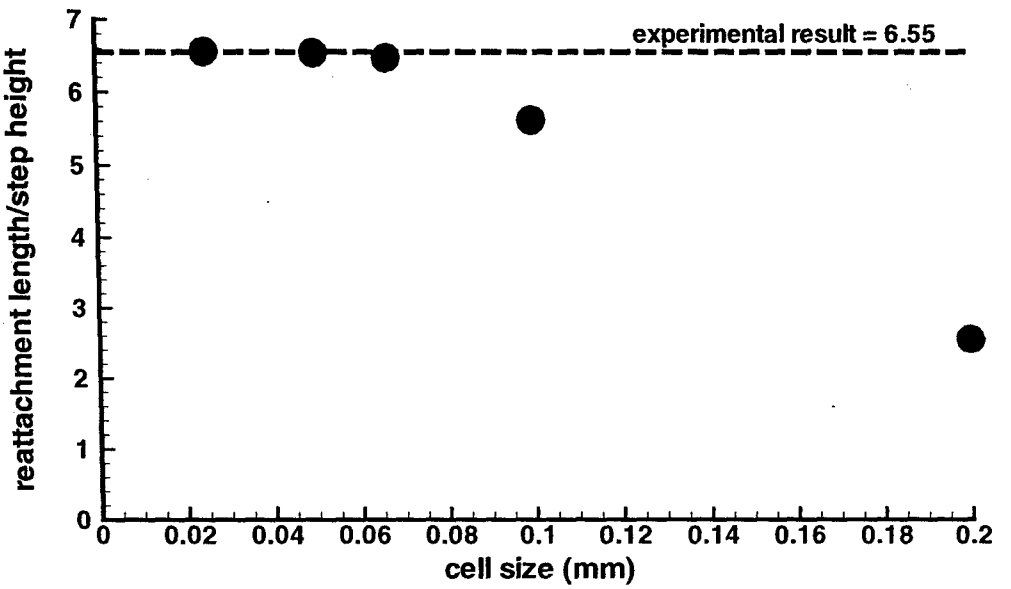

Fig. 7. Normalized reattachment length as a function of grid size. Circles are computational results; dashed line indicates the experimental value of 6.55.

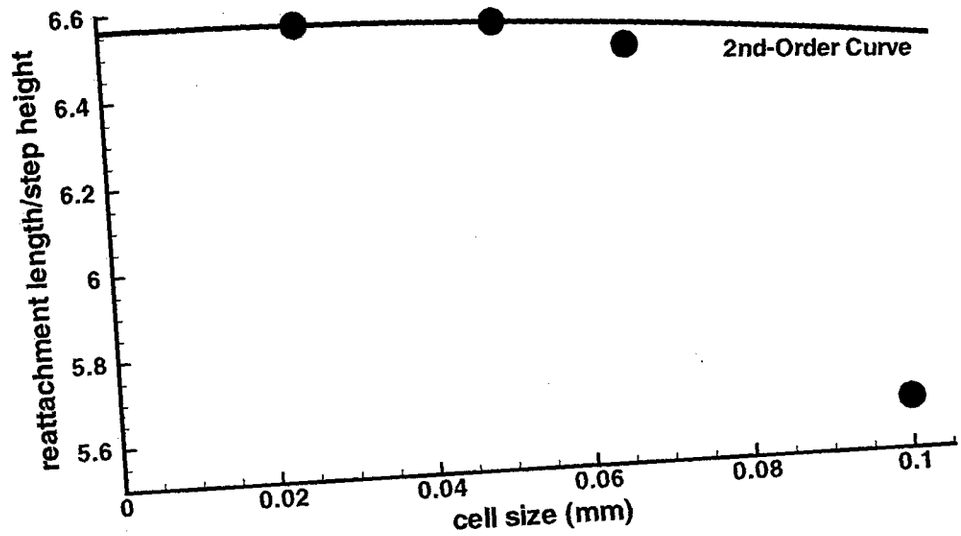

Fig. 8. Normalized reattachment length as a function of grid size for the four finest grids.. Solid curve indicates the ideal second-order convergence curve derived from the two finest grids.