Analysis of the numerical dissipation rate of different Runge-Kutta and velocity interpolation methods in an unstructured collocated finite volume method in OpenFOAM®

E.M.J. Komen[a,*], E.M.A. Frederix[a], T.H.J. Coppen[a], V. D'Alessandro[b], J.G.M. Kuerten[c]

[a]*Research and Innovation department, Nuclear Research and consultancy Group NRG, P.O. Box 25, 1755 ZG Petten, The Netherlands*

[b]*Dipartimento di Ingegneria Industriale e Scienze Matematiche, Università Politecnica delle Marche, Via Brecce Bianche, 60131 Ancona (AN), Italy*

[c]*Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands*

Abstract

The approach used for computation of the convecting face fluxes and the cell face velocities results in different underlying numerical algorithms in finite volume collocated grid solvers for the incompressible Navier-Stokes equations. In this study, the effect of the following five numerical algorithms on the numerical dissipation rate and on the temporal consistency of a selection of Runge-Kutta schemes is analysed: 1) the original algorithm of Rhie and Chow (1983), 2) the standard OpenFOAM method, 3) the algorithm used by Vuorinen et al. (2014), 4) the Kazemi-Kamyab et al. (2015) method, and 5) the D'Alessandro et al. (2018) approach. The last three algorithms refer to recent implementations of low dissipative numerical methods in OpenFOAM®.

No new computational methods are presented in this paper. Instead, the main scientific contributions of this paper are: 1) the systematic assessment of the effect of the considered five numerical approaches on the numerical dissipation rate and on the temporal consistency of the selected Runge-Kutta schemes within one unified framework which we have implemented in OpenFOAM, and 2) the application of the method of Komen et al. (2017) in order to quantify the numerical dissipation rate introduced by three of the five numeri-

[*] Corresponding author
Email address: komen@nrg.eu (E.M.J. Komen)

cal methods in quasi-DNS and under-resolved DNS of fully-developed turbulent channel flow. In addition, we explain the effects of the introduced numerical dissipation on the observed trends in the corresponding numerical results.

As one of the major conclusions, we found that the pressure error, which is introduced due to the application of a compact stencil in the pressure Poisson equation, causes a reduction of the order of accuracy of the temporal schemes for the test cases in this study. Consequently, application of higher order temporal schemes is not useful from an accuracy point of view, and the application of a second order temporal scheme appears to be sufficient.

Keywords: Rhie-Chow, velocity interpolation methods, time integration schemes, numerical dissipation rate, LES, quasi-DNS, UDNS, OpenFOAM.

1. Introduction

High-fidelity numerical methods are a prerequisite for the numerical simulation of turbulent flows using Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES). The continuity and Navier-Stokes equations conserve mass, momentum, and kinetic energy in the limit of inviscid and incompressible flow. Therefore, it is highly desirable to have numerical methods which conserve mass, momentum, and kinetic energy at the numerical level too. The discrete conservation properties of numerical methods for the numerical solution of the continuity and Navier-Stokes equations are strongly dependent on the way the flow variables are arranged on the grid. On staggered grids, discretisation methods which conserve both mass, momentum, and kinetic energy have been published in quite a number of papers, e.g., Ham et al. (2002) and Morinishi et al. (2004). However, the extension of the staggered mesh approach to complex solution domains is not straightforward.

The collocated grid arrangement offers significant advantages over the staggered grid approach for complex solution domains. Namely, the collocated grid shortens the computational time and reduces the required memory storage, especially for unstructured and curvilinear body fitted grids in three-dimensional computations. Furthermore, collocated grids offer much simpler code implementation, especially when a multigrid procedure is used for acceleration of the convergence rate. This probably explains why the collocated mesh approach is used in commercial codes such as ANSYS-FLUENT (ANSYS-FLUENT, 2011) and STAR-CCM+ (STAR-CCM+, 2013), and in open source codes such as OpenFOAM (OpenFOAM®, 2017) and Code-Saturn (Archambeau et al., 2004). However, the use of a central discretisation stencil, which is a prerequisite for high-fidelity simulations, together with collocated grids results in the well known checkerboard oscillations (see, e.g., Versteeg and Malalasekera (2007) and Ferziger and Perić (1997)). This checkerboard problem is caused by the resulting wider stencil in the Poisson equation for the pressure, which yields a decoupling of nearby grid points. To remedy this problem, Rhie and Chow (1983) proposed an interpolation method for the cell face velocities named Pressure-Weighted Interpolation Method (PWIM). Subsequently, various alternative solutions have been proposed in the literature. An extensive review of alternative velocity interpolation methods published in the literature until 2007 is presented by Pascau (2011). The proposed alternative solutions are similar in the sense that the face velocities are calculated based on a pressure gradient which is taken directly at the cell faces instead of interpolating the adjacent cell center pressure gradients. This approach results in a compact stencil in the pressure Poisson equation. Although this approach provides a remedy to the checkerboard problem, a fourth-order dissipation term is introduced in the pressure Poisson equation which causes dissipation of kinetic energy (see, e.g., Versteeg and Malalasekera (2007) and Ferziger and Perić (1997)).

An additional problem of the collocated arrangement results from the fact that there are two different velocity fields, namely, the convecting velocities at the cell faces and the cell center velocities. These cell center velocities, together with the pressure, form the primary solution variables. The cell face velocities do not have an own transport equation. Instead, they are calculated from the cell center values of the velocity and pressure field. By application of the pressure Poisson equation, the continuity constraint is applied to the cell face velocities. As explained by Morinishi et al. (1998) and Felten and Lund (2006), the primary cell center velocities are only approximately divergence free when a compact stencil for the pressure Poisson equation is used. Furthermore, they explain that this leads to an error in the kinetic energy conservation.

Despite the deficiencies in the kinetic energy conservation as discussed above, Komen et al. (2014) demonstrated that DNS accuracy can practically be achieved for turbulent pipe flow computations using unstructured collocated grids with arbitrary polyhedral cells with typical DNS mesh resolution using OpenFOAM. However, using the same approach, but with typical LES mesh resolution, the introduced numerical dissipation rate may be substantially larger that the SGS dissipation rate. This makes the application of explicit LES modelling ineffective (Castiglioni and Domaradzki (2015), Komen et al. (2017)). Therefore, less dissipative methods for unstructured collocated finite volume CFD solvers are considered to be very valuable for application to high-fidelity analyses in industrial applications with complex solution domains. In this respect, a number of studies have been published in the recent literature with the aim to reduce the numerical dissipation rate in OpenFOAM.

Vuorinen et al. (2014) implemented an incompressible explicit Runge-Kutta (RK) based projection method in OpenFOAM. They used the classical fourth order RK4 and accelerated third order ARK3 time integration schemes. Vuorinen et al. (2014) concluded that the implemented RK projection methods have good computational efficiency and provide low-dissipative alternatives to OpenFOAM's standard implemented PISO method (Issa, 1985) with 1^{st} and 2^{nd} order accurate implicit temporal schemes. Furthermore, they concluded that the standard implemented PISO method is relatively low-dissipative as well. Kazemi-Kamyab et al. (2015) implemented high order implicit RK time integration methods in an incompressible iterated PISO-based solver in OpenFOAM. Furthermore, they introduced a face velocity interpolation method which preserves the formal order of the implemented temporal schemes. D'Alessandro et al. (2018) implemented the incompressible explicit RK based projection algorithm developed by Sanderse and Koren (2012) in OpenFOAM. Furthermore, they implemented a slight variant of the iterated PISO-based procedure developed by Kazemi-Kamyab et al. (2015) together with high order Singly Diagonally Implicit Runge-Kutta (SDIRK) schemes. Both explicit and implicit algorithms were extended to flow with heat transfer. They concluded that the implemented RK-based solvers have very low numerical dissipation properties, which make them suitable for turbulence simulation. It is noted that RK based compressible flow solvers have been implemented in OpenFOAM by Shen et al. (2014) and Modesti and Pirozzoli (2017).

Concerning the development of energy conserving methods in collocated grid solvers, Hicken et al. (2005) developed a fully-conservative method for staggered unstructured grids. They used so-called shift transformations to obtain staggered variables from collocated variables. They applied their method to an adaptive unstructured Cartesian mesh. The extension of their approach to general unstructured collocated meshes seems unlikely, since it is impossible to construct proper shift transformations for such grids (van der Blij, 2007). Shashank et al. (2010) presented an incompressible Navier-Stokes method for a Cartesian collocated grid arrangement which exactly conserves mass, momentum and kinetic energy. They use the wider stencil in the discrete Poisson equation based on second-order central differencing. In order to circumvent the checkerboard problem while solving the discrete Poisson equation, they modify the obtained pressure solution by adding a combination of null space modes to create a smooth final pressure field. To the best of the authors' knowledge, the extension of this approach to unstructured collocated grids is not published. As a generalization of the approach presented by Verstappen and Veldman (2003), Trias et al. (2014) developed a fully-conservative incompressible Navier-Stokes discretisation for an unstructured collocated mesh scheme. They use a fully-conservative regularization of the convective term in order to eliminate possible spurious checkerboard oscillations without introducing numerical dissipation.

Different algorithms can be distinguished for application in unstructured collocated finite volume CFD solvers based on the selected approach for computation of the convecting face fluxes and for computation of the cell face velocities. The main objective of the present study is to determine the effect of five of these algorithm on the numerical dissipation rate and on the temporal consistency of a selection of Runge-Kutta schemes. More specifically, we compare the following five algorithms within one unified framework which we implemented in OpenFOAM: 1) the standard method of Rhie and Chow (1983), 2), the algorithm used by Vuorinen et al. (2014), 3) the standard method implemented in OpenFOAM, 4) the temporally consistent method of Kazemi-Kamyab et al. (2015), and 5) the method applied by D'Alessandro et al. (2018). This assessment will be based on Taylor-Green vortex and lid-driven cavity test cases. We use a uniform orthogonal hexahedral and a distorted hexahedral mesh topology for these test cases. For the least dissipative approach, we subsequently explore whether we can achieve numerical dissipation rates which are smaller than the SGS dissipation rate for explicit LES modelling of turbulent channel flow. Finally, we explore to what extent the accuracy of the quasi-DNS (q-DNS) computations presented in Komen et al. (2014) and Komen et al. (2017), where the standard icoFOAM solver was used, can be improved by application of the least dissipative approach. The secondary objective of this study is to provide a description of the RKFoam solver which we have developed in order to test the considered different algorithms within one unified framework.

2. RKFoam numerical methods

The numerical methods which form the basis of the RKFoam solver which we have developed are described in this section. Our RKFoam solver is based on the open source library OpenFOAM5.0 (OpenFOAM®, 2017). This solver combines a number of well-known algorithms. By introducing a number of algorithm parameters, it is possible to switch from algorithm to algorithm within the same framework, allowing for a systematic comparison. The basic algorithm in the RKFoam solver has the following three iterative levels:

1. a loop over each Runge-Kutta stage i, indicated by $1 \leq i \leq s$;
2. an outer iteration loop for updating the non-linear convective term, indicated by $1 \leq j \leq \ell$;
3. an inner PISO iteration loop for the pressure-velocity coupling, indicated by $1 \leq k \leq m$.

The goal of the second level is to update the non-linear convective term. The goal of the third level is to perform PISO iteration loops (Issa, 1985) in order to solve for the pressure and velocity in a segregated way. The RKFoam algorithm has four key parameters. These parameters control the behaviour of the algorithm. These parameters are:

1. the Runge-Kutta scheme, which can be either Explicit Runge-Kutta (ERK) or Diagonally Implicit Runge-Kutta (DIRK);
2. the way the convective face flux is computed in the momentum equations;
3. the treatment of the non-linearity of this convective face flux. This can either be linearized, or fully non-linear;

4. the method for the computation of the cell face velocity and the corresponding so-called incomplete flux term. This incomplete flux term is the flux which appears in the pressure Poisson equation.

These four parameters will be discussed in the subsequent sections.

2.1. Governing equations

For a Newtonian constant-density flow without gravity and with constant physical properties, the general conservation equations of mass and momentum reduce to respectively (Bird et al. (1960)):

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

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u}\,\mathbf{u}) - \frac{1}{\rho}\nabla p + \nabla \cdot (\nu\,\nabla\,\mathbf{u}), \tag{2}$$

where $\mathbf{u}$ represents the velocity, p the pressure field, ρ the density, and ν the kinematic viscosity. The corresponding filtered Navier-Stokes equations are (see e.g. Durbin and Petterson-Reif (2011), Pope (2000)):

$$\nabla \cdot \hat{\mathbf{u}} = 0, \tag{3}$$

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} = -\nabla \cdot (\hat{\mathbf{u}}\,\hat{\mathbf{u}}) - \frac{1}{\rho}\nabla \hat{p} + \nabla \cdot (\nu\,\nabla\,\hat{\mathbf{u}}) - \nabla \cdot \boldsymbol{\tau}^{SGS}, \tag{4}$$

where $\hat{\mathbf{u}}$ represents the filtered velocity, $\hat{p}$ the filtered pressure field, and $\boldsymbol{\tau}^{SGS}$ the subgrid-scale (SGS) (or: residual) stress tensor defined as

$$\boldsymbol{\tau}^{SGS} = \widehat{\mathbf{u}\,\mathbf{u}} - \hat{\mathbf{u}}\,\hat{\mathbf{u}}. \tag{5}$$

In this formulation, explicit commutator errors are not taken into consideration (van der Bos and Geurts, 2005).

2.2. The PISO pressure correction method

The PISO (Pressure-Implicit with Splitting of Operators) method of Issa (1985) is based on a predictor-corrector approach. For a detailed description of the method, we refer to the literature, see, e.g., Issa (1985), Ferziger and Perić (1997), and Versteeg and Malalasekera (2007). In an analysis of the accuracy of the considered PISO method, Issa (1985) demonstrated that the order of the accuracy is increased by one for each additional corrector stage. As a result, the effort of using more than two corrector stages appears to be unnecessary when the standard temporal schemes up to second order accuracy are used in OpenFOAM. Furthermore, Issa (1985) demonstrated that, although the pressure field after one corrector step is a second order accurate approximation of the pressure field at the next time level, it may still be a poor approximation. Therefore, at least two corrector stages should be used in order to obtain a sufficiently accurate pressure field.

2.3. Temporal discretisation

The governing partial differential equations must be discretized in both space and time. Application of the spatial discretisation, which is presented in section 2.4, to the momentum equations, yields a system of ordinary differential equations of the form

$$\frac{d\mathbf{u}_P}{dt} = \mathbf{F}_P(t, \mathbf{u}_P) - \frac{1}{\rho}(\nabla p)_p \qquad \text{with} \qquad \mathbf{u}_P(t^n) = \mathbf{u}_P^n, \tag{6}$$

where the pressure gradient term is not yet discretised. For discretisation of the time derivative, we have selected a family of ERK and DIRK schemes. At the basis of a Runge-Kutta scheme stands the so-called Butcher tableau (Butcher, 1964). For both ERK and DIRK schemes, the Butcher tableau is given by (see, e.g., Hairer et al. (2008))

$$\begin{array}{c|cccc} c_1 & a_{11} & & & \\ c_2 & a_{21} & a_{22} & & \\ \vdots & \vdots & \vdots & \vdots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

where $a_{i\gamma}$ are the stage weights of stage i and c_i are the quadrature nodes of the scheme with

$$c_i = \sum_{\gamma=1}^{i} a_{i\gamma} \qquad \text{for} \qquad i = 1, ..., s, \tag{7}$$

and $t^i = t^n + c_i\Delta t$. Furthermore, s is the number of stages, and b_γ are the main weights of the applied Runge-Kutta scheme, with

$$\sum_{\gamma=1}^{s} b_\gamma = 1. \tag{8}$$

For the system of ordinary differential equations (6), the Runge-Kutta scheme takes the following form:

$$\mathbf{F}_P^i = \mathbf{F}_P(t^i, \tilde{\mathbf{u}}_P^i) \qquad \text{for} \qquad i = 1, ..., s, \tag{9}$$

where the intermediate solution $\tilde{\mathbf{u}}_P^i$ for stage i at time $t^i = t^n + c_i\Delta t$ is given by

$$\tilde{\mathbf{u}}_P^i = \mathbf{u}_P^n + \Delta t\left(\sum_{\gamma=1}^{i} a_{i\gamma}\mathbf{F}_P^\gamma - \frac{c_i}{\rho}(\nabla\tilde{p})_P^i\right), \tag{10}$$

and the final solution $\mathbf{u}_P^{n+1}$ at time $t^{n+1} = t^n + \Delta t$ by

$$\mathbf{u}_P^{n+1} = \mathbf{u}_P^n + \Delta t\left(\sum_{\gamma=1}^{s} b_\gamma\mathbf{F}_P^\gamma - \frac{1}{\rho}(\nabla p)_p^{n+1}\right). \tag{11}$$

The Runge-Kutta scheme consists of s stages in which s sources are determined. Each of these sources depends on the intermediate solution of $\tilde{\mathbf{u}}_P$ at a certain stage, e.g., $\mathbf{F}_P^1$ depends on the solution for $\mathbf{F}_P$ at quadrature point $t_n + c_1\Delta t$, $\mathbf{F}_P^2$ on the solution for $\mathbf{F}_P$ at $t_n + c_2\Delta t$, etc. Finally, these sources are gathered in a final step to determine the solution at $t^{n+1} = t^n + \Delta t$. Most commonly, Runge-Kutta schemes are used as explicit schemes. This means that the diagonal of the considered Butcher tableau is zero for all i. In Eqs. 10 and 11, the $(\nabla p)_p$ term is treated separately from the $\mathbf{F}_P^\gamma$ term, since it acts as a Lagrangian multiplier (Hairer et al. (2008)). That is, the $(\nabla p)_p$ term has no contribution to the time dependence. It can therefore be treated separately from the $\mathbf{F}_P^\gamma$ term, similar as done in the work of e.g. Sanderse and Koren (2012).

When a DIRK scheme is used, the solution of Eq. 6 at each stage i can be written as

$$\frac{\tilde{\mathbf{u}}_P^i - \mathbf{u}_P^n}{a_{ii}\Delta t} = \mathbf{F}_P^i + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_P^\gamma - \frac{c_i}{\rho a_{ii}}(\nabla\tilde{p})_P^i. \tag{12}$$

This equation will be used further in section 2.4. For explicit time integration, $a_{ii} = 0$, and Eq. 12 should be multiplied by a_{ii}, such that the first term in the RHS of Eq. 12 vanishes. The explicit and implicit temporal schemes used in the present study are indicted in Table 1. For both explicit and implicit schemes, we selected one first order scheme and two well known schemes of higher order.

Table 1: Explicit and implicit time integration schemes tested in this study.

Scheme	**Temporal order**	**Number of stages**	**Ref**
Explicit			
Forward Euler	$O(\Delta t^1)$		Ferziger and Perić (1997)
RK3	$O(\Delta t^3)$	3	Hairer et al. (2008)
RK4	$O(\Delta t^4)$	4	Ferziger and Perić (1997)
Implicit			
Backward Euler	$O(\Delta t^1)$	1	Ferziger and Perić (1997)
Implicit midpoint	$O(\Delta t^2)$	2	Ferziger and Perić (1997)
DIRK2	$O(\Delta t^2)$	2	Ascher et al. (1997)

In DNS and LES, the diffusive terms are frequently treated implicitly, whereas the convective terms are treated explicitly. The main reason for implicit treatment of the diffusive terms is the severe time step restriction originating from these terms near solid surfaces where the grid is substantially refined, when treated explicitly. Since the velocity is generally very small close to the wall, similar time step restrictions are not introduced by the convective terms. In such a situation, so-called implicit-explicit (IMEX) temporal discretisation schemes can be applied. We use diagonally implicit schemes for the implicit part of the IMEX-schemes. More specifically, we use IMEX schemes constructed by Ascher et al. (1997), see Table 2. The naming of the IMEX schemes consists of the three numbers (s, σ, p), where s is the number of implicit scheme stages, σ is the number of explicit scheme stages, and p is the combined order of the scheme.

Table 2: IMEX time integration schemes tested in this study.

Scheme	**Number of implicit stages**	**Number of explicit stages**	**Temporal Order**	**Ref**
Ascher121	1	2	$O(\Delta t^1)$	Ascher et al. (1997)
Ascher122	1	2	$O(\Delta t^2)$	Ascher et al. (1997)
Ascher232	2	3	$O(\Delta t^2)$	Ascher et al. (1997)

2.4. *Spatial discretisation*

The PISO method consists of an implicit momentum predictor step and subsequent pressure corrector steps. The spatial discretisation of these main steps will be explained now.

For a detailed description of the finite-volume discretisation as applied in OpenFOAM, the reader is referred to Jasak (1996) and de Villiers (2006). In this paper, the main lines are summarized. Within OpenFOAM, the equations are solved in a fixed Cartesian coordinate system. The control volumes in the mesh can be of a general polyhedral shape, with a variable number of neighbours, thereby creating an arbitrary unstructured mesh. All faces of the control volumes are flat, and the point P is located at the centroid of control volume V_p. This implies:

$$\int_{V_p} (\mathbf{x} - \mathbf{x}_P)\, dV = \mathbf{0}. \tag{13}$$

Similarly, the centroid $\mathbf{x}_f$ of a face S_f is defined by:

$$\int_{S_f} (\mathbf{x} - \mathbf{x}_f)\, dS = \mathbf{0}. \tag{14}$$

As an example, the integral form of the momentum equation for an arbitrary control volume V_P using forward Euler time integration reads

$$\int_{V_P} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}\, dV = \int_{V_P} -\nabla \cdot (\mathbf{u}\,\mathbf{u})^n\, dV + \int_{V_P} \nabla \cdot (\nu \nabla \mathbf{u}^n)\, dV - \int_{V_P} \frac{1}{\rho} \nabla p^n\, dV, \tag{15}$$

where the convective, diffusive, and pressure terms in this equation are taken at the old time level n. The spatial discretisation for implicit or mixed implicit-explicit methods proceeds in an analogous way.

The numerical solution of Eq. 15 essentially consists of *interpolation* and *integration*. In relation to this, the goal is to have a second-order accurate solution in space. In order to obtain this second-order accuracy, the variation of a general flow variable $\psi = \psi(\mathbf{x}, t)$ has to be linear in space, that is, it is assumed that

$$\psi(\mathbf{x}) = \psi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla\psi)_P, \tag{16}$$

where $\psi_P = \psi(\mathbf{x}_P)$. Application of Gauss divergence theorem together with Eq. 16 in the *spatial integration* of Eq. 15 yields the following discretised form

$$\frac{\mathbf{u}_P^{n+1} - \mathbf{u}_P^n}{\Delta t} V_P = -\sum_f \mathbf{u}_f^n \phi_f^n + \sum_f (\nu \nabla \mathbf{u}^n)_f \cdot \mathbf{n}_f S_f - \frac{1}{\rho}(\nabla p^n)_p V_P. \tag{17}$$

where the convective face flux ϕ_f^n is introduced as $\phi_f^n = \mathbf{u}_f^n \cdot \mathbf{n}_f S_f$, with unit outward normal vector $\mathbf{n}_f$ and area S_f of cell face f.

For the convection term in Eq. 17, the values of $\mathbf{u}_f^n$ are *interpolated* to the cell faces using a selected interpolation scheme, that is,

$$\mathbf{u}_f^n = f_x \mathbf{u}_P^n + (1 - f_x)\mathbf{u}_N^n, \tag{18}$$

where f_x is the interpolation weight. In this paper, we use linear interpolation, which is commonly known as central differencing, and which is second order accurate. In section 4, we use the midpoint scheme which has an interpolation weight always equal to $1/2$. Here, it is assumed that the line connecting P with the centroid N of the neighbouring cell intersects the corresponding face in its center f, which is indeed the case for the orthogonal meshes used in this study. Otherwise, a skewness correction can be used (Ferziger, 1995).

For the diffusion term in Eq. 17, the gradient of the velocity components u_i^n normal to the cell face is needed. When the mesh is orthogonal, i.e., when the vector $\mathbf{d}_f$ from P to the centroid N of the neighbouring cell and the face normal vector $\mathbf{n}_f$ are parallel, the derivative of $\mathbf{u}_f$ normal to the cell face can be discretised as

$$(\nabla \mathbf{u})_f \cdot \mathbf{n}_f S_f = \frac{\mathbf{u}_N - \mathbf{u}_P}{|\mathbf{d}_f|}|S_f|. \tag{19}$$

This approach provides a compact computational stencil. It is noted that, for non-orthogonal meshes, the considered diffusion flux term consists of an orthogonal and non-orthogonal contribution, see Jasak (1996). The discretisation of the pressure gradient term in the momentum equation will be discussed momentarily.

We now introduce velocity $\mathbf{u}_P^{ijk}$. This is the cell-centered velocity solution of Runge-Kutta stage i for outer iteration j and inner PISO iteration k. Below, similar as in the original paper of Issa (1985), we will use stars ($*$) to indicated the PISO iteration level k. For both explicit, implicit, or IMEX methods, the application of the above discussed spatial discretisation of the momentum equation in the *predictor step for Runge-Kutta stage i and outer iteration j* can be written in the form of a linear algebraic equation for cell P, that is,

$$\frac{d\mathbf{u}_P^{ij*}}{dt} = a_P\mathbf{u}_P^{ij*} + \sum_{N=nb(P)} a_N\mathbf{u}_N^{ij*} + \mathbf{r}_P^n - \frac{1}{\rho}(\nabla p)_P^n =$$

$$\mathbf{F}_P(\mathbf{u}_P^{ij*}, t) - \frac{1}{\rho}(\nabla p)_P^n, \tag{20}$$

where the pressure gradient is not yet discretised, and Eq. 17 has been divided by the cell volume V_P. The coefficients a_P and a_N are the diagonal and off-diagonal coefficients of the discretisation matrix, and N represents the centers of the neighboring cells. Both coefficients consist of contributions from the implicit discretisation of the convective and diffusive transport terms. The vector $\mathbf{r}_P^n$ consists of parts of the convective and diffusive terms which are treated explicitly, as well as a contribution from the boundary conditions. When an implicit Runge-Kutta time integration method is applied to this semi-discretized equation, Eq. 20 can be written as follows for each stage i and outer iteration j

$$\frac{\mathbf{u}_P^{ij*} - \mathbf{u}_P^n}{a_{ii}\Delta t} = a_P\mathbf{u}_P^{ij*} + \sum_{N=nb(P)} a_N\mathbf{u}_N^{ij*} + \mathbf{r}_P^n - \frac{c_i}{a_{ii}\rho}(\nabla p)_P^n + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_P^{\gamma}, \tag{21}$$

where the tilde above the intermediate solutions $\tilde{\mathbf{u}}_P^{ij*}$ for stage i has been dropped. For explicit time integration, a_{ii} is zero, and Eq. 21 should be multiplied by a_{ii}. The following equation for the cell-center velocity can be obtained from Eq. 21

$$\tilde{a}_P\mathbf{u}_P^{ij*} = \frac{1}{a_{ii}\Delta t}\mathbf{u}_P^n + \sum_{N=nb(P)} a_N\mathbf{u}_N^{ij*} + \mathbf{r}_P^n - \frac{c_i}{a_{ii}\rho}(\nabla p)_P^n + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_P^{\gamma}, \tag{22}$$

where

$$\tilde{a}_P = \frac{1}{a_{ii}\Delta t} - a_P. \tag{23}$$

Next, the velocity at the cell center in the *first corrector step for Runge-Kutta stage i and* *outer iteration j* can be formulated as

$$\mathbf{u}_P^{ij**} = \frac{1}{\tilde{a}_P}\left(\frac{1}{a_{ii}\Delta t}\mathbf{u}_P^n + \mathbf{H}_P(\mathbf{u}_N^{ij*}) - \frac{c_i}{a_{ii}\rho}(\nabla p)_P^{ij*} + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_P^{\gamma}\right), \tag{24}$$

where

$$\mathbf{H}_P(\mathbf{u}_N^{ij*}) = \sum_{N=nb(P)} a_N \mathbf{u}_N^{ij*} + \mathbf{r}_P^n. \tag{25}$$

Equivalent to Eq. 24 for the cell center velocities, the following equation can be formulated for the cell face velocities

$$\mathbf{u}_f^{ij**} = \frac{1}{\tilde{a}_f}\left(\frac{1}{a_{ii}\Delta t}\mathbf{u}_f^n + \mathbf{H}_f^{ij*} - \frac{c_i}{a_{ii}\rho}(\nabla p)_f^{ij*} + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_f^{\gamma}\right). \tag{26}$$

The approaches which we use for the computation of the individual terms in Eq. 26 are explained in the subsequent section. In order to compute the corrected velocities $\mathbf{u}_f^{ij**}$ and $\mathbf{u}_P^{ij**}$, the Poisson equation for the pressure p^{ij*} in the corrector step should be solved first. This equation is obtained by combining the discretised momentum equation and the continuity equation. The continuity equation for cell P in the corrector step equals

$$\int_{V_P} \nabla \cdot \mathbf{u}^{ij**}\, dV = \sum_f \mathbf{u}_f^{ij**} \cdot \mathbf{n}_f S_f = 0. \tag{27}$$

Substitution of the interface velocity $\mathbf{u}_f^{ij**}$ in Eq. 27 yields the following Poisson equation for the pressure p^{ij*}

$$\sum_f \frac{c_i}{a_{ii}\rho}(\nabla p)_f^{ij*} \cdot \mathbf{n}_f S_f = \sum_f \overline{\phi}_f^{ij*}, \tag{28}$$

where the incomplete flux $\overline{\phi}_f^{ij*}$ is defined as

$$\overline{\phi}_f^{ij*} = \left(\frac{1}{a_{ii}\Delta t}\mathbf{u}_f^n + \mathbf{H}_f^{ij*} + \frac{1}{a_{ii}}\sum_{\gamma=1}^{i-1} a_{i\gamma}\mathbf{F}_f^{\gamma}\right) \cdot \mathbf{n}_f S_f, \tag{29}$$

and where the flux of the normal pressure gradient $(\nabla p)_f^{ij*} \cdot \mathbf{n}_f S_f$ in Eq. 28 can be computed using a discretisation similar to Eq. 19. As explained in the next subsection, the incomplete flux $\overline{\phi}_f^{ij*}$ may have several forms. Once the corrected pressure p^{ij*} is obtained, the divergence free cell face velocity $\mathbf{u}_f^{ij**}$ can be computed using Eq. 26, whereas the cell center velocity $\mathbf{u}_P^{ij**}$ can be obtained from Eq. 24. For an explicit RK scheme, the corrected velocities $\mathbf{u}_f^{ij**}$ and $\mathbf{u}_P^{ij**}$, and the pressure p^{ij*} are the final velocity and pressure fields for stage i. For implicit RK schemes, the same approach can be repeated for additional corrector steps (or: inner iterations) within the PISO procedure. That is, the incomplete flux is re-evaluated for the next inner PISO iteration level, which provides a new pressure Poisson equation that can be used in order to obtain the new corrected pressure and velocity field. This process is repeated until an inner criterion is satisfied. Once this has happened, the inner PISO iteration process is finalized by setting the convective face flux ϕ^{ij}, which is used in the momentum flux term, for the next outer iteration level. In our RKFoam solver, this can be done either using

$$\phi_f^{i(j+1)} = \mathbf{u}_f^{ijm} \cdot \mathbf{n}_f S_f, \tag{30}$$

which we call u_f-based, or

$$\phi_f^{i(j+1)} = I_f\left(\mathbf{u}_P^{ijm}\right) \cdot \mathbf{n}_f S_f, \tag{31}$$

which we call u_P-based. The inner PISO iteration process can now be repeated for a new outer iteration in order to find a pressure and velocity which satisfy the momentum equation updated with the flux $\phi_f^{i(j+1)}$. After completing ℓ outer iterations, the final solutions $\mathbf{u}_P^{i\ell m}, \mathbf{u}_f^{i\ell m}$ and $p_P^{i\ell m}$ are accepted as the Runge-Kutta stage i solutions. At the end of each stage i, the cell center Runge-Kutta source term $\mathbf{F}_P^i$ and the cell face Runge-Kutta source term $\mathbf{F}_f^i$ can be determined. These source terms are used in subsequent Runge-Kutta stages. The cell center Runge-Kutta source term $\mathbf{F}_P^i = \mathbf{F}_P(u_P^{ilm})$ can be evaluated from the definition provided in Eq. 20. One approach to compute the cell face Runge-Kutta source terms $\mathbf{F}_f^i$ consists of linear interpolation of the corresponding cell center values $\mathbf{F}_P^i$. In Kazemi-Kamyab (2013), it is demonstrated that the temporal order of the applied DIRK schemes (3^{rd} - 5^{th}) reduces to first order when this approach is used. In contrast, Kazemi-Kamyab et al. (2015) and Kazemi-Kamyab (2013) demonstrate that the temporal order of these schemes is preserved when the following approach is used to compute the source terms $\mathbf{F}_f^i$: at the end of each stage i, the face velocity $\mathbf{u}_f^{ilm}$ is known from Eq. 26.

Subsequently, $\mathbf{F}_f^i$ can be calculated using Eq. 12, that is,

$$\mathbf{F}_f^i = \frac{\mathbf{u}_f^{ilm} - \mathbf{u}_f^n}{a_{ii}\Delta t} - \frac{1}{a_{ii}} \sum_{\gamma=1}^{i-1} a_{i\gamma} \mathbf{F}_f^\gamma + \frac{c_i}{\rho a_{ii}} (\nabla p^{ilm})_f \tag{32}$$

Once the Runge-Kutta source terms $\mathbf{F}_P^i$ and $\mathbf{F}_f^i$ are computed, we can continue to the next stage i. When the final stage s is completed, the solutions $\mathbf{u}_P^{s\ell m}, \mathbf{u}_f^{s\ell m}, p_P^{s\ell m}$ and $\phi^{s(\ell+1)}$ are accepted as, respectively, the solutions $\mathbf{u}_P^{n+1}, \mathbf{u}_f^{n+1}, p_P^{n+1}$ and ϕ_f^{n+1} at the new time level $n+1$. This completes the description of the RKFoam iterative algorithm.

2.5. *Treatment of the cell face pressure gradient*

In the computation of the cell face velocity $\mathbf{u}_f^{ij**}$, Eq. 26, and the Poisson equation for the pressure p^{ij*}, Eq. 28, the cell face pressure gradient $(\nabla p)_f^{ij*}$ is needed. It seems straightforward to obtain the pressure gradient at the cell faces using interpolation of the cell center pressure gradients, that is, $(\nabla p)_f^{ij*} = I_f\left((\nabla p)_P^{ij*}\right)$. However, the problem with this approach is that it leads to the well known checkerboard oscillations, because the pressure differences are determined between every second node, as explained in for example Versteeg and Malalasekera (2007). Furthermore, interpolation of the cell center pressure gradients results in a larger computational stencil than taking the pressure gradient directly at the cell faces. Therefore, in OpenFOAM, the pressure gradient $(\nabla p)_f^{ij*}$ is directly taken at the cell face instead. This approach is used in all methods which we consider in the present study. As explained in Versteeg and Malalasekera (2007), the calculation of the pressure gradient directly at the cell faces corresponds to the addition of a third-order pressure gradient term to the cell face velocities. By taking the differences between these cell face velocities in the continuity equation, the considered term results in a fourth-order dissipation of pressure in the Poisson equation.

2.6. *Methods for the cell face velocity and incomplete flux computation*

Except for the cell face pressure gradient term $(\nabla p)_f^{ij*}$, the terms in the expression for the cell face velocity $\mathbf{u}_f^{ij**}$ in Eq. 26 are the same as the terms in the formulation of the incomplete flux $\overline{\phi}_f^{ij*}$ in Eq. 29. The computation of the cell face pressure gradient term has already been discussed in the previous section. This section presents the methods which we use for the computation of the remaining terms in the computation of the cell face velocity $\mathbf{u}_f^{ij**}$, which correspond to the individual terms in the incomplete flux $\overline{\phi}_f^{ij*}$.

2.6.1. Standard interpolation method

In the standard method, the individual terms in Eq. 26 are computed using linear interpolation of the corresponding cell center terms, that is,

$$\frac{1}{\tilde{a}_f} = I_f(\frac{1}{\tilde{a}_P}), \quad \mathbf{u}_f^n = I_f(\mathbf{u}_P^n), \quad \mathbf{H}_f^{ij*} = I_f\left(\mathbf{H}_P(\mathbf{u}_N^{ij*})\right), \quad \mathbf{F}_f^{\gamma} = I_f(\mathbf{F}_P^{\gamma}). \tag{33}$$

This interpolation method, together with the approach of taking the cell face pressure gradient directly at the cell faces, corresponds to the original interpolation method of Rhie and Chow (1983).

2.6.2. Consistent interpolation method

A temporally consistent Rhie-Chow method which preserves the order of accuracy of the applied higher order temporal discretisation schemes has been developed by Kazemi-Kamyab et al. (2015) and Kazemi-Kamyab (2013). They used higher order DIRK schemes in their analyses. Kazemi-Kamyab et al. (2015) and Kazemi-Kamyab (2013) used the following interpolations

$$\frac{1}{\tilde{a}_f} = I_f(\frac{1}{\tilde{a}_P}), \quad \mathbf{H}_f^{ij*} = I_f\left(\mathbf{H}_P(\mathbf{u}_N^{ij*})\right), \tag{34}$$

and the Runge-Kutta source terms $\mathbf{F}_f^{\gamma}$ are computed according to Eq. 32. For $\mathbf{u}_f^n$, no interpolation from the corresponding cell center velocities is performed, but the available face values from the previous time step n are taken directly.

2.6.3. OpenFOAM interpolation method

In the standard OpenFOAM implementation, the mass flux at face f in the pressure correction loop in the PISO algorithm is of the form $\phi_f^{ij**} = \phi_{f,1} + \phi_{f,2}$. Here, the first term is the standard mass flux $\phi_{f,1} = \mathbf{u}_f^{ij**} \cdot \mathbf{n}_f S_f$. The second term is a correction term which is specific to OpenFOAM. This term is not documented in the OpenFOAM user manual (OpenFOAM®, 2017) and does not belong to the original PISO method. However, a description of this second term can be found in Vuorinen et al. (2014). According to

Vuorinen et al. (2014), the $\phi_{f,2}$ term is computed on each cell face as

$$\phi_{f,2} = \frac{\gamma}{\Delta t}\left(I_f\left(\frac{1}{\tilde{a}_P}\right)\phi_f^n - I_f\left(\frac{1}{\tilde{a}_P}\mathbf{u}_P^n\right)\cdot\mathbf{n}_f S_f\right). \tag{35}$$

It can be seen that the flux correction is based on this flux and the velocity field at the old time level n, whereas the diagonal coefficient $\tilde{a}_P$ of the discretisation matrix is taken at the time level $t^i = t^n + c_i\Delta t$. The coefficient $\gamma \in [0,1]$ is calculated as

$$\gamma = 1 - \min\left(\frac{|\phi_f^n - \mathbf{u}_f^n \cdot \mathbf{n}_f S_f|}{|\phi_f^n| + \epsilon}, 1\right), \tag{36}$$

where ϵ is a very small number in order to avoid division by zero. The divergence of the face velocity is influenced by $\phi_{f,2}$, since the flux from the interpolated velocity is subtracted from the face flux ϕ_f^n. In Choi (1999), the extra $\phi_{f,2}$ term has been introduced in the flux computation. Vuorinen et al. (2014) have shown that the flux correction $\phi_{f,2}$ acts as a source of numerical dissipation.

2.7. Selected options within the RKFoam solver

By choosing certain combinations of the key parameters which were discussed in the introduction of section 2, special flavors of the algorithm can be uncovered which correspond to published ones. These are listed here:

Table 3: Selected options within the RKFoam solver.

Name	**Ref**	**Momentum flux**	**Linearization**	**Incomplete flux**
Standard u_f-based	Rhie and Chow (1983)	u_f-based	Linearized	Standard
Standard u_P-based	Vuorinen et al. (2014)	u_P-based	Linearized	Standard
Kazemi	Kazemi-Kamyab et al. (2015)	u_f-based	Non-linear	Consistent
OpenFOAM	–	u_f-based	Linearized	OpenFOAM
D'Alessandro	D'Alessandro et al. (2018)	u_P-based	Linearized	Consistent

3. Results for basic test cases

In this section, we first explain the different possible error sources which are present in the applied RKFoam solver. Subsequently, we present an analysis of the effect of the application of the different velocity interpolation methods and a selection of RK temporal schemes on the kinetic energy dissipation in Taylor-Green vortex flow. Next, the order of accuracy of a selection of temporal schemes is determined by execution of a self-convergence study for the Taylor-Green vortex and the lid-driven cavity. Subsequently, we verify the spatial consistency of the applied spatial discretisation. Finally, we present a performance assessment of a selection of time integration schemes based on which we select the preferred temporal scheme for further analyses in section 4.

3.1. Error sources

We make a distinction between the following possible error sources when using the numerical method described in the previous section: spatial and temporal discretisation errors, the pressure error resulting from the application of a compact stencil in the pressure Poisson equation, the splitting error, the linearization error of the convection term, and other possible iteration errors.

Concerning the spatial discretisation, the linear interpolation scheme is second order accurate. Since we are using orthogonal grids in the present study, error contributions due to mesh non-orthogonality and mesh skewness are zero. As demonstrated by Felten and Lund (2006), the applied interpolation scheme causes a kinetic energy error of $O(f_x - \frac{1}{2})\Delta h_f$, where f_x is the interpolation weight and Δh_f is the distance between the two cell centers adjacent of face f. Thus, a kinetic energy error is present when $f_x \neq \frac{1}{2}$. For linear interpolation on uniform grids, $f_x = \frac{1}{2}$, and the convective terms are both second order accurate and kinetic energy conserving. For linear interpolation on non-uniform grids, the convective terms are second order accurate but do not conserve kinetic energy. In contrast, application of midpoint interpolation on non-uniform grids, the convective terms are first order accurate but conserve kinetic energy. That is, a trade off between interpolation accuracy and kinetic energy conservation exists for non-uniform grids. For the test cases in this section, we use uniform grids with the linear interpolation scheme, that is, $f_x = \frac{1}{2}$.

When the compact stencil is used in the pressure Poisson equation, the obtained pressure field leads to mass conserving face fluxes, while the cell center fluxes are only approximately divergence free. As a continuation of the research of Morinishi et al. (1998), Felten and Lund (2006) demonstrated that a kinetic energy conservation error of $O(\Delta t \Delta h^2)$ arises from the fact the cell center velocities do not conserve mass exactly. When projection is used, Felten and Lund (2006) indicated that the $O(\Delta t)$ dependence in this pressure error can be reduced to $O(\Delta t^2)$ dependence by application of the pressure correction method proposed by van Kan (1986).

The splitting error can be neglected for the test cases considered in this section, since we have used five inner corrector steps. Concerning the calculation of the convective face fluxes, a linearization error in principle exists when using implicit time integration. We have used both one, three, and five outer iterations in the computations in this section. We observed that the effect of the number of outer iterations on the results can be practically neglected for the considered test cases. Therefore, in this section, we present the results for one outer iteration for both explicit, IMEX, and implicit time integration schemes. Other possible iteration errors (linear solvers) are excluded because of application of very strict convergence criteria. As a result, it can be concluded that the following two kinetic energy conservation error sources are present in the subsequent analysis: errors introduced by the application of 1) the temporal discretisation and, 2) the compact stencil in the pressure Poisson equation.

3.2. Taylor-Green vortex and lid-driven cavity

The two-dimensional Taylor-Green vortex is a standard test case in order to analyse conservation properties of numerical schemes for incompressible flows. Taylor-Green vortex flow consists of an array of periodic vortices which are described by the following set of equations

$$u(x, y, t) = \sin(x)\cos(y)\, e^{-2\nu t}, \tag{37}$$

$$v(x, y, t) = -\cos(x)\sin(y)\, e^{-2\nu t}, \tag{38}$$

$$p(x, y, t) = \frac{1}{4}(\cos(2x) + \cos(2y))\, e^{-4\nu t}, \tag{39}$$

on a domain of $0 \leq x \leq 2\pi,\ 0 \leq y \leq 2\pi$ and periodic boundary conditions on all four domain boundaries. The Reynolds number Re equals $1/\nu$. In the limit of inviscid flow, $\nu = 0,\ Re = \infty$, and the vortex flow is steady. That is, the total kinetic energy E_{kin} of the Taylor-Green vortex flow remains preserved, i.e., $dE_{kin}/dt = 0$. For viscous flow, $\nu > 0$, and E_{kin} will decay with time, i.e., $dE_{kin}/dt < 0$. The decay rate follows from the analytical solution.

The computations of the Taylor-Green vortex have been executed on a 64 × 64 uniform orthogonal grid. Also mesh resolutions of 32 × 32 and 128 × 128 cells have been used. These meshes provided the same conclusions as the 64 × 64 mesh. The simulation time equals three integral time scales τ, with $\tau = 2\pi$. The timestep Δt has been chosen such that the maximum CFL-number is around 0.5, which is a typical value used in for example LES. The computational details are summarized in Table 4. In the self-convergence study which we performed in order to determine the temporal and spatial consistency, the number of cells, time step, and simulation time are different from the values in Table 4.

Table 4: Computational details for Taylor-Green vortex flow.

Reynolds number Re = $1/\nu$	∞ and 100
Number of cells (x,y)	64×64
Domain(x,y)	$2\pi, 2\pi$
End time t_{end}	$6\pi \approx 18.85$ s
Time step Δt	$0.015\pi \approx 0.05s$
Max CFL	0.5
Spatial scheme convection	Linear (central)
Temporal scheme	Various schemes
Inner iterations	5
Outer iterations	1, 3 and 5
Initial field $t = 0$	Eqs. 37, 38, 39

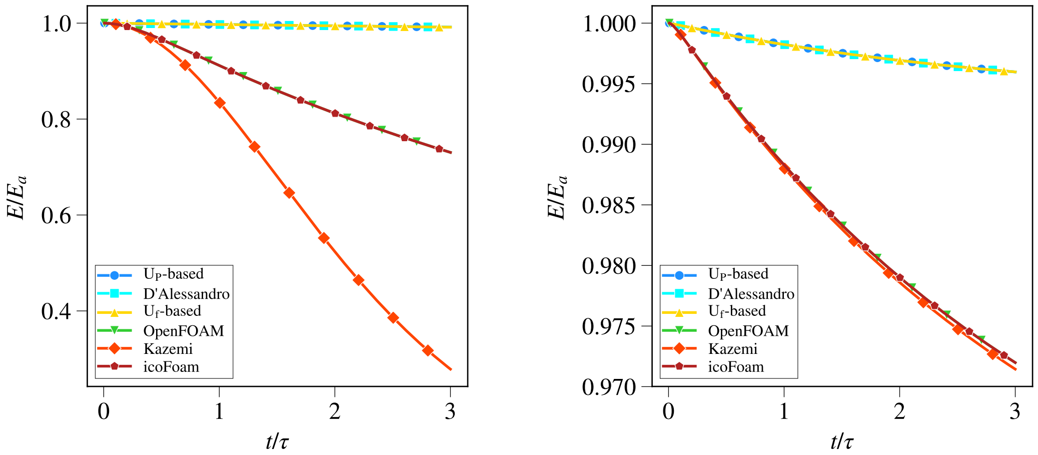
The lid-driven cavity (LDC) also forms a widely used test case for assessment of numerical schemes. The two-dimensional LDC consists of a square cavity containing an incompressible fluid with viscosity ν. The four walls of the cavity have a length L. The upper wall is moving from left to right with a velocity U_{lid}, thereby forming a vortex inside the cavity. The other three walls are static walls. The initial field is at rest. We use a 64×64 and a 256×256 uniform orthogonal grid. The applied boundary conditions at the walls are a zero-gradient condition for pressure and a no-slip condition for the velocity at the walls. The Reynolds number in the lid-driven cavity equals Re = LU_{lid}/ν. In our analysis, the domain length $L = 1$, the lid velocity $U_{lid} = 1$, and the kinematic viscosity ν equals 0.01 and 0.001 respectively. This leads to Reynolds numbers of respectively 100 and 1000. In Table 5, the computational details are summarized. The time step Δt has been incrementally reduced to study the temporal convergence.

Table 5: Computational details for the lid-driven cavity temporal consistency study.

Reynolds number Re = LU_{lid}/ν	100 and 1000
Number of cells (x,y)	64×64 and 256×256
Domain (x,y)	1,1
End time t_{end}	0.1
Time step Δt	Various
Max CFL	0.5
Spatial scheme convection	Linear (central)
Temporal scheme	Various
Inner iterations	5
Outer iterations	1, 3 and 5
Initial field $t = 0$	At rest

3.3. Kinetic energy dissipation

For the five different methods summarized in Table 3, Fig. 1a presents the ratio of the numerical and analytical total kinetic energy (E/E_a) versus time for the inviscid Taylor-Green vortex using the backward Euler time integration scheme. In addition, the corresponding results obtained using the standard icoFoam solver are presented. As expected, the OpenFOAM approach which we have implemented in our RKFoam solver provides the same results as the standard icoFoam solver. The results clearly show that all methods suffer from kinetic energy dissipation, that is, $E/E_a < 1$ due to numerical dissipation. The Kazemi approach appears to be the most dissipative. This method produces a solution which has lost more than 70% of its total kinetic energy after 3 time periods τ. The OpenFOAM method is also very dissipative. Its solution has lost approximately 25% of the total kinetic energy after 3τ. The u_f-based, u_P-based, and D'Alessandro methods are the least dissipative methods and provide practically the same results. Approximately 1% kinetic energy has dissipated numerically after three time periods using these methods. The large difference in the kinetic energy dissipation rate between the u_f-based method and the OpenFOAM method originates from the additional flux term ϕ_2 in the OpenFOAM method. Our results suggest that this term yields a strong dissipative contribution for this case. This observation is consistent with Vuorinen et al. (2014).



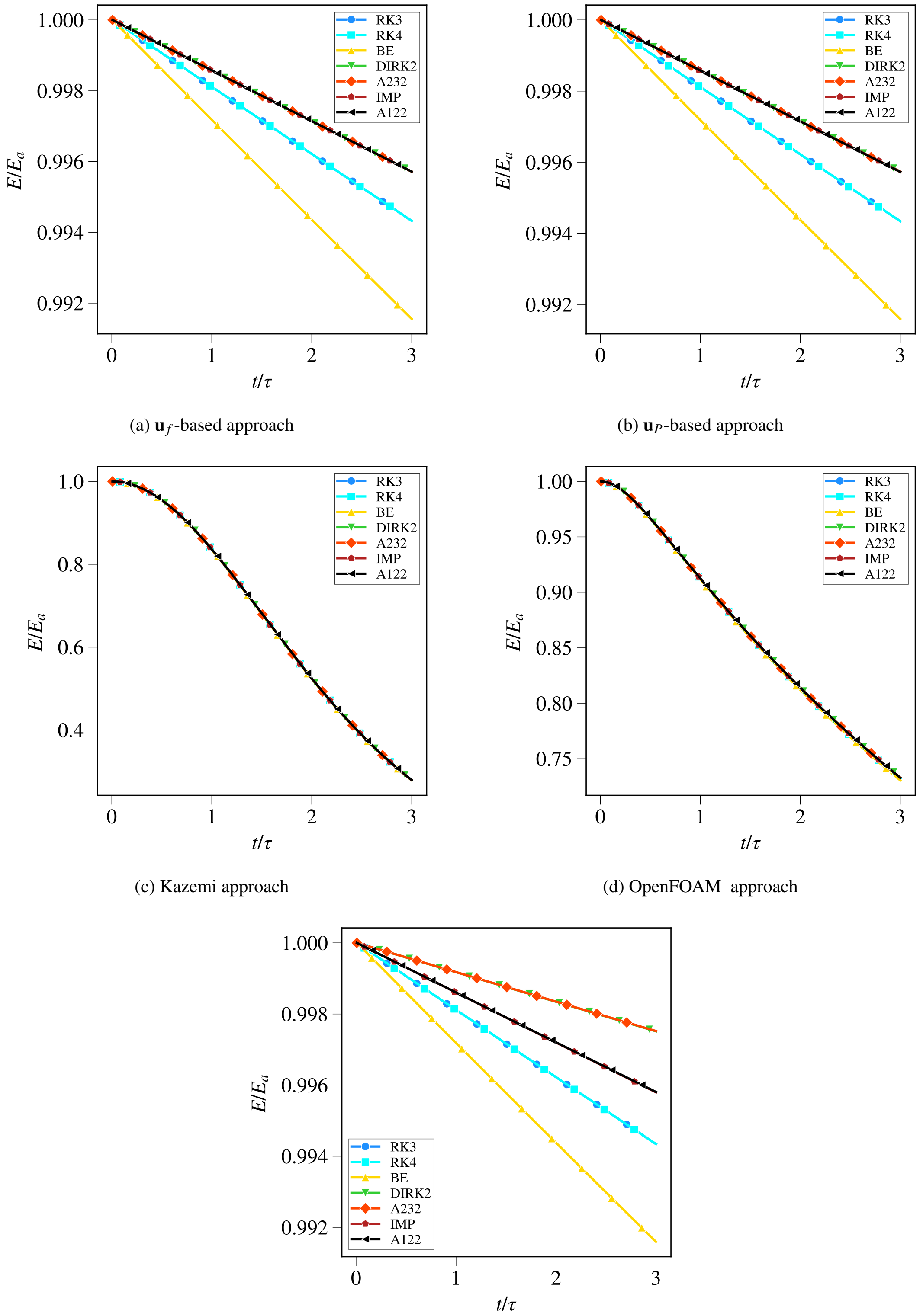
(a) Inviscid Taylor-Green vortex

(b) Taylor-Green vortex at Re = 100

Figure 1: Ratio of the numerical and analytical total kinetic energy versus time for the Taylor-Green vortex.

We have also performed computations for the viscous Taylor-Green vortex at Re = 100, see Fig. 1b, where E_a now represents the analytical solution with viscosity at Re = 100. It can be observed that the Kazemi method is again the most dissipative, closely followed by the OpenFOAM method. The u_f-based, u_P-based, and D'Alessandro methods are again the least dissipative methods. In that sense, the addition of the diffusion term does not result in different conclusions.

Fig. 8 presents the ratio of the numerical and analytical total kinetic energy versus time for the inviscid Taylor-Green vortex using a selection of explicit, implicit, and IMEX time integration schemes for the methods of Table 3. For the u_f-based method, approximately 0.8% of the kinetic energy has dissipated numerically after three time periods when the backward Euler scheme is used, whereas almost 0.6% has dissipated when the explicit RK3 and RK4 schemes are used (Fig. 2a). About 0.4% has dissipated after three time periods when the Ascher122 and Ascher232 IMEX scheme and the implicit midpoint and DIRK2 schemes are used. Practically identical results are obtained for the u_P-based approach (Fig. 2b). Due to the dissipative nature of the Kazemi and OpenFOAM approaches, no differences can be observed between the results obtained with the different applied temporal schemes (Figs. 2c and 2d). This indicates that the temporal discretisation errors play a negligible role in the Kazemi and OpenFOAM approaches for this case. For the D'Alessandro approach, the backward Euler scheme, RK3, RK4, implicit midpoint, and Ascher122 schemes show the same behaviour as for the u_f-based and u_P-based methods (see Figs. 2e, 2a, and 2b). In contrast, the DIRK2 and Ascher232 schemes dissipate less kinetic energy when using the D'Alessandro method. The Ascher122 method provides the same result as the implicit midpoint scheme, which is consistent with the fact that the Ascher122 scheme is based on a combined explicit and implicit midpoint treatment. A similar reasoning holds for the Ascher232 and DIRK2 schemes. Results for the viscous Taylor-Green vortex at Re = 100 are not shown, since they are not fundamentally different.



(a) $\mathbf{u}_f$-based approach

(b) $\mathbf{u}_P$-based approach

(c) Kazemi approach

(d) OpenFOAM approach

(e) D'Alessandro approach

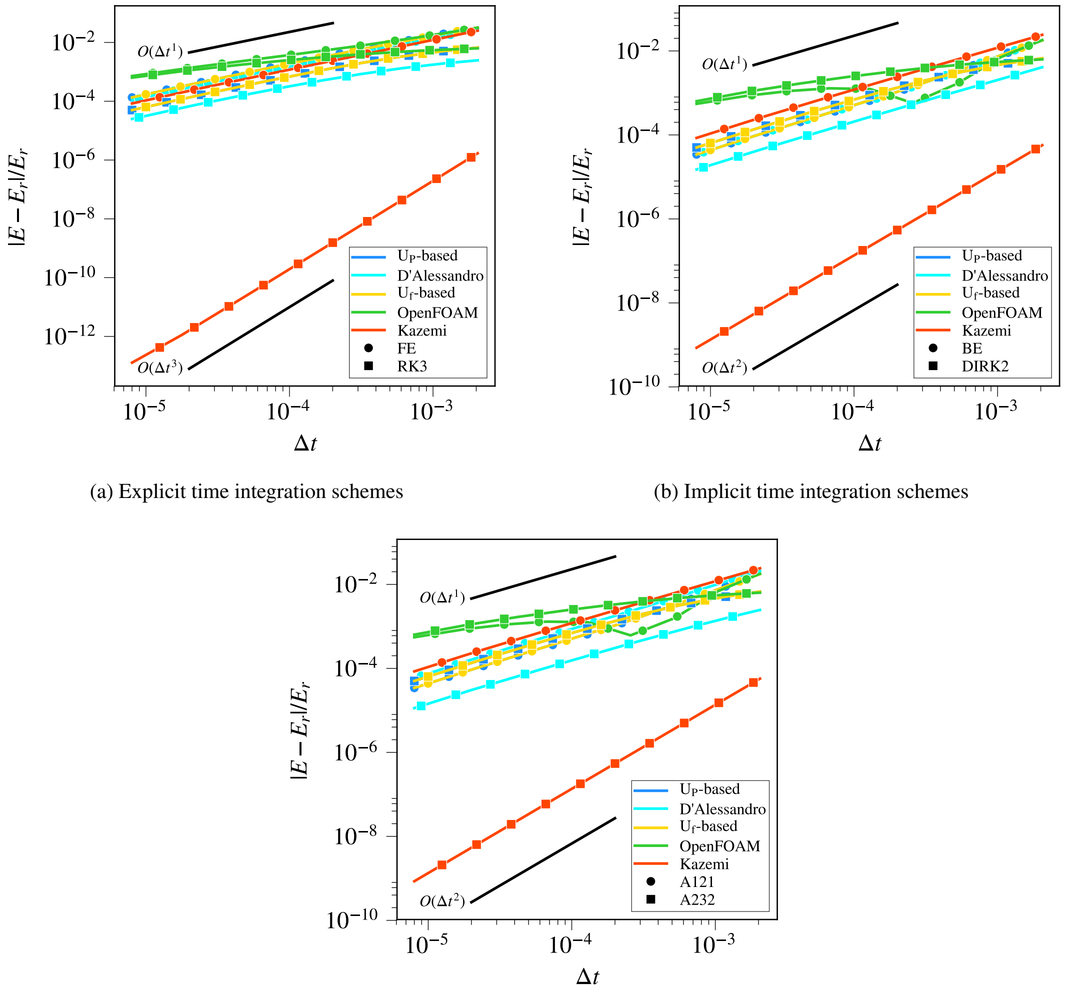
Figure 2: Ratio of the numerical and analytical total kinetic energy versus time for the inviscid Taylor-Green vortex for a selection of time integration schemes.

3.4. Temporal consistency

In order to determine the order of accuracy of a selection of temporal schemes when using the methods in Table 3, we have performed self-convergence analyses for both the Taylor-Green vortex and the lid-driven cavity. In these self-convergence analyses, we have determined the temporally exact solution for the selected mesh resolution as the numerical solution at a very small time step. Subsequently, we have computed a sequence of numerical solutions at larger time steps for the same mesh resolution. Since all solutions are obtained using the same mesh resolution, the temporal errors can be calculated as the difference between the temporally exact solution and the corresponding solutions at larger time steps, so that the spatial error is effectively eliminated. Since the results for the considered Taylor-Green vortex and lid-driven cavity test cases provided the same conclusions, we provide results only for the lid-driven cavity case at Re = 100 for a 64x64 grid resolution, see Fig. 3.

As can be seen in Fig. 3, the Kazemi method reproduces the theoretical order of accuracy of the applied time integration schemes. In contrast, all other applied approaches suffer from a reduction of the temporal order to approximately one. As explained in section 3.1, the following two error sources exist in the kinetic energy conservation: 1) the temporal discretisation error of $O(\Delta t^n)$, where n is the order of the applied temporal discretisation scheme, and 2) the pressure error of $O(\Delta t \Delta h^2)$ resulting from the application of a compact stencil in the pressure Poisson equation. As will be demonstrated shortly, the pressure error dominates over the temporal discretisation error on the applied grid resolution of 64x64. As a result, for all velocity interpolation methods except Kazemi, $O(\Delta t)$ convergence is observed for all applied temporal schemes. For the OpenFOAM velocity interpolation method, the temporal convergence order is even less than one for the applied second and third order temporal schemes. We speculate that this is caused by the additional flux term $\phi_{f,2}$ in the OpenFOAM interpolation method. In the Kazemi method, the theoretical order of the applied temporal schemes is imposed by the consistent treatment of the individual terms in the incomplete flux, as described in section 2.6.2. However, we have already observed that the Kazemi method appears to be the most dissipative method.

In the temporal convergence study using the 64x64 grid, the pressure error dominates over the temporal discretisation error. When a sufficiently refined grid is used, the pressure error vanishes. In this limit, the theoretical order of accuracy of the applied temporal schemes should be retrieved. Figure 4 shows the results of a temporal convergence study using a 256x256 grid resolution. As can be seen in this figure, the formal order of accuracy is obtained for the applied second and third order temporal schemes for all velocity interpolation methods for relatively large time steps. When the time step is decreased, the pressure error gradually starts to dominate over the temporal discretisation error. As a result, the formal order of temporal accuracy gets gradually lost. This situation happens first for the third order RK scheme, subsequently followed by the second order DIRK2 and Ascher232 schemes. Of all considered velocity interpolation methods, the OpenFOAM method looses the formal temporal order first, subsequently followed by the other velocity interpolation methods, except the Kazemi method. Of the velocity interpolations methods other than Kazemi, the method of D'Alessandro shows overall somewhat better performance. For reasons explained above, the Kazemi method maintains the formal order of temporal accuracy. Conform expectation, the formal order of accuracy of the applied first order schemes is reproduced for all applied velocity interpolation methods. We have performed this temporal convergence study also for the lid-driven cavity case at Re = 1000. We found a slight difference in the sense that, at this higher Reynolds number, the pressure error starts to dominate over the temporal discretisation error in a slightly earlier stage when the time step is refined. Since the difference in the results for the two Reynolds numbers are fairly small, we do not present results for Re = 1000.

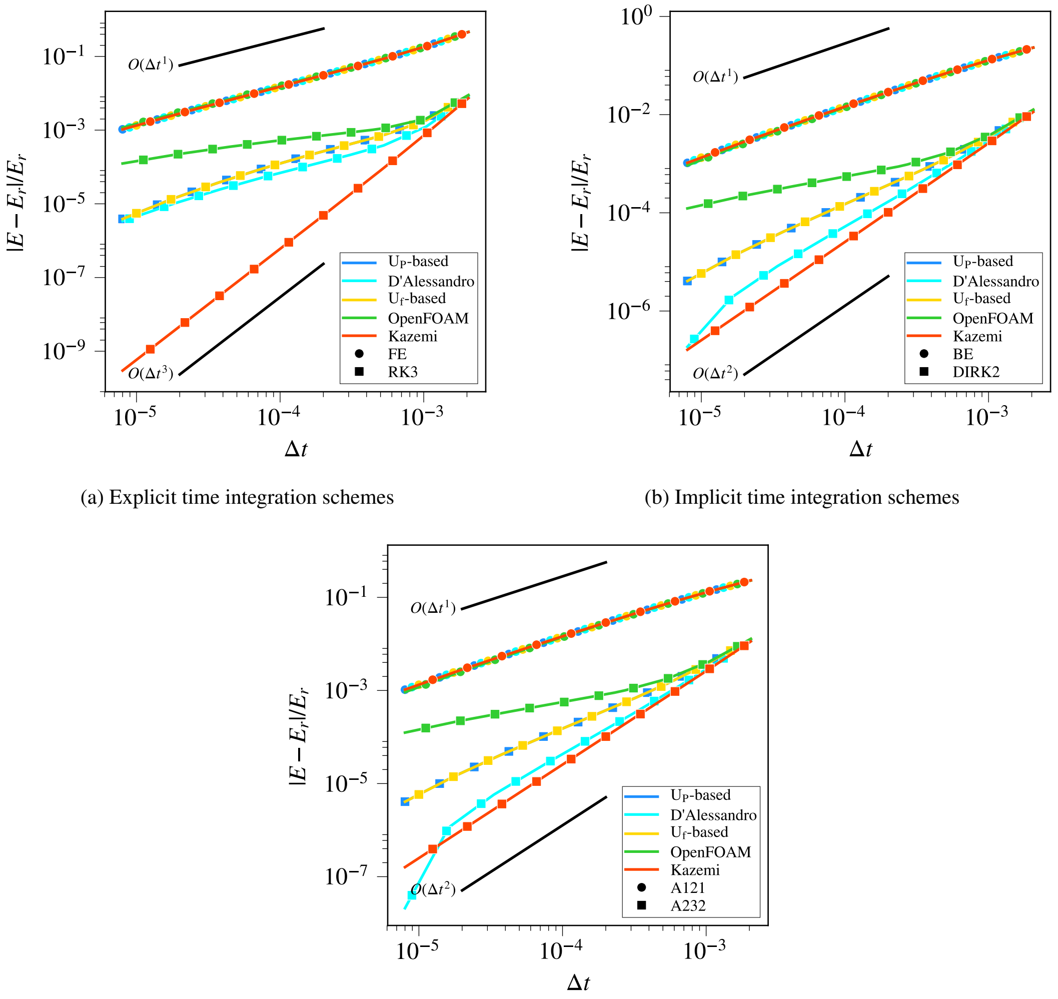


(a) Explicit time integration schemes

(b) Implicit time integration schemes

(c) IMEX time integration schemes

Figure 3: Scaled temporal error in the total kinetic energy as a function of the time step Δt for the lid-driven cavity test case at Re= 100 and 64x64 grid resolution. E_r is the temporally exact solution at very small time step.



(a) Explicit time integration schemes

(b) Implicit time integration schemes

(c) IMEX time integration schemes

Figure 4: Scaled temporal error in the total kinetic energy as a function of the time step Δt for the lid-driven cavity test case at $Re = 100$ and 256x256 grid resolution. E_r is the temporally exact solution at very small time step.

3.4.1. Spatial consistency

We have assessed the spatial consistency of the RKFoam solver by execution of a spatial self-convergence study based on the lid-driven cavity test case at Re = 1000. The spatial error is calculated as the difference between the spatially exact solution at a very fine uniform mesh resolution of 1024x1024 for the applied time step and the corresponding solutions at coarser mesh resolutions. For all computations, the time step size was the same and set in such a way that the maximum CFL number was 0.5. We have tested the u_f-based and D'Alessandro velocity interpolation approaches together with the Backward Euler and DIRK2 temporal schemes. The results are presented in Fig. 5. These results confirm the second order spatial accuracy of the solver while using the linear interpolation scheme.

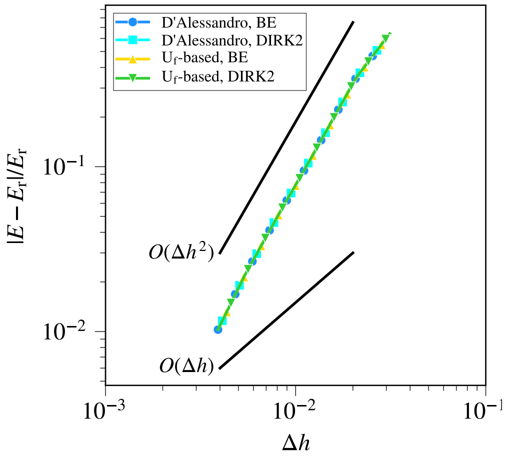


Figure 5: Relative spatial error in the total kinetic energy as a function of the mesh size for the lid-driven cavity test case at Re= 1000.

3.4.2. Performance assessment

We have assessed the computational performance of the following four time integration schemes: the implicit midpoint, DIRK2, Ascher122, and Ascher 232 schemes. For the assessment, we have used the lid-driven cavity test case at Re = 100, for which the computational settings are summarized in Table 6. The computational times are given in Table 7, whereas the corresponding average number of inner iterations per RK stage are given in Table 8.

A number of observations can be made from the results. In general, the viscous term is important for low Reynolds number flows. For such flows, the explicit treatment of the convective terms in IMEX schemes brings little benefit, which is reflected by the present results. Implicit midpoint and Ascher122 have both two stages and are second order accurate. For low Reynolds numbers, both schemes show a similar performance. For higher Reynolds numbers, Ascher122 becomes cheaper, since it requires on average less inner iterations. DIRK2 has two implicit stages, whereas Ascher232 has two implicit stages, followed by a final third explicit stage, but which also requires a pressure Poisson equation to be solved. Both schemes are second order accurate. Although Ascher232 requires less corrector steps per stage for higher Reynolds numbers, this scheme remains more expensive than the DIRK2 scheme. At low Reynolds numbers, the Ascher232 scheme is definitely more expensive. For low Reynolds numbers, the DIRK2 scheme shows the best performance since it requires the smallest number of corrector steps per stage. Overall, Ascher122 is the cheapest scheme for high Reynolds numbers. However, for the D'Alessandro velocity interpolation method, the DIRK2 scheme is less dissipative than the Ascher122 scheme, see Fig 2e. In addition, the DIRK2 scheme shows the best performance for low Reynolds numbers. Therefore, we have selected the DIRK2 scheme for subsequent analyses.

Table 6: Computational details for the lid-driven cavity performance test case.

Reynolds number Re = LU_{lid}/ν	100, 1000 and 10000
Number of cells (x,y)	256×256
Domain (x,y)	1,1
End time t_{end}	1.024 s
Time step Δt	1.024×10^{-3} s
Spatial scheme convection	Linear (central)
Temporal scheme	A122, A232, IMP and DIRK2
Inner iteration convergence criterion	10^{-5}
Outer iterations	1
Initial field $t = 0$	At rest

Table 7: Required computational times (s) for the lid-driven cavity performance test case.

Reynolds number Re= LU_{lid}/ν	100	1000	10000
Implicit midpoint	1426.5	1073.5	929.5
DIRK2	1383.2	1202.9	1061.3
Ascher122	1413.1	1044.9	867.3
Ascher232	1855.0	1441.4	1261.8

Table 8: Required average number of correctors per stage for the lid-driven cavity performance test case.

Reynolds number Re = LU_{lid}/ν	100	1000	10000
Implicit midpoint	6.50	3.01	2.99
DIRK2	5.25	3.58	3.47
Ascher122	6.50	3.01	2.50
Ascher232	5.67	3.04	2.62

3.4.3. *Effect of mesh non-uniformity, non-orthogonality, and skewness*

The unstructured collocated finite volume approach is an attractive option for modelling of flows in complex geometries. Two possible mesh topologies for the execution of LES or (q-)DNS for flows in complex geometries are: a) unstructured polyhedral meshes, and b) block-structured or unstructured hexahedral meshes. In Komen et al. (2014), it is demonstrated that skewness and non-orthogonality errors can be practically neglected when a good polyhedral mesh generator is used. As a result, the accuracy obtained on an unstructured polyhedral mesh approaches the accuracy obtained on an orthogonal hexahedral mesh closely. Therefore, with the goal of performance of LES or (q-)DNS for flow in complex geometries in mind, we also present results obtained using a distorted hexahedral grid topology which includes both non-uniformity, non-orthogonality, as well as skewness. This mesh topology is shown in Fig. 6. Consistent with the analyses presented in the previous sections, we use grid resolutions of 64×64 and 256×256. For these grids, the cell edge grading in the horizontal and vertical mid plane has been selected in such a way that the ratio of the largest and smallest cell edge length equals 4.

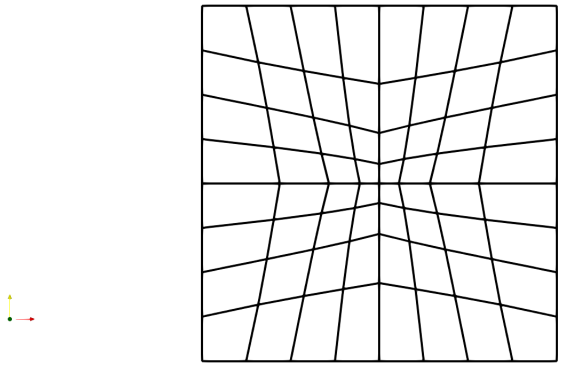
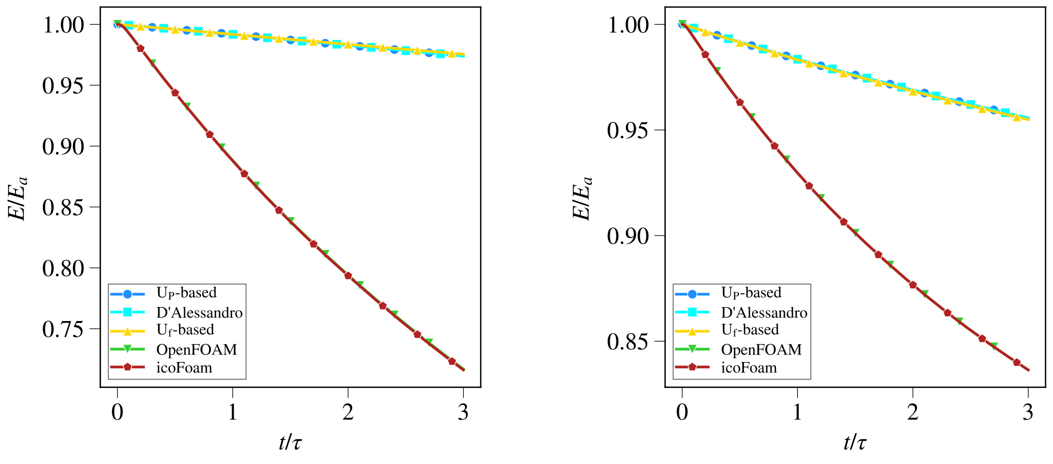


Figure 6: Mesh topology used to study the contribution from mesh non-uniformity, non-orthogonality, and skewness.

As explained in section 3.1, when uniform orthogonal grids are used, two kinetic energy conservation error sources are introduced due the application of 1) the temporal discretisation, and, 2) the compact stencil in the pressure Poisson equation. When the distorted grid topology of Fig. 6 is used, the following three additional error sources are present: a kinetic energy conservation error of $O(f_x - \frac{1}{2})\Delta h_f$, as introduced by the linear interpolation scheme, since the interpolation weight $f_x \neq \frac{1}{2}$ on non-uniform grids. Furthermore, errors are introduced due to mesh non-orthogonality and skewness. Due to those three additional error sources, less favourable results are to be expected for the distorted grid topology of Fig. 6.

For the 64 × 64 distorted hexahedral mesh and backward Euler time integration scheme, the ratio of the numerical and analytical total kinetic energy (E/E_a) versus time is presented in Fig. 7a for the inviscid Taylor-Green vortex. Since the Kazemi method showed stability problems on the distorted mesh, no results are presented for this method. When Fig. 7a is compared with Fig. 1a, it can be concluded that the applied distorted mesh topology decreases the ratio E/E_a from approximately 0.75 to 0.7 after 3 time periods for the OpenFOAM method. For the u_f-based, u_P-based, and D'Alessandro methods, the distorted mesh topology decreases this ratio E/E_a from approximately 0.99 to approximately 0.97. So, it can be concluded that the distorted mesh topology causes a substantial increase of the numerical energy dissipation for this test case.

For the viscous Taylor-Green vortex at Re = 100, comparison of Figs. 7b and Fig. 1b shows that the ratio of the numerical and analytical total kinetic energy versus time decreases from almost 0.97 to 0.84 after 3 time periods for the OpenFOAM method when the distorted mesh is used, whereas this ratio decreases from 0.996 to about 0.96 for the u_f-based, u_P-based, and D'Alessandro methods. Apparently, the consequences of grid distortion are more severe for the viscous case, which can be understood by the fact that the errors due non-orthogonality and skewness in the diffusion term now also play a role. The consequences of non-orthogonality in the diffusion term can be mitigated by application of non-orthogonality corrections, see e.g. Jasak (1996). These non-orthogonality corrections are left outside the scope of this study.

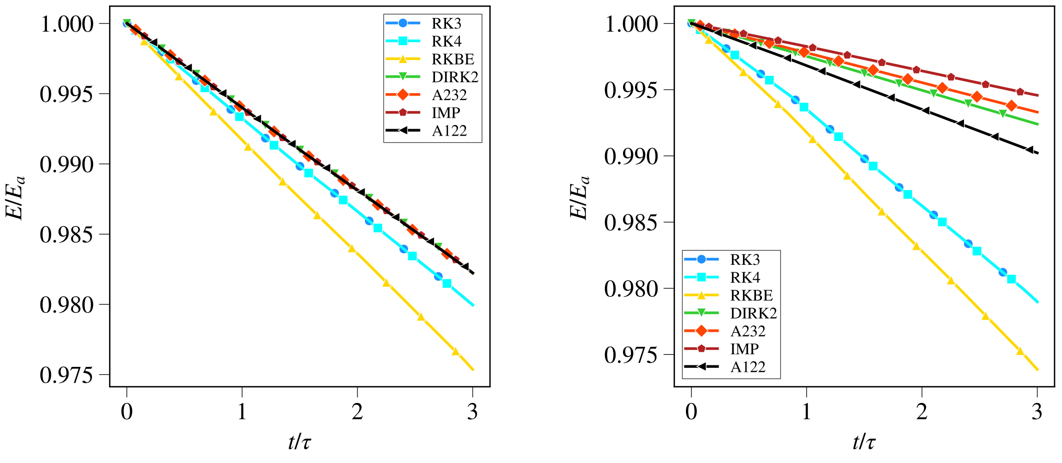


(a) Inviscid Taylor-Green vortex

(b) Taylor-Green vortex at Re = 100

Figure 7: Distorted hexahedral mesh: ratio of the numerical and analytical total kinetic energy versus time for the Taylor-Green vortex.

For the distorted mesh and $\mathbf{u}_f$-based approach, Fig. 8a presents the ratio of the numerical and analytical total kinetic energy versus time for the inviscid Taylor-Green vortex for a selection of time integration schemes. When this figure is compared with Fig. 2a, it can be observed that the applied mesh distortion increases the numerical energy dissipation by a factor of approximately 3 to 4 during the considered time frame for all considered time integration schemes. By comparison of Figs. 8b and 2e, a similar conclusion can be drawn for the D'Alessandro method.



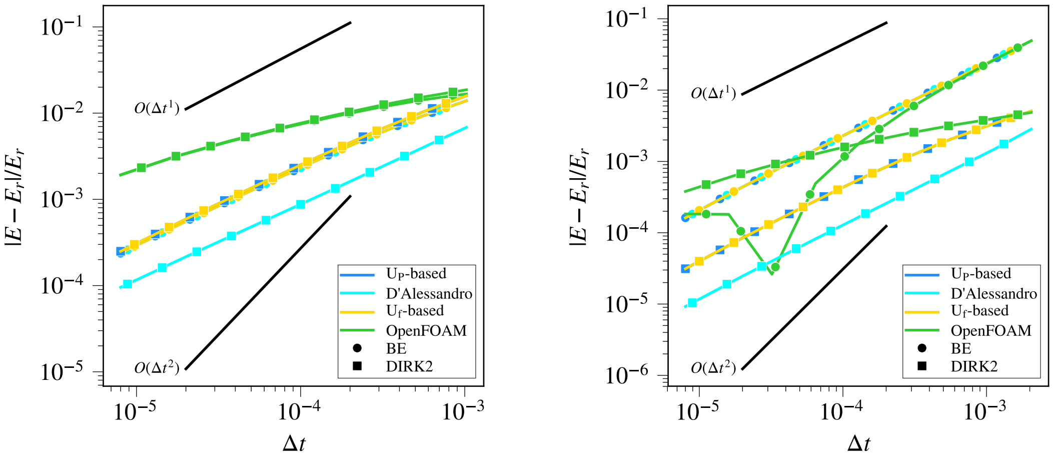
(a) $\mathbf{u}_f$-based approach

(b) D'Alessandro approach

Figure 8: Distorted hexahedral mesh: ratio of the numerical and analytical total kinetic energy versus time for the inviscid Taylor-Green vortex for a selection of time integration schemes.

Figure 9a shows the scaled temporal error in the total kinetic energy as a function of the time step Δt for the lid-driven cavity test case at Re = 100 for the distorted 64 × 64 mesh and the backward Euler and DIRK2 time integration schemes. From a comparison of this figure with Fig. 3b, it can be observed that the order of both temporal schemes equals approximately one for the u_f-based, u_P-based, and D'Alessandro methods together with the uniform orthogonal grid, whereas the temporal order reduces to somewhat less than one for the distorted mesh. Furthermore, the applied distorted mesh also causes a further reduction of the temporal order in case of the OpenFOAM method.

As explained in section 3.4, the pressure error becomes smaller than the temporal error for the uniform orthogonal 256 × 256 grid for relatively large time steps, and, as a result, the formal order of accuracy of the applied temporal schemes was obtained. For the distorted 256 × 256 grid and the u_f-based, u_P-based, and D'Alessandro methods, the temporal order of the second order DIRK2 scheme remains about one, whereas the formal order of accuracy of the first order Backward Euler scheme is obtained. The inability of the DIRK2 scheme to recover the formal temporal order of convergence even on the finer 256 × 256 grid suggests that the pressure error on this distorted hexahedral mesh topology is still dominating the temporal error. The applied temporal schemes show quite peculiar temporal convergence behaviour for the OpenFOAM method. We speculate that this is due to the additional flux term ϕ_2 in the OpenFOAM method.



(a) 64x64 grid resolution

(b) 256x256 grid resolution

Figure 9: Distorted hexahedral mesh and implicit time integrations schemes: Scaled temporal error in the total kinetic energy as a function of the time step Δt for the lid-driven cavity test case at Re= 100. E_r is the temporally exact solution at very small time step.

3.4.4. *Conclusions from the basic test cases*

From the analyses using the uniform mesh topology, we draw the following conclusions:

- one outer iteration is sufficient for the considered test cases, since the differences in the results obtained using one, three, and five outer iterations are practically negligible;
- the results of the u_f-based and u_P-based methods are practically the same;
- the results of the D'Alessandro method closely resemble the results obtained with the u_f-based and u_P-based methods. However, the DIRK2 and Ascher232 temporal schemes perform slightly better while using the D'Alessandro approach;
- the consistent Kazemi method is the only method which preserves the formal order of accuracy of the applied time integration schemes for all meshes used. However, it is the most dissipative method.
- the OpenFOAM interpolation method is also very dissipative.
- the u_f-based, u_P, and D'Alessandro methods are the least dissipative methods.

- for the lid-driven cavity test case at Re = 100, the pressure error is larger than the temporal discretisation error on a 64x64 grid resolution. Consequently, in line with the theory of Felten and Lund (2006), $O(\Delta t)$ convergence is observed for all velocity interpolation methods except the Kazemi method. At a 256x256 grid resolution, the pressure error is smaller than the temporal discretisation error for all velocity interpolation methods for relatively large time steps. As a result, the formal temporal order is obtained for this situation. When the time step is gradually refined, the pressure error gradually starts to dominate over the temporal discretisation error. As a result, the formal temporal order gradually gets lost.

In addition to the kinetic energy conservation errors due to the temporal discretisation and the compact stencil in the pressure Poisson equation, additional errors are introduced due to mesh non-uniformity, non-orthogonality, and skewness for the distorted hexahedral mesh topology of Fig. 6. From the results obtained for the presented Taylor-Green and lid-driven cavity test cases, it can be concluded that these three additional error sources increase the numerical energy dissipation substantially and further reduce the order of the applied temporal schemes for the considered velocity interpolation methods.

Based on these conclusions, we excluded the considered consistent Kazemi method from the analyses in the subsequent section. Furthermore, we excluded the u_P-based method, since the cell face velocities which are obtained using this method are not guaranteed to be divergence free. Although very dissipative, we kept the OpenFOAM interpolation method, since it is the standard method implemented in OpenFOAM.

4. Results for turbulent channel flow at $Re_\tau = 180$

First, the computational settings which we have used for this turbulent channel flow case are presented. Next, we explain the method which we have used in order to quantify the numerical dissipation rate in the computations. Subsequently, the LES and UDNS results obtained for the considered channel flow case are discussed, and finally, the corresponding q-DNS results are presented.

4.1. Computational settings

As a more complex test case, we have selected fully-developed turbulent channel flow at a frictional Reynolds number $Re_\tau = u_\tau h/\nu$ of 180, where h is half the channel height, ν the kinematic viscosity, and u_τ the frictional velocity which is defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} = \sqrt{\nu\left(\frac{\partial\overline{u}}{\partial y}\right)_{wall}}, \tag{40}$$

where τ_w is the wall shear stress and $\overline{u}$ the mean streamwise velocity. As a reference, we use the DNS data of Vreman and Kuerten (2014). They used a computational domain size of (4π, $4\pi/3$, 2), which is also used in this study. The flow in our computations is driven by a fixed pressure gradient forcing term in the streamwise momentum equation. The standard no-slip boundary condition is used at the walls, whereas periodic boundary conditions are applied in both streamwise and spanwise directions. We use the approach applied by de Villiers (2006) to create an initial field.

We use the WALE SGS model (Nicoud and Ducros, 1999) for the LES computations in this section. In addition, we have performed corresponding Under-resolved DNS (UDNS), or, *no-model* LES computations, for the same LES grid. Furthermore, for typical DNS mesh resolution, we present quasi-DNS (q-DNS) results in this section. We deliberately speak about q-DNS results instead of DNS results, since somewhat lower accuracy may be expected for the numerical method which we use in the present study. In Komen et al. (2017), three different typical LES mesh resolutions have been used for the analysis of trends in LES of wall-bounded turbulent flows using collocated grid solvers like OpenFOAM. Based on this analysis, we have selected the so-called *60-grid* mesh for the present study. Table 9 presents the cell sizes of this orthogonal mesh in wall units which are calculated using

$$\Delta_i^+ = \Delta_i \frac{u_\tau}{\nu},\ \ i = x, y, z, \tag{41}$$

where $\Delta_i,\ i = x, y, z$, represent the actual cell sizes in respectively the streamwise, spanwise, and wall normal directions. The q-DNS computations are performed using a so called *9-grid* which is also presented in Table 9.

The main computational settings which we have used for the LES, UDNS, and q-DNS

Table 9: Computational grids for the turbulent channel flow at $Re_\tau = 180$: 60-grid for LES, and the 9-grid for q-DNS. A stretching ratio of 1.05 is used in the wall normal direction in order to stretch the cells from Δy^+_{wall} to Δy^+_{bulk}.

Grids	Type	Δx^+	Δy^+_{wall}	Δy^+_{bulk}	Δz^+	N_x	N_y	N_z	N_{total}
60-Grid	LES	60	0.35	10	20	38	121	38	155.952
9-Grid	q-DNS	9	0.25	4.5	4.5	251	152	168	5.566.176

computations in this section are summarized in Table 10. We use the midpoint scheme for momentum interpolation, because it conserves kinetic energy (Felten and Lund, 2006). Since the mesh is uniform in the streamwise and spanwise directions, and a small stretching ratio of 1.05 is used in the wall-normal direction, the linear and midpoint schemes provide nearly identical results here. We use the velocity interpolation method of D'Alesandro, since it performed best for the basic test cases. In addition, we use the standard U_f-based and OpenFOAM methods, since they correspond to respectively the classical method of Rhie and Chow (1983) and the standard method in the OpenFOAM platform. We have tested the influence of the number of inner and outer iterations, and it was found that 2 inner iterations and 1 outer iteration are sufficient.

Table 10: Computational settings for the LES, UDNS, and q-DNS channel flow computations at $Re_\tau = 180$.

	LES/UDNS	**q-DNS**
Domain	$(4\pi, 4\pi/3, 2)$	$(4\pi,4\pi/3,2)$
Grid	*60-Grid*	*9-Grid*
Averaging time	80 FFT	80 FFT
Max CFL	0.4	0.4
Spatial scheme convection	midpoint	midpoint
Temporal scheme	DIRK2	DIRK2
Inner iterations	2	2
Outer iterations	1	1

4.2. Method for quantification of the numerical dissipation rate

A detailed description of the method which we use for the quantification of the numerical dissipation rate can be found in Komen et al. (2017). For convenience, the basic idea behind this method is summarized here. The applied method is based on analysis of the budget terms for respectively the mean turbulent kinetic energy per unit mass $k = \frac{1}{2}\Sigma_i \overline{u'_i u'_i}$ and the individual components $\overline{u'u'}$, $\overline{v'v'}$ and $\overline{w'w'}$ of k, where u'_i is defined by the Reynolds decomposition of the instantaneous velocity u_i in a mean and a fluctuating component, that is, $u_i = \overline{u_i} + u'_i$. The transport equations for these budget terms can be obtained from standard text books such as Durbin and Petterson-Reif (2011). Following Durbin and Petterson-Reif (2011) or Hoyas and Jimenez (2008), the transport equation for the $\overline{u'_i u'_j}$ component of the Reynolds stress tensor for an incompressible flow with no temperature and density effects can symbolically be written as

$$\frac{D\overline{u'_i u'_j}}{Dt} = \frac{\partial \overline{u'_i u'_j}}{\partial t} + \overline{u_k}\frac{\partial \overline{u'_i u'_j}}{\partial x_k} = P_{ij} + \epsilon_{ij} + T_{ij} + \Pi^s_{ij} + \Pi^d_{ij} + D^m_{ij}, \tag{42}$$

where the right hand side terms denote the production rate, dissipation rate, turbulent transport rate, pressure strain rate, pressure diffusion rate, and molecular diffusion rate of $\overline{u'_i u'_j}$ respectively. Subsequently, the budget terms for the mean turbulent kinetic energy k can be obtained by computing half of the sum of the budget terms of $\overline{u'u'}$, $\overline{v'v'}$ and $\overline{w'w'}$. The line over the quantities in Eq. 42 indicates Reynolds-averaging. The continuity constraint, $\Sigma_i \frac{\partial u'_i}{\partial x_i} = 0$, implies that the pressure strain rate term $\Pi^s_{i.i}$ has zero trace, that is, $\Sigma_i \Pi^s_{i,i}(y^+) = 0$ for all y^+. This corresponds to a physical interpretation where the pressure strain rate term $\Pi^s_{i.j}$ redistributes turbulent kinetic energy over the three coordinate directions during the turbulence cascade process without changing the total amount of mean turbulent kinetic energy.

The method used to determine the numerical dissipation rate can be summarized as follows: the transport equation for k is symbolically written as

$$R_k + C_k - \left(P_k + \epsilon_k + T_k + \Pi^d_k + D^m_k\right) = 0, \tag{43}$$

where R_k indicates the Reynolds-averaged time rate of change of k, and C_k the Reynolds-averaged convective transport rate of k. Equation 43 will be contaminated by numerical errors. Consequently, a non-zero residual will remain in this equation, that is,

$$R_{k,\Delta} + C_{k,\Delta} - \left(P_{k,\Delta} + \epsilon_{k,\Delta} + T_{k,\Delta} + \Pi^d_{k,\Delta} + D^m_{k,\Delta}\right) = -\epsilon^{num}_{k,\Delta}, \tag{44}$$

where Δ in the subscripts denotes the numerical counterpart of the exact term. The following error sources are present in the considered non-zero residual: Firstly, the discretisation and iteration errors discussed in section 3.1 are introduced, because the primary flow variables are computed with a finite volume spatial discretisation and a generic temporal discretisation. Secondly, numerical errors are present in the computation of the spatial derivatives in the budget terms, to which the computed primary flow variables serve as input. Finally, statistical errors arise due to the application of the Reynolds-averaging process. As we will show in the subsequent section, the non-zero residual in Eq. 44 corresponds to the numerical dissipation rate of k in the presented UDNS and q-DNS computations. That is, in a statistically steady turbulent channel flow, the numerical dissipation rate of k is equal to the sum of the budget terms of k.

4.3. Computation of statistical quantities

The procedure that we use for the computation of the statistical quantities in Eq. 44 is in principle identical to the statistical procedure commonly used in DNS analyses. The method that we use to compute the averaged quantities in Eq. 44 is second-order accurate. This is consistent with the applied discretisation schemes. Schranner et al. (2015) investigated the importance of the application of higher-order schemes for the computation of the budget terms. They observed that second order and higher-order schemes provide practically identical results while using a second order accurate CFD method. Therefore, second-order accurate numerical schemes were also used by Castiglioni and Domaradzki (2015) for computation of the numerical dissipation rate.

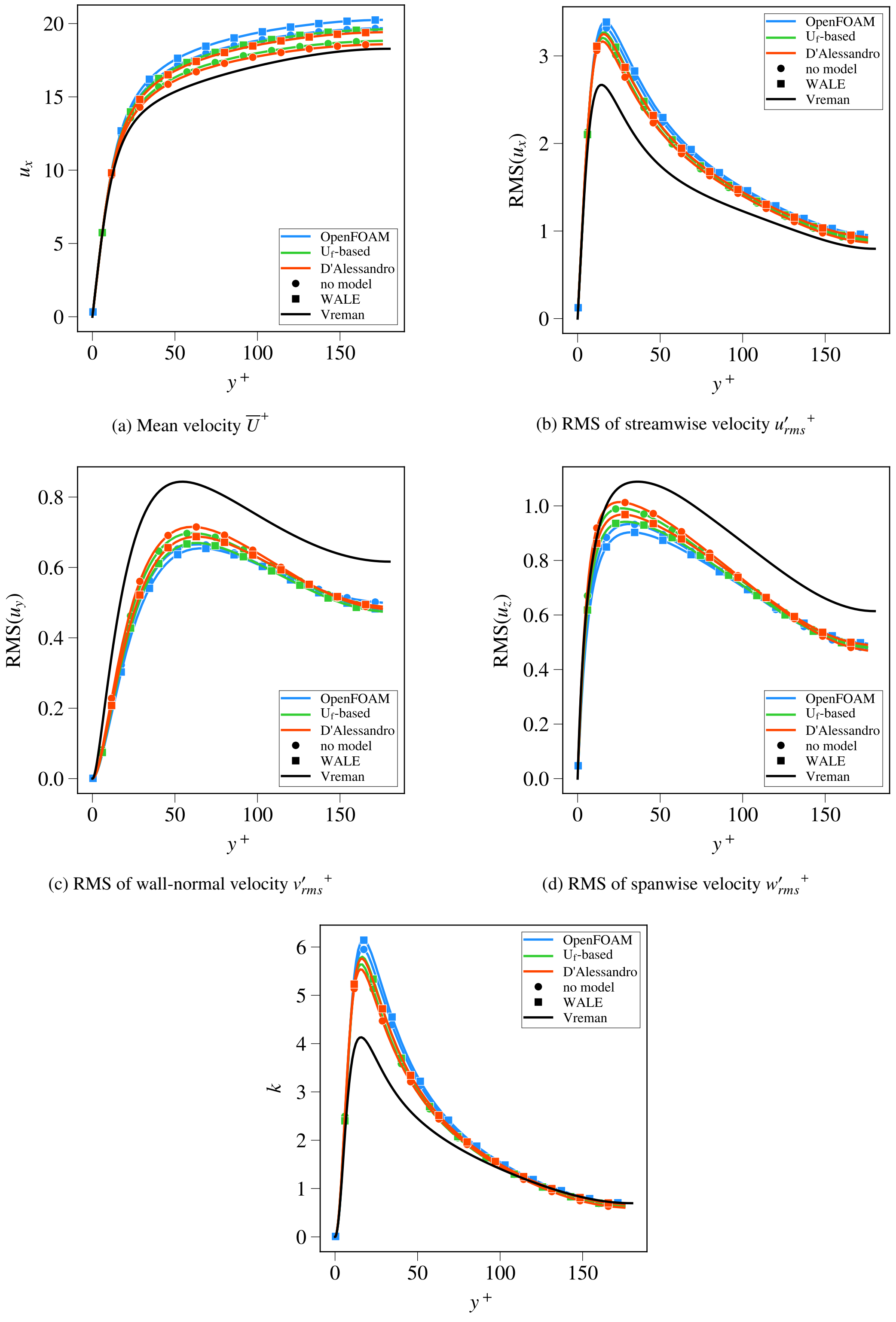
The averaging time which we used for obtaining the statistics was always taken such that the statistical uncertainties are smaller than 0.5% for the mean velocity and smaller than 1% for the RMS velocities and energy budgets. Typically 80 Flow Through Times (FTT) were needed for time averaging, where FTT is defined as FTT = L/u_{mean}, L the length of the channel, and u_{mean} the mean velocity. This approach guarantees that the statistical errors are negligible compared to the numerical dissipation rate of k in our LES and UDNS computations, and that the non-zero residual in Eq. 44 corresponds to this numerical dissipation rate.

4.4. LES and UDNS results

For the three selected velocity interpolation methods, Fig. 10a presents the WALE LES and corresponding *no-model* LES results for the non-dimensional mean velocity. We compare with the reference DNS data of Vreman and Kuerten (2014). As can be concluded from this figure, all RKFoam results overpredict the mean velocity for y^+ larger than approximately 15. This trend has been observed and explained in the literature earlier, see for example Hadžiabdić (2006) and Komen et al. (2017). Namely, the present application of a dissipative numerical method together with a *60-Grid* results in unphysical super low speed streaks in the near wall region. Consequently, the near wall velocity gradients will be underestimated, resulting in too small friction velocities. Therefore, larger flow rates are required in the present computations in order to obtain the desired frictional Reynolds number Re_τ. Consequently, the mean velocity is overpredicted away from the wall, and the linear law is followed exactly in the near wall region for all considered computations. Later in this section, we will demonstrate that the OpenFOAM velocity interpolation method is the most dissipative method among the considered three methods, whereas the method of D'Alessandro is the least dissipative method. Therefore, the OpenFOAM interpolation method shows the largest overpredictions for the mean bulk velocity, and the D'Alessandro method shows the smallest overpredictions. Furthermore, for each velocity interpolation method, it can be concluded that the difference between the WALE LES results and the corresponding *no-model* LES results is fairly small. In principle, the applied eddy-viscosity LES model is designed based on the concept that the effective viscosity is equal to the sum of the molecular and SGS viscosity, that is, $\nu_{eff} = \nu + \nu_{SGS}$. However, numerical dissipation is present in our computations, and the effective viscosity equals $\nu_{eff} = \nu + \nu_{SGS} + \nu_{num}$ in our LES computations, whereas this effective viscosity equals $\nu_{eff} = \nu + \nu_{num}$ in our *no-model* LES computations. As we will see later, the numerical viscosity ν_{num} is a few times larger than the SGS viscosity ν_{SGS}. As a result, the effective viscosity is too large in both the WALE and the *no-model* LES computations, where the WALE computations suffer more from the too large effective viscosity than the *no-model* LES computations. Consequently, the *no-model* LES data are in somewhat better agreement with the reference DNS data than the WALE LES data.

The WALE and *no-model* LES predictions for the non-dimensional RMS velocities are presented in Figs. 10b through 10d. It can be observed that all computations overpredict the streamwise RMS velocities and underpredict the wall normal and spanwise RMS velocities, where the OpenFOAM velocity interpolation method yields the largest overpredictions and underpredictions respectively. Again, the velocity interpolation method of D'Alessandro provides the closest agreements. Similar as for the mean velocity predictions, the difference between the WALE LES predictions and the *no-model* LES predictions is also quite small for the RMS velocities, where the *no-model* LES yields again somewhat better agreement with the reference DNS data than the WALE LES. The explanation for the observed underpredictions and overpredictions of the RMS velocities will be given momentarily.

Figure 10e shows that all computations overpredict the mean turbulent kinetic energy. This overprediction of the mean turbulent kinetic energy corresponds to an overprediction of the mean kinetic energy, which in turn is a direct consequence of the overprediction of the mean velocity. Also for the mean turbulent kinetic energy, we consistently observe that the *no-model* LES predictions are in somewhat better agreement with the reference DNS data than the WALE LES predictions. Furthermore, the OpenFOAM method yields the largest overpredictions, and the D'Alessandro method provides the smallest overpredictions.



(a) Mean velocity $\overline{U}^+$

(b) RMS of streamwise velocity ${u'_{rms}}^+$

(c) RMS of wall-normal velocity ${v'_{rms}}^+$

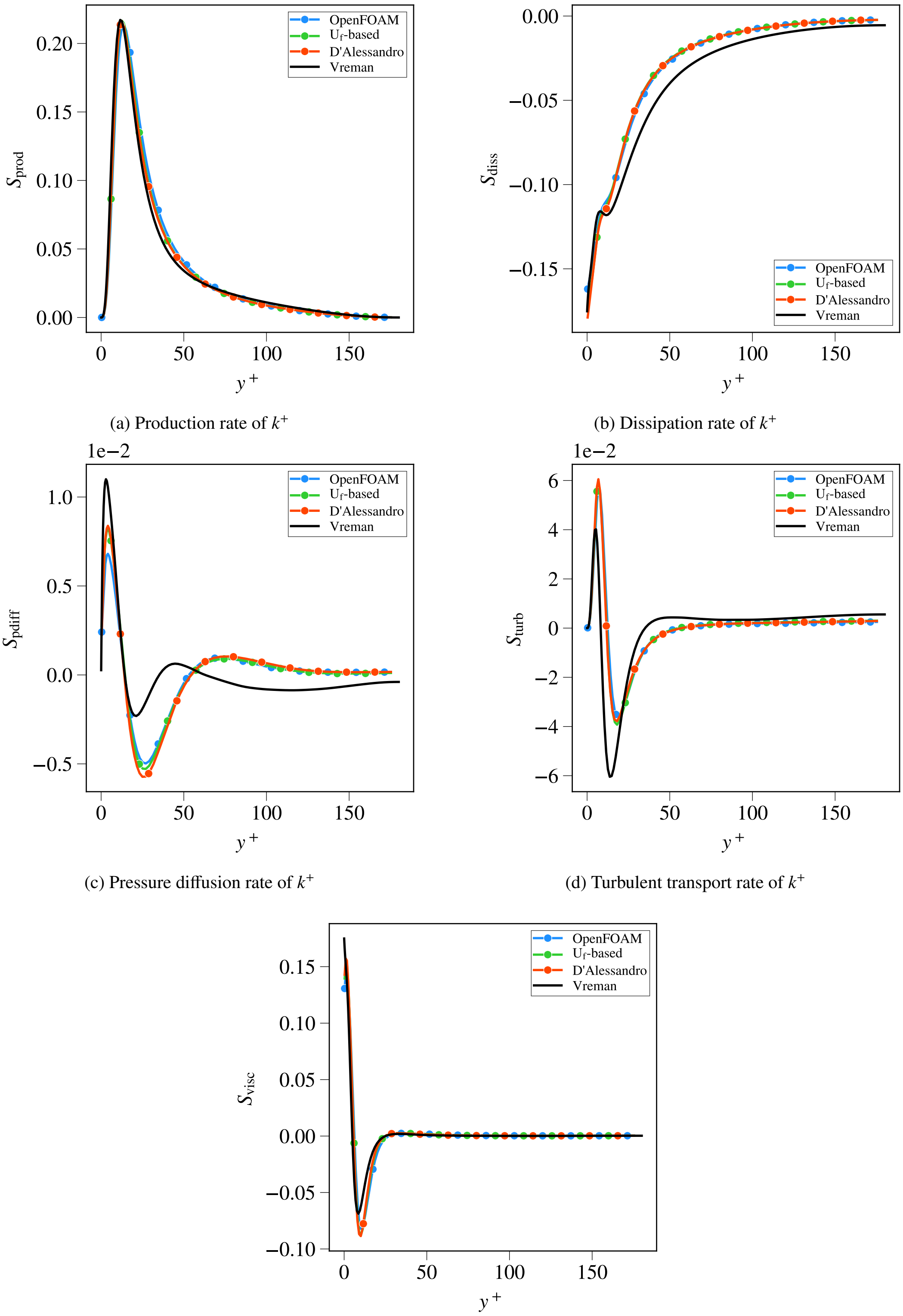
(d) RMS of spanwise velocity ${w'_{rms}}^+$

(e) Mean turbulent kinetic energy k^+

Figure 10: Reference DNS data and RKFoam LES and UDNS data for the mean velocity, RMS velocities, and mean turbulent kinetic energy profiles for fully developed turbulent channel flow at $Re_\tau = 180$.

Figure 11 presents the reference DNS data and corresponding RKFoam UDNS data for the non-dimensional budget terms for the mean turbulent kinetic energy k^+. A scaling of u_τ^4/ν has been used. Generally, it can be concluded that the difference between the results obtained with the three different velocity interpolation methods is very small for all budget terms. As can be seen in Fig. 11a, the production rate is very well predicted by all three UDNS computations. Except for the region close to the wall, the magnitude of the dissipation rate is somewhat underpredicted (Fig. 11b). This results from the fact that accurate computation of the spatial gradients of the velocity fluctuations in the dissipation rate term requires higher mesh resolution than the computation of the $-2\overline{u'v'}\,\partial\overline{u}/\partial y$ term in the production rate term. Figures 11c through 11e present the UDNS results for the three transport terms. Substantial differences between the UDNS results and the DNS data can be observed for the pressure diffusion rate and turbulent transport rate for the applied *60-Grid*. In contrast, the molecular diffusion rate is relatively well predicted.

In a turbulent channel flow, the production term of the $\overline{v'v'}$ and $\overline{w'w'}$ Reynolds stresses is zero. Therefore, the production rate of the $\overline{u'u'}$ stress equals twice the production rate of k^+. This production rate of k^+ is well predicted using the applied *60-Grid* (Fig. 11a). The pressure strain rate terms of the considered Reynolds stresses redistribute the turbulent kinetic energy from the streamwise to the spanwise and wall normal directions. The pressure strain rate terms of the $\overline{u'u'}$, $\overline{v'v'}$, and $\overline{w'w'}$ stresses are presented in Fig. 12. All three pressure strain rate terms are underpredicted on the *60-Grid*. Consequently, too little turbulent kinetic energy is transferred from $\overline{u'u'}$ to $\overline{v'v'}$ and $\overline{w'w'}$. As a result, the *net* total production rate of $\overline{u'u'}$ is overpredicted, whereas the *net* total production rate of $\overline{v'v'}$ and $\overline{w'w'}$ is underpredicted while using the *60-Grid*. This explains why the *60-Grid* results in an underprediction of the RMS of the spanwise and wall normal velocities and in an overprediction of the RMS of the streamwise velocity.



(a) Production rate of k^+

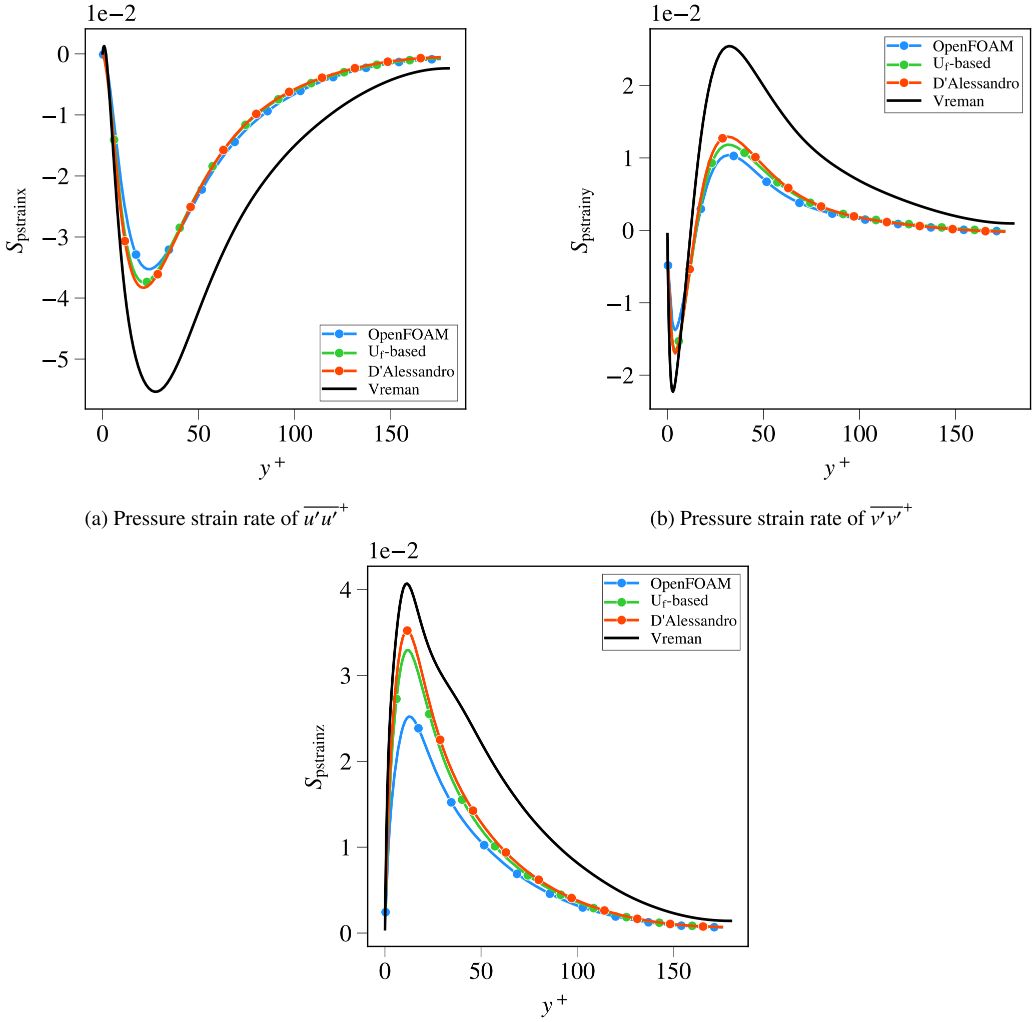
(b) Dissipation rate of k^+

(c) Pressure diffusion rate of k^+

(d) Turbulent transport rate of k^+

(e) Molecular diffusion rate of k^+

Figure 11: Reference DNS data and RKFoam UDNS data for budget terms for the mean turbulent kinetic energy k^+ for fully developed turbulent channel flow at $Re_\tau = 180$.



(a) Pressure strain rate of $\overline{u'u'}^+$

(b) Pressure strain rate of $\overline{v'v'}^+$

(c) Pressure strain rate of $\overline{w'w'}^+$

Figure 12: Reference DNS data and RKFoam UDNS data for the pressure strain rate term of respectively the $\overline{u'u'}$, $\overline{v'v'}$, and the $\overline{w'w'}$ Reynolds stress components for fully developed turbulent channel flow at $Re_\tau = 180$.

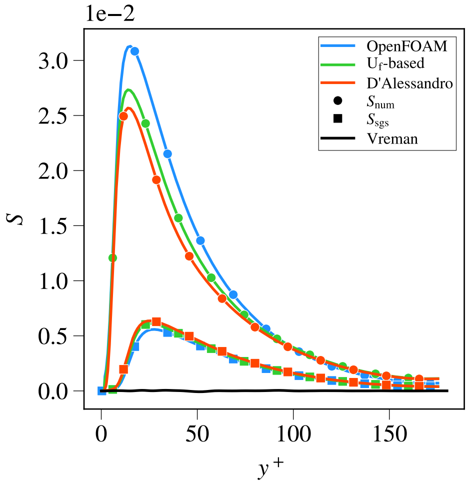


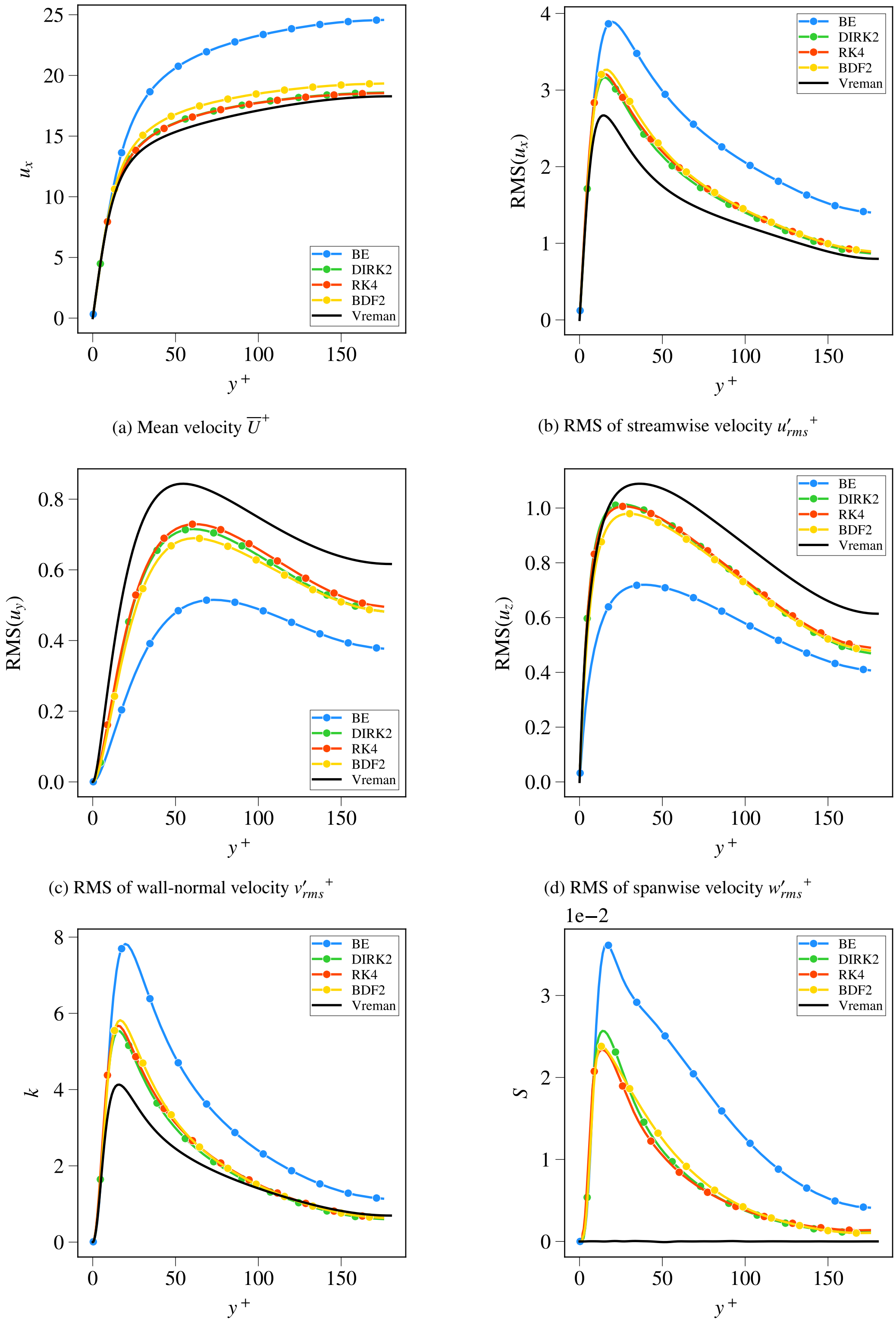
Figure 13: Reference DNS data and RKFoam UDNS data for the numerical dissipation rate of the mean turbulent kinetic energy k^+, and RKFoam LES data for the SGS dissipation rate of k^+ for channel flow at $Re_\tau = 180$.

For statistically steady turbulent channel flows, the distribution of the numerical dissipation rate $\epsilon^{num}_{k,\Delta}(y^+)$ across the height of the channel in our RKFoam UDNS computations is equal to the distribution of the sum of the budget terms of k, see Eq. 44. The distribution of this budget sum is presented in Fig. 13. For the reference data of Vreman and Kuerten (2014), the budget sum is practically zero. In contrast, there is a net budget source for the turbulent kinetic energy in the RKFoam UDNS computations. Therefore, in order to maintain the statistically steady state conditions, an equal amount of numerical dissipation rate should be present in these UDNS computations, compensating for this net source. In the region with significant numerical dissipation, we can observe that the OpenFOAM velocity interpolation method yields the largest numerical dissipation. The D'Alessandro method provides the smallest numerical dissipation rate. These observations are consistent with related observations found earlier.

From Fig. 10, we concluded that the differences between the WALE LES results and the *no-model* LES results are fairly small. Therefore, it is plausible to assume that the amount of numerical dissipation in the LES computations is of the same order of magnitude as in the *no-model* LES computations. The SGS dissipation rate in the WALE LES computations is also shown in Fig. 13. From this figure, it can be concluded that the SGS dissipation rate is substantially smaller than the numerical dissipation rate. This explains why the application of an explicit LES model has little effect in the considered settings.

For the D'Alessandro velocity interpolation method, we have performed additional UDNS computations using our RKFoam solver with the following three time integration schemes: Backward Euler (BE), DIRK2, and RK4. Furthermore, we performed a complementary computation using the standard available icoFOAM solver with the OpenFOAM velocity interpolation method and the implicit second order BDF2 temporal scheme (Ferziger and Perić, 1997). The results are presented in Fig. 14. From this figure, it can be concluded that the applied second order and fourth order temporal schemes provide improved results compared to the first order BE scheme. However, the fourth order RK4 scheme does not perform better than the second order DIRK2 scheme. From the analyses performed for the basic test cases in the previous section, we have concluded that the pressure error results in a reduction of the temporal order of the applied temporal schemes for mesh resolutions used in practice. This explains why the fourth order RK4 scheme provides results which are practically identical to those of the applied second order DIRK2 scheme. Finally, it can be concluded that the RKFoam solver with the DIRK2 scheme provides somewhat better results than the standard icoFOAM solver with the BDF2 scheme.

From the LES and UDNS analysis presented in this section, we have concluded that all three considered numerical methods suffer from substantial numerical dissipation. That is, the SGS dissipation rate is smaller than the numerical dissipation rate. Of the three considered numerical methods, the method of D'Alessandro provides consistently, both qualitatively and quantitatively, somewhat better results, whereas the standard OpenFOAM method performs consistently worse.



(a) Mean velocity $\overline{U}^+$

(b) RMS of streamwise velocity ${u'_{rms}}^+$

(c) RMS of wall-normal velocity ${v'_{rms}}^+$

(d) RMS of spanwise velocity ${w'_{rms}}^+$

(e) Mean turbulent kinetic energy k^+

(f) Sum of the budget terms of k^+

Figure 14: Reference DNS data and RKFoam and icoFOAM UDNS data, using different temporal schemes, for the mean velocity, RMS velocities, mean turbulent kinetic energy k, and sum of budgets of k for fully developed turbulent channel flow at $Re_\tau = 180$.

4.5. q-DNS results

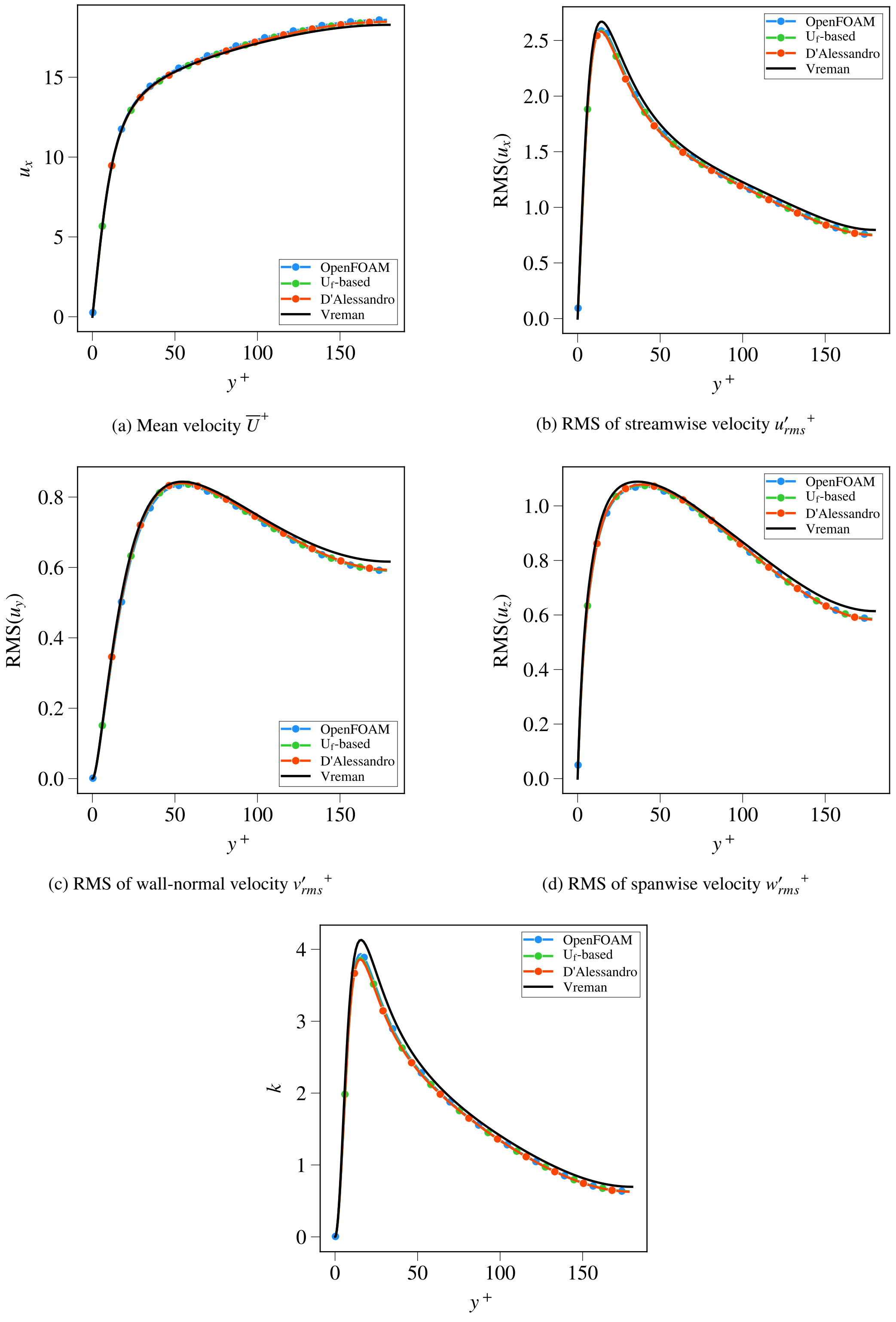
For the three considered velocity interpolation methods, Fig. 15 presents respectively the mean velocity, the RMS velocities, and the mean turbulent kinetic energy k as obtained using the *9-Grid*. From these results, we conclude that the differences between the velocity interpolation methods are practically negligible. This finding is in contrast with the corresponding LES and UDNS results presented in the previous section and the results for the basic test cases in section 3, where the applied velocity interpolation methods showed clear differences. This can be explained as follows: the pressure error and the temporal discretisation error become very small for the applied DNS-type mesh resolution and corresponding small time step. Consequently, the differences between the results obtained with the different velocity interpolation methods become practically negligible. Furthermore, from a comparison with the reference DNS data of Vreman and Kuerten (2014), it can be observed that DNS quality is nearly obtained. Some small deficiencies can still be observed in the presented q-DNS results. Especially the underprediction of the RMS of the spanwise and wall normal velocities, and the underprediction of the peak of the streamwise RMS velocity and the turbulent kinetic energy k can clearly be observed.

The budget terms of k are presented in Fig. 16. Generally, these budget terms are well predicted by the present q-DNS computations. Small differences with the reference DNS data can be observed for the dissipation rate and pressure diffusion rate of k. Figure 17 presents the sum of the budget terms. A wiggle can be observed in this budget sum near $y^+ = 90$. This is attributed to a sudden jump in the wall-normal stretching ratio from 1.05 to 1.0, as the cells have a wall-normal size of 4.5 wall units for $y^+ > 90$, see also Komen et al. (2017). The numerical viscosity in the q-DNS computation can be computed using

$$\nu_{num} = \nu \frac{\epsilon_{k,\Delta}^{num}}{\epsilon_{k,\Delta}}. \tag{45}$$

This indicates that the numerical viscosity can be computed from the sum of the budget terms of the mean turbulent kinetic energy k divided by the resolved dissipation rate of k. The numerical viscosity is presented in Figure 17b. Since this resolved dissipation rate has small values around $y^+ = 90$, the considered wiggle in the budget sum is amplified and shows up as a more pronounced wiggle in the numerical viscosity around $y^+ = 90$.

As explained in section 3.1, the temporal discretisation and the compact stencil in the pressure Poisson equation are the two sources of the kinetic energy conservation error in the present analyses. Using the standard available icoFOAM solver, Komen et al. (2017) performed computations for the following two settings: *no-model* LES, the velocity interpolation method of OpenFOAM, the second order BDF2 temporal scheme (Ferziger and Perić, 1997), the *9-Grid*, and time step size corresponding to Courant numbers of respectively 0.4 and 0.2. Figure 16 in Komen et al. (2017) shows the corresponding numerical dissipation rate. As can be concluded from this figure, the numerical dissipation rate shows less than a linear decrease with time step instead of a quadratic decrease. This observation is consistent with the temporal convergence of the OpenFOAM method obtained in the temporal consistency study for the lid driven cavity test case on a 64x64 grid resolution (Fig. 3). This demonstrates that the error introduced by the application of the compact stencil in the pressure Poisson equation is the dominant source in the kinetic energy conservation error when using a *9-Grid* resolution.



(a) Mean velocity $\overline{U}^+$

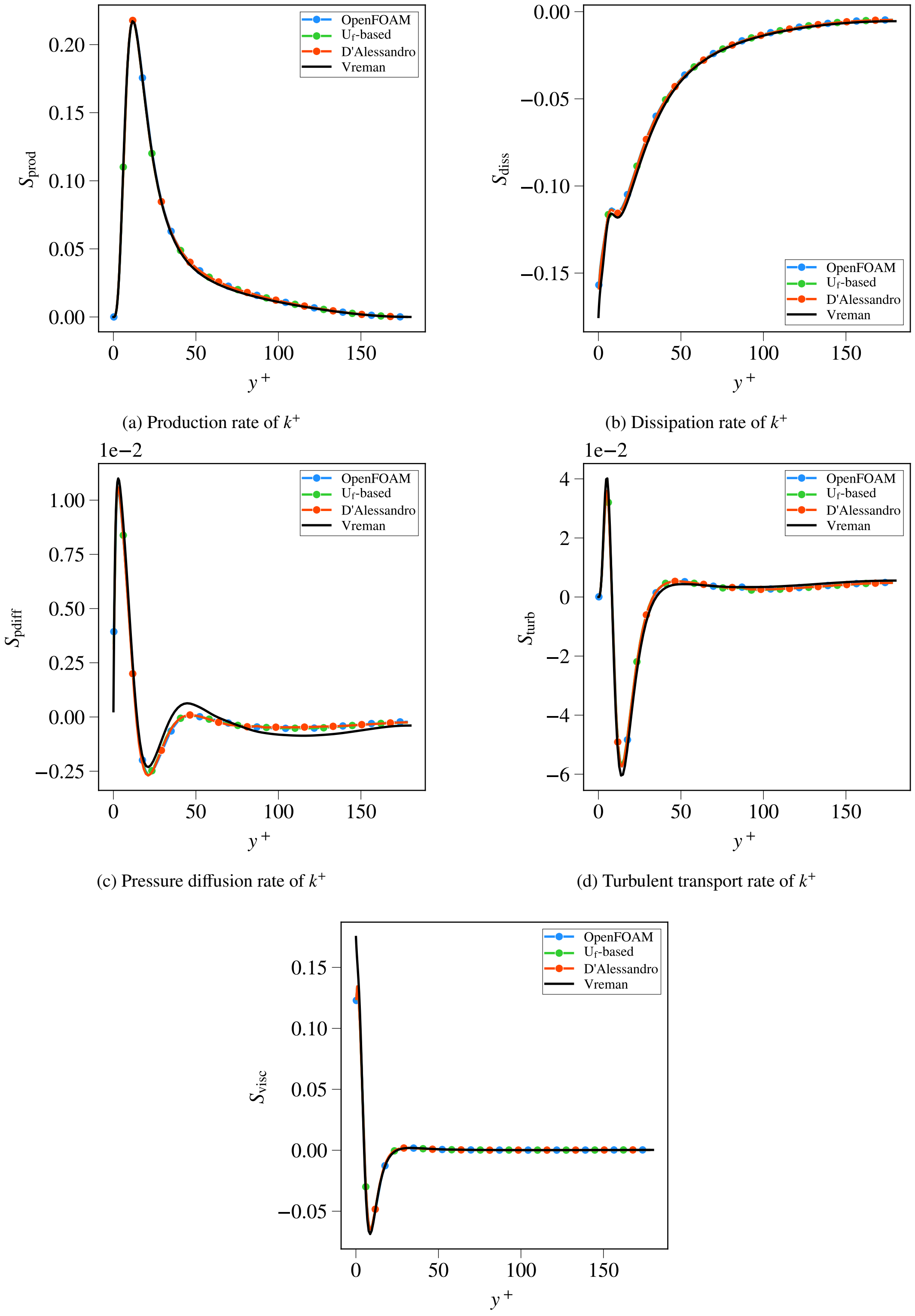
(b) RMS of streamwise velocity ${u'_{rms}}^+$

(c) RMS of wall-normal velocity ${v'_{rms}}^+$

(d) RMS of spanwise velocity ${w'_{rms}}^+$

(e) Mean turbulent kinetic energy k^+

Figure 15: Reference DNS data and RKFoam q-DNS data for the mean velocity, RMS velocities, and mean turbulent kinetic energy profiles for fully developed turbulent channel flow at $Re_\tau = 180$.



(a) Production rate of k^+

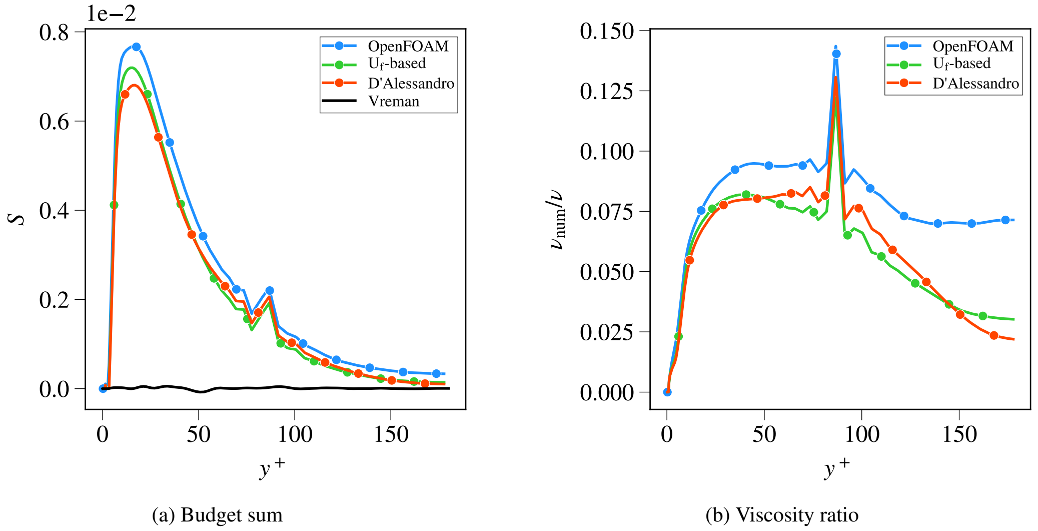
(b) Dissipation rate of k^+

(c) Pressure diffusion rate of k^+

(d) Turbulent transport rate of k^+

(e) Molecular diffusion rate of k^+

Figure 16: Reference DNS data and RKFoam q-DNS data for budget terms for the mean turbulent kinetic energy k^+ for fully developed turbulent channel flow at $Re_\tau = 180$.



(a) Budget sum

(b) Viscosity ratio

Figure 17: Reference DNS data and RKFoam q-DNS data for the sum of the budget terms for the mean turbulent kinetic energy k^+ at $Re_\tau = 180$ and the corresponding q-DNS viscosity ratio $\nu_{num}(y^+)/\nu$.

5. Conclusions

Together with a general Butcher tableau for application of Runge-Kutta time integration schemes, we have implemented the following five different numerical methods within one unified framework in the unstructured collocated finite volume code OpenFOAM: 1) the original algorithm of Rhie and Chow (1983) (u_f-based), 2) the standard OpenFOAM method, 3) the algorithm used by Vuorinen et al. (2014) (u_P-based), 4) the Kazemi-Kamyab et al. (2015) method, and 5) the D'Alessandro et al. (2018) approach. We have determined the effect of these different numerical methods on the numerical dissipation rate and on the temporal consistency of a selection of Runge-Kutta schemes using Taylor-Green vortex and lid-driven cavity test cases. Subsequently, we have performed q-DNS, LES, and corresponding *no-model* LES computations of fully-developed turbulent channel flow at $Re_\tau = 180$ using the methods of respectively Rhie and Chow (1983), D'Alessandro et al. (2018), and the standard OpenFOAM method. We used the method of Komen et al. (2017) in order to quantify the numerical dissipation rate introduced by these three numerical methods in the turbulent channel flow simulations. Furthermore, we explained the consequences of the introduced numerical dissipation on the observed flow trends.

From the Taylor-Green vortex and lid-driven cavity test cases as computed with the uniform hexahedral mesh topology, we have observed that one outer iteration for updating the non-linear convective face fluxes appears to be sufficient. Furthermore, the methods of respectively Rhie and Chow (1983) and Vuorinen et al. (2014) yield very similar results. The consistent Kazemi-Kamyab et al. (2015) method is the only method which preserves the formal order of accuracy of the applied time integration schemes for any grid. However, it was shown to be the most dissipative method. The OpenFOAM interpolation method is also very dissipative. The Rhie and Chow (1983), Vuorinen et al. (2014), and D'Alessandro et al. (2018) methods are the least dissipative methods. However, the DIRK2 and Ascher232 temporal schemes perform slightly better while using the D'Alessandro approach. For the lid-driven cavity test case at Re = 100, the pressure error, which is introduced due to the application of a compact stencil in the pressure Poisson equation, is larger than the temporal discretisation error on a 64x64 grid. Consequently, all applied temporal schemes show $O(\Delta t)$ convergence for all considered numerical methods except the Kazemi method. In contrast, at a 256x256 grid, the pressure error is smaller than the temporal discretisation error for all considered numerical methods for relatively large time steps. As a result, the formal temporal order of accuracy is obtained. However, when the time step is refined, the pressure error gradually starts to dominate over the temporal discretisation error, and the formal temporal order gets lost. For the distorted hexahedral mesh topology, the additional errors introduced due to mesh non-uniformity, non-orthogonality, and skewness increase the numerical energy dissipation substantially and result in a further reduction of the order of the applied temporal schemes for the considered velocity interpolation methods.

From the LES and UDNS channel flow cases for which we used the algorithm of Rhie and Chow (1983), the standard OpenFOAM, and the D'Alessandro et al. (2018) method, we draw the following conclusions: 2 inner iterations and 1 outer iteration appear to be sufficient. Furthermore, all three methods suffer from substantial numerical dissipation in the sense that the SGS dissipation rate is smaller than the numerical dissipation rate. In addition, the method of D'Alessandro provides consistently somewhat better results and shows the smallest amount of numerical dissipation. In contrast, the standard OpenFOAM method provides the least accurate results.

For the q-DNS channel flow cases performed on the *9-Grid*, all three velocity interpolation methods yield practically identical results. This is in contrast with the LES and UDNS cases obtained using the *60-Grid*, where the applied velocity interpolation methods show clear differences. DNS quality is almost obtained in the presented q-DNS analysis.

For the test cases in this study, the pressure error results in a reduction of the order of the temporal schemes. Consequently, application of higher order temporal schemes is not useful from an accuracy point of view, and the application of a second order temporal scheme appears to be sufficient.

It will be worthwhile to investigate whether alternative methods can be implemented in the unstructured collocated finite volume method in OpenFOAM in order to reduce the numerical dissipation rate further. This will be especially needed for explicit LES type analyses, where it is desirable to have a situation where the numerical dissipation rate is substantially smaller than the SGS dissipation rate. We expect that it will be worthwhile to investigate whether the symmetry-preserving discretisation method developed by Verstappen and Veldman (2003) can be implemented in OpenFOAM. Trias et al. (2014) have generalized this method to the unstructured collocated finite volume method.

6. Acknowledgement

The authors are grateful to respectively Dr. V. Vuorinen from Aalto University School of Engineering for providing the OpenFOAM implementation of the method published in Vuorinen et al. (2014) and to Dr. ir. A.H. van Zuijlen from Delft University of Technology for making the OpenFOAM implementation of the method published in Kazemi-Kamyab et al. (2015) available.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

References

ANSYS-FLUENT, 2011. User guide, Version 14.0. ANSYS Inc.. Lebanon.

Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code saturne: A finite volume code for the computation of turbulent incompressible flows: Industrial applications. International Journal on Finite Volumes 1, 1 – 62.

Ascher, U., Ruuth, S., Spiteri, R., 1997. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Applied Numerical Mathematics 25, 151–167.

Bird, R., Stewart, W., Lightfoot, E.N., 1960. Transport phenomena. John Wiley & Sons Inc.

van der Blij, F., 2007. Cell shapes suitable for the shift transformation. Master's thesis. Rijksuniversiteit Groningen, The Netherlands.

van der Bos, F., Geurts, B., 2005. Commutator errors in the filtering approach to large-eddy simulation. Physics of Fluids 17, 035108–035108–20.

Butcher, J., 1964. On Runge-Kutta processes of high order. Journal of the Australian Mathematical Society 4, 179–194.

Castiglioni, G., Domaradzki, J., 2015. A numerical dissipation rate and viscosity in flow simulations with realistic geometry using low-order compressible Navier-Stokes solvers. Computers & Fluids 119, 37–46.

Choi, S., 1999. Note on the use of momentum interpolation method for unsteady flow. Numerical Heat Transfer: Part A: Applications 36:5, 545–550.

D'Alessandro, V., Binci, L., Montelpare, S., Ricci, R., 2018. On the development of OpenFOAM solvers based on explicit and implicit high-order Runge-Kutta schemes for incompressible flows with heat transfer. Computer Physics Communications 222, 14 – 30.

Durbin, P., Petterson-Reif, B., 2011. Statistical Theory and Modelling for Turbulent Flow. Wiley and Sons, Ltd.

Felten, F., Lund, T., 2006. Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow. Journal of Computational Physics 215, 465–484.

Ferziger, J., 1995. Large-eddy and direct simulation of turbulent flows. In Lecture notes for the short course on Calculations of turbulent flows. University of Hamburg. Hamburg, Germany.

Ferziger, J., Perić, M., 1997. Computational Methods for Fluid Dynamics. Springer Verlag, Berlin.

Hadžiabdić, M., 2006. LES, RANS and Combined Simulation of Impinging Flows and Heat Transfer. Ph.D. thesis. Delft University of Technology.

Hairer, E., Norsett, S., Wanner, G., 2008. Solving Ordinary Differential Equations I. Nonstiff problems. Springer-Verlag.

Ham, F., Lien, F., Strong, A., 2002. A fully conservative second-order finitie difference scheme for incompressible flow on nonuniform grids. Journal of Computational Physics 177 (1), 117 – 133.

Hicken, J., Ham, F., Militzer, J., Koksal, M., 2005. A shift transformation for fully conservative methods: turbulence simulation on complex, unstructured grids. Journal of Computational Physics 208, 704 – 734.

Hoyas, S., Jimenez, J., 2008. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. Physics of Fluids 20, 1 – 8.

Issa, R., 1985. Solution of the implicitly dicretised fluid flow equations by operator-splitting. Journal of computational physics 62, 40–65.

Jasak, H., 1996. Error analysis and estimation for the finite volume method with applications to fluid flows. Ph.D. thesis. Imperial College of Science, Technology and Medicine.

van Kan, J., 1986. A second-order accurate pressure correction method for viscous incompressibl flow. SIAM Journal on Scientific and Statistical Computing 7, 870–891.

Kazemi-Kamyab, V., 2013. High Order Time-Accurate Partitioned Simulation of Unsteady Conjugate Heat Transfer. Ph.D. thesis. Delft Technical University.

Kazemi-Kamyab, V., van Zuijlen, A., Bijl, H., 2015. Analysis and application of high order implicit Runge-Kutta schemes to collocated finite volume discretization of the incompressible Navier-Stokes equations. Computers & Fluids , 107–114.

Komen, E., Camilo, L., Shams, A., Geurts, B., Koren, B., 2017. A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows. Journal of Computational Physics 345, 565–595.

Komen, E., Shams, A., Camilo, L., Koren, B., 2014. Quasi-DNS capabilities of open-FOAM for different mesh types. Computers & Fluids 96, 87–104.

Modesti, D., Pirozzoli, S., 2017. A low-dissipative solver for turbulent compressible flows on unstructured meshes, with OpenFOAM implementation. Computers & Fluids 152, 14–23.

Morinishi, Y., Lund, T., Vasilyev, O.V., Moin, P., 1998. Fully conservative higher order finitie difference schemes for incompressible flow. Journal of Computational Physics 143, 90–124.

Morinishi, Y., Vasilyev, O., Ogi, T., 2004. Fully conservative finite difference scheme in cylindrical coordinates for incompressible flow simulations. Journal of Computational Physics 197, 686–710.

Nicoud, F., Ducros, F., 1999. Subgrid-scale stress modeling based on the square of the velocity gradient tensor. Flow Turbulence and Combustion 62, 183–200.

OpenFOAM®, 2017. User guide, Version 5.0. OpenFOAM® Foundation.

Pascau, A., 2011. Cell face velocity alternatives in a structured colocated grid for the unsteady Navier-Stokes equations. International Journal for Numerical Methods in Fluids 65, 812–833.

Pope, S., 2000. Turbulent Flows. Cambridge University Press.

Rhie, C., Chow, W., 1983. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA J. 21, 1525–1532.

Sanderse, B., Koren, B., 2012. Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations. Journal of Computational Physics 231, 3041–3063.

Schranner, F., Domaradzki, J., Hickel, S., Adams, N., 2015. Assessing the numerical dissipation rate and viscosity in numerical simulations of fluid flows. Computer & Fluids 114, 84–97.

Shashank, Larsson, J., Iaccarino, G., 2010. A co-located incompressible Navier-Stokes solver with exact mass, momentum and kinetic energy conservation in the inviscid limit. Journal of Computational Physics 229, 4425–4430.

Shen, C., Sun, F., Xia, X., 2014. Implementation of density-based solver for all speeds in the framework of OpenFOAM. Journal of Computational Physics 185, 2730–2741.

STAR-CCM+, 2013. User guide, Version 8.06. CD-adapco. London.

Trias, F., Lehmkuhl, O., Oliva, A., Pérez-Segarra, C., Verstappen, R., 2014. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. Journal of Computational Physics 258, 246–267.

Verstappen, R., Veldman, A., 2003. Symmetry-preserving discretization of turbulent flow. Journal of Computational Physics 187, 343–368.

Versteeg, H., Malalasekera, W., 2007. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Second Edition, Pearson Education Limited.

de Villiers, E., 2006. The potential of Large Eddy Simulation for the Modeling of wall bounded flows. Ph.D. thesis. Imperial College of Science, Technology and Medicine.

Vreman, A., Kuerten, J., 2014. Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$. Physics of Fluids 26, 015102, 1 – 21.

Vuorinen, V., Keskinen, J., Duwig, C., Boersma, B., 2014. On the implementation of low-dissipative Runge-Kutta projection methods for time dependent flows using OpenFOAM®. Computers & Fluids 93, 153–163.