Journal of Computational and Applied Mathematics 228 (2009) 373–390

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A splitting technique of higher order for the Navier–Stokes equations

Jörg Frochte, Wilhelm Heinrichs *

Universität Duisburg-Essen, Ingenieurmathematik, Universitätsstr. 3, D-45117 Essen, Germany

ARTICLE INFO

Article history:
Received 14 April 2008
Received in revised form 15 August 2008

Keywords:
Navier–Stokes equations
Finite element methods
Recovery technique
Higher order in time
Postprocessing
Preconditioning

ABSTRACT

This article presents a splitting technique for solving the time dependent incompressible Navier–Stokes equations. Using nested finite element spaces which can be interpreted as a postprocessing step the splitting method is of more than second order accuracy in time. The integration of adaptive methods in space and time in the splitting are discussed. In this algorithm, a gradient recovery technique is used to compute boundary conditions for the pressure and to achieve a higher convergence order for the gradient at different points of the algorithm. Results on the 'Flow around a cylinder's- and the 'Driven Cavity's-problem are presented.

1. Introduction

The time dependent incompressible Navier–Stokes equations are given by:

$$\frac{\partial v}{\partial t} + (v \cdot \nabla)v - \nu\nabla^2 v + \nabla p = f \quad \text{in } \Omega,\ t \in [0, \hat{t}] \tag{1}$$

$$\nabla \cdot v = 0 \quad \text{in } \Omega,\ t \in [0, \hat{t}] \tag{2}$$

$$v = h \quad \text{on } \partial\Omega,\ t \in [0, \hat{t}], \tag{3}$$

$$v = v_0 \quad \text{for } t = 0,\ \text{in } \Omega. \tag{4}$$

The solution of these equations on the time interval $[0, \hat{t}]$ are the velocity v of a Newtonian fluid with the kinematic viscosity ν and the pressure p in a domain Ω. We assume that Ω is a bounded domain in $\mathbb{R}^2$ and that its boundary $\partial\Omega$ is polygonal. The boundary conditions are given by a function h on $\partial\Omega$.

We start by introducing a splitting technique for the Navier–Stokes equations with finite elements which is related to the one published by Haschke and Heinrichs [11,12] for spectral methods. We call this algorithm the *base splitting algorithm* and introduce it in Section 2. Like many other splitting techniques it seems to be restricted to the second order in time. An overview of several splitting approaches as well as their orders in time, which are estimated by numerical results or an analytic proof for a given smoothness of the solution, can be found in [7]. The major reason for the restriction to second order is the fact that – by using the approach published in [11] – it has not been possible so far to construct a stable pressure extrapolation of an order higher than one, especially for higher Reynolds numbers.

To negotiate this problem we choose a hierarchy of finite element spaces in Section 5, and conveniently nest two base splitting steps. The result is a technique of higher order in time that generally reduces the CPU costs compared to the base splitting with the same number of unknowns. So this technique can alternatively be seen as a postprocessing or a preconditioning splitting step. First results on this subject are already presented in [9].

* Corresponding author.
E-mail addresses: joerg.frochte@uni-due.de (J. Frochte), w.heinrichs@uni-essen.de (W. Heinrichs).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.09.028

Time step in the base splitting

1. Compute a guess ($\bar{p}^{m+1}$) for the pressure
2. Based on the pressure compute an intermediate velocity $\tilde{v}^{m+1}$
3. Solve the Poisson equation

$$-\nabla^2 p_{update} = -\frac{\beta_0}{\Delta t}\nabla \cdot \tilde{v}^{m+1} \tag{5}$$

$$n \cdot \nabla p_{update} = n \cdot 0 \quad \text{on } \partial\Omega \tag{6}$$

   for the pressure and velocity update
4. Apply the update by

$$p^{m+1} = \bar{p}^{m+1} + p_{update} \tag{7}$$

$$v^{m+1} = \tilde{v}^{m+1} + \frac{\Delta t}{\beta_0}\nabla p_{update} \tag{8}$$

Box I.

For splitting techniques the boundary conditions for the pressure are always a challenge with a long history see e.g. [26,18]. For the boundary conditions of the second-order pressure equation we refer to the discussion in the papers of Karniadakis et al. [16], Maday et al. [19] and Timmermanns et al. [20,23]. For the computation of the pressure boundary conditions in Section 2 we used a variation of the formulation given in [16] by Karniadakis.

To use this technique we have to evaluate the laplacian operator with linear finite elements, which leads to a couple of problems that can be avoided using a gradient recovery technique. Beyond this such a gradient recovery technique can be used at different points of the algorithm to increase the accuracy in space, because for linear finite elements the convergence rate of the gradient is only of first order. Some gradient recovery techniques like the Z^2 gradient recovery [27] are less accurate at the edges of Ω where we want to compute boundary conditions. So in Section 3 we develop a new gradient recovery technique for this splitting with better recovery results at $\partial\Omega$.

To test our splitting scheme on appropriate examples we use two different strategies. One quite common way to get appropriate examples consists of choosing a velocity/pressure pair $(v; p)$ and setting the right-hand side and the boundary conditions so that $(v; p)$ fulfills the Navier–Stokes equations. With this strategy it is easy to compare the finite element solution with the exact one. We tested the splitting on some examples of this type and present the results on two of them in this paper. However, a solution $(v; p)$ chosen in this way has in general no physical meaning. So beyond this we tested the algorithm on some standard CFD problems in Section 6, the 'Flow around a cylinder's- and the 'Driven Cavity's-problem.

2. The stabilised base splitting

For the approximation of $\frac{\partial}{\partial t}$ we use a BDF scheme of third order. The leading coefficient of the BDF scheme is denoted with β_0 and the time step size with Δt. Similar to the splitting for spectral methods [11] one time step of the splitting follows this scheme given in Box I.

The finite element spaces for the velocity and the pressure are chosen to fulfill the inf-sup-condition, so we use triangle Taylor–Hood elements with linear and in the context of the postprocessing also with quadratic base functions.

In [11] $\bar{p}^{m+1}$ is chosen $\bar{p}^{m+1} := \bar{p}^m$. This choice has two disadvantages.

The first one concerns the boundary conditions. With this choice the pressure function is stuck to unnatural homogeneous Neumann boundary conditions. Unfortunately it is problematic to integrate fitted boundary conditions in the Poisson problem (5) and (6) because a too high accuracy is necessary to compute stable adapted boundary conditions. The reason for the need for high accuracy is that, particularly for a small time step size, an erasement of all trusted decimal places is possible if p does not change too much, $p^{m+1} - p^m = p_{update} \approx 0$ on $\partial\Omega$.

The second disadvantage concerns e.g. functions of the type $v(x, y, t) = z(t) \cdot w(x, y)$. Because of a kind of memory effect for such functions, this method is not unconditionally stable for finite elements. In such cases the structure of v does not change in time. So the same e.g. mesh based errors that appear when solving the Poisson problem (5) and (6) are added stepwise to the pressure in step 4 of the algorithm.

For small time step sizes the factor $\frac{\beta_0}{\Delta t}$ on the right side of (5) amplifies this effect which is compensated in (8) but not in (7).

For the used linear finite elements this leads to an unstable algorithm. To avoid this, we have to look for a way to compute p without a memory effect. To do this we use (1) together with the fact that v is divergence free. If we apply $\nabla\cdot$ to (1) the linear terms $\frac{\partial v}{\partial t}$ and $\nu\nabla^2 v$ on the left side are eleminated. To evaluate the non-linear term we use the velocity and the right side f of the last time step. It turned out that a mixed-formulation of v^m and f^{m+1} is unstable in some cases as well as using an extrapolation of a higher order for the velocity. After all we end up with a Poisson equation for the pressure guess. This technique prevents the mentioned memory effect and the associated unstablity. The price is another Poisson equation