

STABLE FOURTH-ORDER STREAM-FUNCTION METHODS FOR INCOMPRESSIBLE FLOWS WITH BOUNDARIES*

Thomas Y. Hou

*Department of Applied and Computational Mathematics, California Institute of Technology,
Pasadena, CA, 91125 USA
Email: hou@acm.caltech.edu*

Brian R. Wetton

*Department of Mathematics, University of British Columbia, Vancouver, BC, V6T 1Z2, Canada
Email: wetton@math.ubc.ca*

Abstract

Fourth-order stream-function methods are proposed for the time dependent, incompressible Navier-Stokes and Boussinesq equations. Wide difference stencils are used instead of compact ones and the boundary terms are handled by extrapolating the stream-function values inside the computational domain to grid points outside, up to fourth-order in the no-slip condition. Formal error analysis is done for a simple model problem, showing that this extrapolation introduces numerical boundary layers at fifth-order in the stream-function. The fourth-order convergence in velocity of the proposed method for the full problem is shown numerically.

Mathematics subject classification: 65M12, 76D05.

Key words: Incompressible flow, Stream-function formulation, Finite difference methods.

1. Introduction

Numerical methods based on the vorticity stream-function formulation of the Navier-Stokes equations have been used with success for flows in two-dimensions (2D). There are many references to second-order finite difference methods applied to different problems: the driven cavity problem [20]; flow past a cylinder [16]; and flow in tubes with occlusions [14] for example. Considerable effort has been spent in developing higher order finite difference methods for the vorticity stream-function formulation, including the early work in [1, 7, 10, 11]. These authors use some combination of compact differencing and one-sided differencing near the boundary to develop fourth-order methods. More modern approaches [3, 9] use the stream-function directly, without reference to vorticity. These methods also use compact differencing. In this paper, we develop a new fourth-order method for the time dependent Navier-Stokes and Boussinesq Equations that uses a wide stencil rather than compact differencing. This approach offers additional flexibility in choice of time stepping and applicability in mapped grids over other methods. Formal analysis of a simple model problem indicates that the method will give fourth-order accuracy in computed velocities. This is demonstrated for the full problem in numerical convergence studies.

There are several reasons to consider higher order methods. Of course, the underlying reason is that we wish to get an approximation to a flow to a given accuracy or resolution using less computational time and less storage space. This can manifest itself in many ways. In some cases, high accurate benchmark solutions are required [11]. In other cases, the higher

* Received March 11, 2008 / Revised version received May 3, 2008 / Accepted August 6, 2008 /

order accuracy is required to resolve delicate phenomena [1, 16]. The results of Kreiss [17] and Chang [6] show that fourth-order methods are in some sense “optimal”. Their results show that fourth-order methods outperform second-order methods for 2D model convection-diffusion problems. They also show that if only modest accuracy (1%) is required that fourth-order methods are at least as efficient as higher order finite difference methods and more efficient than second-order and pseudo-spectral methods. Of course, if very high accurate solutions are needed, pseudo-spectral methods are the best to use.

There are two basic approaches to the construction of higher order finite difference methods for the time-dependent Navier-Stokes equations and more generally, any convection diffusion problem with smooth solutions. One approach is the use of compact differencing which gets its name from the fact that the resulting method is based on a difference stencil no larger than that needed for a second-order method. There are many variants of this approach including [7, 10, 11]. The method in [1] combines compact differencing for the stream-function equation with an integral constraint technique to solve for the boundary vorticity. Strictly speaking, what we will call compact differencing from now on in this paper is actually a restricted subset known as Operator Compact Implicit (OCI) differencing in which only the physical unknowns are used on grid points (see [4] for a good discussion). In another type of compact differencing, called simply Compact Implicit (CI), derivatives of the unknowns are introduced at each grid point. This is the approach taken in [3]. The introduction of the extra unknowns effectively widens the difference stencil and extrapolation or one sided differencing must be used to eliminate the derivative values on the boundary (see [11]) in higher order methods, so CI are similar in spirit to the wide methods discussed below. We will not pursue CI methods in this paper.

The approach taken in this work is to use standard fourth-order differencing using a wider difference stencil than needed for second-order accuracy. We call these wide methods to distinguish them from compact methods. For periodic problems this approach is very natural, but for problems with boundaries, these methods require values on more grid points outside the boundary than can be eliminated using the boundary conditions of the problem. The authors propose below a method of “extrapolating” the values outside the grid using values inside the grid based on their earlier work in vorticity boundary conditions for second-order schemes [12]. The values at grid points outside the boundary are related to the values inside the boundary in such a way that they satisfy the boundary conditions to fourth-order accuracy. For stream-function methods, it is the no-slip condition that is matched to fourth-order to give velocities that are fourth-order accurate.

In the next section, the 2D Navier-Stokes and Boussinesq approximations are introduced. Then, wide and compact schemes are introduced and applied to a 1D model problem. Error analysis of the wide scheme for the model problem is conducted, using matched interior and numerical boundary layer errors as presented in [22]. This is followed by a section on the specification of fourth-order wide methods for the 2D incompressible fluid flow equations. These methods are then applied to two 2D example problems in fluid flow where the fourth-order convergence in velocities (and temperature for the Boussinesq flow example) of the method is verified.

2. The Navier-Stokes and Boussinesq Equations

The 2D Navier-Stokes equations in primitive variables (velocity $\mathbf{u} = (u, v)$ and pressure p) are

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \nu \Delta \mathbf{u}$$

with the incompressibility condition $\nabla \cdot \mathbf{u} = 0$. We can take the third component of the curl of the above equations to get the vorticity form of the Navier-Stokes equations:

$$\omega_t + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega \quad (2.1)$$

where $\omega = v_x - u_y$ is the vorticity. Using the incompressibility condition, we can introduce a stream function ψ that satisfies

$$\psi_x = -v; \psi_y = u; \Delta \psi = -\omega. \quad (2.2)$$

The stream function can be thought of as an auxiliary variable for getting the velocities from the vorticity to update (2.1). This is the typical approach taken in vorticity stream-function methods, where the stream-function values also provide expressions for the boundary vorticity. One can also eliminate the vorticity completely in favour of the stream function to obtain the stream function formulation of the Navier-Stokes equations:

$$\Delta \psi_t + \psi_y \Delta \psi_x - \psi_x \Delta \psi_y = \nu \Delta \Delta \psi. \quad (2.3)$$

This approach has had some recent popularity [3]. In this formulation, the boundary conditions $\psi = 0$ (no flow) and $\partial \psi / \partial n = 0$ (no slip) corresponding to homogeneous Dirichlet boundary conditions on the velocity ($\mathbf{u} = 0$) are easily seen to be appropriate and do not need to be reinterpreted as vorticity boundary conditions. The authors have found that by considering numerical methods for the stream-function equations directly, greater insight into the analysis and efficient implicit time integration techniques can be obtained [12]. This formulation of the Navier-Stokes equations is used as the basis for the numerical methods in the rest of the paper.

Also discussed in this work are the Boussinesq approximations to buoyancy driven flow. In this approximation, the temperature θ is an additional variable that is convected with the flow and diffuses:

$$\theta_t + \psi_y \theta_x - \psi_x \theta_y = \mu \Delta \theta. \quad (2.4)$$

In this work, we consider Dirichlet boundary data for θ . There is a gravity (assumed to be in the negative y direction) driven buoyancy force which modifies the Navier-Stokes equations (2.3) as follows:

$$\Delta \psi_t + \psi_y \Delta \psi_x - \psi_x \Delta \psi_y = \nu \Delta \Delta \psi + \theta_x. \quad (2.5)$$

3. Compact and Wide Schemes for a 1D Model Problem

In this section we derive and discuss compact and wide schemes for a 1D problem written in terms of a stream function. No slip and no flow boundary conditions are applied and the method of extrapolated boundary conditions for wide schemes is presented. Formal error analysis is conducted for the model problem, showing that the wide scheme has smooth fourth-order errors in the interior and fifth-order numerical boundary layers in the stream function. Numerical verification that this approach leads to stable schemes that converge with fourth-order in velocities is given. The effect of using boundary conditions of different orders is considered.

3.1. Notation for difference operators

We consider discrete approximations on a regular grid with grid spacing h in space using capital letters with subscripts to denote the grid location, *i.e.*

$$\Psi_i(t) \approx \psi(ih, t).$$

Difference methods using the method of lines are developed, that is ODE's for $\Psi(t)$ are derived which are later discretized in time. Centered differencing is used throughout this paper and D_k and \tilde{D}_k denote the second and fourth-order centered difference approximations of the k 'th derivative, respectively. For example,

$$D_2\Psi_i = \frac{\Psi_{i-1} - 2\Psi_i + \Psi_{i+1}}{h^2}$$

$$\tilde{D}_2\Psi_i = \frac{-\Psi_{i-2} + 16\Psi_{i-1} - 30\Psi_i + 16\Psi_{i+1} - \Psi_{i+2}}{12h^2}.$$

For problems in more than one direction superscripts are used to indicate the direction of the derivatives, *i.e.*, D_2^x .

3.2. A one-dimensional model problem

The following model problem for $\psi(x, t)$ is considered on the interval $x \in [0, 1]$

$$\psi_{xxt} = \nu\psi_{xxxx} \tag{3.1}$$

with no-flow boundary conditions

$$\psi(0) = \psi(1) = 0$$

and no-slip boundary conditions

$$\frac{\partial\psi}{\partial x}(0) = \frac{\partial\psi}{\partial x}(1) = 0.$$

These equations and boundary conditions model the diffusion terms in Eq. (2.3) in a reduced dimensional setting. Wide and compact schemes are derived below and tested on the above problem. A standard second-order method is presented below for comparison.

3.3. Second-order scheme

A second-order semi-discrete (*i.e.*, method of lines) scheme for (3.1) is

$$D_2\dot{\Psi} = \nu D_4\Psi,$$

where the dot denotes time derivative. Boundary conditions for the operator D_2 are the no-flow conditions $\Psi_0 = \Psi_N = 0$ where $N = 1/h$. We use Thom's boundary conditions

$$\Psi_{-1} = \Psi_1 \tag{3.2}$$

at the lower wall and $\Psi_{N+1} = \Psi_{N-1}$ at the upper wall to eliminate the values outside the boundary needed in the stencil for the operator D_4 . The method above describes implicitly the evolution of the values of Ψ on the interior grid points. The convergence of this second-order scheme for this model problem follows from the results in [12] for the 2D nonlinear problem.

3.4. Wide schemes

Wide schemes are built using standard fourth-order difference operators:

$$\tilde{D}_2 \dot{\Psi} = \nu \tilde{D}_4 \Psi \quad (3.3)$$

If the method above is to describe the evolution of the values of Ψ on the interior grid points, then the values of Ψ at one and two grid points outside the boundary must be related to the interior values. This is done below using the method of extrapolated boundary conditions.

3.5. Extrapolated boundary conditions

To eliminate one point outside the boundary, the following formulas

$$\begin{aligned} \Psi_{-1} &= \Psi_1 \quad \text{Thom} \\ \Psi_{-1} &= 3\Psi_1 - \Psi_2/2 \quad \text{Wilkes} \\ \Psi_{-1} &= 6\Psi_1 - 2\Psi_2 + \Psi_3/3 \quad \text{Fourth-Order} \end{aligned} \quad (3.4)$$

are second-, third-, and fourth-order in terms of the no-slip condition respectively, *i.e.*, the terms in (3.4) can be written as

$$(\Psi_3/3 - 2\Psi_2 + 6\Psi_1 - 10\Psi_0/3 - \Psi_{-1})/(4h) \approx \frac{\partial\psi}{\partial x}(0) + \frac{h^4}{20} \frac{\partial^5\psi}{\partial x^5}(0) \quad (3.5)$$

(verify by Taylor series) and so (3.4) can be seen to be a fourth-order accurate approximation of $\partial\psi/\partial x(0) = 0$. In this form, one can see how to modify (3.4) for non-zero values of ψ or its normal derivative on the boundary. The names for the second- and third-order extrapolations of the no-slip conditions are the commonly used names for these boundary conditions when interpreted as vorticity stream-function boundary conditions. Similar formulas can be written for the upper wall. Formula (3.4) is used to define a fourth-order version of the discrete operator \tilde{D}_2 near the boundary. This operator must be inverted at each time step in an explicit scheme and (3.4) has the nice property that it does not widen the stencil of \tilde{D}_2 . This is also true of the 2D methods described in section 4.

In order to eliminate further points outside the boundary the same procedure is used. The following is a fourth-order (based on the no-slip condition) extrapolation

$$\Psi_{-2} = 40\Psi_1 - 15\Psi_2 + 8\Psi_3/3. \quad (3.6)$$

A simple calculation shows that (3.4) and (3.6) imply

$$\tilde{D}_1 \Psi_0 = 0, \quad (3.7)$$

where as discussed in section 3.1, \tilde{D}_1 denotes the fourth-order, centered difference approximation of the first derivative. Thus we can equivalently consider (3.4) and (3.7) as the two numerical boundary conditions. This leads to a simplification in the presentation of the error analysis of this scheme presented in section 3.7 below.

3.6. Compact scheme

Below an approach to developing compact schemes for time dependent problems using the method of lines is presented. The resulting ODE can then be discretized in time in a number

of ways. This approach is slightly different and we feel clearer than the approach taken by other authors (see, *e.g.*, [7]), where the construction is done for a fixed time stepping scheme. In our approach, compact schemes are derived by eliminating error terms from second-order discretizations using expressions obtained by taking derivatives of the original equations. The derivation looks a little different but gives identical results as the polynomial approach used in [10] and elsewhere. We call the methods derived with this approach “natural” compact differencing. It is essentially an extension of the method of deferred corrections [15] to time-dependent problems.

Using the same stencil as the second-order method, the most accurate direct discretization of (3.1) is

$$\tilde{D}_2 \dot{\Psi} = \nu D_4 \Psi$$

which still has a second-order error term

$$e_2 := \frac{\nu h^2}{6} \frac{\partial^6 \psi}{\partial x^6}$$

which comes from the use of D_4 instead of \tilde{D}_4 . This error cannot be eliminated on the compact stencil directly, but taking two derivatives of the original equation (3.1) gives

$$\nu \frac{\partial^6 \psi}{\partial x^6} = \psi_{xxxxxt}.$$

Now the second-order error term e_2 can be corrected using

$$\frac{h^2}{6} D_4 \dot{\Psi}.$$

Even though only second-order differencing is used above, the expression is equal to e_2 up to fourth-order accuracy because of the extra factor of h^2 . If this correction is applied, the following fourth-order accurate compact differencing method is obtained:

$$\left(\tilde{D}_2 + \frac{h^2}{6} D_4\right) \dot{\Psi} = \nu D_4 \Psi. \quad (3.8)$$

The idea here is that errors involving higher order spatial derivatives are replaced with lower order spatial derivatives of the time derivative term (or the data for time independent problems). In a special method for the Euler equations [18], the fourth-order errors vanish due to an orthogonality condition. In general, though, correction terms must be derived as above.

The compact method derived above (3.8) must be augmented with one of the boundary conditions discussed above to eliminate the values on grid points outside the boundary.

3.7. Formal error analysis

3.7.1. Compact scheme

The error analysis of the compact scheme is straight forward since errors are regular (smooth in space and time). It can be shown using the techniques of [12, 22] that the scheme (3.8) with boundary condition (3.4) has discrete solutions with the following property

$$\Psi_i(t) = \psi(ih, t) + h^4 \psi^{(4)}(ih, t) + \mathcal{O}(h^6), \quad (3.9)$$

where $\psi^{(4)}(x, t)$ is a smooth function independent of h that satisfies

$$\psi_{xxt}^{(4)} = \nu\psi_{xxxx}^{(4)} + \frac{\nu}{90} \frac{\partial^8 \psi}{\partial x^8} \quad (3.10)$$

with zero initial conditions and boundary conditions $\psi^{(4)}(0, t) = 0$ and

$$\psi_x^{(4)}(0, t) = -\frac{1}{20} \frac{\partial^5 \psi}{\partial x^5}(0, t) \quad (3.11)$$

and a similar condition at $x = 1$. Note that Eq. (3.10) and the boundary condition (3.11) for the error term are the same as the underlying problem with forcing from the truncation errors in the interior discretization and the boundary conditions. The error expansion (3.9) can be made to satisfy the discrete equations to arbitrarily high accuracy. The form (3.9) shows that fourth-order convergence can be expected in Ψ and its differences (the higher order terms can also be shown to be smooth).

It is clear here that the order of accuracy of the discrete boundary condition should be measured in terms of the accuracy it has to the no-slip condition (there has been much historical confusion on this point). If the lower order condition (3.2), which can be interpreted as a second-order approximation of the no-slip condition [12], is used with the compact interior scheme (3.8), global smooth errors of second-order will result. This is verified computationally below.

3.7.2. Wide Scheme

We consider now the error analysis of the fourth-order scheme (3.3) with boundary extrapolations (3.4) and (3.7). The error expansion for this scheme (showing only the highest order terms of each type) is

$$\Psi_i(t) = \psi(ih, t) + h^4\psi^{(4)}(ih, t) + A(t)h^5\kappa^i + \dots, \quad (3.12)$$

where $\psi^{(4)}(x, t)$ and $A(t)$ are smooth functions and $0 < \kappa < 1$ all independent of h . The term involving κ above is a numerical boundary layer at $x = 0$. There is also a boundary layer at $x = 1$ that is not shown in the expansion above. The expansion (3.12) should solve the discrete equations to arbitrarily high order. Inserting (3.12) into (3.3) we see immediately that

$$\tilde{D}_4\kappa^i = 0.$$

We can write \tilde{D}_4 as $D_4(I - \frac{h^2}{6}D_2)$ and see that κ^i must be root of $(I - \frac{h^2}{6}D_2)\kappa^i$,

$$-\frac{1}{6}\kappa + \frac{4}{3} - \frac{1}{6}\frac{1}{\kappa} = 0.$$

The roots come in reciprocal pairs (one for each boundary), $\kappa = 4 - \sqrt{15} \approx 0.1270$ and its reciprocal. The smooth interior error leads to

$$\psi_{xxt}^{(4)} = \nu\psi_{xxxx}^{(4)} - \frac{1\nu}{360} \frac{\partial^8 \psi}{\partial x^8}.$$

$\psi^{(4)}$ has zero initial conditions and $\psi^{(4)}(0, t) = 0$. The two boundary extrapolations (3.4) and (3.7) are matched by $A(t)$ and $\psi_x^{(4)}(0)$. Since $\tilde{D}_1 = D_1(I - \frac{h^2}{6}D_2)$ we see that the numerical boundary layer does not enter (3.7) at highest order. We find

$$\psi_x^{(4)}(0, t) = +\frac{1}{180} \frac{\partial^5 \psi}{\partial x^5}(0, t).$$

Now from (3.4) we find that

$$A(t) = -\frac{1}{18C} \frac{\partial^5 \psi}{\partial x^5}(0, t),$$

where $C = (\kappa^3/3 - 2\kappa^2 + 6\kappa - 10/3 - 1/\kappa)/4$. Similar conditions apply at $x = 1$.

The structure of (3.12) shows that differences of this scheme converge with fourth-order accuracy in the interior (away from the boundary layers). $\tilde{D}_1\Psi$ converges to ψ_x (velocities) with fourth-order accuracy even up to the boundary. This is verified in the section below, and the equivalent fourth-order methods for 2D fluid flow are also shown numerically to have this property. However, values of $\tilde{D}_2\Psi$ converge to ψ_{xx} (vorticity) only with third-order accuracy near boundaries.

3.8. Computational results

The above schemes were tested on the one-dimensional model problem (3.1) using $\nu = 0.01$ and initial data consisting of a combination of the first three even and the first three odd modes of the problem. The computed solutions are compared to the exact solution at time 1. Maximum norm errors to the exact solution are shown in Table 1 (for comparison the maximum value of the exact solution at time 1 is about 3.6). The second-order scheme uses the second-order Thom boundary conditions. The fourth-order wide scheme uses the fourth-order extrapolated boundary conditions presented above. The effect of using different order boundary conditions for the compact scheme is shown in Table 2. The time stepping used is standard fourth-order Runge Kutta (4RK) with time step $k = \min(h, \frac{h^2}{8\nu})$. With these small time steps, the errors from the time discretization are smaller than the last digit quoted in the results in Tables 1 and 2 so these results describe the errors in the spatial discretization. The numbers in brackets in Tables 1 and 2 show the approximate convergence rates given by $\log_2(e(N)/e(2N))$ where $e(N)$ is the error in the computation using step size $h = 1/N$.

Table 1: Max norm errors (and convergence rates) to the exact solution of computations for the one dimensional Stokes flow model problem

N	Second-Order	Wide 4th Order	Compact
16	0.514 (2.03)	0.914e-1 (3.78)	0.310 (4.50)
32	0.126 (1.99)	0.664e-2 (3.83)	0.137e-1 (4.77)
64	0.317e-1 (1.98)	0.467e-3 (3.92)	0.502e-3 (4.83)
128	0.804e-2 (1.99)	0.307e-4 (3.97)	0.177e-4 (4.79)
256	0.202e-2	0.196e-5	0.641e-6 (4.68)
512			0.250e-7 (3.85)
1024			0.178e-8

Table 2: Max norm errors (and convergence rates) to the exact solution using the compact differencing method and boundary conditions of different order on the one dimensional Stokes flow model problem

N	Thom	Wilkes	Fourth-Order
16	0.578e-1 (2.45)	0.578 (2.75)	0.310 (4.50)
32	0.106e-1 (2.12)	0.862e-1 (2.89)	0.137e-1 (4.77)
64	0.244e-2 (2.02)	0.116e-1 (2.95)	0.502e-3 (4.83)
128	0.602e-3 (2.00)	0.150e-2 (2.98)	0.177e-4 (4.79)
256	0.151e-3	0.190e-3	0.641e-6

The results in Table 1 show that the wide difference scheme converges with fourth-order and give errors much smaller than the second-order scheme, especially for large N . For small N the compact method appears to be converging at an order higher than fourth, but this is just due to the fact that the fourth-order error term in the interior is quite small and hard to resolve. More resolved calculations with simpler initial data confirm the fourth-order convergence.

The results of Table 2 show the importance of using fourth-order boundary conditions to obtain the best performance from a fourth-order method. When using a second-order boundary condition, the errors are smaller if a fourth-order scheme is used compared to a second-order in the interior (compare with the first column in Table 1) but the convergence rate is dominated by the boundary error and is only second-order. The third-order boundary condition (which is second-order in terms of the vorticity) gives third-order convergence. The fourth-order results are duplicated from Table 1 and show convergence of fourth-order when fully resolved as discussed above.

For time independent problems, the use of fourth-order vorticity stream-function boundary conditions in a compact fourth-order scheme leads to fourth-order convergence [7] in Boussinesq flow. Their boundary condition is fourth-order accurate in the boundary vorticity. Our results show that this can be weakened to a boundary condition accurate to fourth-order in the no-slip condition. This extra freedom may lead to more stable schemes. In other work [10] third-order boundary conditions (the Wilkes boundary conditions which are second-order in boundary vorticity) are used with compact fourth-order differencing in the interior. This author sees better performance in this method than for second-order methods for Navier-Stokes flow, but we believe from the tests shown in Table 2 that much superior performance could be obtained if a fourth-order boundary condition was used.

4. Wide Fourth-Order Methods for the Navier-Stokes and Boussinesq Equations

Fourth-order wide schemes for the Navier-Stokes equations and Boussinesq equations for a simple geometry (a periodic channel) are presented. Numerical stability tests are performed and appropriate fourth-order time stepping techniques are discussed. Numerical verification of fourth-order convergence is given in section 5 where computations using these methods are performed on two model problems.

4.1. Semi-discrete equations

We consider flows in a simple domain, a unit periodic channel, *i.e.* $(x, y) \in [0, 1] \times [0, 1]$ with periodic conditions in x and $\mathbf{u} = 0$ on $y = 0$ and $y = 1$. A regular $N \times N$ grid with grid spacing $h = 1/N$ in both directions is laid on the channel and semi-discrete approximations (discrete in space only) $\Psi_{i,j}(t) \approx \psi(ih, jh, t)$ are considered as before. The interior discretization in space for the Navier-Stokes equations (2.3) is derived first. Using the notation of section 3.1 the fourth-order discrete Laplacian operator is defined as

$$\tilde{D}_h = \tilde{D}_2^x + \tilde{D}_2^y$$

Higher derivative terms ($\Delta\Delta$, $\frac{\partial}{\partial x}\Delta$ and $\frac{\partial}{\partial y}\Delta$) are calculated as follows. The biharmonic term $\Delta\Delta$ is written out

$$\frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2\partial y^2} + \frac{\partial^4}{\partial y^4}.$$

Each term is approximated using centered fourth-order differencing. The same is done to the other higher order derivatives. The approximation now becomes

$$\tilde{\Delta}_h \dot{\Psi} = -\tilde{D}_1^y \Psi (\tilde{D}_3^x + \tilde{D}_1^x \tilde{D}_2^y) \Psi + \tilde{D}_1^x \Psi (\tilde{D}_3^y + \tilde{D}_1^y \tilde{D}_2^x) \Psi + \nu \tilde{B}_h \Psi, \quad (4.1)$$

where

$$\tilde{B}_h = \tilde{D}_4^x + 2\tilde{D}_2^x \tilde{D}_2^y + \tilde{D}_4^y.$$

In this scheme, stream-function values up to three grid points away are needed. The values outside the boundary are eliminated using extrapolation of interior values. This is discussed in the next section.

Fourth-order centered difference approximations of the Boussinesq equations are generated in a straight forward way. The differences in the discrete temperature equation involve values of the temperature up to two grid points away.

4.2. Boundary conditions

For the Navier-Stokes equations no-flow $\psi = 0$ and no-slip $\partial\psi/\partial y = 0$ boundary conditions were assumed on the walls $y = 0$ and $y = 1$ and periodic conditions in x . The interior discretization requires stream-function values on the boundary. These are supplied directly by the no-flow conditions on the lower wall $\Psi_{i,0} = 0$ for all i and on the upper wall $\Psi_{i,N} = 0$ for all i . The fourth-order scheme requires stream-function values up to two grid points outside the domain. These are eliminated in terms of interior values by extrapolation up to fourth-order in the no-slip condition as in section 3.5:

$$\Psi_{i,-1} = 6\Psi_{i,1} - 2\Psi_{i,2} + \Psi_{i,3}/3 \quad (4.2)$$

$$\Psi_{i,-2} = 40\Psi_{i,1} - 15\Psi_{i,2} + 8\Psi_{i,3}/3. \quad (4.3)$$

Similar formulas for the upper wall are applied.

We now turn to the specification of temperature boundary conditions. In the calculations described in section 5 Dirichlet conditions are given for the temperature: $\theta = 1$ on the bottom wall and $\theta = 0$ on the top wall. These can be applied directly to obtain $\Theta_{i,0} = 1$ and $\Theta_{i,N} = 0$ for all i as discrete boundary conditions.

The wide fourth-order differencing requires extra discrete boundary conditions for the temperature. At the lower boundary the value required one grid point outside the boundary is eliminated by extrapolation from interior values, just as was done above for the values of the stream-function outside the domain:

$$\Theta_{i,-1} = 4\Theta_{i,0} - 6\Theta_{i,1} + 4\Theta_{i,2} - \Theta_{i,3}. \quad (4.4)$$

This boundary condition is satisfied up to fourth-order and can be considered to be an approximation of the specified Dirichlet temperature data. The approximation order is measured in the same way as that of reflection boundary conditions for a staggered grid scheme [13].

4.3. Investigation of stability

In this section the stability of the semi-discrete scheme (4.1) with extrapolated boundary conditions is investigated. To do this, we determine the spectrum of the method applied to a linearized problem. This analysis also allows us to choose time steps for the above schemes

appropriately. Eq. (2.3) is linearized about Poiseuille flow with unit maximum velocity in the periodic channel ($u = 4y(1 - y)$ and $v = 0$):

$$\Delta\psi_t + 4(y - y^2)\Delta\psi_x - 8\psi_x = \nu\Delta\Delta\psi. \quad (4.5)$$

A semi-discrete wide approximation of this problem similar to (4.1) with extrapolated boundary conditions is considered and since this problem is linear, it can be written as follows

$$A\dot{\Psi} = B\Psi.$$

The spectrum of the discrete system can be examined by computing the generalized eigenvalues λ and eigenvectors Ψ of

$$A\Psi = \lambda^{-1}B\Psi.$$

This can be done efficiently for the periodic channel geometry by using the FFT to partially reduce the problem (see the remark at the end of section 4.4) and then finding eigenvalues for each Fourier component using the QZ algorithm. This was done for $\nu = 0.001$ (corresponding to a Reynolds number (Re) of 1000 based on the full width of the channel) for several values of h and it was observed that

$$|\text{Im}(\lambda)| \leq 1.34/h \quad (4.6)$$

and

$$-10\nu/h^2 \leq \text{Re}(\lambda) \leq 0. \quad (4.7)$$

Since the eigenvalues all have negative real part, the semi-discrete method for (4.5) is asymptotically stable. This is reasonable since $\text{Re}=1000$ is below the critical Reynolds number for Poiseuille flow [8] and so the continuous problem (4.5) has no growing modes. When the linearized convection terms in (4.5) are dropped (this corresponds to Stokes flow), the eigenvalues of the semi-discrete scheme are pure real and satisfy (4.7). Note that this result is not a convergence proof for the method applied to the linearized problem, since this computation gives no measure of how close to orthogonal the set of eigenvalues of the matrix $A^{-1}B$ is.

For a standard centered second-order stream-function discretization using Thom's boundary conditions (3.2) as described in [12] the spectra lies in $|\text{Im}(\lambda)| \leq 1/h$ and $-8\nu/h^2 \leq \text{Re}(\lambda) \leq 0$ so the use of fourth-order differencing and extrapolated boundary conditions does not increase the size of the region containing the spectrum very much.

We rely on more direct computational evidence to see the stability of the temperature boundary conditions in the approximation of the Boussinesq equations. This is given by the computations in section 5.

4.4. Time stepping

It is certainly possible to use simple second-order time integration techniques to discretize our fourth-order semi-discrete equations (4.1), using quite small time steps to ensure that the second-order time integration errors are smaller than the fourth-order spatial errors. However, we discuss two fourth-order time-stepping methods in this section that can be used.

A simple fourth-order time integration technique that can be used for the fourth-order methods is standard fourth-order Runge Kutta (4RK). In this case, an implicit system (corresponding to the inversion of the discrete Laplacian in (4.1) must be solved at every sub-step. This can be done efficiently as described below using cyclic reduction and repeated penta-diagonal solves for every Fourier component. A necessary condition for stability of 4RK applied to the wide

fourth-order version of (4.5) with step size k is that all the eigenvalues λ must fit into the stability region of the method. Using a diagram of the stability region of 4RK (*e.g.*, in [5], p.109) and the eigenvalue limits (4.6) and (4.7) a fully discrete method using 4RK will be formally stable when

$$k \leq 2h \tag{4.8}$$

and

$$k \leq h^2/(10\nu). \tag{4.9}$$

Condition (4.8) is a CFL stability condition on the convective terms and is modified to $k \leq 2h/M$ when a Poiseuille flow with maximum velocity M is given.

By using an explicit method on the convection terms, we cannot get away from a stability restriction like (4.8) and in the computations described later, we assume a CFL condition is satisfied

$$k = \alpha h/M, \tag{4.10}$$

where M is the maximum velocity in the calculation. Under assumption (4.10) condition (4.9) coming from the diffusive terms will be violated for h sufficiently small. This “diffusive limit” occurs because the diffusive terms are stiff and we are trying to handle them with an explicit method. In many computations of interest, ν is quite small (corresponding to high Re) and this diffusive limit is not reached for moderate values of h . The scheme in [9] is designed for this regime. For problems where ν is not so small (like those in section 5), it is advantageous to handle the diffusive terms implicitly to avoid the restrictive condition (4.9). Our discretization allows all options for time-stepping.

In order to avoid the limitations of a diffusive stability limit we consider a hybrid semi-implicit fourth-order method. Fourth-order backward differencing (BDF) is used for the diffusive terms and the nonlinear terms are extrapolated using data at previous times steps. The method, which we denote as fourth-order semi-BDF (4SBDF) is given for the wide scheme as follows:

$$\begin{aligned} & \tilde{\Delta}_h \left(\frac{25}{12} \Psi^n - 4\Psi^{n-1} + 3\Psi^{n-2} - \frac{4}{3} \Psi^{n-3} + \frac{1}{4} \Psi^{n-4} \right) / k \\ & = \tilde{B}_h \Psi^n + (4N(\Psi^{n-1}) - 6N(\Psi^{n-2}) + 4N(\Psi^{n-3}) - N(\Psi^{n-4})), \end{aligned}$$

where a superscript is used to denote the time level and N is shorthand for the wide fourth-order difference calculation of the nonlinear convection terms. The linear implicit equations are solved efficiently using FFT reduction as described below. The values at the first three time steps Ψ^1 , Ψ^2 and Ψ^3 are initialized using three steps of 4RK starting from the initial data Ψ^0 . The method 4SBDF has the desirable property that the stability region of its explicit part contains part of the imaginary axis (so it will be asymptotically stable for the convection terms under a suitable CFL condition) and the stability region of implicit part contains the negative real axis (and so will handle the diffusion terms stably for any choice of time step k). This scheme and other combination implicit-explicit (IMEX) schemes are described in [2].

Both of the schemes above have linear, positive definite symmetric systems to solve at every time step or sub-step. In the periodic channel, this is done by taking a Fast Fourier Transform in the horizontal direction. This divides the original problem into banded problems (penta-diagonal for 4RK and septa-diagonal for 4SBDF) for each discrete Fourier component. The resulting operation count is $\mathcal{O}(N^2 \log N)$ for each time step which is near the optimal $\mathcal{O}(N^2)$.

5. Numerical Results

In this section, the fourth-order methods are tested on two examples to verify the accuracy and convergence rates of the methods. Both examples are in the 2D periodic channel geometry for which the methods were discussed in the previous section. The first test is for Navier-Stokes flow and the next for Boussinesq flow. The examples are discussed in detail below. Standard second-order methods are used to compute these example flows as well, and the results are compared to the fourth-order methods to evaluate their performance.

In a final example, the fourth-order methods are applied to a high Rayleigh number convection flow and are used to resolve the fine scale flow structures generated.

5.1. Perturbed Poiseuille Flow

Our first test case is a Navier-Stokes flow with initial data

$$\psi_0 = y^2(3 - 2y) + 16y^2(1 - y)^2 \sin(2\pi x)e^y/(2\pi),$$

where the term $y^2(3 - 2y)$ is Poiseuille flow and the other term represents a perturbation. The net flow through the channel is kept constant at 1 and flow is computed out to time 2. The viscosity ν is taken to be 0.01 and the maximum velocity observed in computations is 2, giving a Re of 200. This Reynolds number is well below the critical one for instabilities in Poiseuille flow [8] and so we expect the perturbation to decay in time, which is observed in the computations.

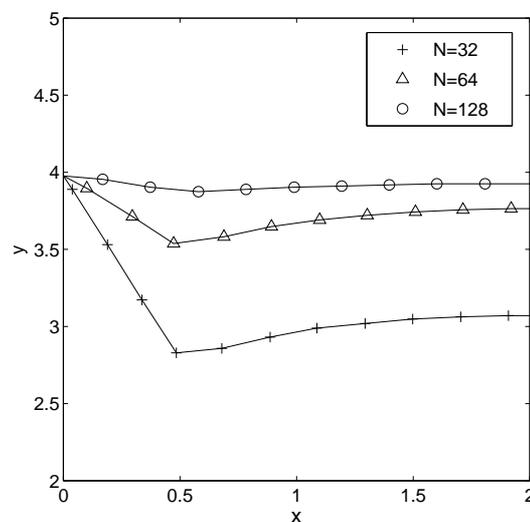


Fig. 1. Pointwise velocity convergence order estimates for the wide fourth-order 4SBDF approximation of the perturbed Poiseuille test case for $N=32$ (A), 64 (B) and 128 (C)

The fourth-order finite difference method proposed in section 4 is applied to this test problem. We computed with stable time steps $k = h/8$ using 4SBDF time stepping for $N = 1/h$ of 16, 32, 64, 128, and 256. The convergence estimates shown in Fig. 1 demonstrate the fourth-order pointwise velocity convergence of the method. Since the exact solution of our problem is not known, we define the estimated pointwise velocity error at the N grid as the difference to

the velocities at the next finest grid ($2N$), *i.e.*, $\|\mathbf{U}(N) - \mathbf{U}(2N)\|_\infty$, where the discrete velocities \mathbf{U} are computed using fourth-order differencing of the stream-function based on (2.2). The norm $\|\cdot\|_\infty$ denotes the maximum norm over all values on the discrete grid. If a very small time step $k = h/16$ is used 4RK calculations up to $N = 128$ can be done stably. Fourth-order velocity errors very similar to those obtained above using 4SBDF time stepping can be seen using 4RK.

The rate of convergence of the vorticity is not resolved in this calculation. At the finest grid for which the convergence rate was approximated ($N = 128$) convergence at less than fourth-order was observed. Third-order convergence is expected in vorticity due to numerical boundary layers from the results of the analysis of the model problem in section 3. A discussion of the size of the errors and a comparison with second-order methods is given in section 5.3 below.

5.2. Convection problem

We now turn to a test problem for the Boussinesq equations. The initial stream function is set to zero (no flow) and a smoothed horizontal interface is placed between regions of high temperature (1) and low temperature (0) initially

$$\theta_0(x, y) = \frac{\tanh((y - 0.3)/\delta) + \tanh(0.3/\delta)}{\tanh((1 - 0.3)/\delta) + \tanh(0.3/\delta)} + 4\epsilon y(1 - y) \sin(2\pi x). \quad (5.1)$$

The sin term in (5.1) is a perturbation to make something happen. The layer thickness δ is taken to be 0.1 and the perturbation size ϵ is taken to be 0.2. We take $\nu = 0.01$ and $\mu = 0.01$ (corresponding to a Rayleigh number of 10,000 and Prandtl number of 1) and calculate out to time 10. The perturbation in the temperature layer grows and periodic convection cells develop. The maximum velocity observed was about 0.4.

The fourth-order scheme was applied to this problem with $k = h/2$, using the temperature boundary condition (4.4). Fourth-order pointwise convergence in velocity is observed as in the example above and fourth-order temperature convergence is clearly seen in Fig. 2.

5.3. Comparison to second-order methods

When doing numerical flow computations, one generally has one of two aims: to minimally resolve some flow to get some practical information or to get high accurate solutions to use as benchmark solutions or to resolve some very delicate phenomena. We choose (somewhat arbitrarily) a criteria for minimal convergence that the estimated relative error of velocity is less than 5% of the maximum velocity in the computation. For the perturbed Poiseuille test case, where the maximum velocity is 2, the method has minimally converged at grid N when

$$\|\mathbf{U}(N) - \mathbf{U}(2N)\|_\infty \leq 0.1.$$

A similar condition for the convection problem, where the maximum velocity observed is 0.4, can be posed. The stringent convergence criteria for high accuracy is that the velocity error is less than 0.1%.

For the fourth-order schemes described to the problems above, minimal convergence was reached for the perturbed Poiseuille problem at $N = 16$ and for the convection problem at $N = 8$ (recall N is the number of grid lines in each direction). These results are compared to standard second-order centered stream-function methods, using Wilkes boundary conditions

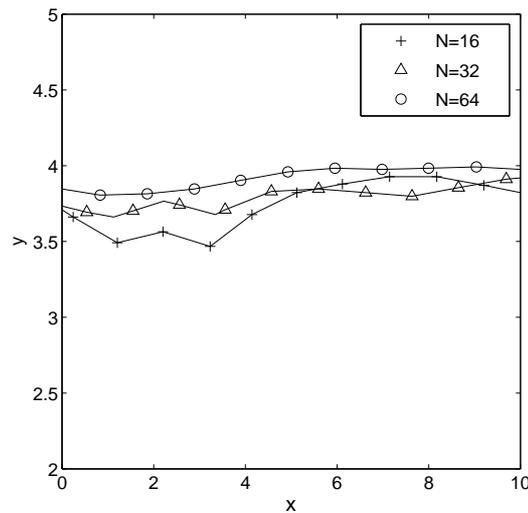


Fig. 2. Pointwise temperature convergence order estimates for the wide fourth-order 4SBDF approximation of the convection problem for $N=16$ (A), 32 (B), and 64 (C)

and a second-order hybrid time stepping technique CNLF (see [12] for details). The second-order methods reached minimal convergence for the perturbed Poiseuille problem at $N = 32$ and for the convection problem at $N = 16$. These results are consistent with the observations of other authors [6, 19] that minimal resolution can be obtained with half the number of grid lines in each direction if fourth-order is used instead of second-order differencing. Thus minimal resolution can be obtained with a quarter (one eighth for 3D computations) the number of grid points. What these and other authors fail to point out is that fourth-order methods involve more work for the same number of grid points. The amount of computational time to achieve minimal resolution is reduced by a factor of 2 for the perturbed Poiseuille calculation and 3/2 for the convection problem when fourth-order methods are used instead of second. This is not as significant a savings as one would expect from being able to use a quarter as many grid points. The storage savings that one hopes for by using a fourth-order method and less grid points are also offset by the extra storage required for the solution of the wider implicit systems in the case of the Navier-Stokes equations. Recall that a linear system (for the implicit handling of the Stokes operator) is solved at every time step for each discrete fourier component after a cyclic reduction of the problem in the periodic direction. Precomputing factorizations of these linear solvers requires storage proportional to $5N^2$ for second-order methods (pentadiagonal matrices) and $7N^2$ for fourth-order methods (septadiagonal matrices). Thus, using half as many grid lines for a fourth-order method instead of a second-order one results in a savings of 7/20 in storage and not 1/4. The point here is that the use of fourth-order methods to minimally resolve flows without delicate structure in 2D gives about a factor of two improvement over the use of second-order methods in computational time and storage. This improvement is not huge, but in many cases computational resources are pushed to the limit to resolve phenomena of physical interest, so any improvement in performance is welcome.

If we want to reach the more stringent convergence criteria of 0.1% error, the fourth-order methods will dramatically outperform second-order methods. To converge in this sense, the fourth-order methods require $N = 32$ for both the model computations and the second-order methods require $N = 128$. This results in a computational time reduction by factors of 35 for

the perturbed Poiseuille problem and 11 for the convection problem when fourth-order methods are used instead of second. This is a significant improvement.

5.4. A high Rayleigh number convection problem

In this section, our fourth-order method is applied to a more difficult convection problem to demonstrate its ability to capture complicated structure. As in section 5.2, we begin with zero flow and a perturbed temperature layer

$$\theta_0(x, y) = \frac{\tanh((y - 0.3)/\delta) + \tanh(0.3/\delta)}{\tanh((1 - 0.3)/\delta) + \tanh(0.3/\delta)} + 4\epsilon y(1 - y)(\sin(2\pi x) + \cos(6\pi x)), \quad (5.2)$$

where $\delta = 0.03$ and $\epsilon = 0.05$. In this example we use $\nu = 0.0001$ and $\mu = 0.0001$ (corresponding to a Rayleigh number of 10^8 and a Prandtl number of 1). It should be noted that real flows in this regime are inherently 3D, so this is a formal calculation to assess the numerical resolution of complex phenomena using the schemes presented in this paper. As in the flow of section 5.2, the perturbation in the temperature layer grows initially, but in this case the diffusion terms are too small to drive the flow to steady convection rolls and the onset to turbulent flow results. Gray scale plots of the temperature are shown in Fig. 3. These are generated using the fourth-order method and $N = 256$. The maximum velocity observed in the computation was about 1. We do not try to estimate pointwise errors or convergence rates for this calculation, since many more grid points would have to be used to reach the asymptotic regime of convergence. However, we have some confidence in the computational results since they respect the monotonicity of the maximum temperature value in time. Calculations with a standard second-order method and $N = 256$ cannot resolve the flow and exhibit oscillations (which lead to an increase in the maximum of the calculated temperature).

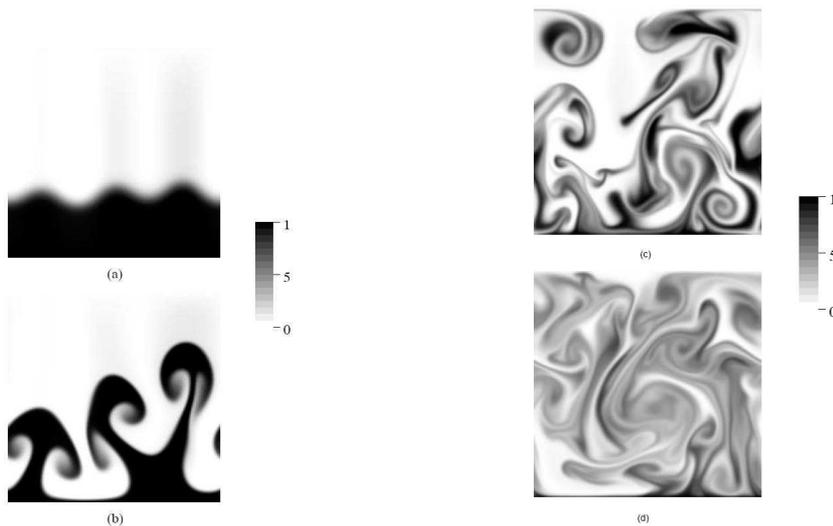


Fig. 3. Computed temperature values for the high Rayleigh number convection problem at times 1(a), 2.5(b), 4(c) and 5.5(d)

6. Summary

The authors have proposed a fourth-order finite difference method for the stream-function formulation of the Navier-Stokes and Boussinesq equations. Stream-function values outside the boundary were eliminated by appropriate extrapolation of the interior values. The stability and fourth-order pointwise velocity and temperature convergence of the methods has been verified numerically.

It was found that minimal accuracy could be obtained using fourth-order methods with half of the number of grid lines in each direction compared with second-order methods. This results in roughly a factor of 2 savings in computational time and storage with our implementations. To obtain highly accurate solutions, the fourth-order methods are much more efficient.

The advantages of the stream-function formulation are limited to 2D. It is possible to define a vector stream function that can describe 3D incompressible flows, but this approach has not received much attention. In simple counting terms, in 2D, primitive variable (velocity and pressure) formulations require three unknowns (two velocity components and pressure) compared to the single stream function. In 3D primitive variable methods have four unknowns, while the stream-function approach has three components. The large advantage of stream-function methods is lost, but the difficulties remain such as higher order derivatives in the formulation. In addition, the 3D stream-function approach involves a non-local constraint that is difficult to deal with when computing in domains with boundaries. Some effort has been made to develop higher order finite difference methods for primitive variable formulations, see [21] and references therein. The exact nature and order of the errors from these schemes could be understood using the same asymptotic error analysis used in this paper.

A limitation of the methods proposed in this paper is that they are only derived for rectangular domains. These methods could be easily extended to domains with radial symmetry and conformally mapped domains (as in [1,9,12]) but an extension to more general domains remains to be done for the methods to be useful for many practical applications. Two further studies of interest are: a comparison of our wide methods to compact methods, and an investigation of efficient solvers for implicit discretizations of our method.

Acknowledgements. The first author would like to acknowledge funding from NSF under grants DMS-0713670 and ACI-0204932. The second author would like to acknowledge funding from NSERC Canada that supported this work.

References

- [1] C.R. Anderson and M.B. Reider, A high order explicit method for the computation of flow about a circular cylinder, *J. Comput Phys*, **125** (1996), 207-224.
- [2] U. Ascher, S. Ruuth and B.T.R. Wetton, Implicit-explicit methods for time-dependent partial-differential equations, *SIAM J. Numer. Anal.*, **32** (1995), 797-823.
- [3] M. Ben-Artzi M, J.P. Croisille, D. Fishelov D and S. Trachtenberg, A pure-compact scheme for the streamfunction formulation of Navier-Stokes equations, *J. Comput Phys*, **205** (2005), 640-644.
- [4] A.M. Berger, J.M. Solomon, M. ciment, S.H. Leventhal and B.C. Weinberg. Generalized OCI schemes for boundary layer problems, *Math. Comput.*, **35** (1980), 695-731.
- [5] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer Verlag, Berlin (1988).

- [6] Y.-C. Chang, Comparison of Finite Difference and the Pseudo-Spectral Approximations for Hyperbolic Wquations and Implementation Analysis on Parallel Computer CM-2, Ph.D. Thesis, UCLA (1992).
- [7] S.R.C. Dennis and J.D. Hudson, Compact h^4 Finite-difference approximations to operators of Navier-Stokes type, *J. Comput Phys*, **85** (1989), 390-416.
- [8] P.G. Drazin and W.D. Reid, Hydrodynamic Stability, Cambridge University Press, Cambridge (1981).
- [9] W. E and J.G. Liu, Essentially compact schemes for unsteady viscous incompressible flows, *J. Comput Phys*, **126** (1996), 122-138.
- [10] M.M. Gupta, High accuracy solutions of Navier-Stokes equations, *J. Comput Phys*, **93** (1991), 343-359.
- [11] R.S. Hirsh, High order accurate difference solutions of fluid mechanics problems by a compact differencing technique, *J. Comput Phys*, **19** (1975), 90-109.
- [12] T.Y. Hou and B.T.R. Wetton, Convergence of a finite difference scheme for the Navier-Stokes equations using vorticity boundary conditions, *SIAM J. Numer. Anal.*, **29**, 615-639 (1992).
- [13] T.Y. Hou and B.T.R. Wetton, Second-order convergence of a projection scheme for the incompressible Navier-Stokes equations with boundaries, *SIAM J. Numer. Anal.*, **30** (1993), 609-629.
- [14] H. Huang, Incompressible Viscous Flow in Tubes with Occlusions, Ph.D. Thesis, University of British Columbia (1991).
- [15] H.B. Keller, Numerical Solution of Two-Point Boundary Value Problems, SIAM, Philadelphia (1976).
- [16] H.B. Keller and H. Takami, Numerical studies of steady viscous flow about cylinders, in *Numerical Solutions of Nonlinear Differential Equations* (D. Greenspan, ed.) Wiley, New York (1966).
- [17] H.-O. Kreiss, Comparison of accurate methods for the integration of hyperbolic equations, *Tellus XXIV*, **3** (1972), 199-215.
- [18] J. Lowengrub, M. Shelley, and B. Merriman, High order accurate and efficient methods in vorticity formulation for the Euler equations, *SIAM J. Sci. Comput.*, **14** (1993), 1107-1142.
- [19] S.A. Orszag and M. Israeli, Numerical simulation of viscous incompressible flows, *Ann. Rev. of Fluid Mech*, **6** (1974), 281-318.
- [20] R. Peyret and T.D. Taylor, Computational Methods for Fluid Flow, Springer Verlag, New York (1983).
- [21] J. C. Strikwerda, High-order-accurate schemes for incompressible viscous flow, *Int. J. Numer. Meth. Fl.*, **24** (1997), 715-734.
- [22] B.T.R. Wetton, Analysis of the spatial error for a class of finite difference methods for viscous incompressible flow, *SIAM J. Number. Anal.*, **34** (1997), 723-755.