Journal of Computational Physics 309 (2016) 368–386

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Projection methods for incompressible flow problems with WENO finite difference schemes

Javier de Frutos [a,1], Volker John [b,c], Julia Novo [d,*,2]

[a] Instituto de Investigación en Matemáticas (IMUVA), Universidad de Valladolid, Spain
[b] Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany
[c] Free University of Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany
[d] Departamento de Matemáticas, Universidad Autónoma de Madrid, Spain

ARTICLE INFO

Article history:
Received 22 July 2015
Received in revised form 10 December 2015
Accepted 17 December 2015
Available online 22 December 2015

Keywords:
Incompressible Navier–Stokes equations
Finite difference WENO schemes
Non-incremental projection methods
Incremental projection methods
PSPG-type stabilization

ABSTRACT

Weighted essentially non-oscillatory (WENO) finite difference schemes have been recommended in a competitive study of discretizations for scalar evolutionary convection–diffusion equations [20]. This paper explores the applicability of these schemes for the simulation of incompressible flows. To this end, WENO schemes are used in several non-incremental and incremental projection methods for the incompressible Navier–Stokes equations. Velocity and pressure are discretized on the same grid. A pressure stabilization Petrov–Galerkin (PSPG) type of stabilization is introduced in the incremental schemes to account for the violation of the discrete inf-sup condition. Algorithmic aspects of the proposed schemes are discussed. The schemes are studied on several examples with different features. It is shown that the WENO finite difference idea can be transferred to the simulation of incompressible flows. Some shortcomings of the methods, which are due to the splitting in projection schemes, become also obvious.

1. Introduction

This paper studies projection methods in combination with weighted essentially non-oscillatory (WENO) finite difference methods for discretizing the incompressible Navier–Stokes equations

$$\begin{aligned} \partial_t \boldsymbol{u} - \nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p &= \boldsymbol{f} \quad \text{in } (0,T]\times\Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 \quad \text{in } [0,T]\times\Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_0 \quad \text{in } \{0\}\times\Omega, \end{aligned} \tag{1}$$

in a finite time interval $[0,T]$ and in a bounded domain $\Omega\subset\mathbb{R}^d$, $d\in\{2,3\}$, subject to Dirichlet boundary conditions $\boldsymbol{u}=\boldsymbol{g}$ on $\partial\Omega$. In (1), $\boldsymbol{u}$ is the velocity field, p the pressure, and $\boldsymbol{f}$ a given body force.

[*] Corresponding author.
E-mail addresses: frutos@mac.uva.es (J. de Frutos), volker.john@wias-berlin.de (V. John), julia.novo@uam.es (J. Novo).

[1] The research was supported by Spanish MINECO under grant MTM2013-42538-P and European Cooperation in Science and Technology through ISCH COST Action IS1104.

[2] The research was supported by Spanish MINECO under grant MTM2013-42538-P.

http://dx.doi.org/10.1016/j.jcp.2015.12.041
0021-9991/© 2015 Elsevier Inc. All rights reserved.

The motivation for studying this topic came from the good experience of applying the WENO finite difference method for the numerical solution of scalar evolutionary convection–diffusion–reaction equations in [20]. In this reference, several methods were studied, including a linear and a nonlinear finite element method (FEM) flux-corrected transport (FCT) method combined with the Crank–Nicolson scheme, an essentially non-oscillatory (ENO) scheme of order three and a fifth order WENO scheme combined with total variational diminishing (TVD) Runge–Kutta methods. The ENO and WENO schemes were shown to perform efficiently and to produce accurate approximations with only small under- and overshoots.

With the good performance of the WENO scheme for scalar convection-dominated problems in mind, it is appealing to study the possibility of extending its application to the incompressible Navier–Stokes equations. However, the fact that velocity and pressure are coupled in these equations introduces a new difficulty. To circumvent this difficulty, the consideration of projection methods, which decouple the computation of velocity and pressure, seems to be a natural approach for applying the WENO scheme for the simulation of incompressible flows. In this approach, the basic idea is to use central finite differences to approximate the spatial derivatives except for the nonlinear term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ that is approximated with the WENO scheme.

Projection methods or splitting schemes or pressure correction schemes have been a long tradition for the simulation of incompressible flows, see [12] for an overview. First proposals of so-called non-incremental schemes date back to [7,31] and of incremental methods to [33]. However, the use of projection methods is still an active field of research: as stated in [3]: "As noted by Karniadakis and Sherwin [23], in high Reynolds incompressible flows, splitting methods can be computationally efficient and competitive compared to more expensive coupled methods". In [3] a pressure-correction scheme for the incompressible Navier–Stokes equations was presented which combines a discontinuous Galerkin approximation for the velocity and a standard continuous Galerkin approximation for the pressure. The main interest of the scheme is the reduced computational cost compared with monolithic strategies. Moderate to high Reynolds number flows were studied in [3] with Reynolds numbers up to 21 000. An adaptive finite element method for the incompressible Navier–Stokes equations based on a standard projection scheme (the incremental pressure correction scheme) is studied in [28]. In [14], a stable projection method on non-graded quadtree and octree meshes and arbitrary geometries is presented. The viscous term is treated implicitly with a finite volume approach while the convective term is discretized with a semi-Lagrangian scheme.

The main goal of the present paper consists in proposing and studying numerically the application of WENO finite difference schemes within several projection methods. Besides standard non-incremental and incremental methods, the application of a TVD Runge–Kutta method to the Navier–Stokes equations will be studied. The considered schemes include two variants of the simplest Euler non-incremental scheme, an Euler incremental scheme, an incremental scheme based on BDF2, and the already mentioned incremental scheme based on a TVD Runge–Kutta method. The numerical studies include examples whose solutions possess different features. Although all studied methods are applicable in two and three dimensions, only two-dimensional problems will be presented since, in our opinion, it turned out to be sufficient for illustrating the properties of the studied methods. The examples are not restricted to problems on rectangular domains. All flows are laminar and the meshes are sufficiently fine to resolve the important scales of the flow fields.

Velocity and pressure are approximated with finite differences on the same grid. It is well known that in the Euler non-incremental scheme certain pressure stability can be expected, regardless of the particular discrete velocity–pressure spaces chosen, see [13,25]. On the other hand, using the incremental schemes with this discretization, one encounters oscillations in the numerical solutions. These oscillations are due to the fact that the spatial discretization does not satisfy a discrete inf-sup condition. As a remedy of this problem, we propose to apply a stabilization technique which resembles the pressure-stabilized Petrov–Galerkin method (PSPG) used in finite element discretizations. The idea is to change the mass balance $\nabla \cdot \boldsymbol{u} = 0$ to $\nabla \cdot \boldsymbol{u} - \delta \Delta p = 0$, where δ is an appropriately chosen stabilization parameter. In the context of equal order finite element methods applied in an incremental projection scheme for the transient Stokes equations, this kind of stabilization can be found in [24]. A similar stabilization, but more related to local projection stabilization than to PSPG, was used in [8]. In a variational formulation, instead of adding $\delta(\nabla p_h, \nabla q_h)$, the term $\delta(\nabla p_h - \pi_h, \nabla q_h)$ is added where π_h is the projection of ∇p_h into a certain finite-dimensional space.

The paper is organized as follows. In Section 2 the methods are described in detail. Section 3 is devoted to the presentation of the numerical studies. Finally, Section 4 summarizes the numerical results and gives an outlook.

2. The projection methods

In this section, the studied methods are presented in detail, the application of the WENO scheme is described, and several algorithmic aspects of the methods are discussed, in particular the computation of the divergence at the boundary, the parameter choice in the PSPG-type stabilization, and the extension of the WENO finite difference method to non-rectangular domains (but still Cartesian grids). For simplicity of presentation, the case of homogeneous boundary conditions $\boldsymbol{g} = \boldsymbol{0}$ is considered.

2.1. Euler non-incremental method, type 1 (Eul-ninc1)

Given $\boldsymbol{u}^0$, compute $(\tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{u}^{n+1}, p^{n+1})$, $n = 0, \ldots, N$, with the following algorithm. In the first step, $\tilde{\boldsymbol{u}}^{n+1}$ is obtained by solving the vector-valued convection–diffusion equation

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}}^{n+1}-\boldsymbol{u}^n}{\Delta t}-\nu\Delta\tilde{\boldsymbol{u}}^{n+1}+(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n &= \boldsymbol{f}^{n+1} \text{ in } \Omega,\\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega.\end{aligned} \tag{2}$$

The second equation is of the form

$$\begin{aligned}\frac{\boldsymbol{u}^{n+1}-\tilde{\boldsymbol{u}}^{n+1}}{\Delta t}+\nabla p^{n+1} &= \boldsymbol{0} \text{ in } \Omega,\\ \nabla\cdot\boldsymbol{u}^{n+1} &= 0 \text{ in } \Omega,\\ \boldsymbol{u}^{n+1}\cdot\boldsymbol{n} &= 0 \text{ on } \partial\Omega.\end{aligned} \tag{3}$$

Taking the divergence of this equation leads to an equation for computing p^{n+1}

$$\begin{aligned}-\Delta p^{n+1} &= -\frac{1}{\Delta t}\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} \text{ in } \Omega,\\ \nabla p^{n+1}\cdot\boldsymbol{n} &= 0 \qquad \text{on } \partial\Omega.\end{aligned} \tag{4}$$

Finally, one obtains $\boldsymbol{u}^{n+1}$ from the update

$$\boldsymbol{u}^{n+1}=\tilde{\boldsymbol{u}}^{n+1}-\Delta t\nabla p^{n+1}. \tag{5}$$

Since velocity and pressure are defined both in the nodes of the mesh, the discretization of velocity and pressure does not satisfy a discrete inf-sup condition. Note that (4) has the form of the pressure stabilization Petrov–Galerkin (PSPG) method, which is popular in finite element methods for stabilizing pairs of spaces that do not satisfy the discrete inf-sup condition. Thus, this scheme possesses an inherent stabilization with respect to the discrete inf-sup condition. The stabilization parameter in (4) is Δt and it becomes small for small time steps, see [8,13]. The similarity of the stabilization in this scheme with the PSPG stabilization was already observed in [25].

2.2. Euler non-incremental method, type 2 (Eul-ninc2)

In this method, the convective term of the vector-valued convection–diffusion equation changes to $(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n$. Inserting now (5) into this equation gives a method which computes only $(\tilde{\boldsymbol{u}}^{n+1}, p^{n+1})$, $n=1,\ldots,N$, where $\tilde{\boldsymbol{u}}^1, p^1$ have to be given. The vector-valued convection–diffusion equation for the velocity reads as follows

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}}^{n+1}-\tilde{\boldsymbol{u}}^n}{\Delta t}-\nu\Delta\tilde{\boldsymbol{u}}^{n+1}+(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n+\nabla p^n &= \boldsymbol{f}^{n+1} \text{ in } \Omega,\\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega.\end{aligned} \tag{6}$$

The second equation in this scheme is (3) such that the pressure is computed by solving (4). Performing the update (5) is not necessary. Because the pressure Poisson equation in this scheme is of form (4), it contains a stabilization with respect to the discrete inf-sup condition.

2.3. Euler incremental method (Eul-inc)

The definition of this method considers also triplets $(\tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{u}^{n+1}, p^{n+1})$. The equation for computing $\tilde{\boldsymbol{u}}^{n+1}$ has the form

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}}^{n+1}-\boldsymbol{u}^n}{\Delta t}-\nu\Delta\tilde{\boldsymbol{u}}^{n+1}+(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n+\nabla p^n &= \boldsymbol{f}^{n+1} \text{ in } \Omega,\\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega.\end{aligned} \tag{7}$$

Then, the correction is defined by

$$\begin{aligned}\frac{\boldsymbol{u}^{n+1}-\tilde{\boldsymbol{u}}^{n+1}}{\Delta t}+\nabla(p^{n+1}-p^n) &= \boldsymbol{0} \text{ in } \Omega,\\ \nabla\cdot\boldsymbol{u}^{n+1} &= 0 \text{ in } \Omega,\\ \boldsymbol{u}^{n+1}\cdot\boldsymbol{n} &= 0 \text{ on } \partial\Omega.\end{aligned} \tag{8}$$

Substituting $\boldsymbol{u}^n$ from (8) into (7) gives the first equation of the scheme which will be considered here. Let $\tilde{\boldsymbol{u}}^2, p^2, p^1$ be given, compute $(\tilde{\boldsymbol{u}}^{n+1}, p^{n+1})$, $n=2,\ldots,N$, by first solving in each discrete time

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}}^{n+1}-\tilde{\boldsymbol{u}}^n}{\Delta t}-\nu\Delta\tilde{\boldsymbol{u}}^{n+1}+(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n+\nabla(2p^n-p^{n-1}) &= \boldsymbol{f}^{n+1} \text{ in } \Omega,\\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega.\end{aligned} \tag{9}$$

A Poisson equation for the pressure update is derived by taking the divergence of (8)

$$\begin{aligned} -\Delta(p^{n+1}-p^n) &= -\frac{1}{\Delta t}\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} \ \text{in } \Omega, \\ \nabla(p^{n+1}-p^n)\cdot\boldsymbol{n} &= 0 \qquad\qquad \text{on } \partial\Omega. \end{aligned} \tag{10}$$

The method (9), (10) is problematic for velocity and pressure approximations that do not satisfy a discrete inf-sup condition. This problem becomes obvious if the equation has in the limit a steady-state solution. Then $p^{n+1}=p^n$ and from (10) it follows that $\nabla\cdot\tilde{\boldsymbol{u}}^{n+1}=0$. This implied equation is together with (9) a saddle point problem whose well-posedness requires the satisfaction of an inf-sup condition. In fact, we could observe in simulations that steady-state solutions usually could not be reached by using the scheme (9), (10) and time-dependent solutions often showed notable spurious oscillations. A remedy, which was mentioned already in [24, Chapter 7.4], consists in stabilizing this method in the spirit of the PSPG method. To this end, the mass balance of (8) is changed to

$$\nabla\cdot\boldsymbol{u}^{n+1}-\delta\Delta p^{n+1}=0, \tag{11}$$

which gives the pressure Poisson equation

$$\begin{aligned} -\Delta\left(\left(1+\frac{\delta}{\Delta t}\right)p^{n+1}-p^n\right) &= -\frac{1}{\Delta t}\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} \ \text{in } \Omega, \\ \nabla\left(\left(1+\frac{\delta}{\Delta t}\right)p^{n+1}-p^n\right)\cdot\boldsymbol{n} &= 0 \qquad\qquad \text{on } \partial\Omega. \end{aligned} \tag{12}$$

The considered method Eul-inc solves (9) and (12). The choice of the stabilization parameter δ will be discussed in Section 2.7.

An alternative option consists in replacing the nonlinear convective term in (7) by $(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n$ and to compute $\boldsymbol{u}^n$ after having computed the pressure from (12). In our experience, this approach gives very similar results to the method (9), (12) such that we omit the discussion of the alternative approach for the sake of brevity.

2.4. BDF2 incremental method (BDF-inc)

This method uses a second order time stepping scheme and a second order extrapolation of the convective term. It is given by

$$\begin{aligned} \frac{3\tilde{\boldsymbol{u}}^{n+1}-4\boldsymbol{u}^n+\boldsymbol{u}^{n-1}}{2\Delta t} &- \nu\Delta\tilde{\boldsymbol{u}}^{n+1}+2(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n \\ -(\tilde{\boldsymbol{u}}^{n-1}\cdot\nabla)\tilde{\boldsymbol{u}}^{n-1}+\nabla p^n &= \boldsymbol{f}^{n+1} \ \text{in } \Omega, \\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega, \end{aligned} \tag{13}$$

and the correction is defined by

$$\begin{aligned} \frac{3\boldsymbol{u}^{n+1}-3\tilde{\boldsymbol{u}}^{n+1}}{2\Delta t}+\nabla(p^{n+1}-p^n) &= \boldsymbol{0} \ \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u}^{n+1} &= 0 \ \text{in } \Omega, \\ \boldsymbol{u}^{n+1}\cdot\boldsymbol{n} &= 0 \ \text{on } \partial\Omega. \end{aligned} \tag{14}$$

Substituting $\boldsymbol{u}^n$ and $\boldsymbol{u}^{n-1}$ from (14) into (13) yields the following scheme. Given $\tilde{\boldsymbol{u}}^3, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{u}}^1, p^3, p^2, p^1$, compute $(\tilde{\boldsymbol{u}}^{n+1}, p^{n+1})$, $n=3,\ldots,N$, by first solving in each discrete time

$$\begin{aligned} \frac{3\tilde{\boldsymbol{u}}^{n+1}-4\tilde{\boldsymbol{u}}^n+\tilde{\boldsymbol{u}}^{n-1}}{2\Delta t} &- \nu\Delta\tilde{\boldsymbol{u}}^{n+1}+2(\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n \\ -(\tilde{\boldsymbol{u}}^{n-1}\cdot\nabla)\tilde{\boldsymbol{u}}^{n-1}+\frac{\nabla(7p^n-5p^{n-1}+p^{n-2})}{3} &= \boldsymbol{f}^{n+1} \ \text{in } \Omega, \\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{0} \quad \text{on } \partial\Omega. \end{aligned} \tag{15}$$

Deriving the pressure Poisson equation from (14) leads to the same instability problem as discussed for the method Eul-inc and we encountered the same bad observations in numerical studies as described in Section 2.3. Again, a remedy consists in applying the PSPG-like stabilization (11) to the equation for the mass balance in (14). Then, the update for the pressure is computed by solving

$$\begin{aligned} -\Delta\left(\left(1+\frac{3\delta}{2\Delta t}\right)p^{n+1}-p^n\right) &= -\frac{3}{2\Delta t}\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} \ \text{in } \Omega, \\ \nabla\left(\left(1+\frac{3\delta}{2\Delta t}\right)p^{n+1}-p^n\right)\cdot\boldsymbol{n} &= 0 \qquad\qquad \text{on } \partial\Omega. \end{aligned} \tag{16}$$

The method BDF-inc solves (15) and (16). Apart from the form of the convective term, BDF-inc is the same method that was studied in [3].

Using $2(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - (\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}$ in (13) as convective term gave very similar results for the examples considered in Section 3 to the method (15), (16) such that the discussion of this approach will be omitted.

2.5. TVD Runge–Kutta incremental method (RK-inc)

This method transfers the idea of using TVD Runge–Kutta methods for the simulation of scalar convection–diffusion problems to projection methods. To the best of our knowledge, this transfer cannot be found in the literature so far.

The method of Heun for discretizing the equation $\dot{u} = F(t,u)$ is given by

$$\begin{aligned} k_1 &= F(t_{n-1}, u_{n-1}), \\ k_2 &= F(t_{n-1} + \Delta t, u_{n-1} + \Delta t k_1), \\ u_n &= u_{n-1} + \frac{\Delta t}{2}(k_1 + k_2). \end{aligned}$$

It is a second order method. Its implementation in the context of projection schemes may be done as follows. Given $\boldsymbol{u}^1, p^1$, one computes in each time step $n = 2, \ldots, N$, the first stage for an intermediate velocity by

$$\begin{aligned} \boldsymbol{k}_1 &= \nu\Delta\boldsymbol{u}^n - (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \nabla p^n + \boldsymbol{f}^n, \\ \tilde{\boldsymbol{u}}^{n,*} &= \boldsymbol{u}^n + \Delta t \boldsymbol{k}_1. \end{aligned}$$

Concerning the projection into the divergence-free space, the same remarks are in order as for the method Eul-inc, i.e., one has to apply a stabilization because the discrete velocity and pressure do not satisfy an inf-sup condition. This stabilization can be performed in the same way as for Eul-inc, giving for the first stage the pressure equation

$$\begin{aligned} -\Delta\left(\left(1 + \frac{\delta}{\Delta t}\right)p^{n,*} - p^n\right) &= -\frac{1}{\Delta t}\nabla \cdot \tilde{\boldsymbol{u}}^{n,*} && \text{in } \Omega, \\ \nabla\left(\left(1 + \frac{\delta}{\Delta t}\right)p^{n,*} - p^n\right) \cdot \boldsymbol{n} &= 0 && \text{on } \partial\Omega, \end{aligned} \tag{17}$$

and the velocity after the first stage is computed by

$$\boldsymbol{u}^{n,*} = \tilde{\boldsymbol{u}}^{n,*} - \Delta t \nabla\left(p^{n,*} - p^n\right).$$

The computation of the second stage starts with

$$\begin{aligned} \boldsymbol{k}_2 &= \nu\Delta\boldsymbol{u}^{n,*} - (\boldsymbol{u}^{n,*} \cdot \nabla)\boldsymbol{u}^{n,*} - \nabla p^{n,*} + \boldsymbol{f}^{n+1}, \\ \tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{u}^n + \frac{\Delta t}{2}(\boldsymbol{k}_1 + \boldsymbol{k}_2) \end{aligned}$$

and the equation for the pressure update, which includes the PSPG-type stabilization, has the form

$$\begin{aligned} -\Delta\left(\left(1 + \frac{\delta}{\Delta t}\right)p^{n+1} - p^{n,*}\right) &= -\frac{1}{\Delta t}\nabla \cdot \tilde{\boldsymbol{u}}^{n+1} && \text{in } \Omega, \\ \nabla\left(\left(1 + \frac{\delta}{\Delta t}\right)p^{n+1} - p^{n,*}\right) \cdot \boldsymbol{n} &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{18}$$

From this equation, p^{n+1} is computed. The new velocity is obtained by

$$\boldsymbol{u}^{n+1} = \tilde{\boldsymbol{u}}^{n+1} - \Delta t \nabla\left(p^{n+1} - p^{n,*}\right).$$

2.6. Evaluating the convective term with the WENO scheme

For the numerical discretization of the convective term, the finite difference weighted essentially non-oscillatory (WENO) method presented in [30] will be used. For simplicity of presentation, the method will be described for a scalar convective term.

The main idea in the finite difference WENO scheme is to approximate the spatial derivatives of the solution $\partial_x u(t, x_i, y_j)$, $\partial_y u(t, x_i, y_j)$ using a convex combination of several finite differences formulae with appropriate weights. For simplicity, the approximation of the one-dimensional function $b(t,x)\partial_x u(t,x)$ at $x = x_i$ on an equidistant mesh with grid size h will be considered. A WENO scheme of fifth order with a stencil of six nodes, which is used in many applications [30], defines the approximation of $u(t, x_i)$ in the case $b(t, x_i) \geq 0$ as follows [15]

$$
\begin{aligned}
(u_h^i)_{x,1} &= \frac{1}{h}\left(-\frac{u_h^{i-1}}{3} - \frac{u_h^i}{2} + u_h^{i+1} - \frac{u_h^{i+2}}{6}\right),\\
(u_h^i)_{x,2} &= \frac{1}{h}\left(\frac{u_h^{i-2}}{6} - u_h^{i-1} + \frac{u_h^i}{2} + \frac{u_h^{i+1}}{3}\right),\\
(u_h^i)_{x,3} &= \frac{1}{h}\left(-\frac{u_h^{i-3}}{3} + \frac{3}{2}u_h^{i-2} - 3u_h^{i-1} + \frac{11}{6}u_h^i\right),\\
(u_h^i)_x &= \omega_1 (u_h^i)_{x,1} + \omega_2 (u_h^i)_{x,2} + \omega_3 (u_h^i)_{x,3},
\end{aligned}
$$

where the weights ω_i are given by

$$
\omega_i = \frac{\alpha_i}{\alpha_1 + \alpha_2 + \alpha_3}, \quad i = 1, 2, 3,
$$

with

$$
\alpha_i = \frac{d_i}{(c_e + \beta_i)^2}, \quad i = 1, 2, 3, \quad d_1 = 3/10,\ d_2 = 3/5,\ d_3 = 1/10.
$$

The parameter c_e is introduced to avoid the denominator to become 0. In the numerical studies $c_e = 10^{-6}$ was used. The role of the parameter c_e has been studied in several papers, see for example [1]. The value $c_e = Ch^2$ implies that the order of accuracy of the WENO reconstruction near discontinuities of the function is 3. However, for smooth solutions the order of accuracy is 5 if c_e is a constant. In practice, a fixed value line $c_e = 10^{-5}$ or $c_e = 10^{-6}$ is commonly used. In the numerical examples studied in Section 3, we could observe only negligible differences between solutions computed with $c_e = 10^{-6}$ and $c_e = h^2$. The values β_i are the so-called smooth indicators of the stencil [15]

$$
\begin{aligned}
\beta_1 &= \frac{13}{12}\left(\bar{u}_h^i - 2\bar{u}_h^{i+1} + \bar{u}_h^{i+2}\right)^2 + \frac{1}{4}\left(3\bar{u}_h^i - 4\bar{u}_h^{i+1} + \bar{u}_h^{i+2}\right)^2,\\
\beta_2 &= \frac{13}{12}\left(\bar{u}_h^{i-1} - 2\bar{u}_h^i + \bar{u}_h^{i+1}\right)^2 + \frac{1}{4}\left(\bar{u}_h^{i-1} - \bar{u}_h^{i+1}\right)^2,\\
\beta_3 &= \frac{13}{12}\left(\bar{u}_h^{i-2} - 2\bar{u}_h^{i-1} + \bar{u}_h^i\right)^2 + \frac{1}{4}\left(\bar{u}_h^{i-2} - 4\bar{u}_h^{i-1} + 3\bar{u}_h^i\right)^2,
\end{aligned}
$$

where $\bar{u}_h^i = \left(u_h^i - u_h^{i-1}\right)/h$ are the cell averages of the first spatial derivative. For $b(t,x) < 0$, one has to compute the following quantities

$$
\begin{aligned}
(u_h^i)_{x,1} &= -\frac{1}{h}\left(-\frac{u_h^{i+1}}{3} - \frac{u_h^i}{2} + u_h^{i-1} - \frac{u_h^{i-2}}{6}\right),\\
(u_h^i)_{x,2} &= -\frac{1}{h}\left(\frac{u_h^{i+2}}{6} - u_h^{i+1} + \frac{u_h^i}{2} + \frac{u_h^{i-1}}{3}\right),\\
(u_h^i)_{x,3} &= -\frac{1}{h}\left(-\frac{u_h^{i+3}}{3} + \frac{3}{2}u_h^{i+2} - 3u_h^{i+1} + \frac{11}{6}u_h^i\right),\\
\beta_1 &= \frac{13}{12}\left(\bar{u}_h^i - 2\bar{u}_h^{i-1} + \bar{u}_h^{i-2}\right)^2 + \frac{1}{4}\left(3\bar{u}_h^i - 4\bar{u}_h^{i-1} + \bar{u}_h^{i-2}\right)^2,\\
\beta_2 &= \frac{13}{12}\left(\bar{u}_h^{i+1} - 2\bar{u}_h^i + \bar{u}_h^{i-1}\right)^2 + \frac{1}{4}\left(\bar{u}_h^{i+1} - \bar{u}_h^{i-1}\right)^2,\\
\beta_3 &= \frac{13}{12}\left(\bar{u}_h^{i+2} - 2\bar{u}_h^{i+1} + \bar{u}_h^i\right)^2 + \frac{1}{4}\left(\bar{u}_h^{i+2} - 4\bar{u}_h^{i+1} + 3\bar{u}_h^i\right)^2,
\end{aligned}
$$

where now $\bar{u}_h^i = \left(u_h^{i+1} - u_h^i\right)/h$. All other quantities are the same as in the case $b(t, x_i) \geq 0$.

To compute the WENO approximation at and near the boundary, values on the boundary are extended in a constant way off Ω.

In [20], also a simple upwind scheme and an essentially non-oscillatory (ENO) extrapolation of the convective term were studied. In our experience, these approximations give less accurate results than the WENO approximation, see also Example 3.2 for a brief discussion of this topic. Similarly, results obtained with central differences as discretization for the convective term will be discussed in Examples 3.4 and 3.5.

Remark 1. This remark provides some comments on the stability and consistency of the proposed methods. The numerical analysis of the Euler and BDF2 projection methods in the space-continuous case or combined with finite element methods has been carried out by several authors, the reader is referred to [12] for a survey. The literature about projection methods in combination with finite difference schemes is rather scarce. In [2] a second order projection method is described and numerically (but not analytically) checked using a specialized higher order Godunov method for differencing the nonlinear convective terms. We are unaware of analysis for Runge–Kutta type projection methods as the one proposed here. A rigorous analysis of the method, even the semidiscrete, space-continuous scheme, is out of the scope of this paper. However, supported by the numerical experience we guess that the proposed method can be proven to be second order in time. On the other hand, the analysis (consistency) of the WENO scheme has been developed in [1]. In [1] the order of the WENO reconstructions is proved to be 5 in smooth regions and 3 near discontinuities. The analysis of the WENO schemes for a one-dimensional scalar conservation law with the third order TVD Runge–Kutta scheme as temporal discretization, assuming sufficient regularity for the solution and an appropriately chosen CFL condition, can be found in [15]. Altogether, as it can be concluded from the above discussion, the analysis of even the simplest method proposed in this paper (Euler projection method for nonlinear Navier–Stokes equations with fifth order WENO finite difference approximation of the convective term) seems to be a difficult task and it is at any rate out of the scope of the present paper.

Remark 2. As in the vast majority of papers studying WENO schemes, we have chosen an explicit evaluation of the convective term. However, one can wonder if a semi-implicit evaluation $u^n \cdot \nabla\tilde{u}^{n+1}$ or $\tilde{u}^n \cdot \nabla\tilde{u}^{n+1}$ is also possible. In [11] several predictor–corrector schemes with semi-implicit WENO evaluation of convective terms are considered. The authors state that a semi-implicit approach which does not involve a separate predictor step is also possible but there is the danger that the stencil determined by the weights at time t^n will cross the shock at time t^{n+1} and leads to instability. Several explicit and semi-implicit/predictor–corrector methods are compared in [11]. The semi-implicit methods are shown to reduce the CFL stability restriction from the value 1/2 (in the explicit methods) to the value 1. However, although the admissible time step is doubled by using semi-implicit methods, when considering the cost of semi-implicit methods it is evident in [11] that they do not outperform the explicit schemes.

2.7. Algorithmic aspects

For problems where an initial condition of the velocity is known, the discrete initial condition just took the values of the given initial condition at the nodes of the grid.

In all methods that need initialization steps, these steps were performed with the method Eul-ninc1.

The Laplacian of $\Delta\tilde{\boldsymbol{u}}^{n+1}$ was always discretized with the usual five points central finite differences.

For computing the divergence of $\tilde{\boldsymbol{u}}^{n+1}$, which is needed in the right-hand sides of the pressure Poisson equations, central finite differences were used in all interior nodes. The values of $\nabla\cdot\tilde{\boldsymbol{u}}^{n+1}$ at the boundary nodes determine, via the pressure Poisson equations (4), (12), (16), (17), and (18), essentially the pressure at the boundary. Projection schemes are known to compute discrete pressures which might exhibit boundary layers. We could observe that different approaches for computing $\nabla\cdot\tilde{\boldsymbol{u}}^{n+1}$ at the boundary in fact might notably influence the accuracy of the computed solutions, see Example 3.2 for a brief discussion. The following approaches were studied:

(1) It is set $\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} = 0$ in nodes at Dirichlet boundaries. There is no equation for the divergence of $\tilde{\boldsymbol{u}}^{n+1}$. However, at Dirichlet boundaries, it is $\tilde{\boldsymbol{u}}^{n+1} = \boldsymbol{u}^{n+1}$. Since $\boldsymbol{u}^{n+1}$ is assumed to be pointwise divergence-free, this property is in this method also transferred to $\tilde{\boldsymbol{u}}^{n+1}$ at Dirichlet boundaries.
(2) The function $\tilde{\boldsymbol{u}}^{n+1}$ is extrapolated constantly in the required direction and the central difference at the boundary node is applied.
(3) One-sided differences are used at the boundary. This approach is equivalent with extrapolating $\tilde{\boldsymbol{u}}^{n+1}$ linearly off the domain and applying a central difference.

Note that for a rectangular domain with the boundaries in coordinate direction, the divergence in the corners is determined by the values at the boundary. To avoid errors from finite difference approximations using Dirichlet boundary conditions, $\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} = 0$ was set at all corners also in methods 2 and 3. Method 1 does not depend on computed values. Computed values possess less impact in method 2 than in method 3 in the sense that the extrapolated values in method 2 do not depend on computed values.

The gradient of the pressure was computed with central differences in all nodes, extending the pressure off the domain in a constant way. Note that in the case of Dirichlet boundary conditions, the values of the gradient of the pressure in boundary nodes do not enter the schemes, such that in this case even the computation of the gradient of the pressure at the boundary can be omitted.

After having computed $\boldsymbol{u}^{n+1}$ in the scheme Eul-ninc1 from (5), the values of $\boldsymbol{u}^{n+1}$ at the boundary nodes were set to the values of the boundary condition at the nodes.

The choice of the stabilization parameter δ in the pressure Poisson equations (12), (16), (17), and (18) is based on known results and experience from finite element methods. For the diffusion-dominated regime, an optimal finite element error analysis for the transient Stokes equations is obtained for $\delta = \delta_0 h^2$, see [21], where h is the mesh width (diameter of the mesh cells). Also in [24], where the PSPG-type stabilization was used in an incremental scheme for solving the transient Stokes equations, a stabilization parameter of this form was used. The experience for smooth solutions in [21] and also our experience with the projection methods is that δ_0 should be chosen to be small. For the convection-dominated regime, there is a proposal for the steady-state Oseen equations to choose $\delta = \delta_0 \min\{h, h^2/\nu\}$, e.g., see [26, p. 475]. Thus, the asymptotic choice of δ seems to be clear for the diffusion-dominated case, but the situation for the convection-dominated regime seems to be unclear. All numerical results which will be presented in Section 3 are for laminar flow problems on meshes which are sufficiently fine to resolve all important scales of the flow fields. In this situation, one is, in some sense, in the transition zone from diffusion-dominated to convection-dominated problems. We found out with numerical studies that usually $\delta = \delta_0 h^2$ was often a better choice than $\delta = \delta_0 h$, see Example 3.1 for an illustration. Here, δ_0 can be chosen to be small for smooth solutions and larger, up to the order of unity, for more complicated solutions. The concrete choice will be given for all presented results.

The solution of the pressure Poisson equations is only determined up to an additive constant due to the Neumann boundary conditions. Fixing one value and using the direct sparse solver umfpack [9], we could not observe any difficulties to compute a solution. Afterwards, this solution was projected into $L_0^2(\Omega)$.

Examples 3.4 and 3.5 consider problems defined in non-rectangular domains. The basic idea for extending the WENO finite difference method to such domains consists in including Ω in a tensor product domain $\tilde{\Omega}$. Then, the equations are approximated in $\tilde{\Omega}$ while setting the values at the nodes in $\tilde{\Omega}\backslash\overline{\Omega}$ to zero.

It should be noted that the WENO methodology in conjunction with finite differences can only be applied to uniform or smooth meshes, see [30, Section 3.3]. For this reason all the meshes considered in the numerical studies are uniform meshes.

All simulations were performed with two codes to double check the results. One of the codes was MooNMD [18].

3. Numerical studies

This section presents several numerical studies for assessing the proposed methods. We think it to be important not only to show the favorable properties of these methods but also to present their shortcomings.

A number of problems in two dimensions will be considered. In Example 3.1, a problem will be studied where the temporal error dominates. With this example, the order of convergence with decreasing time step can be assessed. An analytical solution is considered in Example 3.2 to study the accuracy of the computed solutions in different norms. Example 3.3 presents results for a regularized driven cavity. In this example, the velocity possesses a boundary layer. The last two examples consider problems in domains which are not rectangular. Example 3.4 studies a flow across a step and Example 3.5 a flow around a cylinder.

Grids consisting of squares were used in the simulations.

Example 3.1 *(A problem with small spatial discretization error (similar to [19])).* Let $T = 1$ and $\Omega = (0, 1)^2$. The right-hand side $\boldsymbol{f}$, the initial condition $\boldsymbol{u}_0$, and the non-homogeneous Dirichlet boundary conditions are chosen such that

$$\boldsymbol{u} = \begin{pmatrix} t^3 y \\ t^2 x \end{pmatrix}, \quad p(x, y) = tx + y - \frac{t+1}{2}$$

is the solution of the Navier–Stokes equations (1). The solution is linear with respect to the spatial variables. Thus, one expects mainly errors from the temporal discretization and the splitting.

The simulations were performed on a grid with mesh cells of diameter $h = \sqrt{2}/128$. Among the methods for computing the divergence of $\tilde{\boldsymbol{u}}^{n+1}$ at the boundary, methods 2 and 3 gave similar results for the velocity, with one exception. In all numerical tests which we performed with BDF-inc and method 3, the simulations blew up. The velocity errors for methods 2 and 3 were much lower and the order of convergence considerably higher than for method 1. With respect to the pressure error, method 2 gave the most accurate results. Below, results obtained with method 2 are presented. The PSPG stabilization parameter in the incremental schemes was set to be $\delta = 0.001h^2$.

In this example, also results for the diffusion-dominated regime are shown, see Fig. 1. Concerning the velocity errors, the schemes Eul-inc and BDF-inc (with the exception of the error at the final time) converge with an order of approximately 1.5, whereas a first order convergence is observed for Eul-ninc1 and Eul-ninc2. The order of convergence for the pressure is for the non-incremental schemes around 0.8 and for the incremental schemes approximately 1.

Results for a convection-dominated case are depicted in Fig. 2. One can observe some order reduction for the BDF-inc and Eul-inc schemes. The orders for the convergence of the velocity errors is somewhat below 1.5. Nevertheless, the most accurate results were obtained with BDF-inc. The results computed with RK-inc belong to the least accurate ones.

Fig. 3 presents results for the convection-dominated case and a stabilization parameter of the PSPG method which scales like the mesh width h and not like h^2. Comparing these results with the corresponding results from Fig. 2, it can be seen that in particular for finer time steps, the results with $\delta = 0.001h^2$ are more accurate.

It can be observed, also in other examples, that with respect to the $L^2(\Omega)$ error of the velocity, Eul-ninc1 is considerably more accurate than Eul-ninc2 if method 2 is used for computing $\nabla \cdot \tilde{\boldsymbol{u}}^{n+1}$ at the boundary. With method 3 there were generally only much smaller differences.

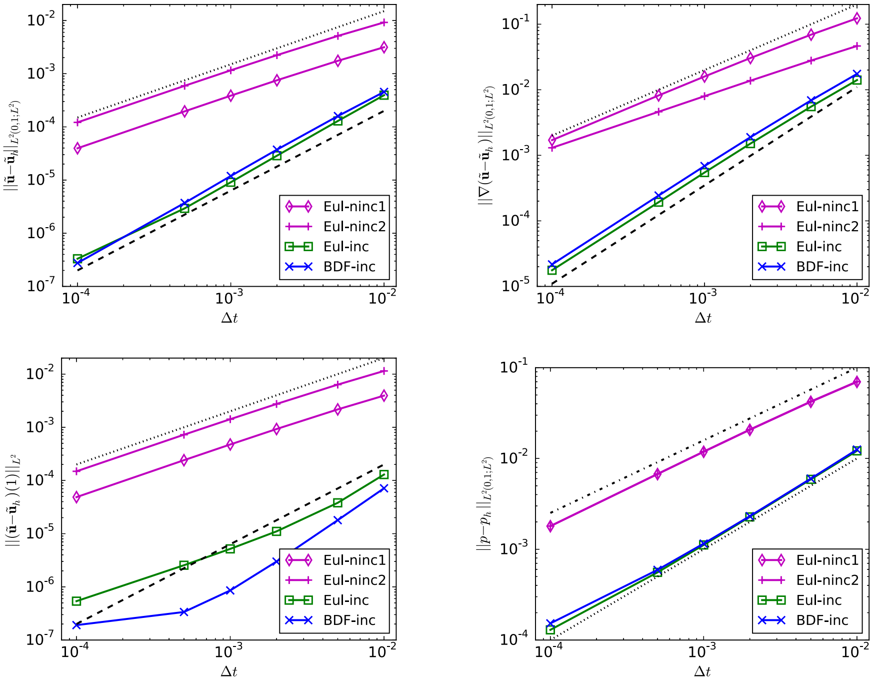

Fig. 1. Example 3.1, results for $\nu = 1$ and $\delta = 0.001h^2$. The asymptotics are of order 0.8 (dash-dotted line), 1 (dotted line), and 1.5 (dashed line). All simulations with the scheme RK-inc blew up.

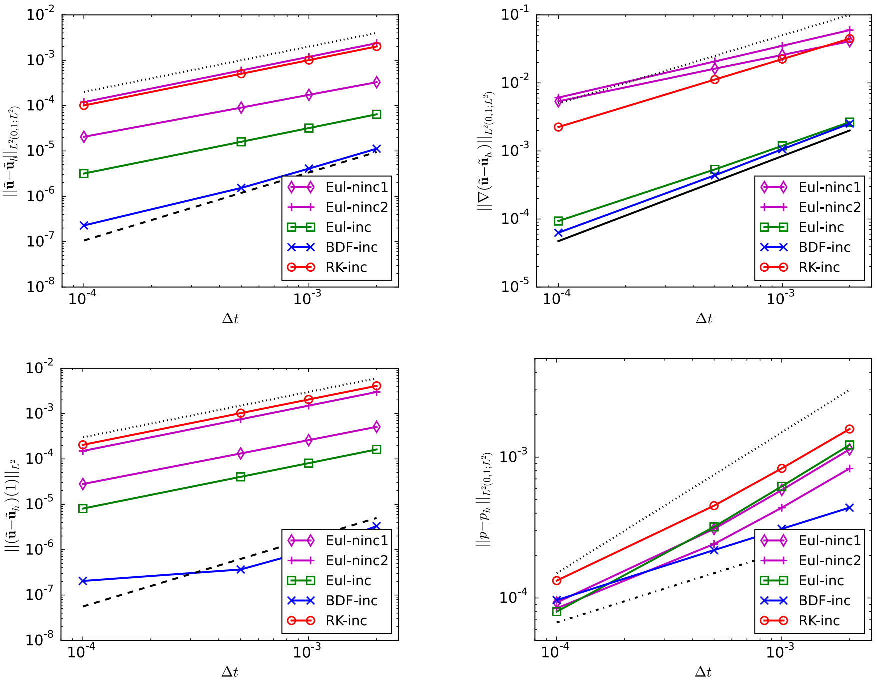

Fig. 2. Example 3.1, results for $\nu = 10^{-3}$ and $\delta = 0.001h^2$. The asymptotics are of order 0.5 (dash-dotted line), 1 (dotted line), 1.25 (continuous line), and 1.5 (dashed line).

Example 3.2 *(Analytical solution).* An example in the unit square $\Omega = (0, 1)^2$ is considered with the prescribed solution

$$\boldsymbol{u} = 2\pi \sin(t) \begin{pmatrix} \sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ \sin(\pi x)\cos(\pi x)\sin^2(\pi y) \end{pmatrix}, \quad p = 20\sin(t)\left(x^2 y - \frac{1}{6}\right).$$

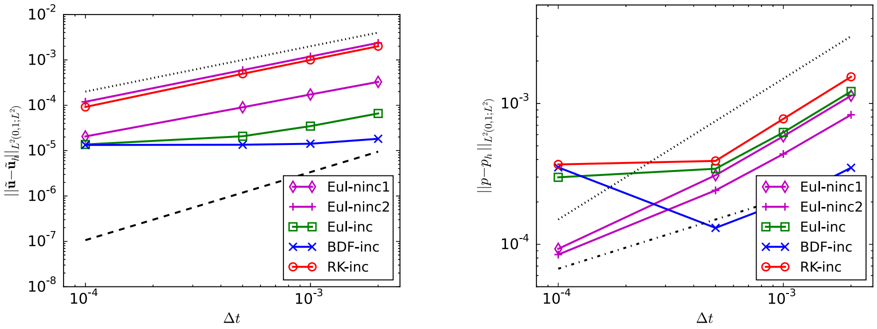

Fig. 3. Example 3.1, results for $\nu = 10^{-3}$ and $\delta = 0.001h$, same axes and asymptotics as in Fig. 2.

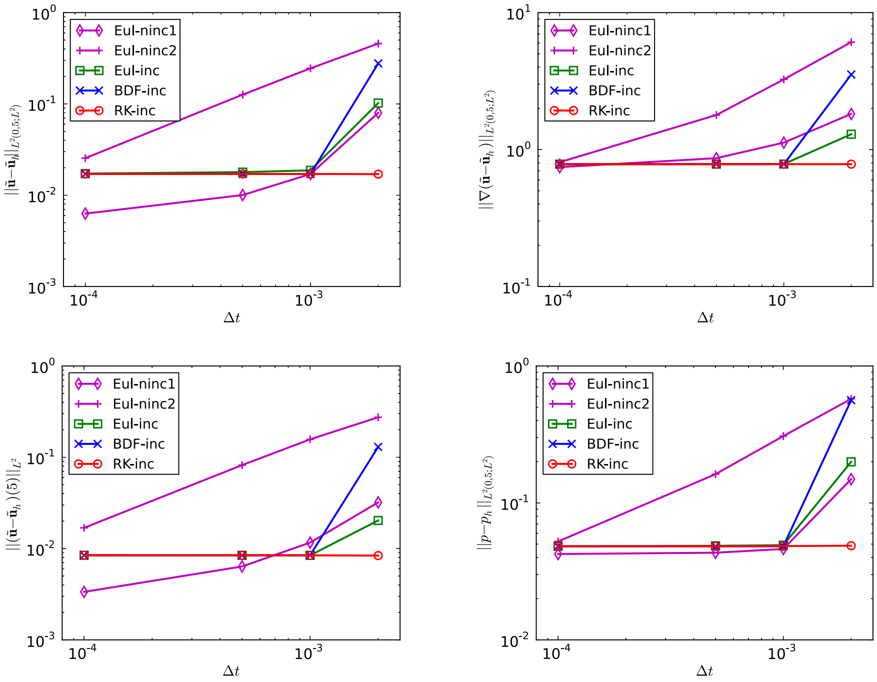

Fig. 4. Example 3.2, results for $\nu = 10^{-3}$, $h = \sqrt{2}/64$ (level 7).

The length of the time interval is chosen to be $T = 5$ and Dirichlet boundary conditions were applied. Only results for the convection-dominated case $\nu = 10^{-3}$ will be presented.

Most of the presented results were computed with method 2 for the computation of $\nabla \cdot \tilde{\boldsymbol{u}}^{n+1}$ at the boundary nodes. For this example as well an illustration of the results obtained with the other methods is given. In addition, a brief comparison of results obtained with the simple upwind stabilization, ENO, and WENO is presented.

In the incremental schemes, the PSPG stabilization parameter $\delta = 0.1h^2$ was used. Errors which are of interest in the error analysis were monitored. Results obtained on a fixed spatial grid and for different time steps are presented in Fig. 4. One can see that on a spatial grid with $h = \sqrt{2}/64$ all incremental schemes give the same results if the time step is sufficiently small. There are differences only for $\Delta t = 2 \cdot 10^{-3}$. For $\Delta t = 5 \cdot 10^{-3}$, the simulations with most of the schemes blew up. In addition, one can see that in the case of small time steps the non-incremental scheme Eul-ninc1 gives more accurate results than all other studied methods for the $L^2(\Omega)$ error of the velocity. Finally, the method RK-inc shows a good behavior in this example in the sense that it reaches the level of the spatial error already for the largest value of Δt while the other methods require smaller values of Δt.

Fig. 5 shows results obtained for a fixed time step and on different spatial grids. The grid level l corresponds to a mesh width $h = \sqrt{2}/2^{l-1}$. Again, all incremental schemes give more or less the same results, save on the finest grid. On level 9, one can see some instabilities. The simulation with BDF-inc even blew up (but with the alternative version of BDF-inc described at the end of Section 2.4 it succeeded and these results are presented). The chosen PSPG stabilization parameter becomes too small on the finest level to stabilize the violation of the discrete inf-sup condition appropriately. In fact, we could observe that simulations with larger parameters, e.g., $\delta = 10h^2$, do not show instabilities and do not blow up. Again,

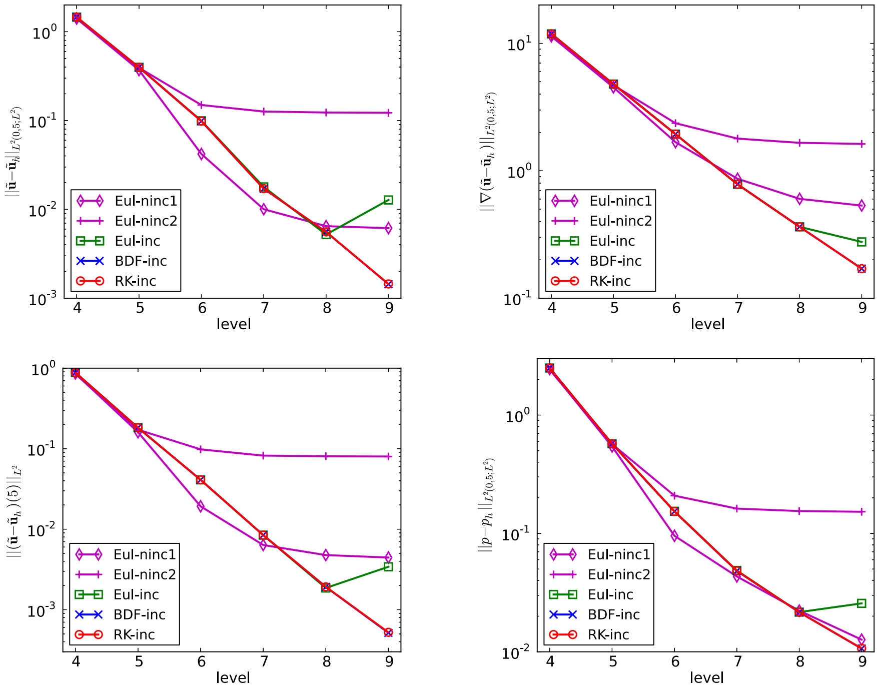

Fig. 5. Example 3.2, results for $\nu = 10^{-3}$, $\Delta t = 5 \cdot 10^{-4}$, BDF-inc on level 9: alternative variant. Curve for BDF-inc is below curve for RK-inc.

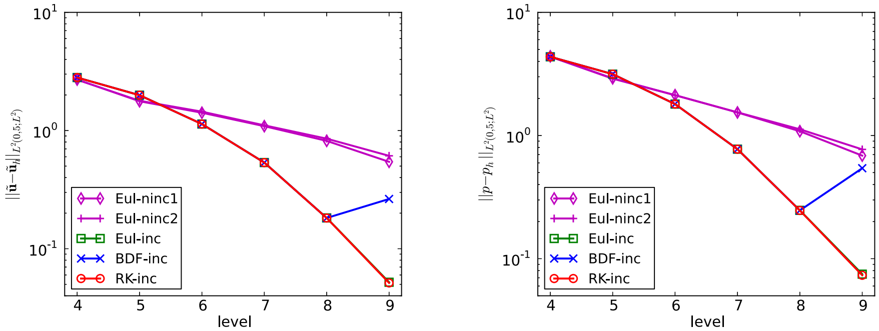

Fig. 6. Example 3.2, results for $\nu = 10^{-3}$, $\Delta t = 5 \cdot 10^{-4}$, method 1 for computing $\nabla \cdot \tilde{\boldsymbol{u}}^{n+1}$. Note the different scaling of the y-axes compared with Fig. 5.

one can see that the results computed with Eul-ninc1 are of similar accuracy as those of the incremental methods, at least on coarser grids.

Some results for the two other methods for computing $\nabla \cdot \tilde{\boldsymbol{u}}^{n+1}$ at boundary nodes are presented in Figs. 6 and 7. For method 1, one can see that all schemes provide less accurate solutions than for method 2. This statement holds in particular for the non-incremental schemes, which produce with method 1 less accurate results than the incremental schemes. The accuracy of the results obtained with methods 2 and 3 is for the most schemes similar. Only with Eul-ninc1, one obtains clearly worse results with method 3.

A representative result is presented which shows that the solutions computed with the WENO scheme are more accurate than the solutions obtained with the ENO scheme and both results are much more accurate than the solutions computed with simple upwinding, see Table 1.

Example 3.3 *(A regularized lid driven cavity example).* Lid driven cavity examples are very popular and there are several proposals for such examples in the literature. Here, a regularized lid driven cavity will be proposed and studied where the velocity can be expected to belong to $C^2(\Omega) \cap C(\overline{\Omega})$, as one needs to get second order accuracy with finite difference methods, and which is consistent in the sense that $\nabla \cdot \boldsymbol{u} = 0$ holds also at the corners of the domain.

The lid driven cavity example is defined in $\Omega = (0,1)^2$. There are no-slip conditions at the boundaries $x = 0$, $x = 1$, and $y = 0$. In the classical lid driven cavity example, e.g., as studied in [10], the velocity at $y = 1$ is chosen to be $\boldsymbol{u} = (1,0)^T$ (the concrete choice of the values at the upper corners is not specified in [10]). At any rate, the boundary condition of the velocity has a jump such that the velocity solution is not yet in $H^1(\Omega)$. Strictly speaking, this setup is not even suited

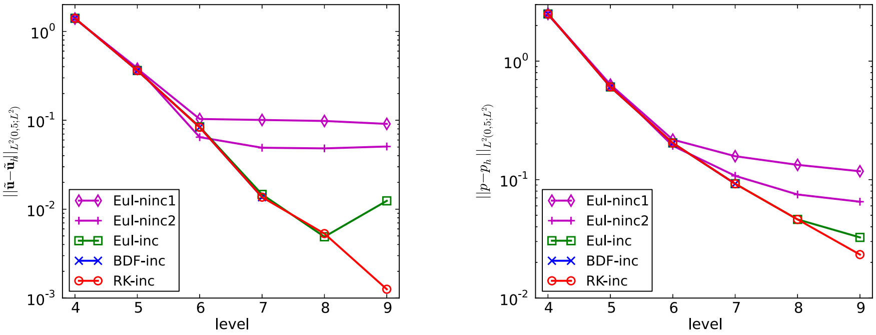

Fig. 7. Example 3.2, results for $\nu = 10^{-3}$, $\Delta t = 5 \cdot 10^{-4}$, method 3 for computing $\nabla \cdot \tilde{\boldsymbol{u}}^{n+1}$. The simulations with BDF-inc blew up on the two finest levels. The scaling of the y-axes is the same as in Fig. 5.

Table 1
Example 3.2, results for $\nu = 10^{-3}$, $h = \sqrt{2}/64$ (level 7), $\Delta t = 5 \cdot 10^{-4}$, divergence computation with method 2.

	Eul-ninc1	Eul-ninc2	Eul-inc	BDF-inc	RK-inc
$\|u - u_h\|_{L^2(L^2)}$					
upwind	1.058	1.020	1.054	1.053	1.053
ENO	1.262e−2	1.277e−1	4.673e−2	4.559e−2	4.559e−2
WENO	1.004e−2	1.263e−1	1.790e−2	1.710e−2	1.711e−2
$\|p - p_h\|_{L^2(L^2)}$					
upwind	1.901	1.839	1.895	1.895	1.895
ENO	4.439e−2	1.645e−1	8.532e−2	8.454e−2	8.470e−2
WENO	4.330e−2	1.621e−1	4.863e−2	4.819e−2	4.828e−2

for finite element methods. In the last years, several regularized driven cavity examples were proposed. As mentioned already, the divergence of the solution should pointwise vanish and the divergence at the corners is determined by the boundary values. The regularized examples we could find in the literature do not meet the requirement of the velocity being divergence-free at the corners. For this reason, we propose a new regularized example, where the velocity at $y = 1$ is given by

$$\boldsymbol{u}(x,1) = \begin{pmatrix} u_1(x) \\ 0 \end{pmatrix},$$

$$u_1(x) = \begin{cases} 1 - \frac{1}{4}\left(1 - \cos\left(\frac{x_1 - x}{x_1}\pi\right)\right)^2 & \text{for } x \in [0, x_1], \\ 1 & \text{for } x \in (x_1, 1 - x_1), \\ 1 - \frac{1}{4}\left(1 - \cos\left(\frac{x - (1 - x_1)}{x_1}\pi\right)\right)^2 & \text{for } x \in [1 - x_1, 1]. \end{cases} \tag{19}$$

It can be checked that $u_1(x)$ is twice continuously differentiable. In the simulations $x_1 = 0.1$ was used.

Numerical results for $\nu^{-1} = \mathrm{Re} = 10\,000$ will be presented. It is known for the classical lid driven cavity example that stationary solutions become unstable at a Reynolds number of around $\mathrm{Re} = 8000$ and that at $\mathrm{Re} = 10\,000$ there is a stable periodic solution, e.g., see [4,6,32]. On the one hand, the driving (19) is a little bit smaller than in the classical setup because (19) tends to zero at the corners. But on the other hand, the considered Reynolds number is considerably larger than the Reynolds number for stable stationary solutions of the classical problem. Altogether, one expects for the studied regularized lid driven cavity example also a stable periodic solution.

The computed solutions will be compared with a solution obtained with a fully implicit finite element approach. In this approach, the inf-sup stable Q_2/P_1^{disc} pair of finite element spaces was used, i.e., the velocity is discretized with a continuous, piecewise biquadratic function and the pressure with a discontinuous piecewise linear function. This pair of finite element spaces is very popular and known to perform well, e.g., see [17]. As discretization in time, the Crank–Nicolson scheme was used. Altogether, for a similar number of degrees of freedom, one expects to obtain more accurate results with this finite element approach than with the studied finite difference methods for two reasons. First, in the finite element approach there is no splitting error and second, the fully nonlinear problem is solved in each discrete time. This expectation was met, e.g., in the numerical studies in [5].

An initial condition is not known for this example. An impulsive start was used for all simulations, i.e., $\boldsymbol{u} = \boldsymbol{0}$ on all internal nodes. The time step was chosen to be $\Delta t = 10^{-3}$ and the PSPG stabilization parameter in the incremental schemes was set to be $\delta = h^2$. The high Reynolds number leads to small scales of the flow for whose resolution a rather fine grid is necessary. Figs. 8 and 9 present results obtained for $h = \sqrt{2}/320$, which led to 206 082 velocity degrees of freedom and 103 041 pressure degrees of freedom (including nodes at Dirichlet boundaries). The velocity exhibits a boundary layer at

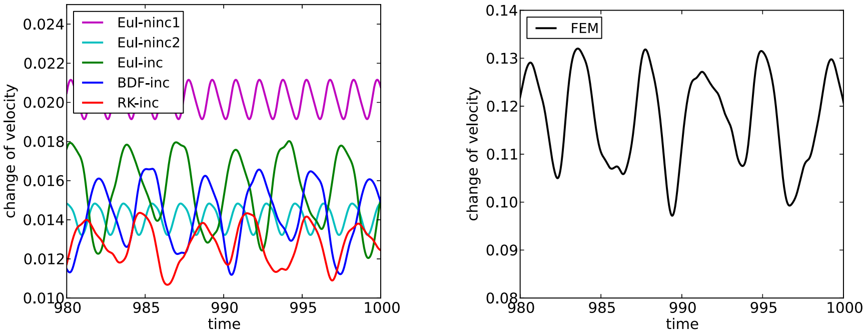

Fig. 8. Example 3.3, change of the Euclidean norm of the computed velocity $\|\tilde{\boldsymbol{u}}^{n+1} - \tilde{\boldsymbol{u}}^{n}\|_{l^2}$.

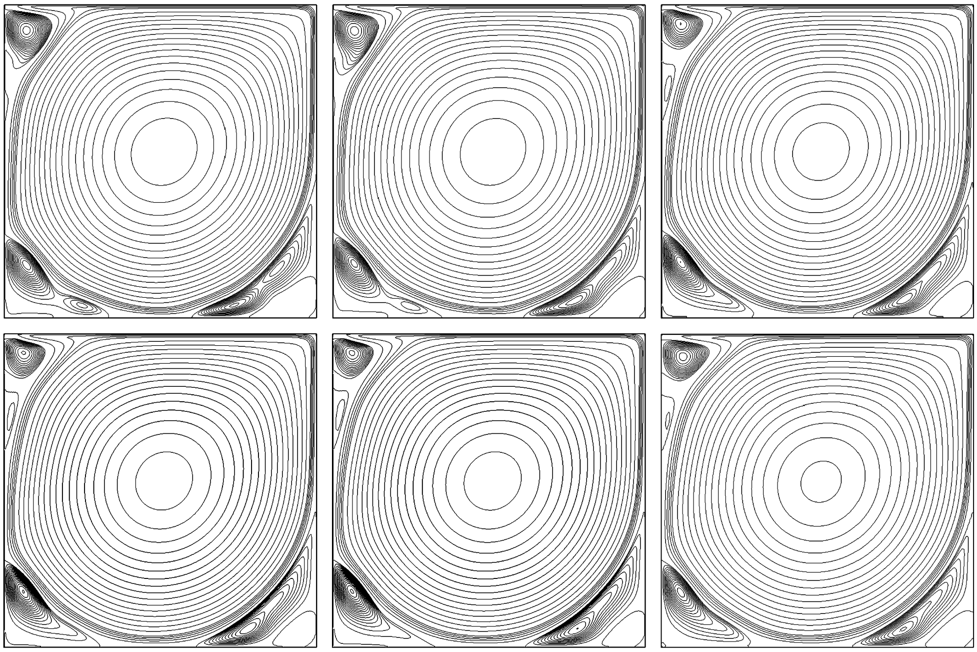

Fig. 9. Example 3.3, streamlines of the velocity for $\mathrm{Re} = \nu^{-1} = 10\,000$, $h = \sqrt{2}/320$: Eul-ninc1, Eul-ninc2, Eul-inc, BDF-inc, RK-inc, reference finite element solution (top left to bottom right), 20 equidistant intervals in $[-0.15, -0.01]$, 6 equidistant intervals in $[-0.01, 0]$, and 40 equidistant intervals in $[0, 0.01]$.

$y = 1$, a large vortex in the center of the domain, and smaller vortices in all corners save the upper right corner. None of the studied schemes reaches a steady state. Results concerning the change of the discrete velocity from time step to time step, see Fig. 8, show that a periodic solution is computed. Comparisons with visualizations of the flow field confirm that the oscillations of the change of the velocity field reflect indeed a periodic behavior of the flow field (not presented in detail for the sake of brevity). The length of the period is considerably longer for the incremental methods. This behavior corresponds better to the results obtained with the finite element method. Snapshots of the solutions obtained with the studied schemes are presented in Fig. 9. To find discrete times where all schemes are at a comparable state in their period, the maximal value of the streamfunction was monitored. Then, discrete times were picked where this value has a local maximum in time and this maximum is attained in the lower left corner of the domain. It can be seen in Fig. 9 that on the one hand the solutions computed with Eul-ninc1 and Eul-ninc2 look similar and on the other hand the solutions obtained with all incremental schemes are almost identical. There are small differences between the results of the non-incremental and the incremental schemes, e.g., in the form of the eddy in the lower left corner. It was found that on a finer grid with $h = \sqrt{2}/640$ the solutions obtained with Eul-ninc1 and Eul-ninc2 are similar to the solutions with the incremental schemes and the reference solution depicted in Fig. 9. In comparing the solutions with results from the literature one should keep in mind that the boundary condition at the top of the cavity is a new proposal. Nevertheless, the snapshots look very similar to the results presented in [4, Figure 9, second row left] (note that the velocity on the upper lid is right to left in [4]).

Altogether the proposed schemes compute qualitatively correct results for this flow with a boundary layer.

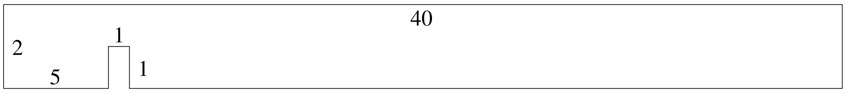

Fig. 10. Example 3.4, sketch of the domain (not to scale).

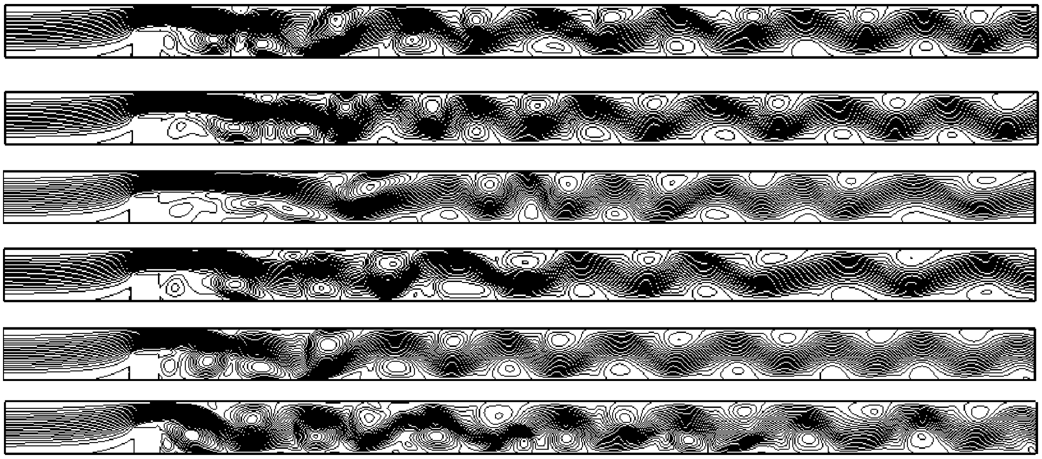

Fig. 11. Example 3.4, streamlines of the velocity at $t = 50$ for $\nu = 10^{-3}$ and $h = \sqrt{2}/32$: Eul-ninc1, Eul-ninc2, Eul-inc, BDF-inc, RK-inc, and reference finite element solution (top to bottom), 41 equidistant streamlines in $[-1, 3]$.

Example 3.4 *(Flow across a step).* The simulation of flows across bodies is a situation which often occurs in applications. Because of the re-entrant corners, the flow across a step is considered to be as one of the more challenging setups.

The setup of this example is similar to [16]. The domain for the simulations is depicted in Fig. 10. At the inlet $x = 0$ and the outlet $x = 40$, the same parabolic profile was prescribed

$$\boldsymbol{u}(t, \boldsymbol{x}) = U(t)\begin{pmatrix} 1.5y(2-y) \\ 0 \end{pmatrix} \tag{20}$$

with

$$U(t) = \begin{cases} \sin(\pi t/8) & \text{if } t \in [0, 4], \\ 1 & \text{if } t \geq 4. \end{cases}$$

Due to the parabolic inflow profile (20), in contrast to the constant profile in [16], the width of the channel was chosen to be smaller here. The viscosity was chosen to be $\nu = 10^{-3}$. With this setup, vortices are created behind the step and a vortex street develops. The results obtained with the studied schemes will be compared with the results computed with the same finite element method as described in Example 3.3.

Snapshots of solutions obtained with $\Delta t = 10^{-3}$ on a grid with squares of diameter $h = \sqrt{2}/32$ and with the PSPG stabilization parameter $\delta = h^2$ for the incremental methods are presented in Fig. 11. This mesh results in 166 530 velocity degrees of freedom and 83 265 pressure degrees of freedom (including boundary nodes and the step). The reference solution with the finite element method was computed with the same discretization as described in Example 3.3 on a grid with $h = \sqrt{2}/128$, which results in 2 599 682/970 752 velocity/pressure degrees of freedom, and with the time step $\Delta t = 10^{-2}$. One can see in Fig. 11 that all schemes compute qualitatively a correct solution. The solutions with the projection schemes look smoother than the reference solution, which indicates that the projection methods introduce a certain amount of numerical viscosity.

Applying in this example central differences for the discretization of the convective term led to a blow-up of the simulations for all projection schemes.

Example 3.5 *(Flow around a cylinder).* This example is a popular benchmark problem defined in [27]. The domain is given by

$$\Omega = (0, 2.2) \times (0, 0.41) / \left\{ (x, y) \mid (x - 0.2)^2 + (y - 0.2)^2 \leq 0.05^2 \right\}$$

and the time interval is $[0, 8]$. At the inlet and the outlet, the prescribed velocity profile has the form

$$\boldsymbol{u}(0, y) = \boldsymbol{u}(2.2, y) = \frac{6}{0.41^2} \sin\left(\frac{\pi t}{8}\right) \begin{pmatrix} y(0.41 - y) \\ 0 \end{pmatrix}.$$

On all other boundaries, no-slip conditions are applied. The initial velocity is $\boldsymbol{u} = \boldsymbol{0}$ and the viscosity is $\nu = 10^{-3}$.

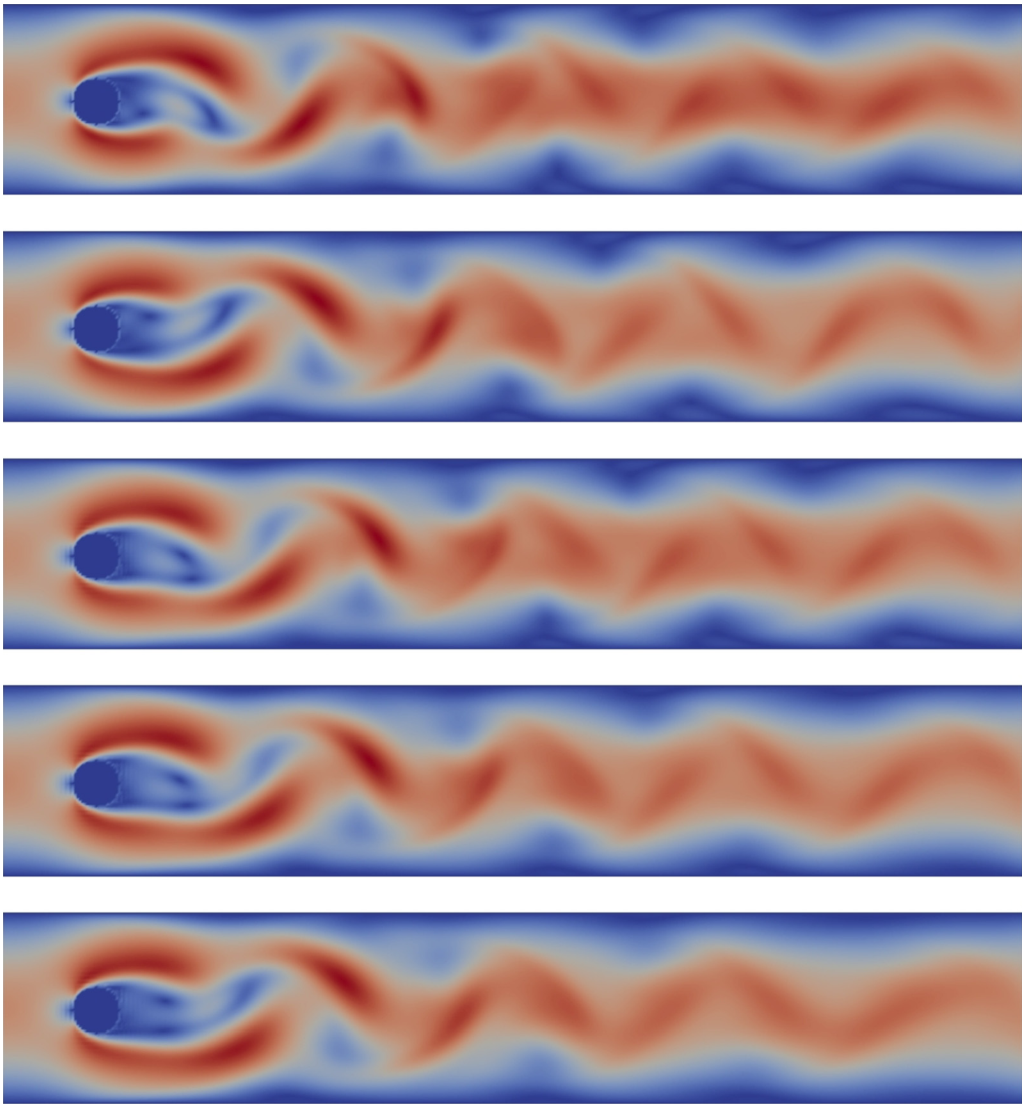

Fig. 12. Example 3.5, velocity field at $t = 6$: Eul-ninc1, Eul-ninc2, Eul-inc, BDF-inc, RK-inc (top to bottom).

While increasing the inflow, a vortex street develops behind the cylinder, starting at around $t = 4$. Important parameters of the flow are the drag coefficient at the cylinder, the lift coefficient, and the difference of the pressure between the front and the back of the cylinder

$$\Delta p(t) = p(t; 0.15, 0.2) - p(t; 0.25, 0.2).$$

The most accurate reference values for the maximal drag and lift coefficient and the pressure difference can be found in [22].

An additional difficulty compared with Example 3.4 is that the domain where the finite difference method has to be switched off is not aligned with the grid lines. Using the strategy explained in Section 2.7 leads thus to a rather inaccurate approximation of the cylinder. Also the calculation of the drag and lift coefficient becomes on the one hand complicated and on the other hand, the results are influenced by the inaccurate approximation of the boundary. Considering the drag coefficient, the formula given in [27] can be reformulated in the form

$$c_{\mathrm{d}} = -20 \int\limits_{\partial\Omega_{\mathrm{cyl}}} \left(\nu(\partial_y u_1 - \partial_x u_2) n_y - p n_x \right) \, ds,$$

where $\partial\Omega_{\mathrm{cyl}}$ is the boundary of the cylinder, u_1 and u_2 are the components of the velocity field, and $(n_x, n_y)^T$ is the unit normal vector pointing outside Ω. In our approach, the integral is evaluated with the midpoint rule approximating the cut of the circle with the mesh cell $\Gamma \subset \partial\Omega_{\mathrm{cyl}}$ having cut points S_1 and S_2 with the segment $[S_1, S_2]$,

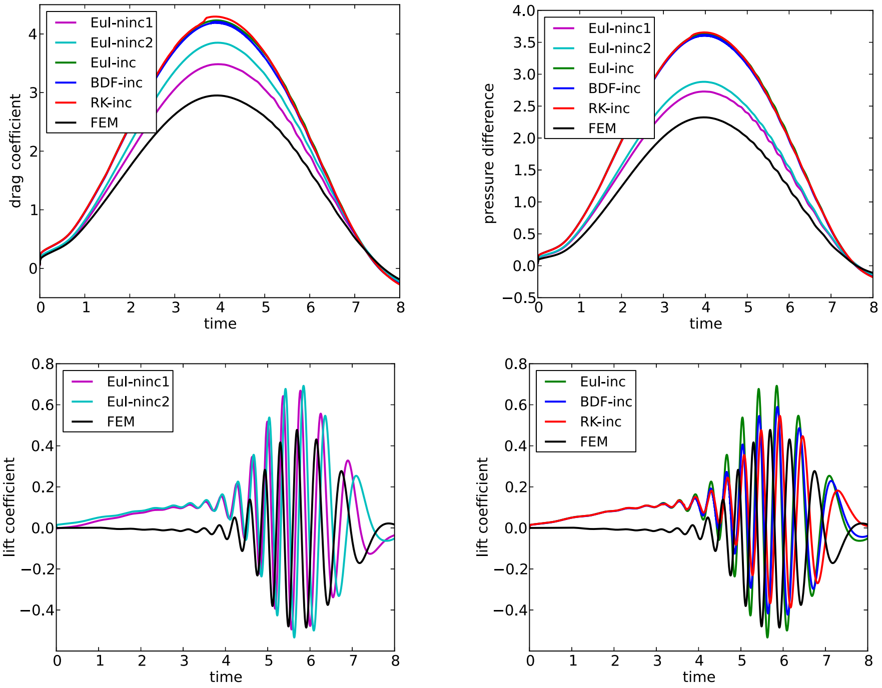

Fig. 13. Example 3.5, computed benchmark coefficients with $h = \sqrt{2}/200$, $\Delta t = 10^{-3}$. The finite element curve is from the reference simulation in [22].

$$\int_\Gamma (\nu(\partial_y u_1 - \partial_x u_2)n_y - pn_x)\, ds \approx \|S_1 - S_2\|_2 \left(\nu(\partial_y u^1 - \partial_x u^2)\tilde{n}_y - p\tilde{n}_x\right)(M),$$

where $M = (S_1 + S_2)/2$, $\|\cdot\|_2$ is the Euclidean norm, the normal vector is approximated by $(\tilde{n}_x, \tilde{n}_y) = (C - M)/\|C - M\|_2$ with $C = (0.2, 0.2)$ being the center of the cylinder. To evaluate $\partial_y u_1$ and $\partial_x u_2$, the bilinear polynomials that interpolate u_1 and u_2 at the four nodes that define the mesh cell were evaluated (note that the nodes in the cylinder has the value $\mathbf{0}$ and thus are a continuous extension of the boundary condition) and then the spatial derivatives were taken and evaluated at M. Two approaches were studied to evaluate the pressure. First the values of the pressure at the vertices of the mesh that do not fall inside the circle were considered and the arithmetic mean of these values was taken as an approximation to the value of the pressure at M. Alternatively, the value of the pressure node closest to M was used. Both variants to incorporate the pressure led qualitatively to the same results, showing quantitatively only insignificant differences. The results presented below are those computed with the first approach.

The stabilization parameter in the incremental schemes was chosen to be $\delta = h^2$. For $\delta = 0.1h^2$, very similar results were obtained.

Fig. 12 presents the flow fields for the studied methods at $t = 6$. It can be seen that all methods predict a vortex street, as it is qualitatively correct.

An assessment of the methods can be performed with the results concerning the benchmark coefficients shown in Fig. 13. These results were obtained at a grid with mesh size $h = \sqrt{2}/200$, leading to around 70 000 velocity degrees of freedom and 35 000 pressure degrees of freedom, and a time step of $\Delta t = 10^{-3}$. All methods give similar results in the sense that all coefficients are overpredicted. The incremental schemes show a large overprediction of the drag coefficient and the pressure difference. A somewhat smaller overprediction was obtained with the non-incremental schemes. Considering the peaks of the lift coefficient, one can conclude that the vortex shedding computed with all schemes is somewhat slower than predicted by the reference solution. Altogether, the methods predicted a qualitatively correct behavior of the flow.

The results presented in Fig. 14 will be used to discuss the question if quantitative correct results can be obtained with the studied methods. The answer is two-fold. Considering the global flow behavior, which is characterized by the vortex shedding, the answer is yes. Using finer meshes in time and space shows that the positions of the peaks of the lift coefficient get closer to the corresponding positions of the reference solution, which indicates an increasing accuracy of predicting the vortex shedding. Considering local properties, like the lift coefficient, quantitative better results cannot be observed. The same statement holds true for the drag coefficient and the pressure difference (not presented for the sake of brevity). Concerning these local properties, it should be noted that the pressure at the boundary possesses a strong impact on all coefficients. In addition, for a similar example of a flow around a cylinder considerable overpredictions of drag and lift coefficients were obtained in [5] also for the incremental Euler scheme in combination with a finite element method. Thus, the reason for the discrepancy to the reference benchmark coefficients is in our opinion three-fold: the approximation

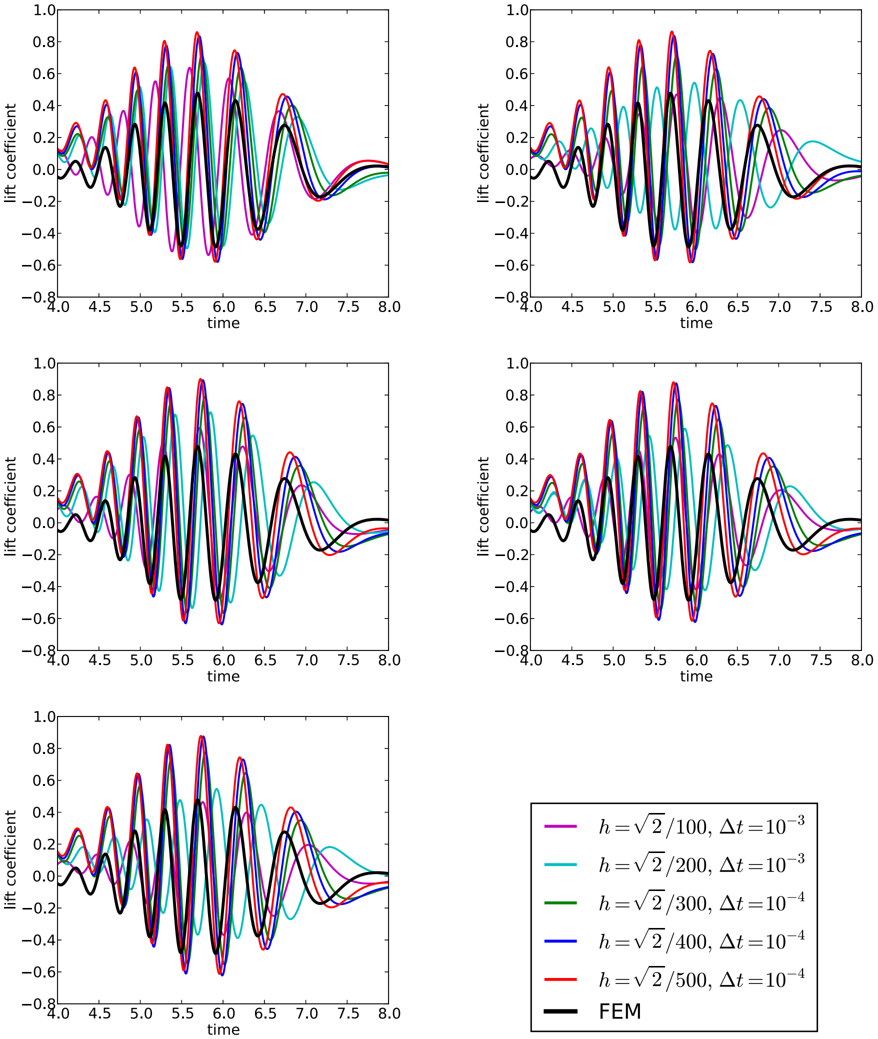

Fig. 14. Example 3.5, lift coefficients on different meshes: Eul-ninc1, Eul-ninc2, Eul-inc, BDF-inc, RK-inc (left to right, top to bottom). The finite element curve is from the reference simulation in [22].

of the domain, the approximations in the calculation of the coefficients, and the problems of projection schemes to compute accurate pressure approximations in a vicinity of boundaries.

Results computed with central differences for discretizing the convective term are presented in Fig. 15. It can be seen that the coefficients are usually considerably less accurate than those obtained with the WENO scheme (note that some ordinates are scaled differently than in Fig. 13). There are also notable oscillations of the pressure difference which cannot be observed in the reference curve. This feature indicates spurious oscillations in the flow field. With respect to these aspects, the results obtained with the WENO scheme are more accurate. Only the intervals of the vortex shedding were computed somewhat more accurately with central differences, with exception of Eul-ninc1.

4. Summary and outlook

This paper studied the use of the WENO finite difference method for the simulation of incompressible flows. Several non-incremental and incremental projection schemes were considered. Velocity and pressure were defined on the same grid. To account for the violation of the discrete inf-sup condition in the incremental schemes, a PSPG-type stabilization was introduced. Numerical studies were performed for examples with different features and different complexity.

In the numerical studies, the proposed methods computed in all cases qualitatively correct solutions. Quantitative differences to prescribed solutions or reference solutions were mainly caused by known shortcomings of projection methods, e.g., the impact of the splitting error and the usually inaccurate approximation of the pressure in a vicinity of boundaries. Nevertheless, differences in the accuracy of the proposed schemes could be observed, e.g., in the period of the solutions in the driven cavity example. The computing times for all studied schemes were in all examples similar. In our opinion, among

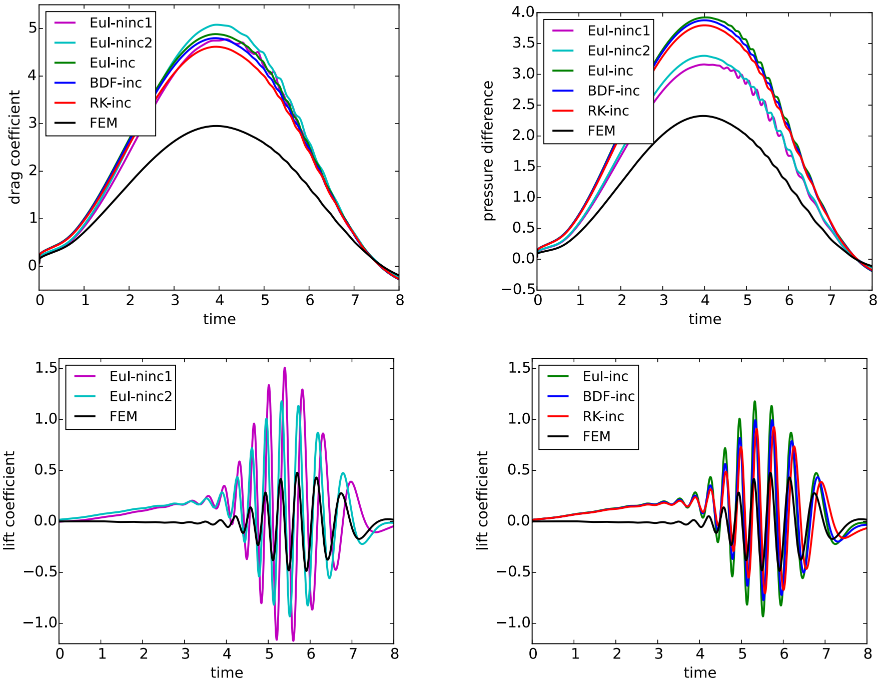

Fig. 15. Example 3.5, computed benchmark coefficients with central differences for discretizing the convective term with $h = \sqrt{2}/200$, $\Delta t = 10^{-3}$. The finite element curve is from the reference simulation in [22].

the considered schemes, the incremental schemes with semi-implicit temporal discretization (Eul-inc, BDF-inc) should be preferred.

As pointed out in [4], the challenge is to find a scheme for the discretization of the convective term that ensures both accuracy and stability. While the stability of first order schemes is very good, their accuracy is poor. On the contrary, second order schemes are more accurate but their stability is in general not ensured. From the numerical studies presented in this paper and from further studies we performed, it turned out that the WENO finite difference scheme is an attractive choice for the convection-dominated regime. The results obtained with this method were never notable less accurate than the results obtained with central finite differences, sometimes even considerably more accurate (Example 3.5), and sometimes the better stability of the WENO scheme was very important (Example 3.4).

To diminish pressure errors near domain boundaries, the projection methods were reformulated in [29] using boundary conditions that allow the pressure to be recovered from the knowledge of the velocity at any fixed time. The application of this approach along with WENO schemes will be one of the subjects of future research.

References

[1] F. Aràndiga, A. Baeza, A.M. Belda, P. Mulet, Analysis of WENO schemes for full and global accuracy, SIAM J. Numer. Anal. 49 (2) (2011) 893–915.

[2] John B. Bell, Phillip Colella, Harland M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.

[3] Lorenzo Botti, Daniele A. Di Pietro, A pressure-correction scheme for convection-dominated incompressible flows with discontinuous velocity and continuous pressure, J. Comput. Phys. 230 (3) (2011) 572–585.

[4] Charles-Henri Bruneau, Mazen Saad, The 2D lid-driven cavity problem revisited, Comput. Fluids 35 (3) (2006) 326–348.

[5] Alfonso Caiazzo, Traian Iliescu, Volker John, Swetlana Schyschlowa, A numerical investigation of velocity–pressure reduced order models for incompressible flows, J. Comput. Phys. 259 (2014) 598–616.

[6] W. Cazemier, R.W.C.P. Verstappen, A.E.P. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids 10 (7) (1998) 1685–1699.

[7] Alexandre Joel Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[8] Ramon Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comput. Phys. 170 (1) (2001) 112–140.

[9] Timothy A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, ACM Trans. Math. Softw. 30 (2) (2004) 196–199.

[10] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (3) (1982) 387–411.

[11] S. Gottlieb, J.S. Muellen, S.J. Ruuth, A fifth order flux implicit WENO method, J. Sci. Comput. 27 (1–3) (2006) 271–287.

[12] J.L. Guermond, P. Minev, Jie Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (44–47) (2006) 6011–6045.

[13] J.-L. Guermond, L. Quartapelle, On stability and convergence of projection methods based on pressure Poisson equation, Int. J. Numer. Methods Fluids 26 (9) (1998) 1039–1053.

[14] Arthur Guittet, Maxime Theillard, Frédéric Gibou, A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive quad/octrees, J. Comput. Phys. 292 (2015) 215–238.

[15] Guang-Shan Jiang, Chi-Wang Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.

[16] V. John, A. Liakos, Time-dependent flow across a step: the slip with friction boundary condition, Int. J. Numer. Methods Fluids 50 (6) (2006) 713–731.

[17] V. John, G. Matthies, Higher-order finite element discretizations in a benchmark problem for incompressible flows, Int. J. Numer. Methods Fluids 37 (8) (2001) 885–903.

[18] V. John, G. Matthies, MooNMD—a program package based on mapped finite element methods, Comput. Vis. Sci. 6 (2–3) (2004) 163–169.

[19] V. John, G. Matthies, J. Rang, A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 195 (44–47) (2006) 5995–6010.

[20] V. John, J. Novo, On (essentially) non-oscillatory discretizations of evolutionary convection–diffusion equations, J. Comput. Phys. 231 (4) (2012) 1570–1586.

[21] V. John, J. Novo, Analysis of the pressure stabilized Petrov–Galerkin method for the evolutionary Stokes equations avoiding time step restrictions, SIAM J. Numer. Anal. 53 (2) (2015) 1005–1031.

[22] V. John, J. Rang, Adaptive time step control for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 199 (9–12) (2010) 514–524.

[23] G.E. Karniadakis, S. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, Oxford University Press, USA, 2004.

[24] A. Prohl, Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier–Stokes Equations, Advances in Numerical Mathematics, B.G. Teubner, Stuttgart, 1997.

[25] R. Rannacher, On Chorin's projection method for the incompressible Navier–Stokes equations, in: The Navier–Stokes Equations II—Theory and Numerical Methods, Oberwolfach, 1991, in: Lecture Notes in Math., vol. 1530, Springer, Berlin, 1992, pp. 167–183.

[26] Hans-Görg Roos, Martin Stynes, Lutz Tobiska, Robust Numerical Methods for Singularly Perturbed Differential Equations, second edition, Springer Series in Computational Mathematics, vol. 24, Springer-Verlag, Berlin, 2008, Convection–diffusion–reaction and flow problems.

[27] M. Schäfer, S. Turek, Benchmark computations of laminar flow around a cylinder, With support by F. Durst, E. Krause, R. Rannacher, in: Flow Simulation with High-Performance Computers II. DFG Priority Research Programme Results 1993–1995, Vieweg, Wiesbaden, 1996, pp. 547–566.

[28] K. Selim, A. Logg, M.G. Larson, An adaptive finite element splitting method for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 209 (212) (2012) 54–65.

[29] D. Shirokoff, R.R. Rosales, An efficient method for the incompressible Navier–Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary, J. Comput. Phys. 230 (23) (2011) 8619–8646.

[30] Chi-Wang Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev. 51 (1) (2009) 82–126.

[31] R. Témam, Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires. II, Arch. Ration. Mech. Anal. 33 (1969) 377–385.

[32] G. Tiesinga, F.W. Wubs, A.E.P. Veldman, Bifurcation analysis of incompressible flow in a driven cavity by the Newton–Picard method, J. Comput. Appl. Math. 140 (1–2) (2002) 751–772.

[33] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 870–891.