Fast Algorithms for Solving High-Order Finite Element Equations for Incompressible Flow

L. Ridgway Scott* Andrew Ilin† Ralph W. Metcalfe‡ Babak Bagheri†

Abstract

We discuss high-order finite element methods to simulate the flow of incompressible viscous fluids. We focus on algorithms for solving the related algebraic equations efficiently using the iterated penalty method for resolving the incompressibility constraint. We show that direct methods may be a suitable choice for solving the resulting linear equations, at least in the two-dimensional case.

Key words: finite element method, incompressible viscous fluids, iterated penalty method, Navier-Stokes problem.

AMS subject classifications: 65M60, 65F05, 65F10.

1 Introduction

High-order finite element methods provide very accurate simulations of the flow of incompressible viscous fluids [7]. Here we focus on various algorithms for solving the related algebraic equations efficiently in the context of Newtonian fluids, that is, ones solving the Navier-Stokes equations [8] which appear subsequently in equation (9).

We discuss the iterated penalty method for resolving the incompressibility constraint [4]. This method allows one to replace an indefinite linear-algebraic system with a positive-definite linear system in many cases. One key observation we make is that it is necessary in some cases to do only a very small number of penalty iterations after an initial start-up phase. This leads to a very efficient algorithm. We also extend the algorithm and analysis in [4] to the case of inhomogeneous boundary conditions.

Another observation we make is that direct methods may be a suitable choice for solving the resulting linear equations, at least in the two-dimensional case. We show that the relative fill-in due to Gaussian elimination is quite small for high-order methods, and decreases with increasing degree. This allows one to solve non-symmetric problems with comparable efficiency to the symmetric case, allowing complex time-stepping schemes to be used.

2 Mixed method formulation

In all of the cases considered here, the formulation will reduce to solving a general mixed method of the form

$$(1)\qquad \begin{aligned} a(u_h,v)+b(v,p_h) &= F(v) \quad \forall v\in V_h\\ b(u_h,q) &= G(q) \quad \forall q\in \Pi_h, \end{aligned}$$

where $F \in V'$ and $G \in \Pi'$ (the "primes" indicate dual spaces [4]). Here V and Π are two Hilbert spaces with subspaces $V_h \subset V$ and $\Pi_h \subset \Pi$, respectively; $u_h \in V_h$ and $p_h \in \Pi_h$ are unknowns, $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are bilinear forms, which satisfy conditions introduced later.

The main new ingredient in the variational formulation of mixed methods with inhomogeneous boundary conditions is that the term G is not zero. In [4], complete details are given only in the case that $G \equiv 0$.

The variational problem (1) is only a slight generalization of (11.1.1) in [4], which had $b(u_h,q) = (g,q)_\Pi$ as the second equation, if we make the translation $g = P_{\Pi_h} G$ where P_{Π_h} denotes the Riesz representation of G in Π_h, that is

$$(2)\qquad (P_{\Pi_h}G,q)_\Pi = G(q) \quad \forall q\in\Pi_h\,.$$

We will assume that $\mathcal{D} : V \to \Pi$ is continuous and that

$$(3)\qquad b(v,p) = (\mathcal{D}v,p)_\Pi \quad \forall p,$$

where $(\cdot,\cdot)_\Pi$ denotes the inner product in Π. Note that the second equation in (1) says that

$$(4)\qquad P_{\Pi_h}\mathcal{D}u_h = P_{\Pi_h}G$$

*Department of Mathematics and Texas Center for Advanced Molecular Computation, University of Houston

†Texas Center for Advanced Molecular Computation, University of Houston

‡Department of Mechanical Engineering, University of Houston

ICOSAHOM'95: Proceedings of the Third International Conference on Spectral and High Order Methods. ©1996 Houston Journal of Mathematics, University of Houston.

where we also use $P_{\Pi_h} g$ to denote the Π-projection of $g \in \Pi$ onto Π_h.

We assume that the bilinear forms satisfy the continuity conditions

$$(5) \qquad \begin{aligned} a(u,v) &\leq C_a \|u\|_V \|v\|_V & \forall u,v \in V \\ b(v,p) &\leq C_b \|v\|_V \|p\|_\Pi & \forall v \in V, p \in \Pi \end{aligned}$$

and the coercivity conditions

$$(6) \qquad \begin{aligned} \alpha \|v\|_V^2 &\leq a(v,v) \quad \forall v \in Z \cup Z_h \\ \beta \|p\|_\Pi &\leq \sup_{v \in V_h} \frac{b(v,p)}{\|v\|_V} \quad \forall p \in \Pi_h. \end{aligned}$$

Here Z and Z_h are defined by

$$(7) \qquad Z = \{v \in V \ : \ b(v,q) = 0 \quad \forall q \in \Pi\}$$

and

$$(8) \qquad Z_h = \{v \in V_h \ : \ b(v,q) = 0 \quad \forall q \in \Pi_h\}$$

respectively.

3 The Navier-Stokes equations

The Navier-Stokes equations for the flow of a viscous, incompressible, Newtonian fluid can be written

$$(9) \qquad \begin{aligned} -\Delta \underset{\sim}{u} + \nabla p &= -R\left(\underset{\sim}{u} \cdot \nabla \underset{\sim}{u} + \underset{\sim}{u}_t\right) \\ \operatorname{div} \underset{\sim}{u} &= 0. \end{aligned}$$

for $\underset{\sim}{x} \in \Omega \subset \mathbb{R}^d$, $t \in [0,T]$, where $\underset{\sim}{u}(t,\underset{\sim}{x})$ denotes the fluid velocity, $p(t,\underset{\sim}{x})$ denotes the pressure, and R denotes the Reynolds number [8]. These equations must be supplemented by appropriate boundary conditions, such as the Dirichlet boundary conditions, $\underset{\sim}{u} = \underset{\sim}{\gamma}$ on $\partial\Omega$, and initial condition $\underset{\sim}{u}(0,\underset{\sim}{x}) = \underset{\sim}{u}_0(\underset{\sim}{x})$.

A complete variational formulation of (9) takes the form: Find $\underset{\sim}{u}$ such that $\underset{\sim}{u}(t,\cdot) - \underset{\sim}{\gamma} \in V$ and $p(t,\cdot) \in \Pi$ such that $\forall t \in [0,T]$

$$(10) \qquad \begin{aligned} a\left(\underset{\sim}{u},\underset{\sim}{v}\right) + b\left(\underset{\sim}{v},p\right) + & \\ R\left(c\left(\underset{\sim}{u},\underset{\sim}{u},\underset{\sim}{v}\right) + \left(\underset{\sim}{u}_t,\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}\right) &= 0 \quad \forall \underset{\sim}{v} \in V, \\ b(\underset{\sim}{u},q) &= 0 \quad \forall q \in \Pi, \end{aligned}$$

where e.g. $a(\cdot,\cdot)$, $b(\cdot,\cdot)$ and $c(\cdot,\cdot,\cdot)$ are given by

$$(11) \qquad a(\underset{\sim}{u},\underset{\sim}{v}) := \int_\Omega \sum_{i,j=1}^{d} \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{x_j}\, d\underset{\sim}{x},$$

$$(12) \qquad b(\underset{\sim}{v},q) := -\int_\Omega \sum_{i=1}^{d} \frac{\partial v_i}{\partial x_i} q\, d\underset{\sim}{x},$$

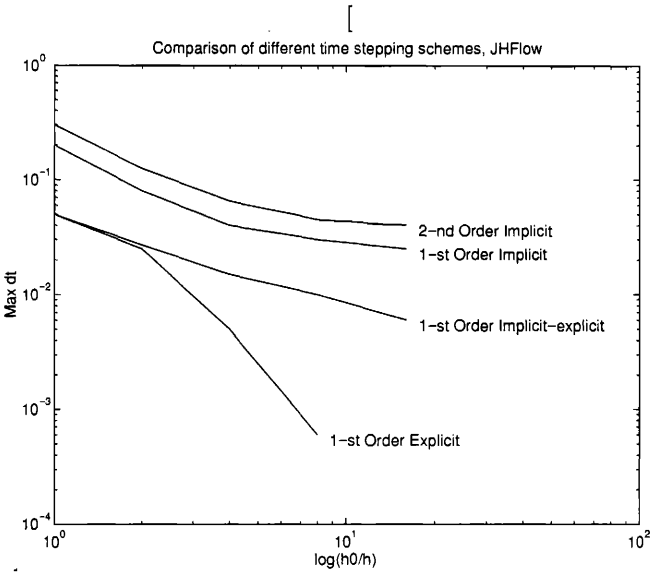


Figure 1: Empirically determined maximum time step size for different time stepping schemes for Jeffrey-Hamel flow for Reynolds number $R = 240$. The explicit scheme is (14), the implicit-explicit scheme is (15), the implicit scheme is (21). To utilize an implicit scheme for nonlinear equation, the fixed-point iterative method has been used.

$$(13) \qquad c(\underset{\sim}{u},\underset{\sim}{v},\underset{\sim}{w}) := \int_\Omega \left(\underset{\sim}{u} \cdot \nabla \underset{\sim}{v}\right) \cdot \underset{\sim}{w}\, d\underset{\sim}{x},$$

and $(\cdot,\cdot)_{\underset{\sim}{L}^2}$ denotes the $L^2(\Omega)^d$-inner-product, $\underset{\sim}{V} = \mathring{H}^1(\Omega)^d$ and $\Pi = \left\{q \in L^2(\Omega) \ : \ \int_\Omega q\, d\underset{\sim}{x} = 0\right\}$. For more details about notation see [4].

One of the simplest first-order time-stepping schemes is

$$(14) \qquad \begin{aligned} a\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right) + b\left(\underset{\sim}{v},p^\ell\right) \quad & + R\, c\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right) \\ & + \tfrac{R}{\Delta t}\left(\underset{\sim}{u}^\ell - \underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2} = 0, \\ b\left(\underset{\sim}{u}^\ell,q\right) \quad & = 0, \end{aligned}$$

where, here and below, $\underset{\sim}{v}$ varies over all V (or V_h) and q varies over all Π (or Π_h) and Δt denotes the time-step size. The scheme is only explicit with respect to the velocity. There is an equation for the pressure and the zero-divergence constraint. The major drawback of explicit schemes such as (14) is the severe limitation on the time step as depicted in Figure 1.

Time-stepping schemes that are implicit with respect to the linear terms and explicit with respect to the nonlinear terms can be more efficient even though a linear system involving $a(\cdot,\cdot)$ form must be solved. The simplest of these is

$$\begin{aligned} a\left(\underset{\sim}{u}^\ell,\underset{\sim}{v}\right)+b\left(\underset{\sim}{v},p^\ell\right) \quad & +Rc\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right) \\ & +\tfrac{R}{\Delta t}\left(\underset{\sim}{u}^\ell-\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}=0, \\ b\left(\underset{\sim}{u}^\ell,q\right) \quad & =0. \end{aligned} \tag{15}$$

A more complex time-stepping scheme could be based on the variational equations

$$\begin{aligned} a\left(\underset{\sim}{u}^\ell,\underset{\sim}{v}\right)+b\left(\underset{\sim}{v},p^\ell\right) \quad & +Rc\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell},\underset{\sim}{v}\right) \\ & +\tfrac{R}{\Delta t}\left(\underset{\sim}{u}^\ell-\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}=0, \\ b\left(\underset{\sim}{u}^\ell,q\right) \quad & =0, \end{aligned} \tag{16}$$

in which the nonlinear term has been approximated in such a way that the linear algebraic problem changes at each time step. It takes the form (1) with a form $\tilde{a}(\cdot,\cdot)$ given by

$$\tilde{a}\left(\underset{\sim}{u},\underset{\sim}{v};\underset{\sim}{U}\right)=a\left(\underset{\sim}{u},\underset{\sim}{v}\right)+ \int_\Omega\left(\frac{R}{\Delta t}\underset{\sim}{u}\cdot\underset{\sim}{v}+\underset{\sim}{U}\cdot\nabla\underset{\sim}{u}\cdot\underset{\sim}{v}\right)d\underset{\sim}{x} \tag{17}$$

where $\underset{\sim}{U}=R\underset{\sim}{u}^n$ arises from linearizing the nonlinear term.

Even though the addition of the $\underset{\sim}{U}$ term makes it non-symmetric, $\tilde{a}(\cdot,\cdot)$ will be coercive for $\frac{R}{\Delta t}$ sufficiently large. In fact, when $\operatorname{div}\underset{\sim}{U}\equiv 0$ then integrating by parts yields

$$\begin{aligned} \int_\Omega \underset{\sim}{U}\cdot\nabla\underset{\sim}{v}\cdot\underset{\sim}{v}\,d\underset{\sim}{x} &= \int_\Omega\sum_{i,j=1}^d U_i\frac{\partial v_j}{\partial x_i}v_j\,d\underset{\sim}{x} = \\ &\int_\Omega\sum_{i,j=1}^d U_i\frac{1}{2}\left(\frac{\partial v_j^2}{\partial x_i}\right)d\underset{\sim}{x} = \\ &-\frac{1}{2}\int_\Omega\sum_{i,j=1}^d\frac{\partial U_i}{\partial x_i}v_j^2\,d\underset{\sim}{x}=0 \end{aligned} \tag{18}$$

for all $v\in V$ so that

$$\alpha\left\|\underset{\sim}{v}\right\|_V^2\le\tilde{a}(\underset{\sim}{v},\underset{\sim}{v})\quad\forall\underset{\sim}{v}\in V \tag{19}$$

for the same choice of $\alpha>0$ as before. Of course, $\tilde{a}(\cdot,\cdot)$ is continuous:

$$\tilde{a}(\underset{\sim}{v},\underset{\sim}{w})\le C_a\left\|\underset{\sim}{v}\right\|_V\left\|\underset{\sim}{w}\right\|_V\quad\forall\underset{\sim}{v},\underset{\sim}{w}\in V \tag{20}$$

but now C_a depends on both $\frac{R}{\Delta t}$ and $\underset{\sim}{U}$.

The main disadvantage of the time-stepping scheme (16) is that it has a time-dependent system of algebraic equations. To avoid this time-dependence and to keep implicitness of the non-linear term approximation, fixed-point iterations have been introduced in the following variational form

$$\begin{aligned} a\left(\underset{\sim}{u}^{\ell,m},\underset{\sim}{v}\right)+b\left(\underset{\sim}{v},p^{\ell,m}\right) \quad & +Rc\left(\underset{\sim}{u}^{\ell,m-1},\underset{\sim}{u}^{\ell,m-1},\underset{\sim}{v}\right) \\ & +\tfrac{R}{\Delta t}\left(\underset{\sim}{u}^{\ell,m}-\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}=0, \\ b\left(\underset{\sim}{u}^{\ell,m},q\right) \quad & =0, \end{aligned} \tag{21}$$

where $m=1,2,...,M$ is the fixed-point iteration index. We write $\underset{\sim}{u}^\ell$ for $\underset{\sim}{u}^{\ell,M}$. The following convergence criterion has been used in our calculations for the fixed-point method:

$$\left\|\underset{\sim}{u}^{\ell,m}-\underset{\sim}{u}^{\ell,m-1}\right\|\le\varepsilon_{\mathrm{fp}}\left\|\underset{\sim}{u}^{\ell,m}\right\|. \tag{22}$$

The simplest initial guess for fixed-point iterations is $\underset{\sim}{u}^{\ell,0}=\underset{\sim}{u}^{\ell-1}$. To reduce the number of fixed-point iterations, the initial guess was calculated using linear interpolation of the solution at two previous time steps:

$$\underset{\sim}{u}^{\ell,0}=2\underset{\sim}{u}^{\ell-1}-\underset{\sim}{u}^{\ell-2}. \tag{23}$$

In practice, higher order time-stepping schemes can be used, including the second-order approximation of the time derivative ([6]):

$$\frac{\partial\underset{\sim}{u}}{\partial t}(t^\ell)\approx\frac{3\underset{\sim}{u}^\ell-4\underset{\sim}{u}^{\ell-1}+\underset{\sim}{u}^{\ell-2}}{2\Delta t}, \tag{24}$$

but the main issues related to solving the resulting linear equations remain the same.

All time-stepping schemes may be written as a problem for $\left(\underset{\sim}{u}^\ell,p^\ell\right)$ which is nearly of the form (1), for example, equation (15) may now be written as

$$\begin{aligned} \tilde{a}\left(\underset{\sim}{u}^\ell,\underset{\sim}{v}\right)+b\left(\underset{\sim}{v},p^\ell\right) \quad & =-Rc\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right) \\ & \;\;+\tau\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}, \\ b\left(\underset{\sim}{u}^\ell,q\right) \quad & =0 \end{aligned} \tag{25}$$

with the more general form $\tilde{a}(\cdot,\cdot)$, namely

$$\tilde{a}(\underset{\sim}{u},\underset{\sim}{v}):=a(\underset{\sim}{u},\underset{\sim}{v})+\tau\left(\underset{\sim}{u},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2} \tag{26}$$

where the constant $\tau=R/\Delta t$ for the first-order time-stepping scheme and $\tau=1.5R/\Delta t$ for the second-order time-stepping scheme. Numerical experiments will be presented subsequently for such a problem. Note that the linear algebraic problem to be solved at each time step is the same.

Note that we have $\underset{\sim}{u}^\ell=\underset{\sim}{\gamma}$ on $\partial\Omega$, that is, $\underset{\sim}{u}^\ell=\underset{\sim}{u}+\underset{\sim}{\gamma}$ where $\underset{\sim}{u}\in V$. The variational problem (25) for $\underset{\sim}{u}^\ell$ can then be written: Find $\underset{\sim}{u}\in V$ and $p\in\Pi$ such that

$$\begin{gathered} \tilde{a}\left(\underset{\sim}{u},\underset{\sim}{v}\right)+b\left(\underset{\sim}{v},p\right)=-\tilde{a}\left(\underset{\sim}{\gamma},\underset{\sim}{v}\right) \\ -Rc\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)+\tau\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{\underset{\sim}{L}^2}\quad\forall\underset{\sim}{v}\in V \\ b\left(\underset{\sim}{u},q\right)=-b\left(\underset{\sim}{\gamma},q\right)\quad\forall q\in\Pi. \end{gathered} \tag{27}$$

This is of the form (1) with

$$F(\underset{\sim}{v}) := -\tilde{a}\left(\underset{\sim}{\gamma}, \underset{\sim}{v}\right) - Rc\left(\underset{\sim}{u}^{\ell-1}, \underset{\sim}{u}^{\ell-1}, \underset{\sim}{v}\right) + \tau\left(\underset{\sim}{u}^{\ell-1}, \underset{\sim}{v}\right)_{L^2} \quad \forall \underset{\sim}{v} \in V$$
$$G(q) := -b\left(\underset{\sim}{\gamma}, q\right) \quad \forall q \in \Pi. \tag{28}$$

Note that the inhomogeneous boundary data $\underset{\sim}{\gamma}$ appears in both right-hand sides, F and G. Thus we are forced to deal with a nonzero G.

A finite element discretization of (10), (15) or (27) is obtained by replacing V by V_h and Π by Π_h satisfying (6). Since the notation is identical if we let V denote either V or V_h (and similarly with Π) we drop all h subscripts from now on.

4 Iterated penalty method

Consider a general mixed method of the form (1). Let $\rho' \in \mathbb{R}$ and $\rho > 0$. The iterated penalty method (IPM) defines $u^n \in V$ and p^n by

$$\begin{aligned} a(u^n, v) + \rho'(\mathcal{D}u^n, \mathcal{D}v)_\Pi \\ = F(v) - b(v, p^n) + \rho' G(\mathcal{D}v) \quad \forall v \in V \\ p^{n+1} = p^n + \rho\left(\mathcal{D}u^n - P_{\Pi_h}G\right) \end{aligned} \tag{29}$$

where $P_{\Pi_h}G$ is defined in (2). Recall also that (4) says that $P_{\Pi_h}\mathcal{D}u = P_{\Pi_h}G$.

The algorithm does not require $P_{\Pi_h}G$ to be computed, only

$$b(v, P_{\Pi_h}G) = (\mathcal{D}v, P_{\Pi_h}G)_\Pi = G(P_{\Pi_h}\mathcal{D}v) \tag{30}$$

for $v \in V$. Suppose that $G(q) := -b(\gamma, q)$. Then

$$G(P_{\Pi_h}\mathcal{D}v) = -b(\gamma, P_{\Pi_h}\mathcal{D}v) = -(\mathcal{D}\gamma, P_{\Pi_h}\mathcal{D}v)_\Pi. \tag{31}$$

If further $\Pi = \mathcal{D}V$, then

$$b(v, P_{\Pi_h}G) = G(P_{\Pi_h}\mathcal{D}v) = -(\mathcal{D}\gamma, \mathcal{D}v)_\Pi \tag{32}$$

for $v \in V$.

One key point of the iterated penalty method is that the system of equations represented by the first equation in (29) for u^n, namely

$$\begin{aligned} a(u^n, v) + \rho'(\mathcal{D}u^n, \mathcal{D}v)_\Pi \\ = F(v) - b(v, p^n) + \rho' G(\mathcal{D}v) \quad \forall v \in V, \end{aligned} \tag{33}$$

will be symmetric if $a(\cdot,\cdot)$ is symmetric, and it will be positive definite if $a(\cdot,\cdot)$ is coercive and $\rho' > 0$.

Suppose that $\Pi = \mathcal{D}V$. Then since $G(q) := -b(\gamma, q)$,

$$\begin{aligned} (P_{\Pi_h}G, q)_\Pi = G(q) = -b(\gamma, q) = \\ -(\mathcal{D}\gamma, q)_\Pi = -(P_{\Pi_h}\mathcal{D}\gamma, q)_\Pi \quad \forall q \in \Pi, \end{aligned} \tag{34}$$

that is, $P_{\Pi_h}G = -P_{\Pi_h}\mathcal{D}\gamma$. Then

$$p^{n+1} = p^n + \rho P_{\Pi_h}\mathcal{D}(u^n + \gamma) \tag{35}$$

since $P_{\Pi_h}\mathcal{D}u^n = \mathcal{D}u^n$.

If we begin with $p^0 = 0$ then, for all $n > 0$,

$$p^n = \rho P_{\Pi_h}\mathcal{D}\sum_{i=0}^{n}\left(u^i + \gamma\right) = P_{\Pi_h}\mathcal{D}w^n, \tag{36}$$

where

$$w^n := \rho\sum_{i=0}^{n}\left(u^i + \gamma\right). \tag{37}$$

Note that

$$\begin{aligned} b(v, p^n) = (\mathcal{D}v, p^n)_\Pi = \\ (\mathcal{D}v, P_{\Pi_h}\mathcal{D}w^n)_\Pi = (\mathcal{D}v, \mathcal{D}w^n)_\Pi \end{aligned} \tag{38}$$

since $P_{\Pi_h}\mathcal{D}v = \mathcal{D}v$.

Thus, the iterated penalty method implicitly becomes

$$\begin{aligned} a(u^n, v) + \rho'(\mathcal{D}u^n, \mathcal{D}v)_\Pi = F(v) \\ - (\mathcal{D}v, \mathcal{D}w^n)_\Pi - \rho'(\mathcal{D}v, \mathcal{D}\gamma)_\Pi \quad \forall v \in V \\ w^{n+1} = w^n + \rho\left(u^n + \gamma\right). \end{aligned} \tag{39}$$

In the case $\rho' = \rho$ this simplifies to

$$\begin{aligned} a(u^n, v) + \rho(\mathcal{D}u^n, \mathcal{D}v)_\Pi = F(v) \\ - (\mathcal{D}v, \mathcal{D}(w^n + \rho\gamma))_\Pi \quad \forall v \in V \\ w^{n+1} = w^n + \rho\left(u^n + \gamma\right). \end{aligned} \tag{40}$$

Thus we see that the introduction of inhomogeneous boundary conditions does not lead to a dramatic change to the formulation of the iterated penalty method. The main difference is that the "pressure" term

$$p = P_{\Pi_h}\mathcal{D}w, \tag{41}$$

where $w := w^n$ for the value of n at which the iteration is terminated, cannot be computed directly since $w \notin V$. That is, $\mathcal{D}w \notin \Pi$. On the other hand, p is not needed at all for the computation of u and can be computed only as required. For example, in a time-stepping scheme it need not be computed every time step.

The "pressure" term can be calculated in various ways. It satisfies the system

$$(p, \mathcal{D}v)_\Pi = (\mathcal{D}w, \mathcal{D}v)_\Pi \quad \forall v \in V. \tag{42}$$

We can write $p = \mathcal{D}z$ for various $z \in V$, with z satisfying the under-determined system

$$(\mathcal{D}z, \mathcal{D}v)_\Pi = (\mathcal{D}w, \mathcal{D}v)_\Pi \quad \forall v \in V. \tag{43}$$

Several techniques could be used to specify z, but one simple one is to use the system (1), namely, to find $z \in V$ and $r \in \Pi$ such that

$$(44)\qquad \begin{aligned} a(z,v)+b(v,r) &= 0 \quad \forall v \in V \\ b(z,q) &= (\mathcal{D}w,q)_\Pi \quad \forall q \in \Pi \end{aligned}$$

which is (1) with $F \equiv 0$ and $G(q) := (\mathcal{D}w,q)_\Pi$.

The iterated penalty method can be used to solve (44), yielding an algorithm of the form

$$(45)\qquad \begin{gathered} a(z^n,v)+\rho\,(\mathcal{D}z^n,\mathcal{D}v)_\Pi = \\ -\left(\mathcal{D}v,\mathcal{D}(\zeta^n-\rho w)\right)_\Pi \quad \forall v \in V \\ \zeta^{n+1}=\zeta^n+\rho\,(z^n-w) \end{gathered}$$

in the case $\rho'=\rho$. This involves inverting the same algebraic system as in (40), so very little extra work or storage is involved.

Note that w^n in (40) as well as ζ^n in (45) do not converge to any finite functions and they might have arbitrarily large values. On the other hand, the iterative penalty method does not require calculation of those variables explicitly since the first equations in (40) and in (45) use $(\mathcal{D}w^n,\mathcal{D}v)$ and $(\mathcal{D}\zeta^n,\mathcal{D}v)$. To avoid the calculation of variables with large entries and to reduce floating-point error, it is useful to rewrite the second equation in (40) and (45) in terms of $(\mathcal{D}w^n,\mathcal{D}v)$ and $(\mathcal{D}\zeta^n,\mathcal{D}v)$, which can be calculated with less error:

$$(46)\qquad \begin{gathered} a(u^n,v)+\rho\,(\mathcal{D}u^n,\mathcal{D}v)_\Pi = F(v) \\ -\left(\mathcal{D}v,\mathcal{D}(w^n+\rho\gamma)\right)_\Pi \quad \forall v \in V \\ (\mathcal{D}v,\mathcal{D}w^{n+1})=(\mathcal{D}v,\mathcal{D}w^n)+\rho(\mathcal{D}v,\mathcal{D}(u^n+\gamma)). \end{gathered}$$

$$(47)\qquad \begin{gathered} a(z^n,v)+\rho\,(\mathcal{D}z^n,\mathcal{D}v)_\Pi = \\ -\left(\mathcal{D}v,\mathcal{D}(\zeta^n-\rho w)\right)_\Pi \quad \forall v \in V \\ (\mathcal{D}v,\mathcal{D}\zeta^{n+1})=(\mathcal{D}v,\mathcal{D}\zeta^n)+\rho(\mathcal{D}v,\mathcal{D}(z^n-w)). \end{gathered}$$

The variables $\mathcal{D}w^n$ and $\mathcal{D}\zeta^n$ are incremented directly in our implementation.

5 Application to Navier-Stokes

The iterated penalty method (39) (with $\rho'=\rho$) for (15) takes the form

$$(48)\qquad \begin{gathered} \tilde a\left(\underset{\sim}{u}^{\ell,n},\underset{\sim}{v}\right)+\rho\left(\operatorname{div}\underset{\sim}{u}^{\ell,n},\operatorname{div}\underset{\sim}{v}\right)_{L^2} \\ = -a\left(\underset{\sim}{\gamma},\underset{\sim}{v}\right)-R\,c\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)+\tau\left(\underset{\sim}{u}^{\ell-1},\underset{\sim}{v}\right)_{L^2} \\ -\left(\operatorname{div}\left(\underset{\sim}{w}^{\ell,n}+\rho\underset{\sim}{\gamma}\right),\operatorname{div}\underset{\sim}{v}\right)_{L^2} \quad \forall \underset{\sim}{v}\in \underset{\sim}{V} \\ \underset{\sim}{w}^{\ell,n+1}=\underset{\sim}{w}^{\ell,n}+\rho\left(\underset{\sim}{u}^{\ell,n}+\underset{\sim}{\gamma}\right) \end{gathered}$$

where either $p^{\ell,0}=0$ (i.e. $\underset{\sim}{w}^{\ell,0}=0$) or $\underset{\sim}{w}^{\ell,0}=\underset{\sim}{w}^{\ell-1,N}$ where N is the final value of n at time-step $\ell-1$. If for some reason $p^\ell=p^{\ell,n}=-P_\Pi \operatorname{div}\underset{\sim}{w}^{\ell,n}$ were desired, it could be computed separately.

For example, algorithm (40) could be used to compute

$$(49)\qquad \begin{gathered} \tilde a\left(\underset{\sim}{z}^n,\underset{\sim}{v}\right)+\rho\left(\operatorname{div}\underset{\sim}{z}^n,\operatorname{div}\underset{\sim}{v}\right)_{L^2} = \\ -\left(\operatorname{div}\left(\underset{\sim}{\zeta}^n-\rho\underset{\sim}{w}^{\ell,n}\right),\operatorname{div}\underset{\sim}{v}\right)_{L^2} \quad \forall \underset{\sim}{v}\in \underset{\sim}{V} \\ \underset{\sim}{\zeta}^{n+1}=\underset{\sim}{\zeta}^n+\rho\left(\underset{\sim}{z}^n-\underset{\sim}{w}^{\ell,n}\right) \end{gathered}$$

starting with, say, $\underset{\sim}{\zeta}^0\equiv 0$. Then $\underset{\sim}{z}^n$ will converge to $\underset{\sim}{z}\in\underset{\sim}{V}$ satisfying $\operatorname{div}\underset{\sim}{z}=P_\Pi \operatorname{div}\underset{\sim}{w}^{\ell,n}=-p^{\ell,n}$. Note that (49) requires the same system to be solved as for computing $\underset{\sim}{u}^{\ell,n}$ in (48), so very little extra code or data-storage is required.

The potential difficulty caused by having inhomogeneous boundary data can be seen for high-order finite elements. For simplicity, consider the two-dimensional case ($d=2$). Let W_h^k denote piecewise polynomials of degree k on a triangular mesh, and let V_h^k denote the subspace of W_h^k consisting of functions that vanish on the boundary. Let

$$(50)\qquad \underset{\sim}{V}_h = V_h^k \times V_h^k,$$

and let $\Pi_h=\operatorname{div}\underset{\sim}{V}_h$. Then each $q\in\Pi_h$ is constrained at all boundary singular [4] vertices, σ_i, in the mesh. On the other hand, inhomogeneous boundary conditions will require the introduction of some $\underset{\sim}{\gamma}\in W_h^k\times W_h^k$. It is known [10] that $\operatorname{div}\left(W_h^k\times W_h^k\right)$ consists of all piecewise polynomials of degree $k-1$, a larger space than Π_h if boundary singular vertices are present in the mesh. On the other hand, if there are no boundary singular vertices, there is no need to form the projection since $\Pi_h=\left\{q\in\operatorname{div}\left(W_h^k\times W_h^k\right)\ :\ \int_\Omega q(x)\,dx=0\right\}$ in this case.

6 Convergence of IPM

The convergence properties of (29) follow from [4].

Theorem 6.1 *Suppose that the forms (1) satisfy (5) and (6) for V_h and $\Pi_h=\mathcal{D}V_h$. Then the algorithm (29) converges to the solution of (1) for any $0<\rho<2\rho'$ for ρ' sufficiently large. For the choice $\rho=\rho'$, (29) converges geometrically with a rate given by*

$$C_a\left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)^2\Big/\rho'.$$

The following stopping criterion follows from [4].

Theorem 6.2 *Suppose that the forms (1) satisfy (5) and (6) for V_h and $\Pi_h=\mathcal{D}V_h$. Then the errors in algorithm (29) can be estimated by*

$$\|u^n-u_h\|_V \le \left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)\|\mathcal{D}u^n-P_{\Pi_h}G\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta} + \frac{C_a^2}{\alpha\beta} + \rho' C_b\right) \|\mathcal{D}u^n - P_{\Pi_h} G\|_\Pi .$$

When $G(q) = -b(\gamma, q)$, then $P_{\Pi_h} G = -P_{\Pi_h}\mathcal{D}\gamma$ and since $\mathcal{D}u^n \in \Pi_h$,

$$(51) \qquad \begin{aligned} \|\mathcal{D}u^n - P_{\Pi_h} G\|_\Pi &= \|P_{\Pi_h}\mathcal{D}(u^n+\gamma)\|_\Pi \\ &\le \|\mathcal{D}(u^n+\gamma)\|_\Pi . \end{aligned}$$

The latter norm is easier to compute, avoiding the need to compute $P_{\Pi_h} G$. We formalize this observation in the following result.

Corollary 6.1 *Under the conditions of Theorem (6.2) the errors in algorithm (29) can be estimated by*

$$\|u^n - u_h\|_V \le \left(\frac{1}{\beta} + \frac{C_a}{\alpha\beta}\right) \|\mathcal{D}(u^n+\gamma)\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta} + \frac{C_a^2}{\alpha\beta} + \rho' C_b\right) \|\mathcal{D}(u^n+\gamma)\|_\Pi .$$

The corollary gives a natural convergence criterion of the iterated penalty method:

$$(52) \qquad \|\mathcal{D}(u^n+\gamma)\|_\Pi \le \varepsilon_{\text{ip}} \|u^n\|.$$

7 Computational examples

All numerical experiments reported in this paper were conducted using the code `Albert` [1]. The space V_h is chosen to be (50) with degree $k \ge 4$. As a test problem we consider the well-studied Jeffrey-Hamel flow in a converging duct [8]. For this flow, a semi-analytical solution is known which is a similarity solution of the form

$$(53) \quad \underset{\sim}{u}(x,y) := 6\nu \frac{u(\text{atan}(y/x))}{x^2+y^2} \underset{\sim}{x}, \quad \underset{\sim}{x} = (x,y) \in \Omega$$

where

$$(54) \qquad u'' + 4u + 6u^2 = C, \quad u(0) = u(\alpha) = 0,$$

and differentiation is with respect to the polar angle ϕ and α is the angle (in radians) made by the two walls of the channel. Figure 2 shows the profile (for $\alpha = \pi/4$) of the radial velocity u at $r = 1$ for two choices of C, 100 and 10,000. These correspond to Reynolds' numbers of 20 and 240, respectively, with R calculated by formula $R = \frac{\max|u|}{\nu}$ for $r = 1$.

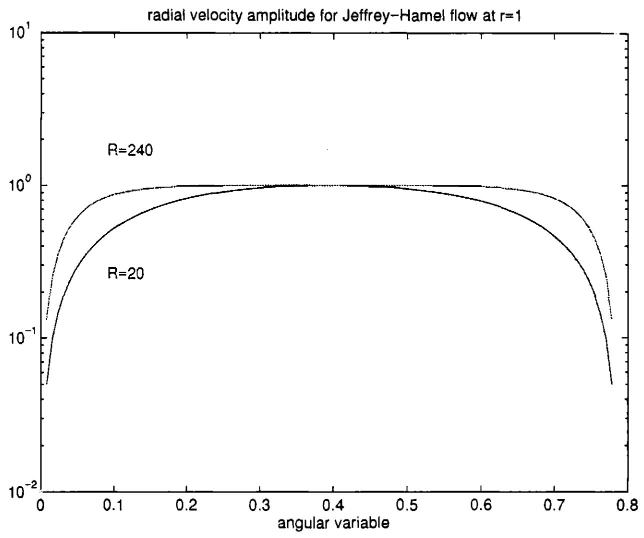


Figure 2: The profile of the angular velocity u at $r = 1$ for two choices of C, 100 and 10,000. These correspond to Reynolds' numbers of 20 and 240, respectively.

The profiles are plotted for $0.01 \le \phi/\alpha \le 0.99$. This allows an assessment of the behavior of the solution in the boundary layer. For example, the angular displacement thickness δ_R^*, which can be defined in this case by

$$(55) \qquad \delta_R^* := \int_0^{\alpha/2} 1 - \frac{u(\phi)}{u(\alpha/2)}\, d\phi,$$

has values $\delta_{20}^* = 0.110$ and $\delta_{240}^* = 0.046$.

For the domain, we consider a slice of a wedge with opening $\alpha = \pi/4$ radians made perpendicular to one of the wedge sides, as shown in Figure 3. This provides a flow with no particular symmetry.

Figure 3 shows the pressure relative "error" in the case of $C = 100$ if the projection P_{Π_h} in (41) is not included in the calculation (left). Note that the error is concentrated around the one boundary-singular vertex. When the projection is included (right), the relative error drops by five orders of magnitude, to about 0.05 percent.

Figure 1 shows the maximum time-step size for stability of the different time-stepping schemes. The implicit schemes are unconditionally stable if one solves the nonlinear equation exactly, but any algorithm used to solve them, such as fixed point iterations, introduces implicitly a stability limit. Our simulations were initialized with $u = 0$. Note that the time-dependent numerical solution converges to the steady state. A set of subsequently refined meshes was used, starting with the coarsest one from Figure 3. Numerical experiments have shown that the second-order

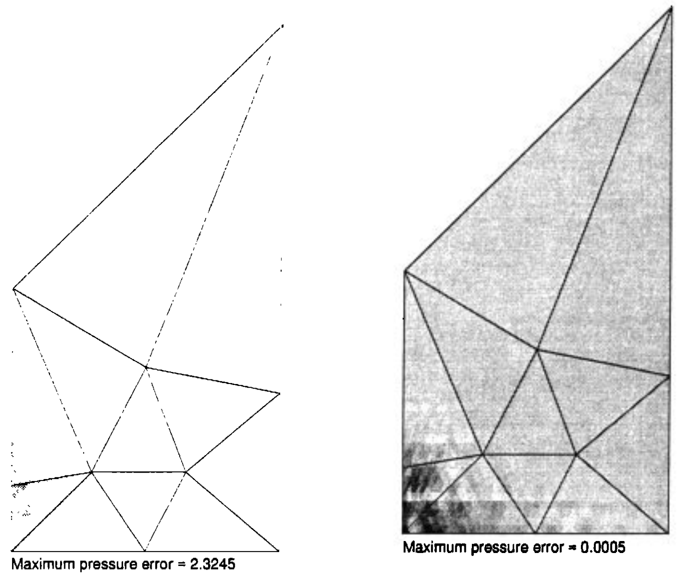

Maximum pressure error = 2.3245
Maximum pressure error = 0.0005

Figure 3: The pressure relative "error" due to a boundary-singular vertex in the case of $C = 100$ if the projection P_{Π_h} is not included in the calculation (left) and if it is included (right). The vertices of the trapezoid are $(1,0)$, $(1,1)$, $(2,2)$, $(2,0)$, in clockwise order starting from the lower-left corner.

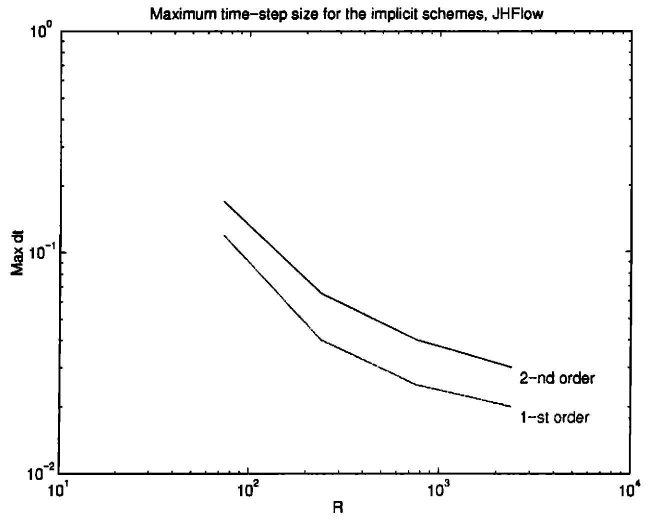


Figure 4: Maximum time-step size for the implicit time stepping schemes using fixed-point iterations as in (48) for Jeffrey-Hamel flow vs Reynolds number. The maximum mesh-step size $h = 0.35$ (two subdivisions of the mesh displayed on Figure 3).

implicit-explicit scheme with 3 fixed-point iterations (21) is the most efficient of the ones we tested.

Figure 4 shows the maximum time-step size for the implicit time-stepping schemes as a function of the Reynolds number. The mesh was generated by twice refining the coarse mesh shown in Figure 3. The number of unknowns $N = 1202$.

8 Using an initial pressure

It is possible to use a good initial guess for w^0 in the iterated penalty method (40). For example, we can use the final w^n from the previous time step or other "outer iteration." Figure 5 shows the reduction in number of iterations needed as a function of time using an initial w^n from the previous time step, for two different Reynolds numbers and for a moderate penalty parameter ($\rho = 100$). Numerical experiments were conducted for the Jeffrey-Hamel problem with zero initial guess. The number of penalty iterations approaches one because the solution approaches steady state.

Figure 6 shows the effect of varying the penalty parameter for Reynolds numbers 20 and 240. Shown are the pressure and velocity error after convergence (at a time of $t = 5$) for the Jeffrey-Hamel problem. We see that if the

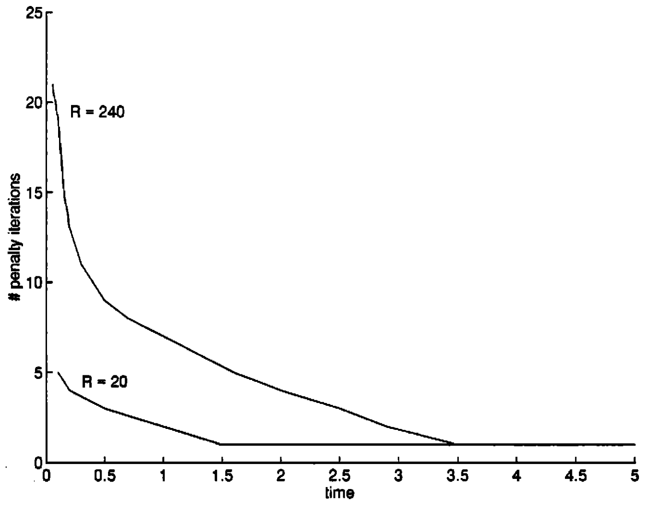
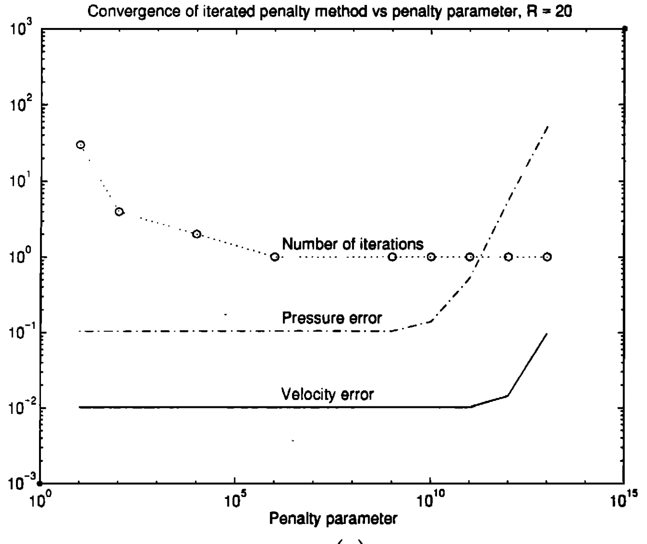
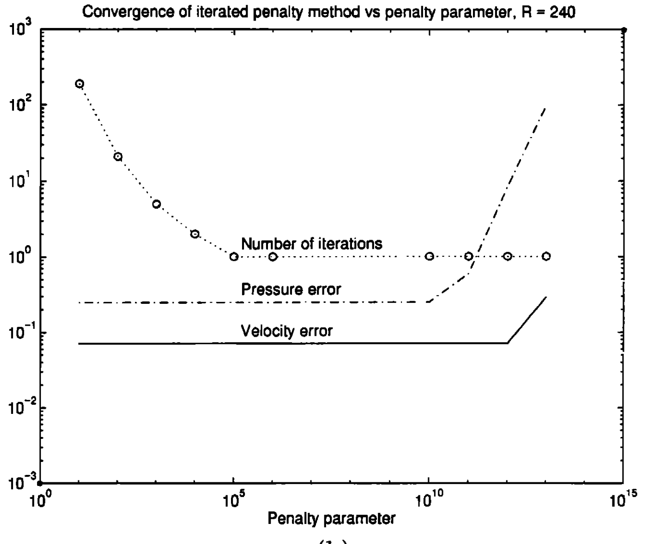


Figure 5: Reduction in number of iterations needed to satisfy (52) as a function of time for two different Reynolds numbers and for penalty parameter $\rho = 100$ using initial w^n from the previous time step for Jeffrey-Hamel flow.



(a)



(b)

Figure 6: Effect of varying the penalty parameter for Reynolds numbers 20 (a) and 240 (b). Shown are the pressure and velocity error after convergence (at a time of $t = 5$) and the number of iterations in the penalty method at the first time step, plotted as a function of the penalty parameter, ρ. Number of penalty iterations for small value of the penalty parameter ρ is proportional to $\frac{1}{\rho}$ (b).

penalty parameter becomes too large, then first the pressure degrades and finally so does the velocity. Also shown is the number of penalty iterations at the first time step when there is no initial guess for w^0. We see that it is large for a small penalty parameter but decreases rapidly to just one iteration for larger penalty parameters. The errors quoted in figures were in the L_∞ norm. We have obtained quantitatively similar error estimates in the L_2 norm.

For truly time-dependent problems, the number of penalty iterations also approaches one for each fixed-point iteration, when the value of penalty parameter is big enough. The uniform flow around a cylinder is a classical time-dependent problem which has been investigated by many researchers (see [3] and references reported therein). For Reynolds numbers larger than some critical value (around 40) vortex shedding occurs behind the cylinder (Figure 7). The time-dependent horizontal component of velocity is given in Figure 8. Plotted are velocity values at six points along the line through the center of the cylinder cross-section at the angle of $\theta = 153°$ with direction of flow for a Reynolds number $R = 100$, a time step size $\Delta t = 0.05$ and a computational mesh with 2831 nodal points. The frequency of the oscillations in terms of the dimensionless Strouhal number is found to be $St = 0.177$.

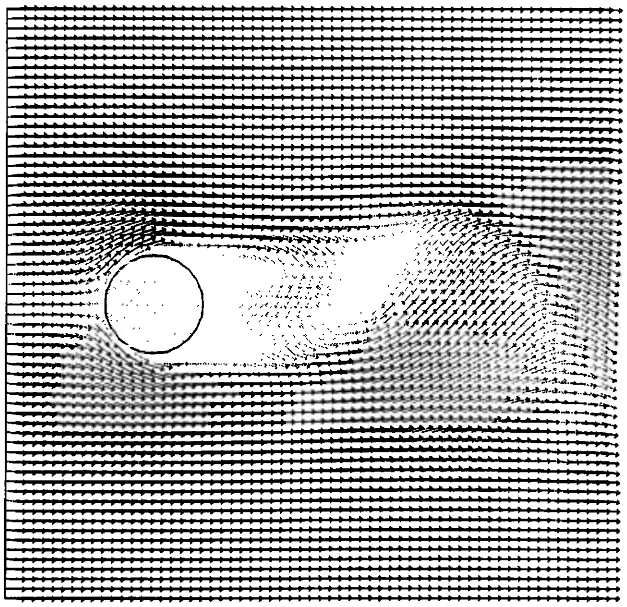


Figure 7: Velocity field for the uniform flow around a cylinder. $R = 100, t = 100, \Delta t = 0.05$. Computational domain $\Omega = [-10, 30] \times [-20, 20]$. Radius of the cylinder $r = 1$. Center of the cylinder is at $(0, 0)$. Plotted is part of the domain $[-3, 9] \times [-6, 6]$. Computational mesh has 2831 nodal points. Initial flow at $t = 0$ was uniform flow except on the cylinder, where it was set to zero.

Authors of [3] found the dimensionless Strouhal number to be $St = 0.16$ (only two digits were reported). The difference in observed St could be a result of the finite computation domain used $\Omega = [-10, 30] \times [-20, 20]$ or by the polygonal approximation of the cylinder (24 sides). A calculation on the domain $\Omega = [-20, 60] \times [-40, 40]$, using a 36-edge polygon for the cylinder representation, a mesh with 12943 nodal points and a time step size $\Delta t = 0.02$, yielded a the dimensionless Strouhal number $St = 0.169$.

The initial time steps need a large number of fixed-point and penalty iterations but when the solution becomes periodic ($t > 50$), each time step requires approximately the same number of fixed-point and penalty iterations. Figure 9 shows the total number of penalty iterations per time step as function of the penalty parameter at time 100. Two different iteration strategies were used. One requires penalty iterations reach convergence according to the criterion (52) with tolerance $\varepsilon_{\rm ip} = 10^{-7}$. Another sets the number of penalty iterations per fixed-point iteration to one and the iterations are forced to satisfy both convergence criteria (22) with tolerance $\varepsilon_{\rm fp} = 10^{-4}$ and (52). The second strategy requires fewer total number of penalty iterations for high values of the penalty parameter and therefore minimizes the number of linear-system solves required. We did simulation of the same flow for higher values of the Reynolds number ($R = 200$). For penalty parameter $\rho = 10^8$ and time step $\Delta t = 0.02$ number of fixed-point iterations equal 3 with one penalty iteration per fixed-point iteration. Like for steady-state flows the value of penalty parameter can not arbitrary large due to the big numerical error.

Since it appears that it is often possible to do only *one* penalty iteration per fixed-point iteration, it is interesting to compare this method with other methods. Algebraically, the system of equations for a mixed method (1) can be written in the form

$$\begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \qquad (56)$$

The iterated penalty method avoids the need to even calculate the matrix B, and only requires the solution of equations of the form

$$(A + \rho D)V = G,$$

which involve only the velocity degrees-of-freedom, where $D = BB^t$ and B^t represents the matrix associated with the divergence operator. Since this is a much smaller system of equations than (56), the iterated penalty method can provide a much more efficient algorithm than ones like (56) which explicitly involve the pressure degrees-of-freedom, such as the Taylor-Hood method [7].

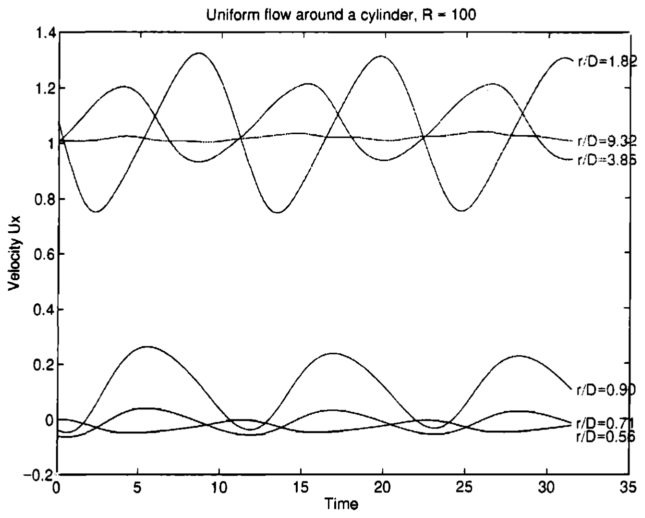


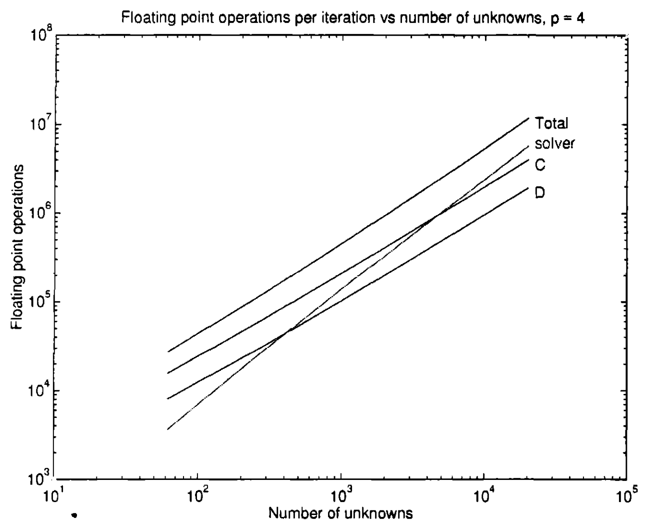
Figure 8: Time-dependent horizontal component of velocity field at six points along the line through the center of the cylinder cross-section at the angle of $\theta = 153°$ with direction of flow. Simulations were for uniform flow around cylinder with diameter $D = 2$, $R = 100$.

9 Direct versus iterative solvers

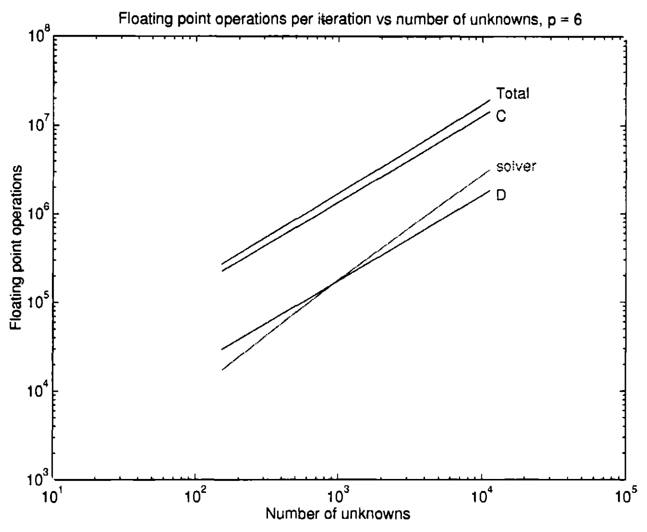
Iterative solvers are often used to solve the linear-algebraic problem that arises in implicit and implicit-explicit schemes for time-dependent partial differential equations. However, direct solvers have various nice features. For example they allow one to solve non-symmetric problems with the same amount of work as for symmetric problems. Thus it is natural to ask what is the relative amount of work for direct versus iterative solvers. We will see that for high-order methods, at least in two dimensions, direct methods have an interesting range of applicability.

Using an ordering strategy related to the "minimum degree" algorithm [9] in the case of symmetric matrices, the growth in the number of fill-ins is as depicted in Figure 10. What is plotted is the ratio of the number of non-zeros in the factored matrix to the number of non-zeros in the original matrix, as a function of the former. The number of non-zeros in the factored matrix is a good measure of the computational complexity for time-stepping problems where the matrix can be factored once and then re-used many times. We see that for a large range of problems, the number of non-zeros in the factored matrix will be less than twice that of the original matrix. This means that using the factored matrix for a direct solution will take less than two steps of typical iterative methods.

The relative cost of solving is made even more apparent



(a)



(b)

Figure 11: Relative cost of solver and form evaluations for finite elements order $p = 4$ (a) and $p = 6$ (b). D denotes the divergence term and C the nonlinear term.

in Figure 11. Shown is the relative cost of the solver and the form evaluations for the 4-th and 6-th order finite-element method. The data reported in Figures 10 and 11 does not change very much as we vary the geometry of the domain and other problem parameters. The line marked D denotes the work for computing the divergence form $b(v,p)$ and the line marked C denotes the work for computing the nonlinear form $c(u,v,w)$. We see that for the 6-th order finite-element method, the nonlinear form dominates the solver by a substantial factor, although the cost for the latter is growing at a faster rate (the forms can be computed in an amount of work that is linear in the number of matrix entries, but the cost of the solver grows supra-linearly).

An interesting question is the effectiveness of increasing the degree of approximation versus decreasing the mesh size. To investigate the dependence of the numerical error on the degree of approximation and number of unknowns, we simulate Jeffrey-Hamel flow. A set of subsequently refined meshes was used, starting with the coarsest one from Figure 3. Figure 12 compares the accuracy gained as a function of the number of non-zeros in the factored matrix, a measure of the dominant part of the computation. For smaller Reynolds numbers, higher degree approximations are more efficient. However, for larger Reynolds numbers, less difference in efficiency is seen as the degree is varied.

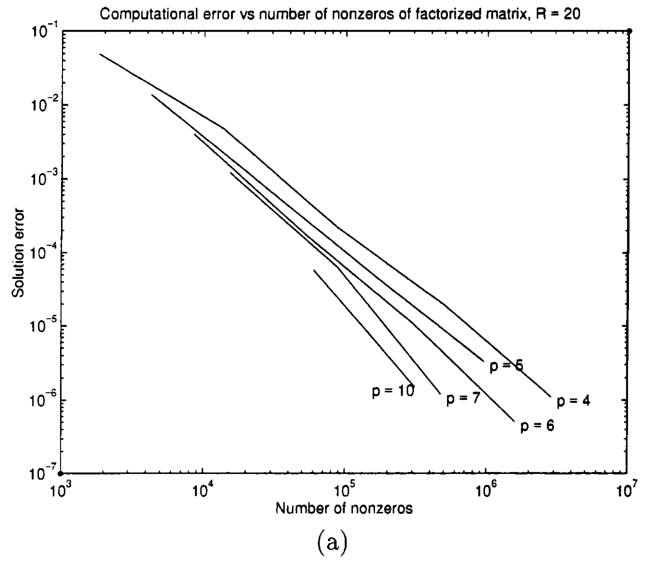
Since the forward- and back-solve is so efficient, it is reasonable to study the efficiency as a function of an estimate of the entire work for a complete time step for the overall algorithm. Figure 13 shows this data for Reynolds numbers 20 and 240. We see that basing the plots on an estimate of the work for a complete time-step tends to make low-order and high-order methods look much more similar in terms of accuracy achieved as a function of work expended. There is a noticeable increase in cost as the Reynolds number increases, due to the increased information content of the solution.

We note that we have omitted plots of the efficiency as a function of run time [2], as the latter is highly dependent on the computer architecture. For cache architectures, run time correlates well with the number of nonzeros in the factorized matrix.

For very large problem sizes, iterative methods will be more efficient. However, the standard multigrid method requires a number of smoothing steps proportional to the penalty parameter [5]. The data in Figures 6 and ?? indicates penalty parameter should be very large.

10 Conclusions

We showed that high-order finite element methods can simulate the flow of incompressible viscous fluids efficiently using the iterated penalty method for resolving the incompressibility constraint. We gave a description of the iter-



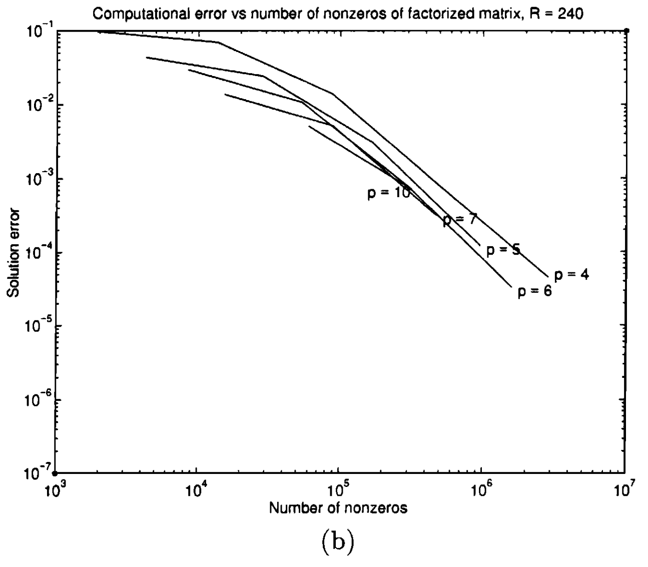
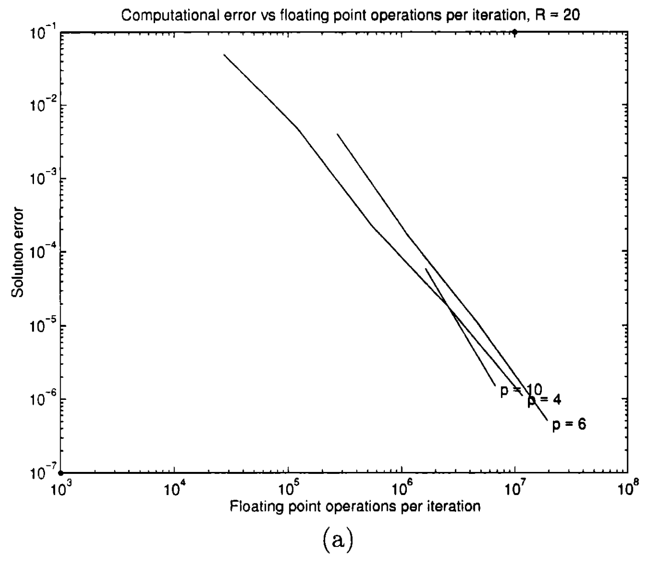


Figure 12: Accuracy in the maximum norm as a function of the number of non-zeros in the factored matrix for Reynolds number = 20 (a) and 240 (b). Numerical solution was computed for Jeffrey-Hamel flow on a set of subsequently refined meshes.



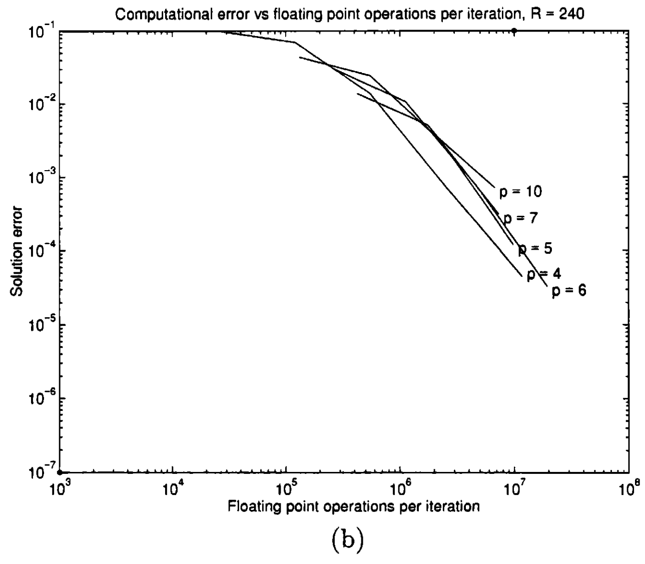


Figure 13: Accuracy as a function of the number of total work in a time step for Reynolds number = 20 (a) and 240 (b).

ated penalty method in the case of inhomogeneous boundary conditions. We showed that direct methods may be a suitable choice for many simulations, at least in the two-dimensional case.

Current work extending the results of this paper involves mesh optimization using error estimators and an extension of these results to three dimensions.

11 Acknowledgments

This work was supported in part by the National Science Foundation through award number DMS9403563 and the Texas Advanced Research Program, Project number 003652-141.

References

[1] The Albert system is available in the compressed tar file albert.tar.Z which can be obtained from http://www.hpc.uh.edu/~babak.

[2] B. Bagheri, S. Zhang, and L. R. Scott. Implementing and using high-order finite element methods. *Finite Elements in Analysis & Design*, 16:175–189, 1994.

[3] M. Braza, P. Chassaing and H. Ha Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *J. Fluid Mechanics*, 165:79–130, 1986.

[4] S. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, 1994.

[5] Z. Cai, C. I. Goldstein and J. E. Pasciak. Multilevel iteration for mixed finite element systems with penalty. *SIAM J. Sci. Comput.*, 14:1072–1088, 1993.

[6] S. D. Conte, C. de Boor. *Elementary Numerical Analysis*, McGraw-Hill, 1981.

[7] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations, Theory and Algorithms.* Springer-Verlag , Berlin, New York, 1986.

[8] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, Pergamon Press, 1959.

[9] S. Pissanetsky, *Sparse Matrix Technology*, Academic Press, 1984.

[10] M. Vogelius and L. R. Scott. Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials. *M^2AN (formerly R.A.I.R.O. Analyse Numérique)*, 19:111–143, 1985.

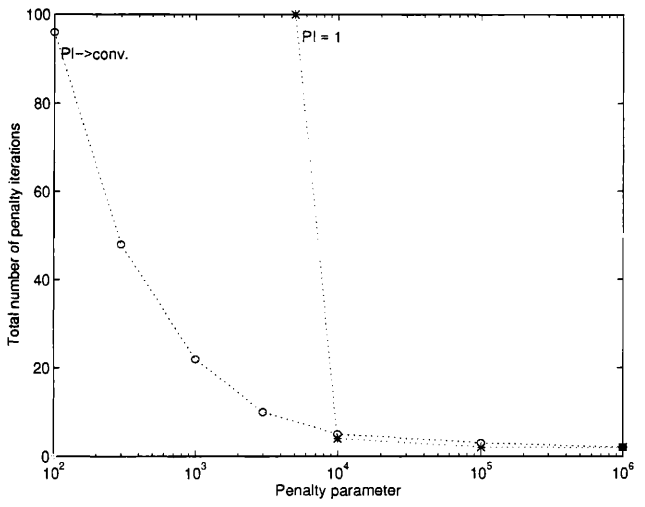


Figure 9: Convergence of iterated penalty method for the uniform flow around a cylinder, as a function of penalty parameter for fully developed periodic flow ($t = 100$). o's indicate a strategy in which the penalty iterations are repeated until they reach convergence criterion (52). *'s indicate a strategy with one penalty iteration per fixed-point iteration and fixed-point iterations are repeated until they satisfy both (22) and (52).

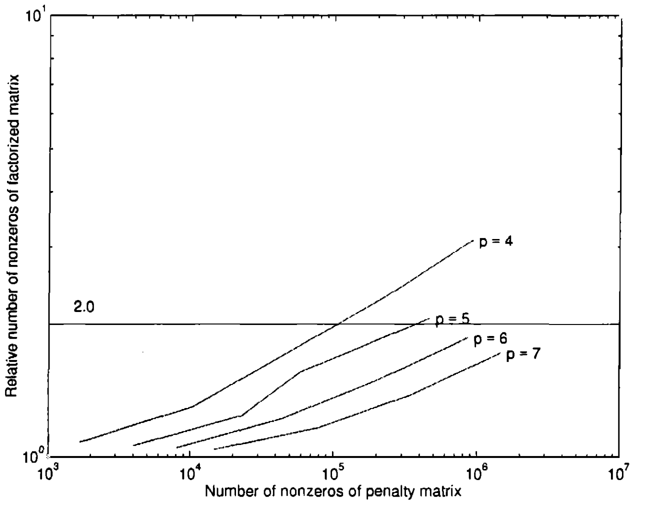


Figure 10: Growth in the number of fill-ins using an ordering strategy related to the "minimum degree" algorithm. Plotted is the ratio of the number of nonzeros in the factored matrix to the number of nonzeros in the original matrix.