Journal of Computational Physics 231 (2012) 3041–3063

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations

B. Sanderse [a,b,*], B. Koren [b,c]

[a] Energy research Centre of the Netherlands (ECN), Petten, The Netherlands
[b] Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
[c] Mathematical Institute, Leiden University, Leiden, The Netherlands

ARTICLE INFO

Article history:
Received 22 July 2011
Received in revised form 9 November 2011
Accepted 18 November 2011
Available online 29 November 2011

Keywords:
Differential–algebraic equations
Incompressible Navier–Stokes equations
Temporal accuracy
Time integration
Runge–Kutta method
Moving meshes

ABSTRACT

This paper investigates the temporal accuracy of the velocity and pressure when explicit Runge–Kutta methods are applied to the incompressible Navier–Stokes equations. It is shown that, at least up to and including fourth order, the velocity attains the classical order of accuracy without further constraints. However, in case of a time-dependent gradient operator, which can appear in case of time-varying meshes, additional order conditions need to be satisfied to ensure the correct order of accuracy. Furthermore, the pressure is only first-order accurate unless additional order conditions are satisfied. Two new methods that lead to a second-order accurate pressure are proposed, which are applicable to a certain class of three- and four-stage methods. A special case appears when the boundary conditions for the continuity equation are independent of time, since in that case the pressure can be computed to the same accuracy as the velocity field, without additional cost. Relevant computations of decaying vortices and of an actuator disk in a time-dependent inflow support the analysis and the proposed methods.

1. Introduction

In this paper we discuss the application of explicit Runge–Kutta methods to the time discretization of the incompressible Navier–Stokes equations. Explicit Runge–Kutta methods are a popular choice for the time discretization of the Navier–Stokes equations, because they are cheap compared to implicit methods if flow problems are not stiff, which is the case for convection-dominated flows not involving solid boundaries. Compared with (explicit) multi-step methods, Runge–Kutta methods have in general better stability properties, do not have a start-up problem, and easily allow for adaptive time stepping, although they generally require the solution to a Poisson equation for the pressure at each stage of the Runge–Kutta method. Examples of Runge–Kutta methods applied to the incompressible Navier–Stokes equations are Wray's third-order method (sometimes combined with an implicit method for the diffusion terms) [1–4], a third-order accurate semi-implicit method [5], and the classic fourth-order method [6,7]. These three- and four-stage methods have the favorable property for convection-dominated flows that the linear stability domain contains part of the imaginary axis.

The application of explicit Runge–Kutta methods to the incompressible Navier–Stokes equations is not straightforward because of the differential–algebraic nature of the equations. It is common practice to explicitly advance the velocity at each stage as if the discretized equations are a system of ordinary differential equations, and subsequently solve a Poisson equation for the pressure to make the velocity field divergence-free. However, it is not clear if and how this approach influences

* Corresponding author at: Energy research Centre of the Netherlands (ECN), Petten, The Netherlands. Tel.: +31 224564668.
E-mail addresses: sanderse@ecn.nl (B. Sanderse), barry.koren@cwi.nl (B. Koren).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.11.028

the temporal order of accuracy of the velocity and pressure. The accuracy of the velocity is often silently assumed to be unaffected by the differential–algebraic nature of the incompressible Navier–Stokes equations, and the temporal accuracy of the pressure is often not reported. A temporally accurate pressure is however of interest in many flow simulations, such as those involving unsteady lift and drag computations or fluid–structure interactions. We therefore thoroughly analyze the accuracy of both velocity and pressure by applying the convergence theory developed for index 2 differential algebraic equations [8,9] to the incompressible Navier–Stokes equations. We discuss the treatment of unsteady boundary conditions and time-varying meshes, which has not been clearly reported in literature so far, and investigate if they influence the order of accuracy.

The outline of this paper is as follows. First, in Section 2 we develop a general formulation for explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations, which includes the case of unsteady boundary conditions for the continuity and momentum equations. Subsequently, in Section 3 we investigate the order conditions for velocity and pressure, and in Section 4 we propose different methods to compute a high-order accurate pressure. In Section 5 we show the results of two test cases which confirm our theoretical findings.

2. Framework

2.1. Introduction

The governing equations for incompressible flow are the conservation of mass and momentum,

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u}, \tag{2}$$

collectively called the incompressible Navier–Stokes equations. For the spatial discretization of (1) and (2) we employ a second-order finite volume method on staggered Cartesian grids, similar to the method of Harlow and Welch [10]. We stress however that the analysis presented in this paper is not restricted to finite volume methods but is equally valid for finite difference and finite element methods, as long as the spatial discretization leads to a semi-discrete system (method of lines) that can be expressed as the following problem:

$$Mu(t) = r_1(t), \tag{3}$$

$$\Omega \dot{u}(t) = -C(u(t)) + \nu D u(t) - G p(t) + r_2(u(t), t), \tag{4}$$

supplemented with suitable initial conditions. $u(t) \in \mathbb{R}^{N_u}$ and $p(t) \in \mathbb{R}^{N_p}$ are vectors with unknowns. In the remainder, their explicit t-dependence will mostly be left out of the notation. M, C, D and G represent the discrete divergence, convection, diffusion and gradient operators, respectively. Note that $r_2 \in \mathbb{R}^{N_u}$; $r_1 \in \mathbb{R}^{N_p}$; $M \in \mathbb{R}^{N_u \times N_p}$; $\Omega, C, D \in \mathbb{R}^{N_u \times N_u}$, $G \in \mathbb{R}^{N_p \times N_u}$. $r_1(t)$ is a vector with boundary conditions for the continuity equation and $r_2(u,t)$ is a vector with boundary conditions and forcing terms for the momentum equation. Ω is a matrix with on its diagonal the finite volume sizes, which are assumed to be independent of t. In finite element methods this is the mass matrix. Eqs. (3) and (4) form a non-autonomous differential algebraic equation (DAE) system of index 2 (see e.g. [8,9,11]), where u plays the role of the differential variable, and p the role of the algebraic variable. Following the literature on DAEs, they can be written as

$$0 = g(u,t), \tag{5}$$

$$\dot{u} = f(u,p,t), \tag{6}$$

where

$$g(u,t) = Mu - r_1(t), \tag{7}$$

$$f(u,p,t) = F(u,t) - Gp, \tag{8}$$

with $F(u,t) = -C(u) + \nu D\, u + r_2(u,t)$. Ω^{-1} has been absorbed in the definition of C, D, G and r_2. The explicit presence of unsteady boundary conditions for the divergence equation, $r_1(t)$, is often omitted in literature, but it will be shown to be important in our discussion. An example of a nonzero $r_1(t)$ is a time-varying inflow condition such as a turbulent inflow field. We assume that

$$L = -g_u(u,t) f_p(u,p,t) \tag{9}$$

is non-singular, so that the problem is indeed of index 2. For the incompressible Navier–Stokes equations

$$L = MG \tag{10}$$

is recognized as the Laplacian operator (independent of u and p), which is actually singular in case of Dirichlet or periodic conditions for the velocity on the entire boundary. A possible remedy against the singular nature is to impose an additional constraint (e.g. setting the average pressure value by replacing one row of the Laplace matrix by ones). An instantaneous equation for the algebraic variable, the pressure, is found by applying the divergence-free constraint to the momentum equation:

$$Lp = MF(u,t) - \dot{r}_1(t), \tag{11}$$

where we have used that M is not depending on t. We note that this equation can be derived similarly if M, G and other operators depend on time, as in the case of time-varying meshes; an additional term $\dot{M}(t)u$ will appear on the right-hand side. The pressure can be eliminated from the system of equations, by solving Eq. (11) and inserting it into (6):

$$\dot{u} = PF(u,t) + GL^{-1}\dot{r}_1(t), \tag{12}$$

where the projection operator P, defined by

$$P = I - GL^{-1}M, \tag{13}$$

is a square matrix that projects velocity fields on the space of divergence-free fields; the divergence of this projection is zero ($MP = 0$). However, when discretizing in time one should *not* start with Eq. (12). Differentiating the constraint, necessary to arrive at (12), effectively lowers the index of the DAE system, and upon discretization its solutions do not necessarily satisfy the constraint (5). It is therefore desirable to use the original DAE system (5) and (6) (the one with highest index), because its solutions will satisfy all the derived lower index systems [12].

The initial conditions at $t = t_0$ should be consistent with Eqs. (5) and (6) and (11):

$$Mu_0 = r_1(t_0), \tag{14}$$
$$Lp_0 = MF(u_0,t_0) - \dot{r}_1(t_0). \tag{15}$$

Eq. (15) expresses that the initial pressure cannot be chosen freely, but has to be calculated based on u_0.

2.2. Explicit Runge–Kutta methods

Application of an explicit Runge–Kutta method to DAE systems is not always straightforward. We employ existing theory on *half-explicit Runge–Kutta methods*, see Hairer et al. [8], to Eqs. (5) and (6), which leads to the following formulation:

$$U_i = u_n + \Delta t \sum_{j=1}^{i-1} a_{ij}(F_j - Gp_j), \quad i = 1,\ldots,s, \tag{16}$$
$$MU_i = r_1(t_i). \tag{17}$$

Here U_i and u_n are approximations to the exact values $u(t_i)$ and $u(t_n)$, respectively, with $t_i = t_n + c_i\Delta t$ and $F_j = F(U_j,t_j)$. $A = (a_{ij})$, b_i and c_i are the coefficients of the Runge–Kutta method, which can be written in compact form using a Butcher tableau:

$$\begin{array}{c|cccc} c_1 & 0 & & & \\ c_2 & a_{21} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & \ldots & a_{s,s-1} & 0 \\ \hline & b_1 & \ldots & \ldots & b_s \end{array} \tag{18}$$

with the convention

$$c_i = \sum_{j=1}^{i-1} a_{ij}. \tag{19}$$

Now that the Runge–Kutta method has been applied, one can obtain a more compact formulation by eliminating the pressure:

$$U_i = u_n + \Delta t \sum_{j=1}^{i-1} a_{ij}PF_j + GL^{-1}(r_1(t_i) - r_1(t_n)), \quad i = 1,\ldots,s. \tag{20}$$

Since $MP = 0$, Eq. (20) satisfies $MU_i = r_1(t_i)$ at all intermediate stages. Note that when Eq. (12) would be integrated in time with a Runge–Kutta method, the last term in (20) would change to $GL^{-1}\sum_j a_{ij}\dot{r}_1(t_j)$, and the constraint is only satisfied if $\dot{r}_1(t) = 0$ (both formulations are equal in that case). Satisfying the incompressibility constraint at the intermediate stages is important to obtain the correct convergence order of the Runge–Kutta method; this will be discussed in some more detail in Section 3. The velocity at the new time step follows as a combination of the stage values:

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_iPF_i + GL^{-1}(r_1(t_{n+1}) - r_1(t_n)). \tag{21}$$

A simpler representation for explicit methods is obtained by introducing the shifted matrix $\widetilde{A}$

$$\widetilde{A} = \begin{pmatrix} a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{s1} & \dots & a_{s,s-1} & 0 \\ b_1 & \dots & b_{s-1} & b_s \end{pmatrix}, \tag{22}$$

and the shifted vectors

$$\tilde{c} = \begin{pmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_{s-1} \\ \tilde{c}_s \end{pmatrix} = \begin{pmatrix} c_2 \\ \vdots \\ c_s \\ 1 \end{pmatrix}, \quad \widetilde{U} = \begin{pmatrix} \widetilde{U}_1 \\ \vdots \\ \widetilde{U}_{s-1} \\ \widetilde{U}_s \end{pmatrix} = \begin{pmatrix} U_2 \\ \vdots \\ U_s \\ u_{n+1} \end{pmatrix}, \quad \tilde{p} = \begin{pmatrix} \tilde{p}_1 \\ \vdots \\ \tilde{p}_{s-1} \\ \tilde{p}_s \end{pmatrix} = \begin{pmatrix} p_2 \\ \vdots \\ p_s \\ p_{n+1} \end{pmatrix}, \tag{23}$$

so that Eqs. (20) and (21) can be written as

$$\widetilde{U}_i = u_n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} P F_j + G L^{-1} (r_1(\tilde{t}_i) - r_1(t_n)), \tag{24}$$

with $\tilde{t}_i = t_n + \Delta t\ \tilde{c}_i$.

The presence of the inverse of the Laplace operator (in the two last terms) makes the computation of $\widetilde{U}_i$ unattractive from a practical point of view. Therefore we rewrite Eq. (24) back into a two-step formulation by introducing a variable which 'looks' like the pressure p. First we substitute $P = I - GL^{-1}M$ in (24), leading to

$$\widetilde{U}_i = u_n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} F_j - G L^{-1} \left(\Delta t \sum_{j=1}^{i} \tilde{a}_{ij} M F_j - (r_1(\tilde{t}_i) - r_1(t_n)) \right). \tag{25}$$

Comparing with the 'exact' equation for the pressure at each stage,

$$L\tilde{p}_i = M\widetilde{F}_i - \dot{r}_1(\tilde{t}_i), \tag{26}$$

it seems natural to introduce a pressure-like variable, ϕ, and the c coefficients and rewrite (25) as the following two steps:

$$\widetilde{U}_i = u_n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} F_j - \tilde{c}_i \Delta t G \tilde{\phi}_i, \tag{27}$$

with $\tilde{\phi}_i$ defined by

$$L\tilde{\phi}_i = \sum_{j=1}^{i} \frac{1}{\tilde{c}_i} \tilde{a}_{ij} M F_j - \frac{r_1(\tilde{t}_i) - r_1(t_n)}{\tilde{c}_i \Delta t}. \tag{28}$$

Eq. (28) is simply the divergence of (27) supplemented with the additional information $M\widetilde{U}_i = r_1(\tilde{t}_i)$. Each $\tilde{\phi}_i$ is a Lagrange multiplier to make $\widetilde{U}_i$ divergence free and each $\widetilde{U}_i$ is independent of the value of $\tilde{\phi}_j$ for $j \neq i$; this is the advantage of using ϕ instead of p. It should be stressed that the presence of the c coefficients in the pressure term is *not* necessary to obtain the correct velocity field. The reason to introduce the c coefficients is that for explicit methods it yields a $\tilde{\phi}_i$ which is a consistent approximation to $\tilde{p}_i$. In case of implicit methods (e.g. methods based on Radau or Lobatto quadrature) it is possible to have a non-trivial first stage ($a_{1j} \neq 0$) with $c_1 = 0$ and then the c-coefficients should *not* be introduced. The most general approach is then to define a pressure-like variable $\rho_i (= c_i \phi_i)$ and not introduce the c coefficients at all. As can be seen in Section 4, this does not complicate obtaining high-order accurate pressure estimates, because the combination $c_i \phi_i = \rho_i$ also occurs in that context. These ρ_i values are still Lagrange multipliers whose value is independent of previous time steps, and consequently the velocity field remains independent of the history of the pressure. On the other hand, this is no longer true when the solution of velocity and pressure is decoupled, for instance by employing a pressure correction method, see e.g. [13,14,4]. In that case the c coefficients appear in the pressure term of the initial velocity estimate and both velocity and pressure depend on pressure values from previous time steps. This is however not important for the explicit methods under consideration here.

When comparing Eqs. (28) with (26), the first term on the right side of (28) is recognized as an approximation to $M\widetilde{F}_i$ and the second term as an approximation to $\dot{r}_1(\tilde{t}_i)$. The second term is clearly a first-order approximation, since

$$\dot{r}_1(\tilde{t}_i) = \frac{r_1(\tilde{t}_i) - r_1(t_n)}{\tilde{c}_i \Delta t} + \mathcal{O}(\Delta t). \tag{29}$$

The first term is also a first-order approximation, which we show by following an argument employed in [15,16]. $\widetilde{F}_i$ in (26) is F evaluated *at* $(\widetilde{U}_i, \tilde{t}_i)$, whereas $\sum_{j=1}^{i} \frac{1}{\tilde{c}_i} \tilde{a}_{ij} F_j$ is an approximation to the *average value* of F from t_n to $\tilde{t}_i$. Assuming that F is con-

tinuous over the interval $[t_n, \tilde{t}_i]$, this average equals the value of F at some point $\hat{t} \in [t_n, \tilde{t}_i]$ (according to the integral version of the mean value theorem) and as such is an $\mathcal{O}(\Delta t)$ approximation to $\widetilde{F}_i$:

$$M\widetilde{F}_i = \sum_{j=1}^{i} \frac{1}{\tilde{c}_i} \tilde{a}_{ij} M F_j + \mathcal{O}(\Delta t). \tag{30}$$

As a consequence the Lagrange multiplier ϕ is a first-order approximation to the pressure p:

$$\tilde{\phi}_i = \tilde{p}_i + \mathcal{O}(\Delta t). \tag{31}$$

Of course, by virtue of the midpoint method, ϕ_i is a second-order approximation to the pressure at $t_n + \frac{1}{2} c_i \Delta t$, as long as the stage order of the method is at least 2 (this will be detailed in Section 4). We will call the approach, where one uses $\tilde{\phi}_s$ as approximation to p_{n+1}, the 'standard' approach. The first-order accuracy of this approach is independent of the particular coefficients of the Runge–Kutta method. It results from the fact that (20) contains an approximation to the integral $\int F \, dt$, whereas (28) contains F evaluated at a certain time instance. This is because the pressure has an *instantaneous* character: its value is such that the velocity field is divergence free at each time instant, and is independent of the pressure at any previous time. The equation for the velocity is, on the contrary, an evolution equation.

An alternative formulation of Eqs. (27) and (28) that can avoid the first-order behavior of the pressure is as follows. Instead of taking a single pressure-like variable at each stage, we take a combination by introducing a Butcher tableau $\widetilde{A}^p$ for the pressure term such that:

$$\widetilde{U}_i = u_n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} F_j - \Delta t \sum_{j=1}^{i} \tilde{a}^p_{ij} G \tilde{\psi}_j, \tag{32}$$

with

$$\sum_{j=1}^{i} \tilde{a}^p_{ij} L \tilde{\psi}_j = \sum_{j=1}^{i} \tilde{a}_{ij} M F_j - \frac{r_1(\tilde{t}_i) - r_1(t_n)}{\Delta t}. \tag{33}$$

$\widetilde{A}^p$ is, like $\widetilde{A}$, lower triangular and should have $\tilde{a}^p_{i,i} \neq 0$ in order to guarantee a unique $\tilde{\psi}$ at each stage. Eqs. (28) and (33) are related by

$$\tilde{\psi} = (\widetilde{A}^p)^{-1} \mathrm{diag}(\tilde{c}_1, \ldots, \tilde{c}_s) \tilde{\phi}. \tag{34}$$

One can therefore obtain $\tilde{\psi}$ from $\tilde{\phi}$, as long as $\widetilde{A}^p$ is invertible. For the choice

$$\widetilde{A}^p = \mathrm{diag}(\tilde{c}_1, \ldots, \tilde{c}_s), \tag{35}$$

we obtain formulation (27) and (28) with $\tilde{\psi} = \tilde{\phi}$, and the accuracy of the pressure is limited to first order. A possible way to obtain a higher order pressure is to choose $\widetilde{A}^p = \widetilde{A}$ (so that there is just a single Butcher tableau). This is simply the original formulation (16) and (17), whose accuracy can be evaluated with the theory of Hairer et al. [8,9].

To conclude, we write down the solution algorithm that we use in practice, obtained by rewriting (27) and (28):

$$\widetilde{V}_i = u_n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} F_j, \tag{36}$$

$$L\tilde{\phi}_i = \frac{1}{\tilde{c}_i \Delta t} (M\widetilde{V}_i - r_1(\tilde{t}_i)), \tag{37}$$

$$\widetilde{U}_i = \widetilde{V}_i - \tilde{c}_i \Delta t G \tilde{\phi}_i, \quad i = 1, 2 \ldots s, \tag{38}$$

optionally followed by Eq. (34). This sequence of first computing a tentative velocity, then the pressure, and finally correcting the tentative velocity is similar to fractional step methods (see e.g. [15]). However, in fractional step methods the diffusive and/or convective terms are often taken implicitly, and a splitting error then results from uncoupling the solution of velocity and pressure. Here all terms are handled explicitly (except the pressure) and consequently there is no splitting error involved. It is therefore unnecessary to solve a coupled system for $\widetilde{U}_i$ and $\tilde{\phi}_i$, as is done for example in Pereira et al. [6]. The half-explicit nature of the method is now clear: the differential variable is advanced with an explicit method (Eqs. (36) and (38)) while the algebraic variable is handled implicitly (Eq. (37)). The implicit equation for the pressure has to be solved at each stage, so in total this results in s Poisson equations. The resulting $\widetilde{U}_s = u_{n+1}$ and $\tilde{\phi}_s = \phi_{n+1}$ (or $\tilde{\psi}_s$) are approximations to $u(t_{n+1})$ and $p(t_{n+1})$. The order of accuracy of $\tilde{\psi}_s$ and $\widetilde{U}_s$ will be considered next.

3. Order conditions

3.1. Local and global error

For general index 2 DAEs of the form (5) and (6) the classical order conditions ('classical' referring to non-stiff ODEs, see e.g. [17]) for the coefficients of the Butcher tableau are not sufficient to guarantee the correct order of accuracy for both the differential and algebraic variable. The work of Hairer et al. [8] and Hairer and Wanner [9] provides local and global error analyses for index 2 DAEs and identifies in which cases *order reduction* can occur. Here we focus on the local error, because for half-explicit methods the error propagation from local to global error is the same as for non-stiff ODEs. For the velocity (the differential variable) this is expressed by the following theorem:

Theorem 1. *Convergence – Brasey and Hairer [18]. Suppose that (9) holds in a neighborhood of the solution (u(t),p(t)) of Eqs. (5) and (6) and that the initial values satisfy (14) and (15). If the coefficients of the half-explicit Runge–Kutta method (32) and (33) satisfy* $\tilde{a}^p_{i,i} \neq 0$ *and* $b_s \neq 0$ *and if the local error satisfies*

$$\delta u(t) = \mathcal{O}(\Delta t^{r+1}), \tag{39}$$

then the method is convergent of order r, i.e.,

$$u_n - u(t_n) = \mathcal{O}(\Delta t^r) \quad \text{for } t_n - t_0 = n\Delta t \leqslant T, \tag{40}$$

with T finite.

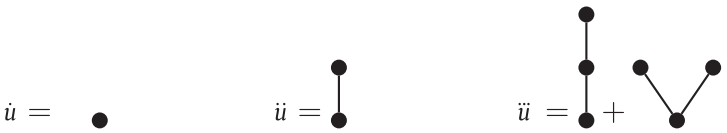
The pressure-like variable $\tilde{\psi}_i$ (including $\tilde{\psi}_s = \psi_{n+1}$) is independent of ψ_n and the order of accuracy of the global error in ψ is therefore given by the order of accuracy of the local error $\delta\psi(t)$, provided that $\delta u(t)$ has at least the same order [8]. The focus of the rest of this section is therefore on the local error of both the u- and p-component.

3.2. A short introduction to trees

For Runge–Kutta methods applied to ODEs of the form $\dot{u} = f(u)$, the local error can be investigated by expanding both the exact and numerical solution in a Taylor series and comparing until which order they agree. This requires that $\ddot{u}, \dddot{u}$, etc. are written in terms of f and its derivatives:

$$\dot{u} = f, \quad \ddot{u} = f_u f, \quad \dddot{u} = f_u f_u f + f_{uu}(f,f). \tag{41}$$

Since f and u are vectors, the first derivatives in this expression should be interpreted as Jacobian matrices, the second derivatives as bilinear maps, and (f,f) as a tensor product. The number of elementary differentials that appear in this process grows rapidly when high orders are compared. With each differential there is an associated order condition. An efficient way to handle the order conditions for ODEs was introduced by Butcher with the concept of rooted trees [17,19]. Given a certain tree the elementary differential and the order condition corresponding to it can be easily written down. For example, (41) becomes in terms of trees



(42)

The order conditions for these trees are (in order of appearance):

$$\sum b_i = 1, \quad \sum b_i c_i = \frac{1}{2}, \quad \sum b_i a_{ij} c_j = \frac{1}{6}, \quad \sum b_i c_i^2 = \frac{1}{3}. \tag{43}$$

In all cases, the summation is over all indices present in the summand.

The extension of the analysis with trees to DAEs was done by Hairer et al. [8] and will be used here. Hairer et al. [8] consider the autonomous index 2 DAE

$$0 = g(u), \tag{44}$$

$$\dot{u} = f(u,p). \tag{45}$$

The non-autonomous system (5) and (6) can be written in this form by adding $\dot{t} = 1$ so that Eqs. (44) and (45) hold by redefining $u := \begin{pmatrix} u \\ t \end{pmatrix}$ and $f := \begin{pmatrix} f \\ 1 \end{pmatrix}$. In the Taylor expansion of the exact and numerical solution, $\dot{p}, \ddot{p}, \ldots$ appear next to $\dot{u}, \ddot{u}, \ldots$ Here we list the first few derivatives (see [8,9]):

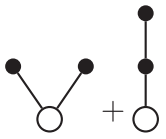
$$\dot{u} = f, \tag{46}$$

$$\dot{p} = (-g_u f_p)^{-1}(g_{uu}(f,f) + g_u f_u f), \tag{47}$$

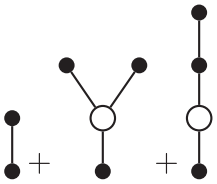
$$\ddot{u} = f_u f + f_p(-g_u f_p)^{-1}(g_{uu}(f,f) + g_u f_u f). \tag{48}$$

For DAEs, the number of differentials grows even more rapidly for higher order derivatives. Trees still provide a compact way to represent these derivatives, when extended to contain both meagre (solid) and fat (open) vertices:

$\dot{u} =$ (49)



$\dot{p} =$ (50)



$\ddot{u} =$ (51)

The *order* of a tree is the number of meagre vertices minus the number of fat vertices [8]. To have a local error of $\mathcal{O}(\Delta t^{r+1})$ (global error $\mathcal{O}(\Delta t^r)$) the order conditions should be satisfied up to and including tree order r. (An exception is the case where a differential corresponding to a tree is of the form $f_p \cdot (\ldots)$. The corresponding order condition then has to be considered for order $r+1$ instead of r [8].) These order conditions can be read again from the trees, as is outlined in Hairer et al. [8] and Brasey and Hairer [18]. As an example, the above trees correspond to the following order conditions:

$$\dot{u}: \quad \sum b_i = 1, \tag{52}$$

$$\dot{p}: \quad \sum b_i \omega_{ij} \omega_{jk} c_k^2 = 2, \quad \sum b_i \omega_{ij} \omega_{jk} a_{kl} c_l = 1, \tag{53}$$

$$\ddot{u}: \quad \sum b_i c_i = \frac{1}{2}, \quad \sum b_i \omega_{ij} c_j^2 = 1, \quad \sum b_i \omega_{ij} a_{jk} c_k = \frac{1}{2}, \tag{54}$$

the summation being again over all indices. In this example we see that next to the classical order conditions (represented by trees with only meagre vertices) additional order conditions appear, corresponding to trees with fat vertices, which include the inverse of matrix A (ω_{ij} denotes the entries of A^{-1}). The order conditions for the p-component are especially difficult due to the presence of $(A^{-1})^2$. Fortunately, some of these additional trees do not pose additional constraints on the coefficients because they reduce to classical order conditions. For example, the last condition in Eq. (54) can be written as

$$\sum b_i \omega_{ij} a_{jk} c_k = b^T A^{-1} A c = \sum b_i c_i = \frac{1}{2}, \tag{55}$$

so it reduces to the classical second-order condition. The additional order conditions that cannot be simplified to classical order conditions are of interest to us. To find these remaining conditions we used the software described in [20]. We found that for the u-component there is no additional tree for order 1, but there is 1 for order 2, there are 4 for order 3 and 17 for order 4. For the p-component there are 2 trees for order 1, 6 for order 2, 21 for order 3, and 81 for order 4.

This large number of additional order conditions can still be considerably reduced when taking into account two important facts, namely that (i) we are considering half-explicit methods and (ii) we are applying these to the Navier–Stokes equations.

3.3. Application to the incompressible Navier–Stokes equations

For the Navier–Stokes equations we know that $f(u,p,t) = F(u,t) - Gp$, which means that $f_p = G$ is a constant matrix and thus all derivatives of f_p, such as f_{pu}, f_{pp}, etc., are zero. Therefore trees which have a meagre vertex as root or as branch and connected to it a fat vertex and at least one other meagre or fat vertex need not be considered. In case G is a function of time, one cannot remove trees with derivatives of the form f_{pu}, f_{puu}, but only those of the form f_{pp}, f_{ppp}, etc.; we will treat this in Section 3.5.

We note that in the case of non-autonomous systems, g_{uu} in Eqs. (47) and (48) consists of g_{uu}, g_{ut} and g_{tt}. g_{uu} and g_{ut} are zero for the Navier–Stokes equations, but $g_{tt} = \ddot{r}_1(t)$ is in general not, and therefore trees that have a fat vertex with more than one meagre vertex connected to it do not vanish. This also covers the case of a time-dependent M matrix. The special case of $\dot{r}_1(t) = 0$, so that $g_{tt} = 0$, will be discussed in Section 4.3.

3.4. Half-explicit methods

For half-explicit methods the construction of the order conditions changes slightly: if a meagre vertex follows a fat vertex then the index changes from a_{jk} to $\tilde{a}_{jk}$ (or from c_j to $\tilde{c}_j$). The trees and order conditions that result with this notation, and after removing all trees containing a derivative of f_p, are shown for the u-component and p-component in Tables 1 and 2,

Table 1
Additional trees and order conditions for u-component up to and including order 4 when f_p = constant.

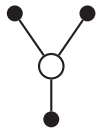
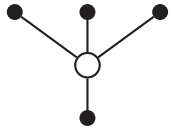
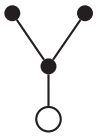
	Tree	Order	Condition	Simplifies to	Differential
1.		$2 \to 3$	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j^2 = 1$	$\tilde{c}_s^2 = 1$	$f_p(-g_u f_p)^{-1} g_{uu}(f,f)$
2.		$3 \to 4$	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j^3 = 1$	$\tilde{c}_s^3 = 1$	$f_p(-g_u f_p)^{-1} g_{uuu}(f,f,f)$
3.		$3 \to 4$	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{a}_{jk} c_k = \frac{1}{2}$	$\sum b_i c_i = \frac{1}{2}$	$f_p(-g_w f_p)^{-1} g_{uu}(f,f_w f)$

Table 2
Trees and order conditions for p-component up to and including order 2 when f_p = constant.

	Tree	Order	Condition	Simplifies to	Differential
4.		1	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k^2 = 2$	$\sum \tilde{\omega}_{si} \tilde{c}_i^2 = 2$	$(-g_w f_p)^{-1} g_{uu}(f,f)$
5.		1	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} c_l = 1$	$c_s = 1$	$(-g_w f_p)^{-1} g_u f_w f$
6.		2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k \tilde{a}_{kl} c_l = \frac{3}{2}$	$\sum \tilde{\omega}_{si} \tilde{c}_i \tilde{a}_{ij} c_j = \frac{3}{2}$	$(-g_w f_p)^{-1} g_{uu}(f,f_w f)$
7.		2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k^3 = 3$	$\sum \tilde{\omega}_{si} \tilde{c}_i^3 = 3$	$(-g_w f_p)^{-1} g_{uuu}(f,f,f)$
8.		2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} a_{lm} c_m = \frac{1}{2}$	$\sum a_{si} c_i = \frac{1}{2}$	$(-g_w f_p)^{-1} g_w f_w f_w f$
9.		2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} c_l^2 = 1$	$c_s^2 = 1$	$(-g_w f_p)^{-1}\ g_w f_{uu}(f,f)$

respectively. Contrary to the u-component, the order of accuracy of the global error of the p-component is equal to the order of the tree plus 1, because there is no power lost when going from local error to global error.

Table 1 shows that for the u-component up to and including order 4 only 3 trees with associated additional order conditions remain. The 'shift' in order indicated by an arrow $r \to r+1$ is due to the form $f_p \cdot (\ldots)$ of the remaining trees, as was mentioned before. The order conditions corresponding to these trees can be simplified for the explicit methods under consideration. Considering that b is the last row of $\tilde{A}$, we can write

$$b^T = (0 \quad \dots \quad 0 \quad 1)\widetilde{A}. \tag{56}$$

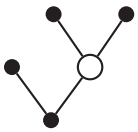
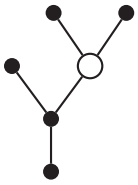
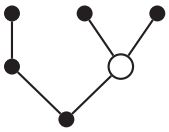
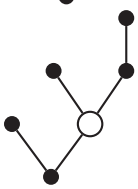
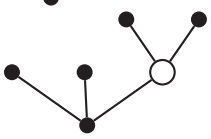
As an example, the order condition for tree number 3 can simplified to

$$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{a}_{jk} c_k = (0 \quad \dots \quad 0 \quad 1)\widetilde{A}\widetilde{A}^{-1}(\tilde{c} \circ (\widetilde{A}c)) = \tilde{c}_s \sum b_i c_i = \sum b_i c_i = \frac{1}{2}, \tag{57}$$

where $\circ$ denotes the elementwise product, i.e., $c = a \circ b$ means $c_i = a_i b_i$. If the Runge–Kutta method satisfies the classical second-order condition $\sum b_i c_i = \frac{1}{2}$ this additional order condition is satisfied. A similar simplification of trees 1 and 2 yields the conditions $\tilde{c}_s^2 = 1$ and $\tilde{c}_s^3 = 1$, which are automatically satisfied, see Eq. (23). In conclusion, the specific form of the pressure term in the Navier–Stokes equations *and* the use of explicit methods leads to the observation that all additional order conditions are trivially satisfied, at least up to and including order 4. This is also true for order 5, for which 6 stages are needed (although methods with order higher than 4 are hardly used for the time integration of the incompressible Navier–Stokes equations). We conjecture that this is true for any order, i.e., when applying an explicit Runge–Kutta method to the incompressible Navier–Stokes equations no additional order conditions appear for the u-component, and consequently no order reduction occurs. On the other hand, order reduction will occur if the continuity equation $M\widetilde{U}_i = r_1(\tilde{t}_i)$ is not satisfied at all intermediate stages, although this does not affect the stability domain of the method. The resulting method can therefore be of interest to compute, for example, steady flows with an artificial time-stepping technique, because stability is then much more important than temporal accuracy. The effect on the order of accuracy of the velocity requires the study of the influence of perturbations in the constraint on the velocity at the end of the time step. Such a study is left as a suggestion for further research; here we focus on methods that satisfy the constraint at each stage, so s Poisson equations are solved for an s-stage method.

For the p-component the above simplification of the additional order conditions is not possible and we have to look in more depth in whether the remaining order conditions can be satisfied. This will be detailed in Section 4.

Table 3
Additional trees and order conditions for u-component up to and including order 4 when f_p is a function of t.

	Tree	Order	Condition	Differential
10.		3	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j^2 = \frac{2}{3}$	$f_{pu}(f, (-g_u f_p)^{-1} g_{uu}(f,f))$
11.		4	$\sum b_i a_{ij} c_j \tilde{\omega}_{jk} \tilde{c}_k^2 = \frac{1}{6}$	$f_u f_{pu}(f, (-g_u f_p)^{-1} g_{uu}(f,f))$
12.		4	$\sum b_i a_{ij} c_j \tilde{\omega}_{ik} \tilde{c}_k^2 = \frac{1}{4}$	$f_{pu}(f_u f, (-g_u f_p)^{-1} g_{uu}(f,f))$
13.		4	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{a}_{jk} c_k = \frac{3}{8}$	$f_{pu}(f, (-g_u f_p)^{-1} g_{uu}(f, f_u f))$
14.		4	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j^3 = \frac{3}{4}$	$f_{pu}(f, (-g_u f_p)^{-1} g_{uuu}(f,f,f))$
15.		4	$\sum b_i c_i^2 \tilde{\omega}_{ij} \tilde{c}_j^2 = \frac{1}{2}$	$f_{puu}(f,f,(-g_u f_p)^{-1} g_{uu}(f,f))$

3.5. Time-dependent operators

The interesting case of a time-dependent gradient operator $G(t)$ can be treated in an analogous fashion. The additional trees that result when f_p is not constant but depends on time leads to the trees shown in Table 3. In contrast to Table 1, the order conditions associated with these trees do *not* reduce to classical order conditions. This table is similar to Table 1 in [18], but with the difference that in that work trees containing f_{pp}, f_{ppp} and f_{ppu} are also present. We present this table here as a reference for practitioners of Runge–Kutta methods for time integration of the incompressible Navier–Stokes equations on time-varying meshes.

For a third order method there is one additional tree, denoted by number 10. Evaluating the condition associated with this tree for a three-stage method, together with the four classical order conditions for third order methods, leads to a solution family with $c_3 = 1$, and c_2 as a free parameter $\left(c_2 \neq 0, c_2 \neq \frac{2}{3}, c_2 \neq 1\right)$:

$$a_{21} = c_2 \qquad a_{31} = \frac{3c_2 - 3c_2^2 - 1}{c_2(2 - 3c_2)} \qquad a_{32} = \frac{1 - c_2}{c_2(2 - 3c_2)}, \tag{58}$$

$$b_1 = \frac{3c_2 - 1}{6c_2} \qquad b_2 = \frac{1}{6c_2(1 - c_2)} \qquad b_3 = \frac{2 - 3c_2}{6(1 - c_2)}. \tag{59}$$

This family excludes Wray's popular third-order method [2]. Wray's method reduces to second order for time-dependent operators, such as moving meshes, and is therefore *not* recommended in this case.

For a fourth order method six additional trees appear due to the time dependency of G. It is proven in [18] that the order condition corresponding to tree number 15 cannot be satisfied with a four-stage, fourth-order method. An example of a five-stage, fourth-order method that satisfies the conditions corresponding to trees 10–15 is the HEM4 method [18]. It does not satisfy the conditions corresponding to trees 6 and 7, so it is second order accurate for the pressure.

Note that the time-dependence of f_p also leads to additional trees for the pressure, next to those already mentioned in Table 2. They are of order 2 or higher.

3.6. A note on space–time errors

At this point it is worthwhile to mention that, apart from the order reduction mechanism discussed above (a result of the differential–algebraic nature of the Navier–Stokes equations), another mechanism for order reduction exists. This mechanism, analyzed for example in [21–24], can result when Runge–Kutta methods are applied to PDEs with time-dependent inflow boundary conditions and the exact boundary values are imposed for the intermediate stages (as we do in the current work, see Eq. (20)). Order reduction then appears when studying the full space–time error, i.e., when simultaneously refining mesh and time-step (for example mesh refinement at a fixed CFL number). A solution to this problem is to not impose any intermediate boundary values but instead obtain these values by integrating the semi-discrete equations at the boundary, using one-sided difference stencils to approximate the spatial derivatives. It is questionable if this is a mathematically valid approach for the incompressible Navier–Stokes equations, but in any case it significantly reduces the allowable time step for stability [24]. Another fix, for linear and non-linear hyperbolic PDEs, is presented in [21,22], which boils down to repeated differentiation of the boundary condition and then cleverly integrating it along with the Runge–Kutta method for the interior points. However, it is questionable if such a fix also works for the incompressible Navier–Stokes equations (a system of DAEs of mixed parabolic-elliptic type), because a change in boundary conditions for the intermediate stages affects the divergence equation. As mentioned before, this requires a study of the effect of perturbations in the divergence-free constraint on the order of accuracy. Furthermore, the fix of [21,22] requires that r_1 can be differentiated with respect to time; this derivative is not always available (as in the case of a turbulent inflow field) or might not even exist (e.g. a boundary condition of the form $e^{1-1/t}$ for $t \geqslant 0$ does not have a proper derivative at $t = 0$). Note that this differentiability of r_1 with respect to time will be encountered again in the next section when discussing the accuracy of the pressure. A cure for DAEs (or even for the specific case of the incompressible Navier–Stokes equations) is, to the authors' knowledge, not yet available. Fortunately, the order reduction from unsteady boundary conditions as described above, generally only manifests itself at very fine grids due to a small coefficient of the leading error term [21]. Moreover, our spatial discretization is second-order accurate so that a possible order reduction in the global error is likely to be overwhelmed by the spatial error. Therefore, in our current work we focus on the temporal error only; we fix the mesh and then refine the time step. We leave the possible interplay of temporal and spatial errors as suggestion for future research.

4. The accuracy of the pressure

4.1. Single Butcher tableau for velocity and pressure (Method 1)

We continue with an order study of the pressure when formulation (27) and (28) with $\widetilde{A}^p = \widetilde{A}$ is employed. As was the case in Table 1, the order conditions in Table 2 can also be simplified in certain cases by employing Eq. (56). However, in contrast to the u-component, additional order conditions remain, even for a second-order method. This is not a surprise when considering that the Lagrange multipliers ϕ and ψ are of a different nature than the pressure p (integral versus point

value). Here we concentrate on obtaining an accurate point value for the pressure. Such a point value is of interest when comparing, for example, a pressure distribution at a certain time instant with an experimentally obtained pressure distribution at the same time instant. However, in other cases, such as computing the displacement of a body due to aerodynamic forces in a CFD code for fluid–structure interaction, the integral value can be a better quantity to use.

We should note that all additional order conditions for the pressure can be circumvented *entirely* by solving an additional Poisson equation, Eq. (11), at t_{n+1}:

$$Lp_{n+1} = MF_{n+1} - \dot{r}_1(t_{n+1}). \tag{60}$$

Given an rth order accurate velocity field u_{n+1}, the resulting pressure p_{n+1} is of the same order of accuracy. However, there are two issues in solving Eq. (60). Firstly it is required that $r_1(t)$ can be differentiated (analytically or numerically), something which is not required in the computation of u and ϕ. In many practical computations, for example involving a prescribed turbulent inflow, $\dot{r}_1(t)$ might not be available. Secondly, solving Eq. (60) amounts to the solution of an additional Poisson equation, which is computationally costly. We will therefore look at the additional conditions of Table 2, which, when satisfied, give a higher order accurate pressure without solving Eq. (60).

4.1.1. Two-stage methods

For two-stage, second-order methods we have the classical conditions $b_1 + b_2 = 1$ and $b_2 c_2 = \frac{1}{2}$. For a second-order accurate pressure the conditions corresponding to trees 4 and 5 have to be satisfied as well. The additional order condition corresponding to tree 4 is

$$\sum_{i=1}^{2} \tilde{\omega}_{2i} \tilde{c}_i^2 = 2, \tag{61}$$

where

$$(\omega_{ij}) = A^{-1} = \begin{pmatrix} \frac{1}{a_{21}} & 0 \\ -\frac{b_1}{a_{21} b_2} & \frac{1}{b_2} \end{pmatrix}. \tag{62}$$

This results in the condition

$$\frac{a_{21} - b_1 c_2}{a_{21} b_2} = 2. \tag{63}$$

The order condition corresponding to tree 5 is

$$c_2 = 1. \tag{64}$$

This latter condition results in $b_1 = b_2 = \frac{1}{2}$ after applying the classical order conditions. However, condition (63) can then not be satisfied, because it reduces to $a_{21} - \frac{1}{2} = a_{21}$. It is therefore not possible to obtain better than first-order accuracy for the pressure with a two-stage explicit method.

4.1.2. Three-stage methods

Butcher [17] lists three cases for which a three-stage, third-order explicit method exists. Only in the 'case I' family there is a solution that allows $c_3 = 1$ (the condition corresponding to tree 5), which is the same as solution family (58) and (59). Evaluating the order condition corresponding to tree 4 for this family leads to

$$\frac{3c_2^2 - 7c_2 + 4}{3c_2 - 2} = 2, \tag{65}$$

which has only one valid solution, being $c_2 = \frac{1}{3}$. The resulting Butcher tableau is

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{3} & \frac{1}{3} & & \\ 1 & -1 & 2 & \\ \hline & 0 & \frac{3}{4} & \frac{1}{4} \end{array}, \tag{66}$$

which satisfies indeed trees 4 and 5. Evaluating Eq. (34) gives

$$p_{n+1} = \tilde{\psi}_3 = -\frac{3}{2}\tilde{\phi}_1 - \frac{3}{2}\tilde{\phi}_2 + 4\tilde{\phi}_3. \tag{67}$$

The conditions corresponding to trees 6, 7 and 8 are not satisfied, and the pressure is at best second-order accurate.

We remark that the occurrence of negative coefficients in (66) and (67) will in general not lead to spurious, non-positive solution behavior, because of the smoothness of incompressible Navier–Stokes solutions. No measures will be taken to make (66), (67), or any following method positive.

4.1.3. Four-stage methods

For explicit four-stage, fourth-order methods the classical order conditions require $c_4 = 1$ (see e.g. [17]), which tree 5 (and 9) automatically satisfy. We found three methods that also satisfy the condition corresponding to tree number 4; they read:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1 & 1 & & & \\ \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & & \\ 1 & -\frac{1}{8} & -\frac{3}{8} & \frac{3}{2} & \\ \hline & \frac{1}{6} & -\frac{1}{18} & \frac{2}{3} & \frac{2}{9} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & & \\ \frac{7}{12} & \frac{91}{192} & \frac{7}{64} & & \\ 1 & \frac{1}{7} & -2 & \frac{20}{7} & \\ \hline & \frac{5}{28} & -\frac{3}{4} & \frac{48}{35} & \frac{1}{5} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ \frac{3}{4} & \frac{3}{4} & & & \\ \frac{5}{9} & \frac{100}{243} & \frac{35}{243} & & \\ 1 & \frac{4}{75} & -\frac{19}{21} & \frac{324}{175} & \\ \hline & \frac{8}{45} & -\frac{16}{63} & \frac{243}{280} & \frac{5}{24} \end{array} . \tag{68}$$

For example, evaluating Eq. (34) for the left tableau gives

$$p_{n+1} = \tilde{\psi}_4 = \frac{1}{2}\tilde{\phi}_1 - 2\tilde{\phi}_2 - 2\tilde{\phi}_3 + \frac{9}{2}\tilde{\phi}_4. \tag{69}$$

As can be readily calculated, none of the above methods satisfies the conditions corresponding to trees 6, 7 and 8. Therefore, with a four-stage, fourth-order explicit method the pressure is, again, at best second-order accurate.

4.2. Reconstructing instantaneous pressure values from time averages (Method 2)

We mentioned in Section 2.2 that $\tilde{\phi}_i$ defined by (28) is only first-order accurate in time but that higher order accurate pressures $\tilde{\psi}_i$ are possible with the generalized formulation (33). With the choice $\widetilde{A}^p = \widetilde{A}$ this led to the order conditions outlined in Section 4.1, and it appeared that only a limited number of three-stage and four-stage methods lead to a second-order accurate pressure. In this section we take a different approach by relating the integral averages $\tilde{\phi}_i$ to the point value p_{n+1}. In fact, this boils down to taking $\widetilde{A}^p \neq \widetilde{A}$, although it is not necessary to explicitly derive $\widetilde{A}^p$.

First we consider the *exact* integration of Eq. (11) from t_n to $\tilde{t}_i$, which reads

$$L\int_{t_n}^{\tilde{t}_i} p(t)\,\mathrm{d}t = M\int_{t_n}^{\tilde{t}_i} F(t)\,\mathrm{d}t - (r_1(\tilde{t}_i) - r_1(t_n)), \tag{70}$$

and we denote the exact average of p over this interval by $\phi(\tilde{t}_i)$ (the exact counterpart of the approximation $\tilde{\phi}_i$):

$$\phi(\tilde{t}_i) = \frac{1}{\tilde{c}_i\Delta t}\int_{t_n}^{\tilde{t}_i} p(t)\,\mathrm{d}t. \tag{71}$$

The challenge is to find a higher order accurate point value p_{n+1} from the time average values $\phi(\tilde{t}_i)$. Such an approximation of point values from integral averages is well-known in the field of Essentially Non-Oscillatory (ENO) conservative finite difference schemes and is called *reconstruction*. We follow [25] to perform this reconstruction, and refer to that work for more details. Denoting the primitive function of $p(t)$ by $P(t)$, we can write

$$\phi(\tilde{t}_i)(\tilde{c}_i\Delta t) = \int_{t_n}^{\tilde{t}_i} p(t)\,\mathrm{d}t = P(\tilde{t}_i) - P(t_n), \quad i = 1, 2, \ldots s. \tag{72}$$

We then construct a polynomial $H(t)$ that interpolates $P(t)$ at the following points:

$$\tilde{t}_{k_1}, \tilde{t}_{k_2}, \ldots, \tilde{t}_{k_m}, \tag{73}$$

where $K = \{k_1, \ldots, k_m\}$ is the set of points that will be used in the interpolation. We always take $k_1 = 0$ and $k_m = s$ (using the convention that $\tilde{t}_0 = t_n$). The other values of k depend on which of the intermediate stages are used in the interpolation. One could take all points, i.e., $K = \{0, 1, \ldots, s\}$, but this is in general not necessary, as we will show later. The derivative of $H(t)$ is denoted by $h(t)$. Then $h(t)$ is an approximation to $p(t)$, and their integrals are the same:

$$\frac{1}{\tilde{c}_i\Delta t}\int_{t_n}^{\tilde{t}_i} h(t)\,\mathrm{d}t = \frac{1}{\tilde{c}_i\Delta t}\int_{t_n}^{\tilde{t}_i} H'(t)\,\mathrm{d}t = \frac{1}{\tilde{c}_i\Delta t}(H(\tilde{t}_i) - H(t_n)) = \frac{1}{\tilde{c}_i\Delta t}(P(\tilde{t}_i) - P(t_n)) = \frac{1}{\tilde{c}_i\Delta t}\int_{t_n}^{\tilde{t}_i} p(t)\,\mathrm{d}t = \phi(\tilde{t}_i). \tag{74}$$

The crucial point in this derivation is that $H(t)$ interpolates $P(t)$ exactly at the points used for the construction of the polynomial. We employ the Lagrange form of the interpolation polynomial, i.e., we write

$$H(t) = \sum_{k\in K} P(\tilde{t}_k)\ell_k(t), \tag{75}$$

where

$$\ell_k(t) = \prod_{j\in K,\ j\neq k} \frac{t - \tilde{t}_j}{\tilde{t}_k - \tilde{t}_j}. \tag{76}$$

For a well-posed polynomial we require that all $\tilde{t}_k$ are distinct. $H(t)$ can be written in terms of the integral values $\phi(\tilde{t}_i)$ by subtracting $P(t_n)$ from both sides and using $\sum_{k\in K}\ell_k(t)=1$:

$$H(t)-P(t_n)=\sum_{k\in K'}\phi(\tilde{t}_k)\;\tilde{c}_k\Delta t\;\ell_k(t), \tag{77}$$

where $K'=\{k_2,\ldots,k_m\}$. Differentiation then leads to

$$h(t)=\sum_{k\in K'}\phi(\tilde{t}_k)\;\tilde{c}_k\Delta t\;\ell'_k(t). \tag{78}$$

Given the values $\phi(\tilde{t}_k)$ and the points $\tilde{t}_k$ this expression can be evaluated at t_{n+1}, which provides the approximation we are looking for:

$$p_{n+1}=h(t_{n+1}). \tag{79}$$

Using $m-1$ ϕ values (the number of elements in K') this results in an $(m-1)$th order accurate pressure. If all ϕ values are used, then $m-1=s$ and one could, in theory, obtain an sth order accurate pressure. Note that, in contrast to ENO schemes, we do not choose the reconstruction points in such a way that the smoothest stencil results.

In practice we cannot use the *exact* average $\phi(\tilde{t}_i)$ to find p_{n+1}. $\phi(\tilde{t}_i)$ is approximated by $\tilde{\phi}_i$, whose order of accuracy depends on the *stage order* of the method. This stage order can be expressed by making use of the so-called *simplifying condition* $C_i(\xi)$ [26]:

$$C_i(\xi):\quad \sum_{j=1}^{s}a_{ij}c_j^{k-1}=\frac{1}{k}c_i^k\quad(1\leqslant k\leqslant\xi), \tag{80}$$

or, in terms of the shifted Butcher tableau:

$$\widetilde{C}_i(\xi):\quad \sum_{j=1}^{s}\tilde{a}_{ij}c_j^{k-1}=\frac{1}{k}\tilde{c}_i^k\quad(1\leqslant k\leqslant\xi). \tag{81}$$

If $C_i(q)$ holds then polynomials of degree lower than q are exactly interpolated at stage i. $C_i(1)$ is equivalent to condition (19). We remark that with the notation of the shifted tableau, $\widetilde{C}_s(q)$ is equal to simplifying condition $B(q)$:

$$B(\eta):\quad \sum_{i=1}^{s}b_ic_i^{k-1}=\frac{1}{k}\quad(1\leqslant k\leqslant\eta). \tag{82}$$

This expresses the order of quadrature of the final step in the Runge–Kutta method, so $\widetilde{C}_s(s)=B(s)$ trivially holds for an s-stage, sth order method. Assuming that $\widetilde{C}_i(q)$ holds, the equation for $\phi(\tilde{t}_i)$ can be written as

$$L\phi(\tilde{t}_i)=\frac{1}{\tilde{c}_i}\sum_{j=1}^{i}\tilde{a}_{ij}MF_j-\frac{r_1(\tilde{t}_i)-r_1(t_n)}{\tilde{c}_i\Delta t}+\mathcal{O}(\Delta t^q), \tag{83}$$

so that the difference between the exact integral and its numerical approximation is $\mathcal{O}(\Delta t^q)$:

$$\phi(\tilde{t}_i)=\tilde{\phi}_i+\mathcal{O}(\Delta t^q). \tag{84}$$

To summarize, it is possible to obtain a higher order accurate pressure at t_{n+1} by combining the average values $\tilde{\phi}_i$ from the different stages. To attain a certain order r requires at least r distinct stages (i.e. with different c_i), and each individual stage i should have stage order r, i.e., satisfy $C_i(r)$. We will now check if this is possible for methods with two, three and four stages.

4.2.1. Two-stage methods

For two-stage, second-order methods $\widetilde{C}_2(2)=B(2)$ is obviously satisfied, but $\widetilde{C}_1(2)$ cannot be satisfied because the equation for $\widetilde{U}_1$ is simply a Forward Euler step, which is first-order accurate (this is always the case for explicit methods).

4.2.2. Three-stage methods

Since $\widetilde{C}_1(2)$ cannot be satisfied, we require $\widetilde{C}_2(2)$ to be satisfied, i.e.,

$$a_{32}c_2=\frac{1}{2}c_3^2, \tag{85}$$

together with the condition $c_3\neq 1$ to have distinct c's. Using the third-order conditions

$$b_3a_{32}c_2=\frac{1}{6},\quad b_2c_2^2+b_3c_3^2=\frac{1}{3}, \tag{86}$$

this leads to $b_1=\frac{1}{4}, b_2=0, b_3=\frac{3}{4}$ and $c_3=\frac{2}{3}$. c_2 can be chosen freely ($\neq 0$), and then determines a_{31} and a_{32}. Wray's popular third-order method [2] falls in this category, with $c_2=\frac{8}{15}$:

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{8}{15} & \frac{8}{15} & & \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}. \tag{87}$$

Another possibility is to take $c_2 = c_3$, which saves an evaluation of boundary conditions and forcing terms:

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}. \tag{88}$$

The interpolation polynomial $h(t)$ from Eq. (78) is independent of c_2 and given by:

$$h(t) = -\left(\frac{2\frac{t-t_n}{\Delta t} - 1}{1 - c_3}\right)\tilde{\phi}_2 + \left(\frac{2\frac{t-t_n}{\Delta t} - c_3}{1 - c_3}\right)\tilde{\phi}_3, \tag{89}$$

and p_{n+1} follows with $c_3 = \frac{2}{3}$ as

$$p_{n+1} = h(t_{n+1}) = -3\tilde{\phi}_2 + 4\tilde{\phi}_3. \tag{90}$$

This equation provides a new way to obtain a second-order accurate pressure by combining two first-order accurate pressures of a three-stage method. It is also valid when Wray's method is used only for the convective terms and an appropriate implicit method for the diffusive terms, such as the method from [1].

If one wants to maximize stability instead of order of accuracy (second-order accuracy is sufficient in many practical applications), one can use three-stage methods that are second order for the velocity. Combining the condition for maximum stability along the imaginary axis ($b_3 a_{32} c_2 = \frac{1}{4}$) with condition (85) yields a family of methods with c_2 and c_3 as free parameters. An example of a low-storage method satisfying these conditions is presented in Perot and Nallapati [27], but it has $c_3 = 1$ so a second-order accurate pressure cannot be obtained. We propose the following alternative method

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\ \hline & 0 & -1 & 2 \end{array}, \tag{91}$$

which we obtained by requiring $b_1 = 0$ and $c_2 = c_3$. The requirement $b_1 = 0$ leads to the same storage requirements as [27], and $c_2 = c_3$ has the advantage that only one intermediate boundary condition evaluation is needed.

4.2.3. Four-stage methods

Four-stage, fourth-order methods have $c_4 = 1$, so that even if $\widetilde{C}_3(2)$ would hold, it cannot be used. We therefore look again for methods that satisfy $\widetilde{C}_2(2)$. Combining condition (85) with the fourth-order conditions leads to $c_3 = \frac{1}{2}$ and two families of solutions result, corresponding to the 'case II' and 'case IV' solutions found by Kutta [17]:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & c_2 & & & \\ \frac{1}{2} & \frac{1}{2} - \frac{1}{8c_2} & \frac{1}{8c_2} & & \\ 1 & \frac{1}{2c_2} - 1 & -\frac{1}{2c_2} & 2 & \\ \hline & \frac{1}{6} & 0 & \frac{2}{3} & \frac{1}{6} \end{array} \qquad c_2 \neq 0, \tag{92}$$

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1 & 1 & & & \\ \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & & \\ 1 & 1 - \frac{1}{4b_4} & -\frac{1}{12b_4} & \frac{1}{3b_4} & \\ \hline & \frac{1}{6} & \frac{1}{6} - b_4 & \frac{2}{3} & b_4 \end{array} \qquad b_4 \neq 0. \tag{93}$$

The upper tableau with $c_2 = c_3 = \frac{1}{2}$ is attractive because it requires only one intermediate evaluation of boundary conditions and forcing terms. In a similar fashion as the three-stage method (Eq. (89) with $\tilde{\phi}_3$ replaced by $\tilde{\phi}_4$), p_{n+1} follows as

$$p_{n+1} = -2\tilde{\phi}_2 + 3\tilde{\phi}_4. \tag{94}$$

Again, this equation provides a new way to obtain a second-order accurate pressure by combining two first-order accurate pressures of a four-stage method.

4.3. Steady boundary conditions for the continuity equation (Method 3)

An important case for the incompressible Navier–Stokes equations is when the boundary conditions for the continuity equation are steady, i.e., Eq. (3) can be written as

$$Mu = r_1, \tag{95}$$

where r_1 is independent of t. Eq. (5) then reads $g(u) = 0$, with g linear in u. This means that g_u = constant and all partial derivatives of g_u (g_{uu},g_{ut},g_{tt}, etc.) are zero. As before, all additional trees for the u-component (Table 1) disappear, but most of the trees for the p-component (Table 2) also vanish. Only trees 5, 8 and 9 remain, and it is possible to find higher order accurate methods for the pressure. For example, with two stages a second-order accurate pressure is possible ($c_2 = 1$), and with four stages a third-order accurate pressure is possible. However, it is not necessary to consider such methods, because in the case g_u = constant the pressure can be computed to the same order of accuracy as the velocity, without additional cost. This can be seen by comparing Eq. (28) for $i = 1$ with Eq. (60):

$$Lp_n = MF_n - \dot{r}_1(t_n), \tag{96}$$

$$L\tilde{\phi}_1 = \frac{\tilde{a}_{11}}{\tilde{c}_1} MF_1 - \frac{r_1(\tilde{t}_1) - r_1(t_n)}{\tilde{c}_1 \Delta t}. \tag{97}$$

Considering that F_1 is equal to F_n (determined at the end of the previous time step) and that $\tilde{a}_{11} = \tilde{c}_1$, these expressions are equal if r_1 is independent of time (or a linear function of t). This means that $\tilde{\phi}_1$ is actually the rth order accurate pressure at t_n, and the additional Poisson solve associated with Eq. (60) can be avoided. An existing implementation could remain unaltered; instead of taking $\tilde{\phi}_s = \phi_{n+1}$, the pressure that makes u_{n+1} divergence free, one should take $\tilde{\phi}_1$ from the next time step to have a higher order accurate pressure at the end of the current time step. We prefer to compute p_{n+1} and then skip the computation of $\tilde{\phi}_1$ in the next time step. This works for any explicit Runge–Kutta method with at least two stages, thus providing a simple way to improve the temporal accuracy of the pressure without increasing computational cost.

4.4. Summary

We classify the methods analyzed so far as follows. The methods that were derived in Section 4.1 (with $\widetilde{A}^p = \widetilde{A}$) will be indicated by M1, the methods derived in Section 4.2 are indicated by M2, and in case Section 4.3 applies we write M3. We then write M*m* S*s* R*r* to indicate that an s-stage explicit Runge–Kutta method of type m is used with order s for the velocity and order r for the pressure. For example, there are three methods of type M1S4R2 and they are given by the tableaux in (68). In case $m = 3$, we can always make r equal to s with the approach of Section 4.2, and any existing s-stage, sth order method can be used. In case $r = 1$ we have the standard approach with $p_{n+1} = \tilde{\phi}_s$, which can be used with any method.

5. Results

5.1. Taylor vortex

The Taylor–Green vortex in two dimensions is an exact solution to the Navier–Stokes equations:

$$u(x,y,t) = -\sin(\pi x)\cos(\pi y)\, e^{-2\pi^2 t/\mathrm{Re}}, \tag{98}$$

$$v(x,y,t) = \cos(\pi x)\sin(\pi y)\, e^{-2\pi^2 t/\mathrm{Re}}, \tag{99}$$

$$p(x,y,t) = \frac{1}{4}(\cos(2\pi x) + \cos(2\pi y)) e^{-4\pi^2 t/\mathrm{Re}}. \tag{100}$$

The domain on which we define the solution is the square $[\frac{1}{4}, 2\frac{1}{4}] \times [\frac{1}{4}, 2\frac{1}{4}]$ with either time-dependent Dirichlet or periodic boundary conditions. Prescribing both inflow and outflow with Dirichlet conditions is normally not a good idea, see e.g. [28], but we did not find any negative effects in this test case. We deliberately do not take the square $[0,2] \times [0,2]$ (or $[-1,1] \times [-1,1]$) as domain, because it leads to $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on the entire boundary, meaning that $r_1(t) = 0$. In all simulations we use Re = 100 and we integrate from $t = 0$ to $t = 1$. The Poisson equation is solved by LU-decomposition of L, which is the most efficient way for this relatively small problem.

As mentioned in Section 2.1, a consistent initial condition requires that p_0 is obtained by solving a Poisson equation (Eq. (15)), but for the explicit methods that are under consideration here, both the velocity field and the pressure field at each time step are not depending on p at the start of a time step, so the initial condition for the pressure is unnecessary.

5.1.1. Verification of spatial accuracy

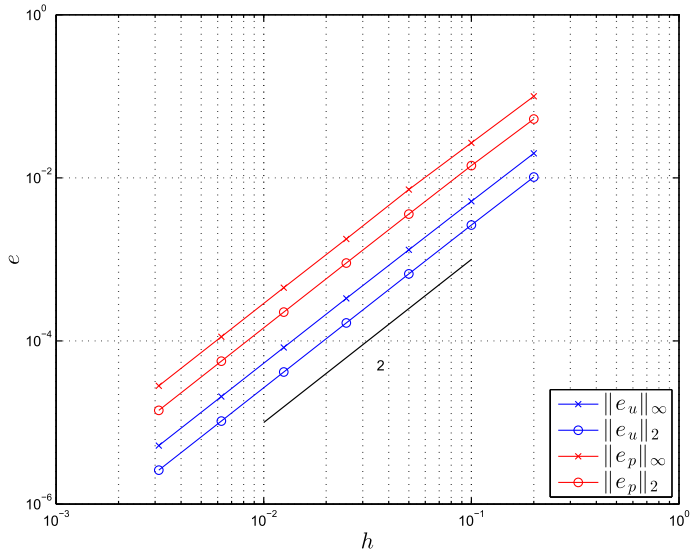
We simulate the Taylor vortex on different uniform grids ranging from 5×5 to 320×320 volumes using a fixed time step of $\Delta t = 10^{-3}$, and then we calculate the error in the resulting velocity and pressure fields by comparing with the exact solution at $t = 1$. We use the classical fourth-order, four-stage Runge–Kutta method for this, which ensures that the temporal errors are negligible compared to the spatial errors.

Figs. 1(a) and (b) show that for both periodic and (unsteady) Dirichlet boundary conditions the error in the u-velocity component and the pressure is second order in the L_2- and L_∞-norm. The error in the v-velocity component is not shown because it is almost indistinguishable from the u-velocity component.

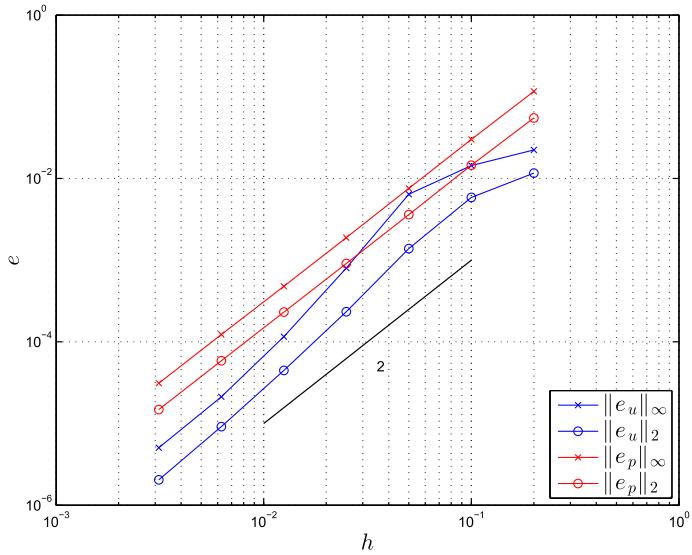
5.1.2. Temporal accuracy

We now take a coarse mesh with 20×20 volumes and vary the time step to investigate the temporal accuracy. The spatial error clearly overwhelms the temporal error and in order to compute the latter, we subtract the solution from a simulation with a small time step ($\Delta t = 10^{-3}$), so that the spatial error is effectively eliminated.

The first test concerns periodic boundary conditions, such that the observations from Section 4.3 apply and the methods are characterized as M3. Four s-stage, sth-order Runge–Kutta methods are tested, with $s = 1, 2, 3, 4$. For $s = 1$ we take Forward Euler, for $s = 2$ modified Euler (explicit trapezoidal, Heun's method), for $s = 3$ Wray's method, and for $s = 4$ the classical fourth-order method. In all cases the number of Poisson solves is the same as the number of stages. Fig. 2(a) shows that with our current approach both pressure and velocity attain the classical order of convergence, whereas the standard method ($p_{n+1} = \tilde{\phi}_s$) leads to only first order convergence of the pressure, see Fig. 2(b). The velocity error is unaffected by the accuracy

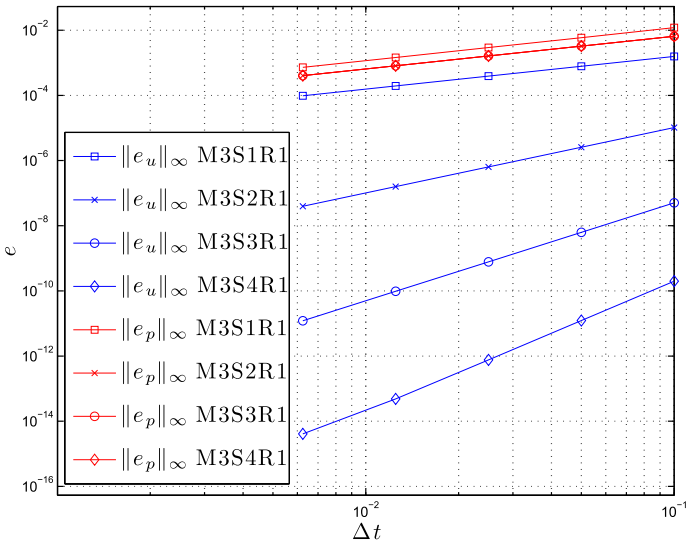


(a) periodic boundary conditions

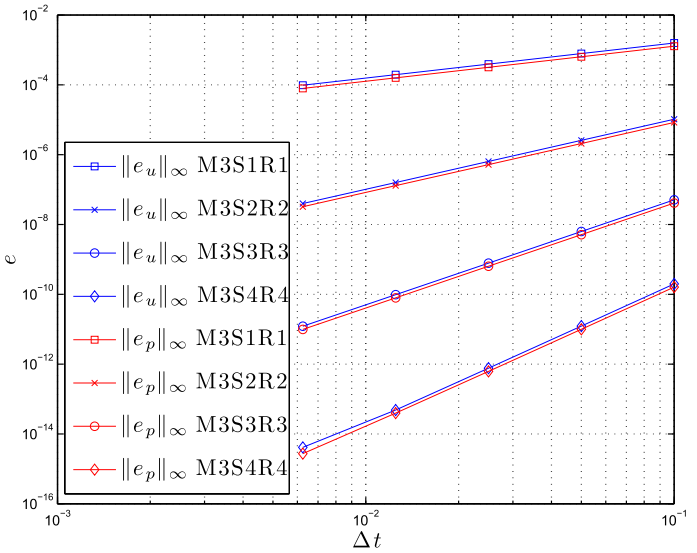


(b) Dirichlet boundary conditions

Fig. 1. Convergence of spatial error for Taylor–Green problem.



(a) Standard method



(b) New method for steady boundary conditions

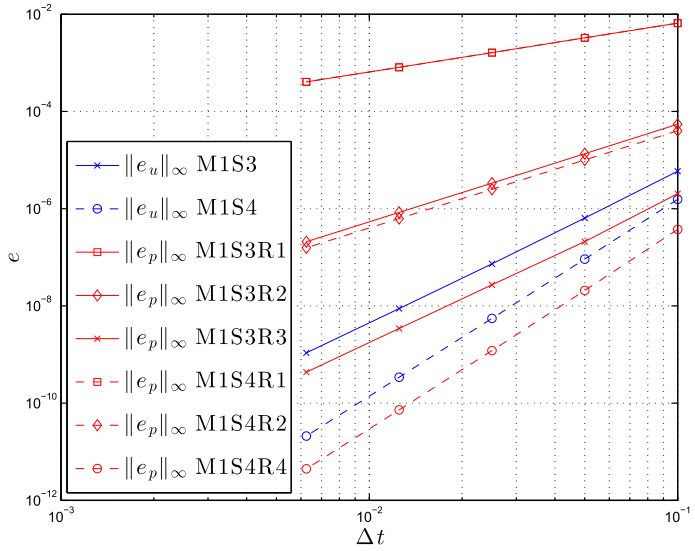
Fig. 2. Convergence of temporal error for Taylor–Green problem with steady boundary conditions.

of the pressure. For this very smooth test case the error of higher order methods does not only converge faster upon time step refinement (as predicted by theory), but the magnitude of the error for the largest time step is also much smaller. We only show here the L_∞-norm because the L_2-norm shows exactly the same behavior. Comparison with Fig. 1(a) shows that for this mesh and boundary conditions the temporal error is much smaller than the spatial error, for all methods (except Forward Euler), even for $\Delta t = 10^{-1}$. In this particular example the two-stage, second-order method is stable and accurate enough and is to be preferred over the more expensive three- and four-stage methods.

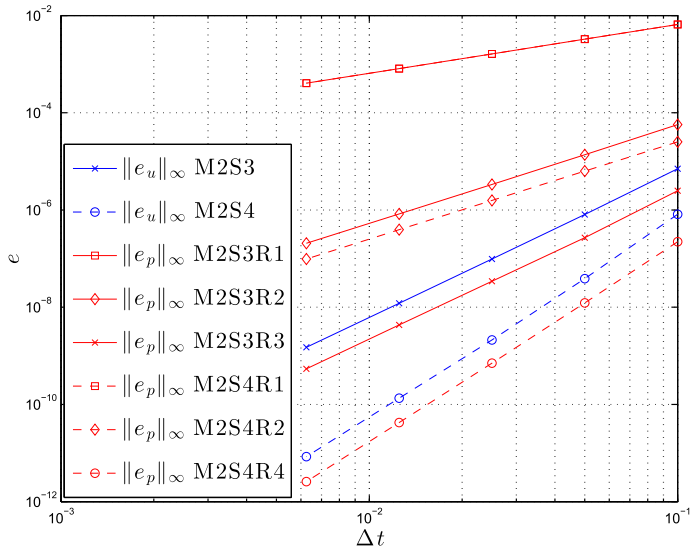
The second test concerns unsteady Dirichlet conditions with methods of type M1. For $s = 1$ and $s = 2$ only $r = 1$ is possible, so we focus on $s = 3$ and $s = 4$ with $r = 2$. For M1S3R2 the only solution is (66) with (67) for the pressure, for M1S4R2 we take the first tableau in (68) (because it has the simplest coefficients) and (69) for the pressure.

The third test concerns unsteady Dirichlet conditions with methods of type M2. As for methods of type M1, we focus on $s = 3$ and $s = 4$. For M2S3R2 we take Wray's method with (90) for the pressure and for M2S4R2 we take (92) with $c_2 = \frac{1}{4}$ and (94) for the pressure.

Figs. 3(a) and (b) then show the order of accuracy of the velocity and pressure for these two methods, in case of the 'standard' approach (R1), our approach (R2) and in case of an additional Poisson solve (R3 or R4). The additional Poisson solve can



(a) New method, with single Butcher array for velocity and pressure (M1)



(b) New method, with reconstruction of instantaneous pressure values from time averages (M2)

Fig. 3. Convergence of temporal error for Taylor–Green problem with unsteady Dirichlet boundary conditions.

be performed because the explicit dependence of r_1 on t is known so that $\dot{r}_1(t)$ can be calculated. The velocity error is in all cases again independent of the particular approach for the pressure. It is confirmed that both methods M1 and M2 indeed lead to a second-order accurate pressure when the proper Butcher tableaux are chosen. The difference in accuracy between the results of M1 and M2 is small, but this depends on the test case under consideration. The computational effort is for both very similar. These second-order schemes greatly improve the accuracy with respect to the standard first-order approach without additional cost. The effort of an additional Poisson solve can only be justified in case higher-order accurate (third or fourth order) pressure solutions are required.

5.2. An actuator disk in an unsteady inflow field

A practically relevant situation with a temporally varying inflow appears when simulating the flow of air through wind turbines operating in a turbulent atmospheric wind field. Such simulations of wind-turbine wakes can contribute to the

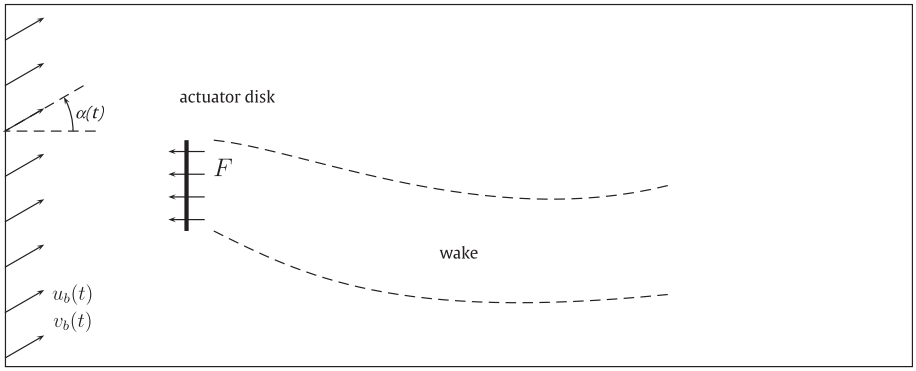


Fig. 4. Actuator disk in an unsteady inflow field.

understanding of the effect of wakes on power production and blade loading. Different methods exist for modeling the effect of the wind turbines on the flow, such as actuator methods and direct methods [29]. In the former, the action of the turbine is modeled with a body force, such as an actuator surface, and in the latter the action of the turbine is modeled by computing the actual flow around it, for example by applying a body fitted grid around the turbine blades. In both cases accurate knowledge of the pressure field is necessary to calculate the forces that act on the blade, such as lift and drag.

Here we study a simplified case of an actuator disk in a laminar flow. The domain is $[0,10] \times [-2,2]$, the Reynolds number is 100 and the thrust coefficient of the turbine is $C_T = \frac{1}{2}$. The actuator disk is located at $x = 2$ and has unit length, see Fig. 4. On all boundaries, except the inflow boundary at $x = 0$, we prescribe the following outflow conditions (see e.g. [28]):

$$y = -2, 2: \qquad \frac{\partial u}{\partial y} = 0 \quad p - \frac{1}{\mathrm{Re}} \frac{\partial v}{\partial y} = p_\infty, \tag{101}$$

$$x = 10: \quad p - \frac{1}{\mathrm{Re}} \frac{\partial u}{\partial x} = p_\infty \qquad \frac{\partial v}{\partial x} = 0. \tag{102}$$

For a verification study of the actuator disk in a laminar flow with steady inflow and these boundary conditions we refer to [30]. In the current test, the inflow conditions are given by:

$$x = 0: \quad u_b(t) = \cos\alpha(t), \quad v_b(t) = \sin\alpha(t), \tag{103}$$

where $\alpha(t) = \frac{\pi}{6}\sin(t/2)$. This describes a time-varying inflow with constant magnitude but changing direction, see Fig. 5.

First we perform a simulation from $t = 0$ to $t = 4\pi$, with a uniform mesh having 200×80 volumes ($\Delta x = \Delta y = 1/20$) and 10,000 time steps ($\Delta t = 4\pi/10000$), using the M2S4R4 method of Eq. (92), again with $c_2 = \frac{1}{4}$. We focus on methods of type M2, because they still allow for some freedom in the choice of the coefficients of the Butcher tableau, in contrast to methods

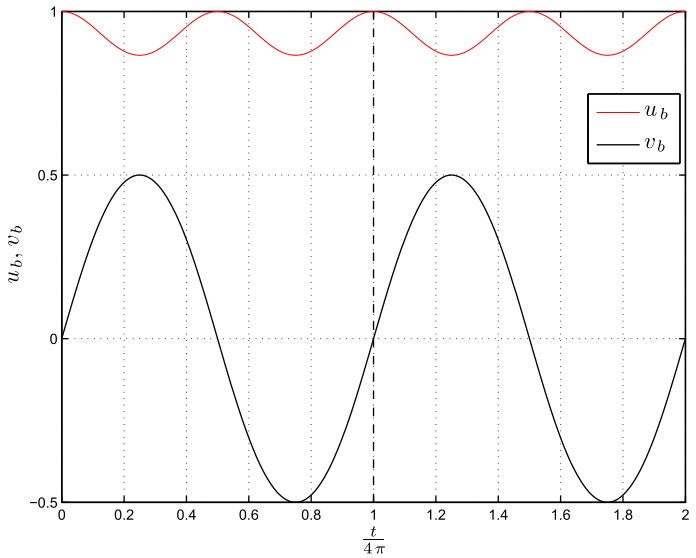


Fig. 5. Inflow as a function of time.

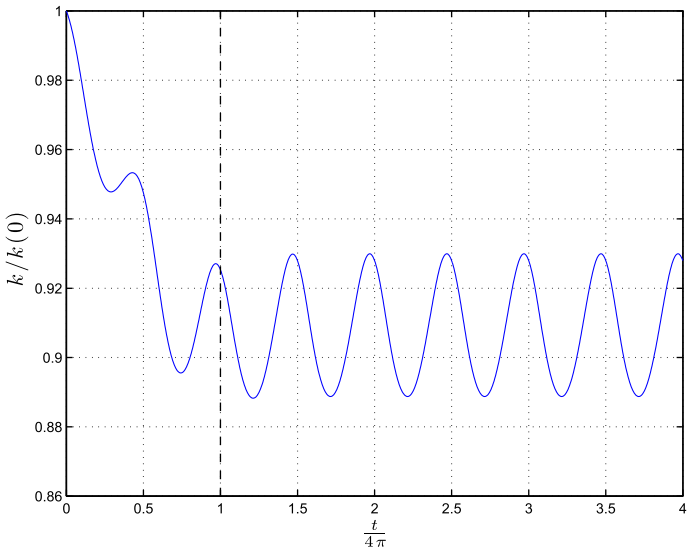


Fig. 6. Kinetic energy, integrated over the entire computational domain, as a function of time.

of type M1. In Fig. 6 the normalized kinetic energy of the flow (integrated over the entire domain) is shown, from which it can be concluded that the flow becomes periodic with period 2π after approximately $t = 4\pi$. The velocity and pressure field at this time instant are shown in Figs. 7 and 8. The wake has been deflected downwards due to the inflow with negative v_b that was present from $t = 2\pi$ to 4π. The presence of the actuator disk is clearly seen in the pressure contours; they are discontinuous across the disk.

Due to the very small time step, these velocity and pressure fields have a negligible temporal error compared to the spatial error, and are therefore used to compute the temporal error in the velocity and pressure field for larger time steps. The resulting convergence of the velocity and pressure error is shown in Fig. 9, for methods (87) and (92). As before, we see that the velocity attains its classical order of accuracy, i.e., third order for the three-stage method, and fourth order for the four-stage method. The pressure can be computed to the same order as the velocity, but this requires an additional Poisson solve and an expression for $\dot{r}_1(t)$. Since $\dot{r}_1(t)$ contains only the normal velocity component on the boundary, it is sufficient to derive the expression for $\dot{u}_b(t)$:

$$\dot{u}_b(t) = -\frac{\pi}{12}\sin(\alpha(t))\cos(t/2). \tag{104}$$

On the other hand, the standard approach is only first order and starts with a large error at large time steps. Our proposed approach, corresponding to the lines M2S3R2 and M2S4R2, does not require any significant additional computational effort (no additional Poisson solve, no evaluation of $\dot{r}_1(t)$), it clearly shows second-order accuracy and starts with a small error already at the largest time step considered. This time step, $\Delta t = 4\pi/200$, is the largest step for which stable solutions could be obtained. It is determined by the convective terms, showing the benefit of explicit Runge–Kutta methods for this test case.

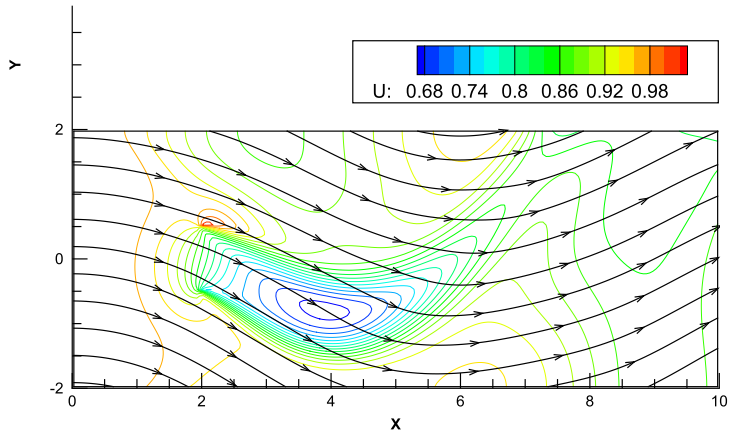


Fig. 7. Streamlines and u-contour lines at $t = 4\pi$.

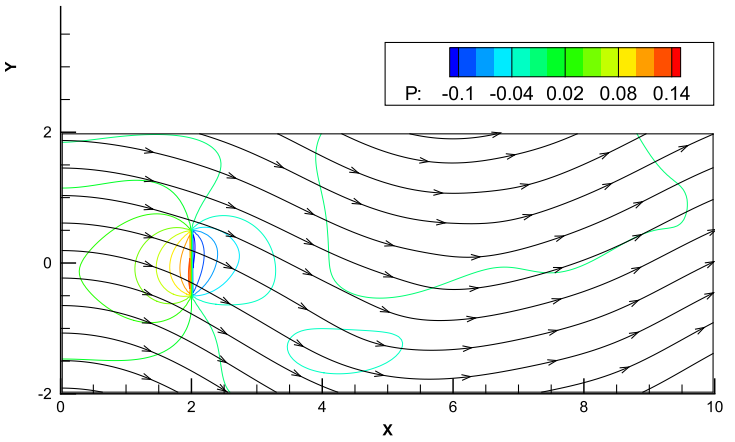


Fig. 8. Streamlines and p-contour lines at $t = 4\pi$.

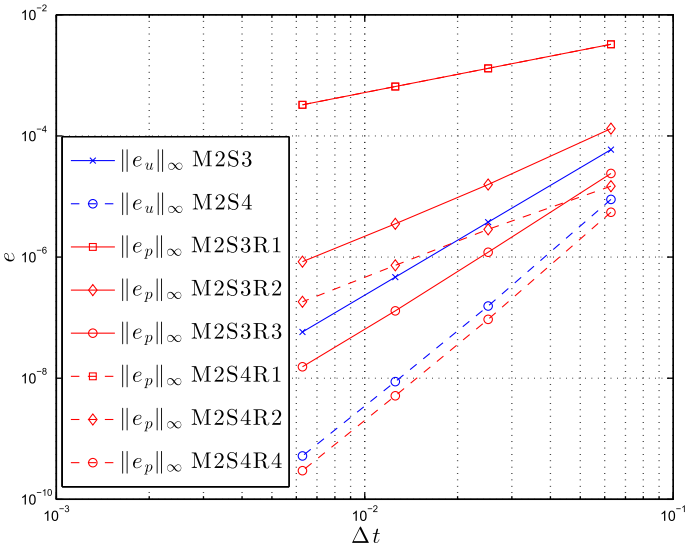


Fig. 9. Velocity and pressure error at $t = 4\pi$ for a selection of methods.

6. Conclusions

In this paper we have analyzed the temporal order of accuracy of the velocity and pressure when explicit Runge–Kutta methods are applied to the incompressible Navier–Stokes equations. It is shown that the order of accuracy of the velocity is not affected by the differential–algebraic nature of the incompressible Navier–Stokes equations and is therefore the same as for non-stiff ordinary differential equations. However, if the semi-discrete equations involve time-dependent operators, then additional order conditions appear for orders higher than two. These conditions restrict three-stage, third-order methods to a one-parameter family of methods, to which the popular method of Wray does not belong. Four-stage, fourth-order methods for time-varying operators *do not exist*, and one has to resort to five stages to achieve fourth order. In any case (time-dependent and time-independent operators) the pressure suffers from the problem that upon time-stepping a time-average pressure is computed, instead of a point value. Therefore, achieving higher than first order accuracy for the pressure imposes additional conditions on the coefficients of the Runge–Kutta method compared to the classical order conditions.

Fortunately, if the boundary conditions for the continuity equation are independent of time, then the pressure can be determined to the same order of accuracy as the velocity, without requiring an additional solution of a Poisson problem.

However, if the boundary conditions for the continuity equation depend on time, then additional order conditions for the pressure appear. These are not satisfied by most existing explicit Runge–Kutta methods, so that the pressure is typically only first-order accurate in time. Using the same Butcher tableau for velocity and pressure, second-order accuracy can be achieved by only one three-stage, and only three four-stage methods.

An alternative approach is to reconstruct instantaneous pressure values from time-average values. We showed that this reconstruction, based on Lagrange polynomials, can be of the same order as the number of stages, but that the stage order of the method limits the accuracy of the pressure. These methods can be interpreted as having a different Butcher tableau for velocity and pressure, in contrast to the foregoing single-Butcher array approach. Three- and four-stage methods with second-order stage order were derived, leading to a much larger class of methods that have second-order accuracy for the pressure. Furthermore, a distinct advantage of this new class of methods is that they can be directly applied to implicit and implicit-explicit (IMEX) Runge–Kutta methods as well.

In all cases considered here third-order accuracy could not be obtained with a three- or four-stage method without resorting to an additional Poisson solve. Such an additional solve is not always straightforward in practical computations, because it requires the derivative of the boundary conditions for the continuity equation with respect to time. If the additional Poisson solve is performed, the proposed second-order accurate methods can be used to provide an accurate and cheap initial guess for iterative methods (such as the conjugate gradient method) to solve this Poisson equation.

Runge–Kutta methods with more than four stages might perhaps lead to higher-order accurate pressures, but such methods are not very relevant from a practical point of view. For example, when considering five-stage methods (requiring five Poisson solves), to obtain third-order accuracy for the pressure, it would be better to employ an additional Poisson solve in a four-stage method, leading to fourth-order accuracy of the pressure (assuming that $\dot{r}_1(t)$ is available). However, methods with more stages than the classical order of accuracy (e.g. a four-stage third-order method) can be of interest from a stability point of view. Such methods can possibly be used with less Poisson solves than the number of stages.

To conclude, we think that the 'best' explicit Runge–Kutta method for many incompressible Navier–Stokes problems is a three-stage method of type M2, that is third order for the velocity and second order for the pressure. It combines stability (includes the imaginary axis), sufficient accuracy (temporal error is in general sufficiently small) and flexibility (c_2 can still be chosen, in contrast to the three-stage method of type M1). A fourth-order method might lead to unnecessarily accurate solutions for the velocity without improving the order of accuracy of the pressure. For time-dependent operators, the three-stage method derived in Section 4.1.2 is to be preferred: it maintains third-order accuracy on time-varying meshes, and is second-order accurate for the pressure. Of course, the coefficients of a Runge–Kutta method can be chosen on other grounds than accuracy only, for example low storage, low dispersion or built-in error estimation with adaptive step-size control. Such arguments have not been considered in this paper.

References

[1] P. Spalart, R. Moser, M. Rogers, Spectral methods for the Navier–Stokes equations with one infinite and two periodic directions, J. Comput. Phys. 96 (1991) 297–324.
[2] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 92 (1991) 369–379.
[3] Y. Morinishi, T. Lund, O. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flows, J. Comput. Phys. 143 (1998) 90–124.
[4] R. Knikker, Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flow, Int. J. Numer. Methods Fluids 59 (2009) 1063–1092.
[5] N. Nikitin, Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 51 (2006) 221–233.
[6] J. Pereira, M. Kobayashi, J. Pereira, A fourth-order-accurate finite volume compact method for the incompressible Navier–Stokes equations, J. Comput. Phys. 167 (2001) 217–243.
[7] N. Kampanis, J. Ekaterinaris, A staggered, high-order accurate method for the incompressible Navier–Stokes equations, J. Comput. Phys. 215 (2006) 589–613.
[8] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods, Springer, 1989.
[9] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Problems, Springer, 1996.
[10] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[11] U. Ascher, L. Petzold, Computer Methods for Ordinary Differential Equations and Differential–Algebraic Equations, SIAM, 1998.
[12] P. Gresho, R. Sani, Incompressible flow and the finite element method, Isothermal Laminar Flow, vol. 2, Wiley, 2000.
[13] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.
[14] S. Armfield, R. Street, An analysis and comparison of the time accuracy of fractional-step methods for the Navier–Stokes equations on staggered grids, Int. J. Numer. Methods Fluids 38 (2002) 255–282.
[15] J. Perot, An analysis of the fractional step method, J. Comput. Phys. 180 (1993) 51–58.
[16] J. Perot, Comments on the fractional step method, J. Comput. Phys. 121 (1995) 190–191.
[17] J. Butcher, Numerical Methods for Ordinary Differential Equations, Wiley, 2003.
[18] V. Brasey, E. Hairer, Half-explicit Runge–Kutta methods for differential–algebraic systems of index 2, SIAM J. Numer. Anal. 30 (1993) 538–552.
[19] J. Butcher, Coefficients for the study of Runge–Kutta integration processes, J. Aust. Math. Soc. 3 (1963) 185–201.
[20] F. Cameron, A Matlab package for automatically generating Runge–Kutta trees, order conditions, and truncation error coefficients, ACM Trans. Math. Software 32 (2006) 274–298.
[21] M. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error, SIAM J. Sci. Comput. 16 (1995) 1241–1252.
[22] S. Abarbanel, D. Gottlieb, M. Carpenter, On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations, SIAM J. Sci. Comput. 17 (1996) 777–782.
[23] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, Springer, 2003.
[24] D. Pathria, The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial boundary value problems, SIAM J. Sci. Comput. 18 (1997) 1255–1266.
[25] C.-W. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, Technical Report NASA/CR-97-206253, ICASE Report no. 97-65, NASA Langley Research Center, 1997.
[26] J. Butcher, Implicit Runge–Kutta processes, Math. Comput. 18 (1964) 50–64.
[27] B. Perot, R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys. 184 (2003) 192–214.

[28] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, 2001.
[29] B. Sanderse, S. van der Pijl, B. Koren, Review of computational fluid dynamics for wind turbine wake aerodynamics, Wind Energ. 14 (2011) 799–819.
[30] B. Sanderse, ECNS: Energy-Conserving Navier–Stokes Solver. Verification of Steady Laminar Flows, Technical Report ECN-E–11-042, Energy research Centre of the Netherlands, 2011.