*fluids*

*Article*

# Fast-Projection Methods for the Incompressible Navier–Stokes Equations

**Carlo De Michele , Francesco Capuano and Gennaro Coppola ***

Dipartimento di Ingegneria Industriale (DII), Università di Napoli "Federico II", 80125 Napoli, Italy;
carlo.demichele2@unina.it (C.D.M.); francesco.capuano@unina.it (F.C.)
* Correspondence: gcoppola@unina.it; Tel.: +39-081-7682534

Received: 10 November 2020; Accepted: 25 November 2020; Published: 27 November 2020

**Abstract:** An analysis of existing and newly derived fast-projection methods for the numerical integration of incompressible Navier–Stokes equations is proposed. Fast-projection methods are based on the explicit time integration of the semi-discretized Navier–Stokes equations with a Runge–Kutta (RK) method, in which only one Pressure Poisson Equation is solved at each time step. The methods are based on a class of interpolation formulas for the pseudo-pressure computed inside the stages of the RK procedure to enforce the divergence-free constraint on the velocity field. The procedure is independent of the particular multi-stage method, and numerical tests are performed on some of the most commonly employed RK schemes. The proposed methodology includes, as special cases, some fast-projection schemes already presented in the literature. An order-of-accuracy analysis of the family of interpolations here presented reveals that the method generally has second-order accuracy, though it is able to attain third-order accuracy only for specific interpolation schemes. Applications to wall-bounded 2D (driven cavity) and 3D (turbulent channel flow) cases are presented to assess the performances of the schemes in more realistic configurations.

**Keywords:** computational fluid dynamics; incompressible flows; projection methods

## 1. Introduction

Numerical discretization of the incompressible Navier–Stokes (NS) equations is nowadays a common practice for the analysis of a variety of fluid-flow phenomena, encompassing both fundamental research and industrial applications. Although computational power has steadily grown over the last decades, the problems tackled by means of numerical simulations are becoming increasingly more challenging, constantly pushing the computational requirements to the limit of the available resources. A typical example is the direct simulation of turbulent flows, where the number of degrees of freedom required to accurately describe the entire range of spatial and temporal scales remains prohibitive for problems of engineering interest [1]. This situation warrants the continuous quest for novel numerical tools and efficient implementations that are able to reduce the amount of computational work needed to solve the discrete flow equations. Although some of the most popular approaches have been well established for several decades, the design of accurate and cost-effective numerical methods is crucial for the success of high-fidelity calculations, and is currently an active field of research [2–4].

The constant-density NS equations constitute a set of differential–algebraic equations (DAEs), wherein mass continuity acts as a divergence-free constraint on the velocity field [5]. A common approach to integrate this set of DAEs in time is by means of so-called projection methods: The momentum equation is time-advanced without satisfying the incompressibility constraint, then a correction is applied to the provisional velocity field to project it onto a divergence-free space without changing its rotational component [6]. A qualifying feature of the method is the coupling with the time-integration procedure. A popular choice for wall-bounded flows is the separate treatment of convective and viscous terms, which are advanced in time by using different time-integration schemes. Diffusion is usually treated implicitly using the Crank–Nicolson method, whereas convection is advanced explicitly with an Adams–Bashforth or a Runge–Kutta (RK) method [7,8]. RK methods are usually preferred for their flexibility, larger stability margins, and small dissipative numerical errors. However, due to the differential–algebraic nature of the underlying equations, RK-based projection schemes require the incompressibility constraint to be enforced at each sub-stage [5], which, in turn, implies the solution of a (typically costly) Pressure Poisson Equation (PPE). As a consequence, the use of multi-stage methods is often associated with a significant increase in computational cost, especially in situations where fast Poisson solvers cannot be employed.

An early attempt to circumvent this drawback was presented by Le and Moin [9], who proposed the enforcement of the divergence-free constraint approximately at intermediate sub-stages by using an explicit estimate of the pressure, whereas the exact projection of the velocity field was performed only at the final step. This procedure allowed a significant reduction in the overall computational cost while producing divergence-free velocity fields at each time step. However, its accuracy has been shown to be limited to the second order, independently of the order of the RK method. A generalization of this procedure was presented by Capuano et al. [10], who considered a different estimate for the approximate pressure inside the stages. It was shown that an ad-hoc linear interpolation formula was able to guarantee third-order accuracy for a three-stage RK method with only one Poisson solve per time step. The accuracy of a class of related fast-projection (FPJ) procedures has been recently analyzed by Karam et al. [11], who proposed a general Taylor-series analysis framework to rationally reduce the number of Poisson equations to be solved within each time step.

In this paper, we build on previous work to analyze a broad class of FPJ methods based on a general family of linear interpolations of the pressure inside the RK stages, and we test their accuracy on benchmarks of increasing complexity.

## 2. Problem Formulation

### 2.1. Continuous Equations and Semi-Discretization

The non-dimensional incompressible NS equations for a homogeneous Newtonian fluid read:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_j^2}, \tag{2}$$

where $u_i$ and $p$ are the Cartesian velocity components and the pressure—here divided by density—respectively. In the Reynolds number, $\mathrm{Re} = \frac{VL}{\nu}$, the values chosen for the reference velocity and length $V$ and $L$ depend on the flow configuration. Due to the incompressibility hypothesis, the continuity equation reduces to the divergence-free kinematic constraint on the velocity field (Equation (1)), with the pressure acting as a Lagrange multiplier enforcing this constraint. Equations (1) and (2) are equipped with suitable initial and boundary conditions (on velocity).

A common practice in the context of space–time partial differential equations (PDE) is to use a semi-discretization technique, where the equations are firstly discretized in space and then integrated in time using a dedicated method for the resulting set of ordinary differential (or differential–algebraic) equations. In our case, upon discretization of the space variables, the system of Equations (1) and (2) can be written as

$$\mathbf{M}\mathbf{u} = \mathbf{r}_1 \tag{3}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{C}(\mathbf{u})\mathbf{u} = -\mathbf{G}\mathbf{p} + \frac{1}{\mathrm{Re}}\mathbf{L}\mathbf{u} + \mathbf{r}_2, \tag{4}$$

where $\mathbf{u}$ is the discrete velocity vector containing the discretization of the three velocity components on the mesh, and $\mathbf{p}$ is the vector of discrete pressure. Denoting with $N_{\mathbf{u}}$ and $N_p$ the number of unknowns on the mesh for velocity and pressure, one has $\mathbf{u} \in \mathbb{R}^{N_{\mathbf{u}}}$ and $\mathbf{p} \in \mathbb{R}^{N_p}$. The matrices $\mathbf{M}$, $\mathbf{C}$, $\mathbf{G}$, and $\mathbf{L}$ are the discrete versions of the divergence, convective, gradient, and diffusion (block Laplacian) operators, respectively, whereas $\mathbf{r}_1$ and $\mathbf{r}_2$ are vectors stemming from the specification of boundary conditions for the continuity and momentum equations. For these quantities, one has $\mathbf{M} \in \mathbb{R}^{N_p \times N_{\mathbf{u}}}$, $\mathbf{C} \in \mathbb{R}^{N_{\mathbf{u}} \times N_{\mathbf{u}}}$, $\mathbf{G} \in \mathbb{R}^{N_{\mathbf{u}} \times N_p}$, $\mathbf{L} \in \mathbb{R}^{N_{\mathbf{u}} \times N_{\mathbf{u}}}$, $\mathbf{r}_1 \in \mathbb{R}^{N_p}$, and $\mathbf{r}_2 \in \mathbb{R}^{N_{\mathbf{u}}}$. In general, $\mathbf{r}_2$ could depend on $\mathbf{u}$ (for Neumann-type boundary conditions) and, when unsteady boundary conditions are considered, both $\mathbf{r}_1$ and $\mathbf{r}_2$ could depend on $t$. However, we leave these explicit dependencies out of our notation, since they are not relevant in the current work. We also note that the analysis presented hereinafter is not restricted to a specific spatial discretization technique as long as the semi-discrete system can be cast as in Equations (3) and (4).

### 2.2. Application of Explicit Runge–Kutta Schemes

In this section, we detail how to further discretize Equations (3) and (4) in time with an explicit (RK) method. The material in this section is based on the analysis reported in [5].

Equations (3) and (4) constitute a system of DAEs, since the continuity equation reduces to a set of algebraic equations for the velocity components on the mesh upon discretization in space. It could be cast as a single system of ordinary differential equations (ODEs) by differentiating Equation (3) with respect to time and substituting it into Equation (4). However, as discussed in [5], this procedure should be avoided in general, since differentiating the constraint changes the index of the DAE system, and a subsequent discretization in time could give a discrete field not satisfying the constraint. When a half-explicit RK procedure (cfr. [12]) is applied, one has the formulation

$$\mathbf{M}\mathbf{u}_i = \mathbf{r}_1 \tag{5}$$

$$\mathbf{u}_i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{i,j} \left( \mathbf{F}_j - \mathbf{G}\mathbf{p}_j \right), \tag{6}$$

with $i = 1, \ldots, s$ ($s$ being the number of stages), plus the final step:

$$\mathbf{M}\mathbf{u}^{n+1} = \mathbf{r}_1 \tag{7}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{j=1}^{s} b_j \left( \mathbf{F}_j - \mathbf{G}\mathbf{p}_j \right), \tag{8}$$

where $\mathbf{F}_j = -\mathbf{C}(\mathbf{u}_j)\mathbf{u}_j + \frac{1}{\mathrm{Re}}\mathbf{L}\mathbf{u}_j + \mathbf{r}_2$. The coefficients $a_{i,j}$ and $b_j$ are the standard RK coefficients, with $\sum b_j = 1$. We will assume that they belong to an explicit RK method, for which $a_{i,j} = 0$ for $i \leq j$.

A straightforward procedure to solve the system of discrete Equations (5) and (6) consists of premultiplying Equation (6) by $\mathbf{M}$ and substituting it into Equation (5), which provides, at each stage $i$, an equation involving the pressures $\mathbf{p}_j$ at previous stages:

$$\boldsymbol{\mathcal{L}} \sum_{j=1}^{i-1} a_{i,j}\mathbf{p}_j = \sum_{j=1}^{i-1} a_{i,j}\mathbf{M}\mathbf{F}_j, \tag{9}$$

where $\mathbf{u}^n$ is assumed to satisfy Equation (3) and $\boldsymbol{\mathcal{L}} = \mathbf{M}\mathbf{G}$ is a discrete Laplace operator acting on pressure variables. Equation (9) can now be substituted into Equation (6) to eliminate the pressure, giving the single equation for the stage $i$:

$$\mathbf{u}_i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{i,j}\boldsymbol{\mathcal{P}}\mathbf{F}_j, \tag{10}$$

where $\mathcal{P}$ is the projection operator $\mathcal{P} = \mathbf{I} - \mathbf{G}\mathcal{L}^{-1}\mathbf{M}$, satisfying $\mathbf{M}\mathcal{P} = \mathbf{0}$. Analogous equations can be written for the final correction step:

$$\mathcal{L}\sum_{j=1}^{s} b_j \mathbf{p}_j = \sum_{j=1}^{s} b_j \mathbf{M}\mathbf{F}_j \tag{11}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{j=1}^{s} b_j \mathcal{P}\mathbf{F}_j. \tag{12}$$

Equations (10) and (12) are in the form of a classical RK procedure, although the presence of the matrix $\mathcal{L}^{-1}$ in the operator $\mathcal{P}$ indicates that the inversion of a linear system for each equation is required, which is given by Equations (9) and (11), respectively. Note that since the RK method is explicit, Equations (9) and (11) involve as unknown only the fields $\mathbf{p}_{i-1}$ and $\mathbf{p}_s$, respectively, since the other pressure fields are known from previous stages. By substituting the known pressures, they can be finally written as

$$\mathcal{L}\mathbf{p}_{i-1} = \mathbf{M}\mathbf{F}_{i-1} \qquad i = 2\ldots s \tag{13}$$

$$\mathcal{L}\mathbf{p}_s = \mathbf{M}\mathbf{F}_s. \tag{14}$$

A total number of $s$ linear systems have to be solved for each time interval, $s-1$, coming from Equation (13), and the last one coming from Equation (14). The property $\mathbf{M}\mathcal{P} = \mathbf{0}$ shows that the discrete constraints of Equations (5) and (7) are satisfied at each stage and at $t^{n+1}$. The variables $\mathbf{p}_j$ represent the discretized pressure inside the stages of the RK procedure, and are consistent with the pressure at time instants $t^n + c_j\Delta t$, where $c_i = \sum_{j=1}^{i-1} a_{i,j}$. The accuracy of these quantities (and of the corresponding velocity vectors $\mathbf{u}_j$ and $\mathbf{u}^{n+1}$) can be evaluated by means of the theory of half-explicit RK methods for DAEs, and is studied in detail in [5].

The algorithm outlined above has the advantage of calculating the pressure field $\mathbf{p}_j$ explicitly at each stage inside the time step. However, it is quite involved, since at each stage, it depends on all the pressure fields of the previous stages, which have to be stored and combined to be used for the calculation of $\mathbf{u}_i$ and $\mathbf{u}^{n+1}$. A more efficient procedure, commonly used in the practical implementation of the projection scheme with an RK method, is to directly solve Equation (9) for a variable proportional to $\sum_{j=1}^{i-1} a_{i,j}\mathbf{p}_j$, and to use this solution to correct a provisional velocity $\mathbf{u}_i^*$ obtained by completely neglecting the contributions of all the pressure fields. For reasons that will be made clear in a moment, we define the new pressure-like variables $\boldsymbol{\phi}$ as

$$\boldsymbol{\phi}_i = \frac{1}{c_i}\sum_{j=1}^{i-1} a_{i,j}\mathbf{p}_j. \tag{15}$$

With these new variables, the pseudo-algorithm for a generic stage $i$ reduces to:

- Calculate a provisional velocity $\mathbf{u}_i^*$ by neglecting all the pressure terms in Equation (10):

$$\mathbf{u}_i^* = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{F}_j. \tag{16}$$

- Solve the linear system of Equation (9), which, expressed in terms of $\boldsymbol{\phi}$, is

$$\mathcal{L}\boldsymbol{\phi}_i = \frac{1}{c_i}\sum_{j=1}^{i-1} a_{i,j}\mathbf{M}\mathbf{F}_j = \frac{1}{c_i\Delta t}\left(\mathbf{M}\mathbf{u}_i^* - \mathbf{r}_1\right). \tag{17}$$

- Correct the velocity field $\mathbf{u}_i^*$ through the equation

$$\mathbf{u}_i = \mathbf{u}_i^* - c_i\Delta t\mathbf{G}\boldsymbol{\phi}_i. \tag{18}$$

The analogous procedure for the final correction step is based on the equations

$$(\mathbf{u}^{n+1})^* = \mathbf{u}^n + \Delta t \sum_{j=1}^{s} b_j \mathbf{F}_j \tag{19}$$

$$\mathcal{L}\boldsymbol{\phi}^{n+1} = \sum_{j=1}^{s} b_j \mathbf{M}\mathbf{F}_j = \frac{1}{\Delta t}\left(\mathbf{M}(\mathbf{u}^{n+1})^* - \mathbf{r}_1\right) \tag{20}$$

$$\mathbf{u}^{n+1} = (\mathbf{u}^{n+1})^* - \Delta t \mathbf{G}\boldsymbol{\phi}^{n+1}. \tag{21}$$

Note that the "true" pressure at $t^{n+1}$ is not required to advance the velocity field; it can can be calculated, if needed, by solving the additional PPE $\mathcal{L}\mathbf{p}^{n+1} = \mathbf{M}\mathbf{F}^{n+1}$.

The advantage of the formulation involving $\boldsymbol{\phi}$ is that, at each stage, one has only one pressure-like variable, which acts as a Lagrange multiplier for the enforcement of the algebraic constraint. However, the question arises about the physical meaning of the pressure-like variable $\boldsymbol{\phi}$, and about its relation with the "true" pressure $p$. In this respect, we follow the observation reported in [5], where the authors note that Equation (17) can be compared to the equation for the exact pressure obtained by taking the divergence of the momentum equation, Equation (4):

$$\mathcal{L}\mathbf{p} = \mathbf{M}\mathbf{F}. \tag{22}$$

Since the term $\frac{1}{c_i}\sum_{j=1}^{i-1} a_{i,j}\mathbf{F}_j$ can be seen as an approximation of the average value of $\mathbf{F}$ from $t^n$ to $t^n + c_i\Delta t$, it is a first-order approximation of $\mathbf{F}_i$. As a consequence, the pressure-like variable $\boldsymbol{\phi}$ is a first-order approximation of the pressure. A further key observation, which is relevant for our work, is that by virtue of the midpoint rule, it is also a second-order approximation of the pressure evaluated at $t = t^n + c_i\Delta t/2$, provided that the stage order of the method is at least 2. The theory presented provides the starting point for the construction of approximate procedures that avoid the solution of a Poisson equation at each stage inside the time step, reducing the total number of PPEs from $s$ to 1.

### 2.3. Fast-Projection Methods

The idea behind FPJ methods is to modify the fractional-step procedure detailed in the previous section in such a way to avoid the costly solution of the PPE for $\boldsymbol{\phi}_j$ at each stage, while keeping a relatively high order of accuracy for the velocity field. This can be achieved by using an approximation of the local pseudo-pressure; the incompressibility constraint required by the RK method is obviously perturbed inside the stages, but if the explicit solution of the final PPE for $\boldsymbol{\phi}^{n+1}$ is retained, the divergence-free character of $\mathbf{u}^{n+1}$ is ultimately preserved. Care must be taken to ensure that the nominal order of accuracy of the velocity and pressure fields is retained.

The first attempt to implement an FPJ method is due to Le and Moin [9], who simply considered the pseudo-pressure $\boldsymbol{\phi}^n$ as frozen inside the time step from $t^n$ to $t^{n+1}$. This method (hereinafter denoted as LM) uses a rough approximation for the pressure inside the stages, resulting in a deterioration of the accuracy of the global method, which is of the second order. As a generalization of LM, Capuano et al. [10] considered a particular linear interpolation of $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$ to estimate the pressures at each stage inside the time step from $t^n$ to $t^{n+1}$. This interpolation was shown to provide higher-order accuracy with respect to LM and lower absolute values of the errors. During the preparation of this manuscript, a further FPJ method has been published [13], where a different linear interpolation of the pseudo-pressure is used. It turns out that the interpolation analyzed by the authors fits one of the methods considered in this paper, which allowed for independent evaluations of its performances.

To have an estimate of the pseudo-pressure $\boldsymbol{\phi}_i$ inside the stages of the time interval $(t^n, t^{n+1})$, we construct an interpolation formula that is based on the values $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$. These two values are assumed to belong to a solution of a PPE, since the calculation of the pressure field for the correction of the velocity at the end of the time step cannot be avoided if one wants a divergence-free velocity field. A straightforward approach consists of a linear interpolation that provides $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$ when evaluated at $t^n$ and $t^{n-1}$, respectively, i.e., the formula

$$\boldsymbol{\phi}(\bar{t}) = \boldsymbol{\phi}^n + \delta\boldsymbol{\phi}\bar{t}, \tag{23}$$

where $\bar{t} = (t - t^n)/\Delta t$ and $\delta\boldsymbol{\phi} = \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n-1}$. Using this function to extrapolate the pseudo-pressure inside the next time step at the stages $t_i = t^n + c_i\Delta t$, one has the approximations

$$\widetilde{\boldsymbol{\phi}}_i = \boldsymbol{\phi}^n + c_i\delta\boldsymbol{\phi}. \tag{24}$$

This extrapolation is based on the pseudo-pressure $\boldsymbol{\phi}$, which is a first-order approximation of the pressure. The accuracy of this procedure is hence expected to be spoiled, as in LM. In an attempt to interpolate the true pressure $\mathbf{p}$, in place of $\boldsymbol{\phi}$, Capuano et al. [10] adopted a different approach in which the values $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$ are assumed as located at the time instants $t = t^n - \Delta t/2$ and $t = t^{n-1} - \Delta t/2$, where they are second-order approximations of the pressure. This assumption gives

$$\boldsymbol{\phi}(\bar{t}) = \mu\boldsymbol{\phi} + (1 + \bar{t})\,\delta\boldsymbol{\phi}, \tag{25}$$

where $\mu\boldsymbol{\phi} = (\boldsymbol{\phi}^n + \boldsymbol{\phi}^{n-1})/2$. This interpolation is then used to calculate the approximate values $\widetilde{\boldsymbol{\phi}}_i$ as the values of the interpolant at time instants $t = t^n + c_i\Delta t/2$, which gives the prediction

$$\widetilde{\boldsymbol{\phi}}_i = \mu\boldsymbol{\phi} + \left(1 + \frac{c_i}{2}\right)\delta\boldsymbol{\phi}. \tag{26}$$

It was shown in [10] that this estimate is able to retain the third-order accuracy for a three-stage, third-order RK procedure.

To have a more general treatment, here, we use an interpolation in which $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$ are assumed to be located at general time instants $t = t^n - \beta\Delta t$ and $t = t^{n-1} - \beta\Delta t$, and the values $\beta = 0$ and $\beta = 1/2$ give the two linear interpolations (23) and (25), as already discussed. As a result, the following general formula for $\boldsymbol{\phi}$ is obtained:

$$\boldsymbol{\phi}_\beta(\bar{t}) = \mu\boldsymbol{\phi} + \left(\frac{1+2\beta}{2} + \bar{t}\right)\delta\boldsymbol{\phi}. \tag{27}$$

The linear interpolation (27) can now be used to calculate the approximate pseudo-pressures at the $i$-th stage inside the time step $(t^n, t^{n+1})$ as an evaluation of $\boldsymbol{\phi}_\beta(\bar{t})$ at a general time instant $\bar{t} = \alpha c_i$, with $\alpha$ ranging form 0 to 1, the cases (24) and (26) corresponding to $\alpha = 1$ and $\alpha = 1/2$, respectively. This assumption gives the general position

$$\widetilde{\boldsymbol{\phi}}_i = \boldsymbol{\phi}_\beta(\alpha c_i) = \mu\boldsymbol{\phi} + \left(\frac{1+2\beta}{2} + \alpha c_i\right)\delta\boldsymbol{\phi}. \tag{28}$$

Equation (28) is a two-parameter family of approximations that gives an expression of the most general linear interpolation of the pseudo-pressure $\boldsymbol{\phi}$ inside the stages as a function of the values $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$. The calculated pseudo-pressure $\widetilde{\boldsymbol{\phi}}_i$ is used as an approximation of $\boldsymbol{\phi}_i$ to avoid the solution of the PPE in Equation (17). The proposed FPJ algorithm is finally articulated as follows:

- Given the velocity field $\mathbf{u}^n$ and the values of the pseudo-pressures $\boldsymbol{\phi}^n$ and $\boldsymbol{\phi}^{n-1}$, the velocity fields at the stages inside the time interval $(t^n, t^{n+1})$ are directly calculated as

$$\mathbf{u}_i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{F}_j - c_i\Delta t\mathbf{G}\widetilde{\boldsymbol{\phi}}_i, \tag{29}$$

where $\widetilde{\boldsymbol{\phi}}_i$ is given by Equation (28) for a particular choice of the parameters $\alpha$ and $\beta$. This evaluation substitutes the steps given by Equations (16)–(18) and avoids the solution of the PPE in Equation (17).

- The final projection step is made as in Equations (19)–(21).

The case $\alpha = \beta = 0$ gives $\widetilde{\boldsymbol{\phi}}_i = \boldsymbol{\phi}^n$, i.e., the LM procedure. The case $\alpha = 1$, $\beta = 0$ gives the standard prediction of Equation (24), also used by Aithal and Ferrante [13], and the case $\alpha = \beta = 1/2$ gives the procedure proposed in [10]. These three methods will be denoted by FPJ-0, FPJ-1, and FPJ-2, respectively, and in the subsequent sections, they are analyzed for various RK methods and in different flow configurations. A general assessment of the formula proposed in Equation (28) is also presented.

## 3. Results

### 3.1. Taylor–Green Vortices

The family of pseudo-pressure interpolations illustrated in the previous section is firstly analyzed by applying the proposed FPJ schemes to the two-dimensional Taylor–Green problem, for which the following analytical solution exists:

$$u_{ex}(x,y,t) = -\cos(x)\sin(y)e^{-2t/\mathrm{Re}}, \tag{30}$$

$$v_{ex}(x,y,t) = \sin(x)\cos(y)e^{-2t/\mathrm{Re}}, \tag{31}$$

$$p_{ex}(x,y,t) = -\frac{1}{4}\left(\cos(2x) + \cos(2y)\right)e^{-4t/\mathrm{Re}}. \tag{32}$$

The viscous NS equations are solved in a 2D square domain with length $2\pi$, with initial conditions obtained by setting $t = 0$ into Equations (30)–(32) and periodic boundary conditions. For the present analysis, the value $\mathrm{Re} = 100$ has been chosen. The equations are discretized in space with a standard pseudo-spectral method on a uniform mesh of $20^2$ points, and the convective terms are expressed by using the so-called skew-symmetric form to ensure spatial conservation of quadratic invariants [14,15]. Time integration is performed by using six different RK methods: the three-stage third-order Heun, Kutta, and Wray schemes, the classical fourth-order RK4 scheme, and the four-stage $3p5q$ and five-stage $3p6q$ pseudo-symplectic schemes presented in [16,17]. The latter methods are here considered with the aim of extending the analysis to nonstandard RK schemes. It is expected that the perturbations to the divergence-free constraint inside the stages alter the improved energy-conservation properties of the pseudo-symplectic schemes. However, since this effect is not relevant for our study, it is not considered here, and a dedicated analysis of this topic is left for possible future investigations. Integration in time is performed up to the final time $T_f = 2$ with different time steps to evaluate the temporal order of accuracy. Our numerical tests show that, except for a slight deviation of the Heun scheme in conjunction with the FPJ-2 procedure, all the considered schemes give very similar results for what concerns the errors on both velocity and pressure fields, with error curves usually indistinguishable in our plots. For this reason, in what follows, only the results for RK4 and Heun schemes are reported.

In Figure 1a, the temporal order of accuracy is shown for the velocity field. The $L_\infty$ norm of the differences between numerically computed and analytical solutions at the final time $T_f$ is employed. It is found that the FPJ-0 and FPJ-1 procedures have second-order accuracy in time, whereas FPJ-2 has third-order accuracy. In addition, the figure shows that the absolute values of the errors are reduced by two or three orders of magnitude for the FPJ-2 method. In Figure 1b, the maximum divergence inside the stages of the final time step is reported for the RK4 and Heun methods. The figure shows that the FPJ-2 method always has the lowest maximum divergence, also confirming the better performances in terms of satisfaction of the divergence-free constraint.

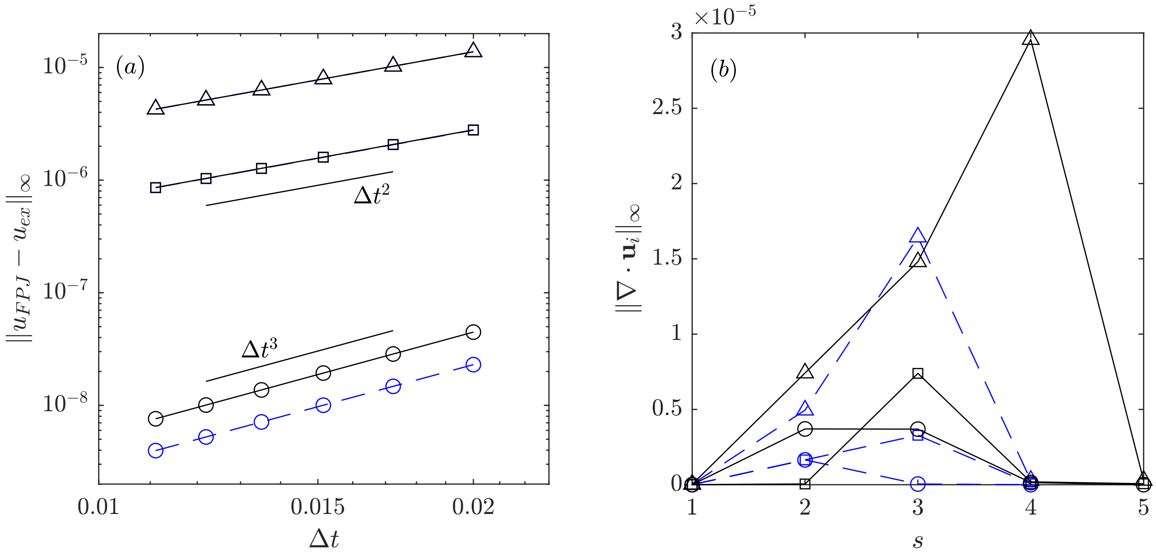

**Figure 1.** Results for the two-dimensional Taylor–Green problem for the fast-projection (FPJ)-0 (△), FPJ-1 (□), and FPJ-2 (○) methods: (**a**) order of accuracy of the velocity field; (**b**) maximum divergence inside the stages. Dashed (blue) lines refer to the Heun method, and continuous (black) lines refer to the RK4 method.

In Figures 2 and 3, a parametric study is reported to assess the accuracy of the whole family of methods expressed by Equation (28). In Figure 2, various quantities are reported, both for the RK4 and the Heun method, as functions of $\alpha$ for the one-parameter family of schemes obtained by restricting Equation (28) to the case $\alpha = \beta$. In this one-parameter family, the FPJ-0 and FPJ-2 schemes are both present and correspond to the values $\alpha = 0$ and $\alpha = 0.5$. In Figure 2a,b, the $L_\infty$ norm of the error on velocity and pressure, respectively, is reported as a function of $\alpha$. The pressure error is calculated as the difference between the calculated value $\boldsymbol{\phi}^n$ and the analytical solution $p^n$ (square symbols) or $p^{n-1/2}$ (round symbols). The plots nicely confirm the strong reduction of the errors between the FPJ-0 and FPJ-2 methods (from $4.3 \times 10^{-6}$ to $7.6 \times 10^{-9}$ on velocity). Figure 2c depicts the maximum divergence inside the stages as a function of $\alpha$, and Figure 2d reports the order of accuracy (on velocity) calculated on the simulation data at each value of $\alpha$. The gain in order of accuracy for the FPJ-2 procedure is clearly shown by the plot in Figure 2d. A similar analysis is reported in Figure 3, where the one-parameter family obtained by restricting Equation (28) to the case $\alpha = 1 - \beta$ is now considered. In this case, the value $\alpha = 1$ corresponds to the FPJ-1 method, whereas, as usual, the case $\alpha = 0.5$ corresponds to the FPJ-2 method. The data reported in Figure 3 confirm all the observations made for the case $\alpha = \beta$. In this case, the reduction in the velocity errors between the FPJ-1 and FPJ-2 methods is from $8.6 \times 10^{-7}$ to $7.6 \times 10^{-9}$.

In Figure 4, a general parametric study of the procedure is presented by independently varying the parameters $\alpha$ and $\beta$. The map shows the magnitude of the $L_\infty$ norm of the error in velocity in the $\alpha - \beta$ plane. Time integration is performed by using the standard RK4 method, but similar results are obtained with all the considered RK methods. The families $\alpha = \beta$ and $\alpha = 1 - \beta$ are depicted as continuous lines, whereas the scheme $\alpha = \beta = 1/2$ is highlighted by the filled dot at the center of the plot. The fully two-dimensional analysis reveals that the FPJ-2 method is a particular member of a family of schemes determined by the values of $\alpha$ and $\beta$ lying on a straight line, which we estimated as $\beta = -2/3\alpha + 5/6$ (dashed line). An inspection of this family of schemes confirms that all its members have third-order accuracy and have performances very similar to that of the FPJ-2 method. A theoretical analysis of this new family of methods is not attempted here, and is left for future work.

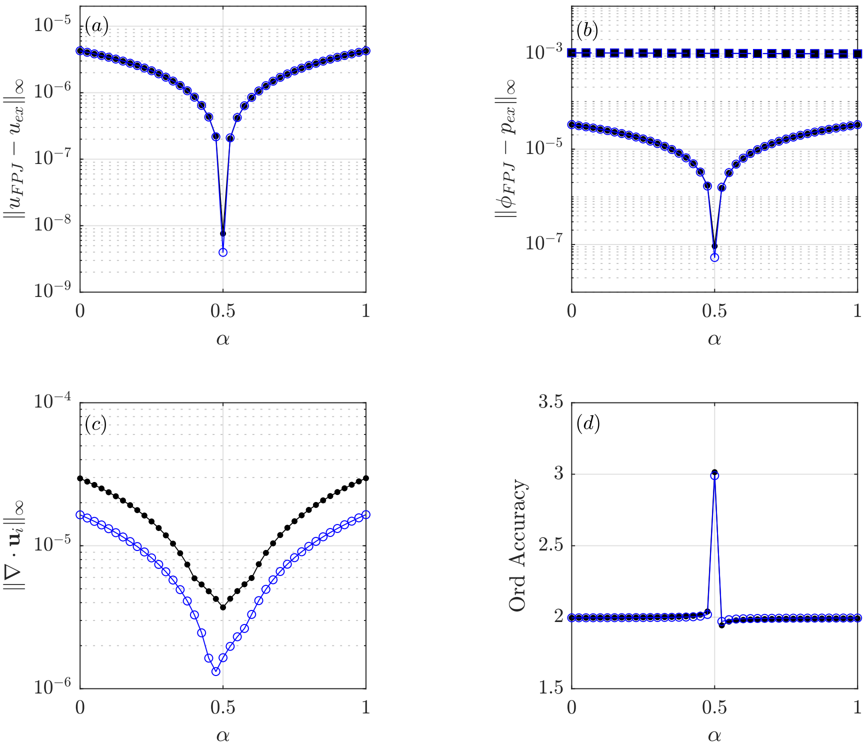

**Figure 2.** Parametric study of the case $\alpha = \beta$. (**a**) Velocity error, (**b**) pressure error (see text), (**c**) maximum divergence inside the stages, (**d**) order of accuracy. Empty circles refer to the Heun method, and filled circles refer to RK4.

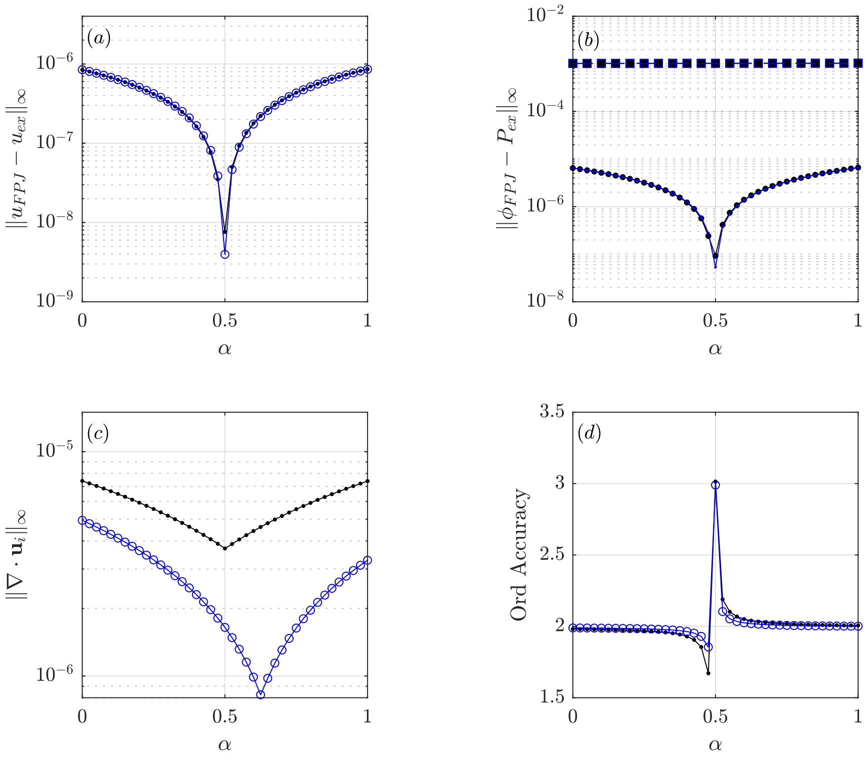

**Figure 3.** Parametric study of the case $\alpha = 1 - \beta$. (**a**) Velocity error, (**b**) pressure error (see text), (**c**) maximum divergence inside the stages, (**d**) order of accuracy. Empty circles refer to the Heun method, and filled circles refer to RK4.

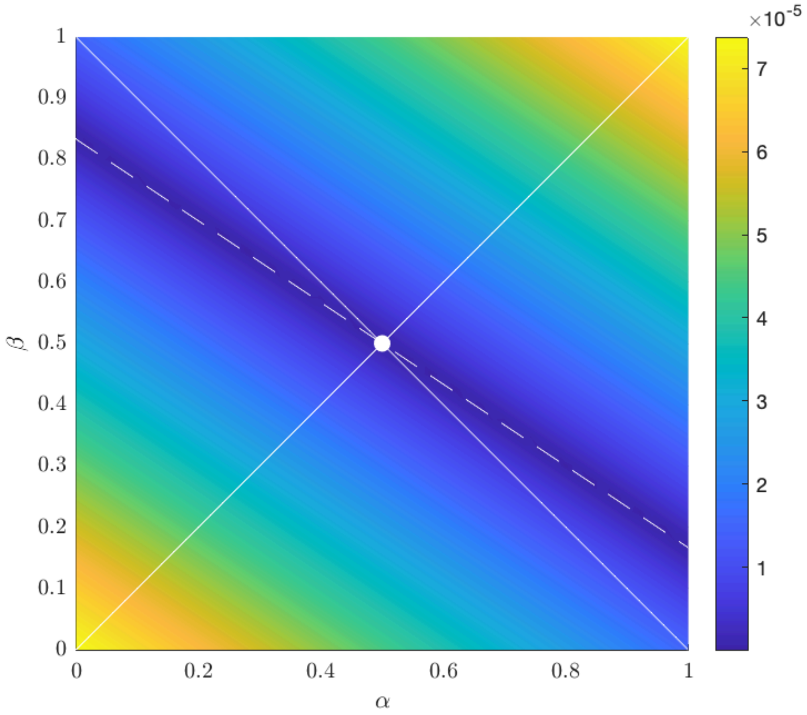

**Figure 4.** Magnitude of the $L_\infty$ norm of the error in velocity in the $\alpha - \beta$ plane using the RK4 method. The continuous lines represent the families $\alpha = \beta$ and $\alpha = 1 - \beta$; the dashed line $\beta = -2/3\alpha + 5/6$ describes the most accurate methods; the filled dot at the curves' intersection marks the FPJ-2 method $\alpha = \beta = 1/2$.

### 3.2. *Lid-Driven Cavity Flow*

The lid-driven cavity flow is a well-known test problem for the assessment of numerical discretizations of NS equations. It has been used in many studies to evaluate the performances of novel numerical methods and, as a consequence, it has also been the subject of many publications aiming at providing accurate benchmark results (e.g., [18,19]). The problem is that of a two-dimensional viscous incompressible flow inside a rigid square cavity in which the top wall moves at a constant velocity parallel to itself. Both the side of the cavity and the velocity of the top lid are assumed to be equal to one in non-dimensional units. For moderate Reynolds numbers, after an initial transient, the problem reaches a steady state with a central core vortex and, depending on the magnitude of the Reynolds number, several corner and side vortices.

The spatial discretization of the problem is here performed with a classical second-order Harlow–Welch procedure on a staggered uniform mesh, which guarantees a robust control of the pressure odd–even decoupling and global preservation of kinetic energy by convective terms ([14,15]). Equations were advanced in time with the classical RK4 scheme, and the three FPJ schemes illustrated in Section 3.1 were employed. The results presented here are relative to the case Re $= 1000$. Spatial discretization was performed with 200 nodes on each dimension and the time step was set such that the Courant number based on the lid velocity was 0.8. The flow was integrated with a full projection scheme, which is assumed as the reference solution, and with the FPJ-0, FPJ-1, and FPJ-2 methods. In Figure 5, the stationary numerical solution is reported as streamlines (left) and magnitude of the horizontal and vertical components of velocity through the (vertical and horizontal, respectively) centerlines of the cavity (right). The stream function contour values are the same as in Table 7 of [19] (also reported in Table III of [18]), provided that a change in the sign is taken into account due to a different convention on the direction of the moving wall. The horizontal and vertical components of velocity in Figure 5 (right) are plotted together with the data reported in Tables I and II of [18]. The output of the various FPJ methods is indistinguishable at plotting accuracy in both cases.

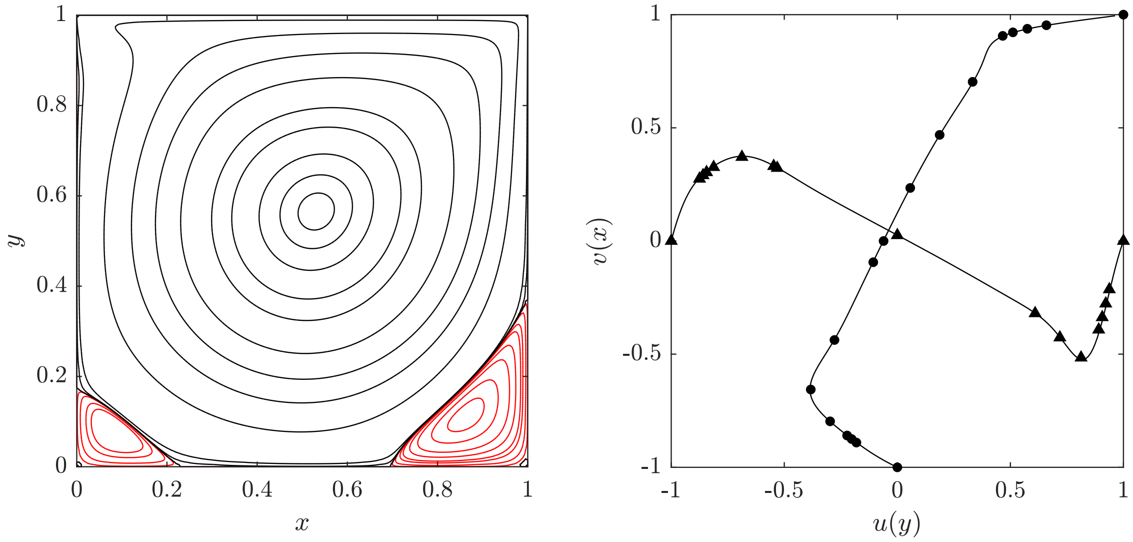

**Figure 5.** Streamlines (**left**) and magnitude of the horizontal and vertical components of velocity through the centerlines of the cavity (**right**) for the lid-driven cavity flow at Re $= 1000$. Black and red streamlines refer to negative and positive values of the streamfunction, respectively. Triangles and circles refer to the vertical and horizontal velocity components of the reference solution in Tables I and II of [18].

In Figure 6, the differences between the values computed with the various FPJ methods and the full-projection method for the horizontal (left) and vertical (right) components of velocity are reported. The plot also shows that for this problem, a remarkable reduction in the errors is obtained with the FPJ-2 method as compared to the FPJ-0 and FPJ-1 methods. The maximum errors on the u-velocity component for FPJ-0, FPJ-1, and FPJ-2 are $2.6 \times 10^{-8}$, $5.3 \times 10^{-9}$, and $9.3 \times 10^{-11}$, respectively, confirming the reduction of two to three orders of magnitude of the maximum error.

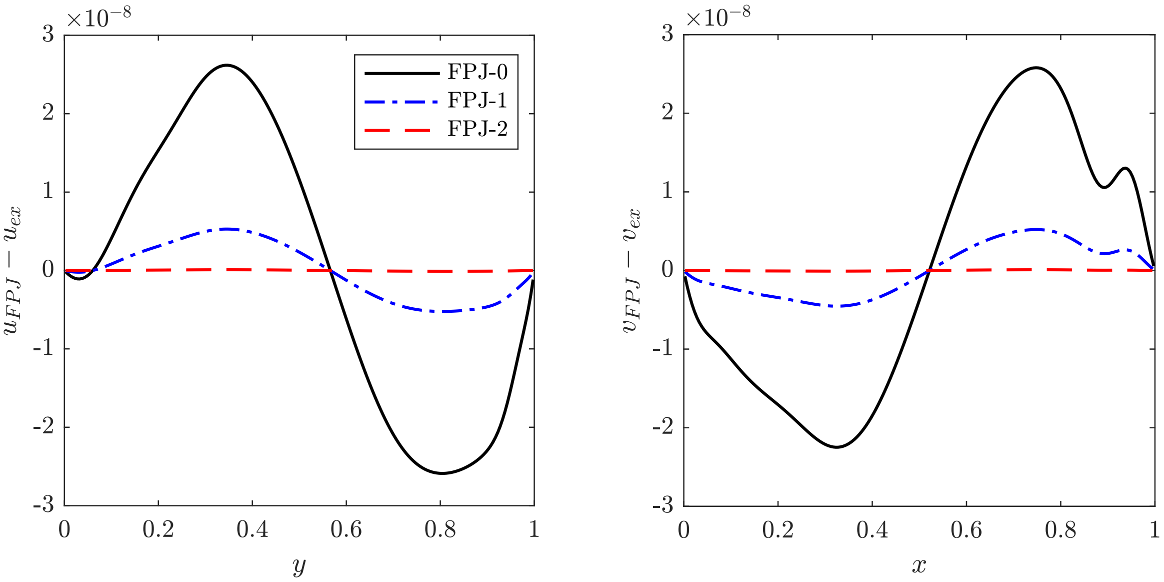

**Figure 6.** Difference between the FPJ and full-projection methods for the horizontal (**left**) and vertical (**right**) components of the velocity along the centerlines of the cavity.

### 3.3. Turbulent Channel Flow

As a last test, the accuracy of the FPJ methods was verified on a fully turbulent configuration. The channel flow is a common choice because, while having simple geometry and boundary conditions, it is key to the investigation of wall-bounded turbulence. Here, we pick $\mathrm{Re}_\tau = \frac{\delta u_\tau}{\nu} = 180$ as in Moser et al. [20], where $\delta$ is half the channel height, $\nu$ the kinematic viscosity, and $u_\tau = \sqrt{\tau_w/\rho}$ the friction velocity, with density $\rho$ and shear stress at the wall $\tau_w$. For reference, Table 1 reports a literature survey of Direct Numerical Simulation (DNS) studies for this value of $\mathrm{Re}_\tau$. Following the usual convention, $x$ identifies the stream-wise direction, $y$ the wall-normal direction, and $z$ the span-wise one; the corresponding velocity components are $u$, $v$, and $w$, respectively. The non-dimensionalized quantities are marked with a plus sign ('+') apex.

In order to run the simulations, the methods were implemented in a pre-existing open-source Fortran 90 incompressible parallel solver, Incompact3D, which was first presented in the paper by Laizet and Lamballais [21] and, later on, by Laizet and Li [22]. It has been extensively used in the academic field because of its versatility and because of the modifiable source code that can be adapted to the needs of the current research. The code employs sixth-order compact schemes for spatial discretization and, among other schemes, the Wray (RKW) method for time integration.

**Table 1.** Overview of DNS simulations of channel flow, $\mathrm{Re}_\tau = 180$. The averaging time $T$ has been non-dimensionalized with the characteristic time $\tau$. The mean velocity has been divided by $u_\tau$, and the root mean square velocities by $u_\tau^2$.

| | | | | **Centerline Values** | | |
|---|---|---|---|---|---|---|
| **Database** | $\mathrm{Re}_\tau$ | $T$ | $U^+$ | $u_{\mathrm{rms}}$ | $v_{\mathrm{rms}}$ | $w_{\mathrm{rms}}$ |
| Incompact3D, exact | 177.1 | 180 | 18.49 | 0.7949 | 0.6132 | 0.6081 |
| Incompact3D, FPJ-2 | 176.9 | 180 | 18.52 | 0.7998 | 0.6154 | 0.6143 |
| Moser, Kim, and Mansour [20] | 178.1 | ? | 18.30 | 0.8140 | 0.6118 | 0.5893 |
| FD2, Vreman and Kuerten [23] | 180.0 | 200 | 18.25 | 0.7974 | 0.6149 | 0.6139 |
| S2, Vreman and Kuerten [23] | 180.0 | 200 | 18.28 | 0.7971 | 0.6166 | 0.6140 |
| Abe, Kawamura, and Matsuo [24] | 180.0 | 40 | 18.64 | 0.8054 | 0.6368 | 0.6041 |
| del Álamo and Jiménez [25] | 185.6 | 50 | 18.28 | 0.7892 | 0.6062 | 0.6068 |
| Kozuka, Seki, and Kawamura [26] | 180.0 | 3.1 | 18.55 | 0.8084 | 0.6410 | 0.6280 |

When applying the approximation methods, particular attention was paid to the peculiar three-register low-storage implementation of the RKW method, which led to a slight variation with respect to the algorithm described in Section 2.2. The three variables that are used at each stage $i$ are the velocity field evaluated at the previous stage $\mathbf{u}_i^*$ and the values of $\mathbf{F}_{i-1}$ and $\mathbf{F}_{i-2}$ (for $i > 1$). These are sufficient because of the special property of the RKW method:

$$\begin{aligned} a_{i,j} &= a_{i-1,j} && \text{for each } i > 3 \text{ and } j < i-2 \\ b_j &= a_{s,j} && \text{for each } j < s-1. \end{aligned} \tag{33}$$

The evolution equation used to update the velocity field from one stage to the next is

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta t \eta_i \mathbf{F}_{i-1} + \Delta t \gamma_i \mathbf{F}_{i-2} - (c_{i-1} - c_{i-2})\Delta t \mathbf{G}\hat{\boldsymbol{\phi}}_{i-1} \tag{34}$$

with $\eta_i = a_{i,i-1}$, $\eta_s = b_s$, $\gamma_i = a_{i,i-2} - a_{i-1,i-2}$, $\gamma_s = b_{s-1} - a_{s-1,s-1}$. The pseudo-pressures $\hat{\boldsymbol{\phi}}_i$ in this scheme are not the same as those that were previously found, but it is possible to find the relationship between them by comparing the values of $(\mathbf{u}_i - \mathbf{u}_{i-1})/(\Delta t)$ obtained from Equation (34) and from Equations (16) and (18):

$$\begin{aligned} \textstyle\sum_{j=1}^{i-1} a_{i,j}\mathbf{F}_j - \sum_{j=1}^{i-2} a_{i-1,j}\mathbf{F}_j - (c_{i-1}\mathbf{G}\boldsymbol{\phi}_{i-1} - c_{i-2}\mathbf{G}\boldsymbol{\phi}_{i-2}) &= \eta_i \mathbf{F}_{i-1} + \gamma_i \mathbf{F}_{i-2} - (c_{i-1} - c_{i-2})\mathbf{G}\hat{\boldsymbol{\phi}}_{i-1} \\ \textstyle\sum_{j=1}^{i-2} \left(a_{i,j}\mathbf{F}_j - a_{i-1,j}\mathbf{F}_j\right) - \gamma_i \mathbf{F}_{i-2} &= (c_{i-1}\mathbf{G}\boldsymbol{\phi}_{i-1} - c_{i-2}\mathbf{G}\boldsymbol{\phi}_{i-2}) - (c_{i-1} - c_{i-2})\mathbf{G}\hat{\boldsymbol{\phi}}_i - 1 \\ \textstyle\sum_{j=1}^{i-3} \left(a_{i,j}\mathbf{F}_j - a_{i-1,j}\mathbf{F}_j\right) &= \mathbf{G}(c_{i-1}\boldsymbol{\phi}_{i-1} - c_{i-2}\boldsymbol{\phi}_{i-2} - (c_{i-1} - c_{i-2})\hat{\boldsymbol{\phi}}_{i-1}). \end{aligned} \tag{35}$$

The left-hand side of the last equation is null because of the property in Equation (33) and, consequently,

$$c_{i-1}\boldsymbol{\Phi}_{i-1} - c_{i-2}\boldsymbol{\Phi}_{i-2} = (c_{i-1} - c_{i-2})\hat{\boldsymbol{\Phi}}_{i-1}. \tag{36}$$

Substituting in Equation (28), the new formula is

$$\tilde{\hat{\boldsymbol{\Phi}}}_i = \mu\boldsymbol{\Phi} + \left(\frac{1+2\beta}{2} + \alpha\frac{c_i^2 - c_{i-1}^2}{c_i - c_{i-1}}\right)\delta\boldsymbol{\Phi}. \tag{37}$$

Analogous transformations need to be applied to other schemes written in low-storage or unconventional forms, and should be derived on a per-case basis. Once this procedure has been implemented, it is possible to compare the results of the standard method involving the resolution of all the pressure equations with those obtained using the third-order accurate FPJ-2 method.

Both cases have the same geometry, discretization, and initial conditions. The domain in the stream-wise, normal, and span-wise directions has size $4\pi \times 2 \times \frac{4}{3}$, and has been discretized with a $128 \times 129 \times 128$ mesh, as in [20]. For the time step, the value $\Delta t = 0.01$ was chosen. The flow was initialized with the laminar solution and a superposition of random noise to trigger transition to turbulence. Once a statistically steady state was reached (as confirmed by a stable value of the friction coefficient), statistics were collected every tenth of the typical turnover time of the large-scale eddies, defined as $\tau = \delta/u_\tau$, so that the samples could be considered as statistically independent. In order to make sure that convergence was reached, 1800 samples were collected; as usual, the flow field was averaged along the homogeneous directions ($x - z$ planes) and with respect to $y = 0$.

The results obtained with both methods are in very good agreement with classical results from literature. For comparison, two reference datasets were considered: The one associated with the work of Moser, Kim and Mansour (MKM) [20], which has the same spatial resolution as in the present work, and the one listed as S2 in [23] (VK), which uses a finer mesh. Both datasets were obtained using spectral methods.

In Figure 7 (left), very good agreement can be observed in terms of the mean velocity profile, except from a slight deviation in the center of the channel. Both methods demonstrate that they correctly evaluate the root-mean-square velocities (Figure 7, right) and kinetic energy dissipation rate (Figure 8, left). In order to better distinguish the results produced by the different approaches, relative differences are shown on the right-hand side of Figure 8, which were computed by dividing the difference by the value found with the exact method. The discrepancy in the case of the velocity profile is always under $3 \times 10^{-3}$; slightly larger values are reached when considering the other quantities; however, this might be due in part to them approaching zero and consequently increasing the relative difference.

As expected, the use of the FPJ procedure brought about a reduction in the wall-clock time of the simulation. A careful analysis performed using 256 CPUs on the GALILEO supercomputer (CINECA) showed that the average time step took approximately $3.024 \times 10^{-2}$ s for the standard method, and about $2.243 \times 10^{-2}$ s for the FPJ one, i.e., a reduction of almost 26%. This specific value is closely related to the algorithm employed by Incompact3D, which makes use of an optimized direct Poisson solver; we expect higher CPU time savings in situations where the PPE solution takes a larger fraction of the overall algorithm.

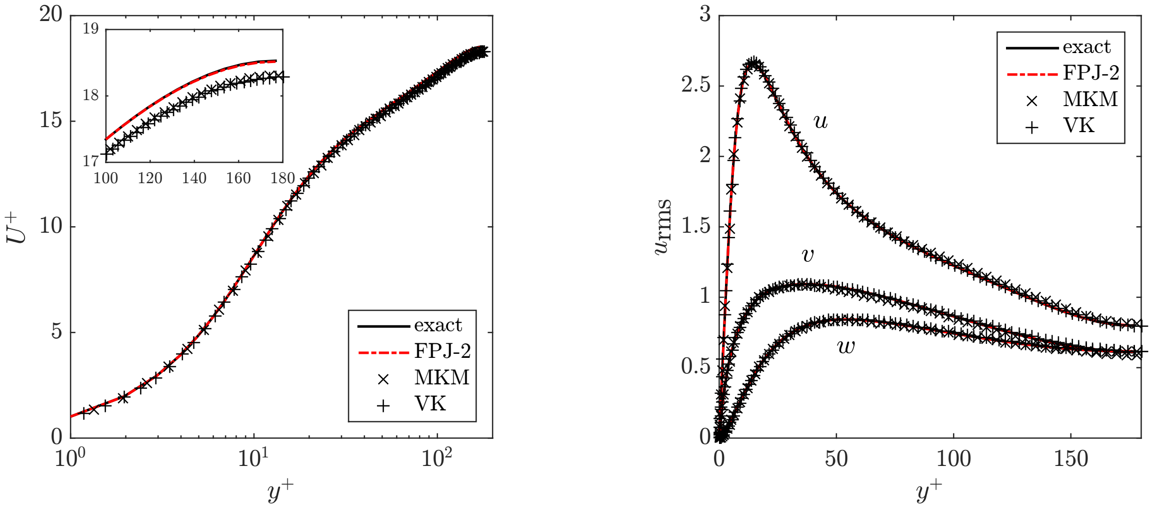

**Figure 7.** Comparison of the mean stream-wise velocity (**left**) and root-mean-square velocities (**right**) for the channel flow with published data at $\mathrm{Re}_\tau = 180$.

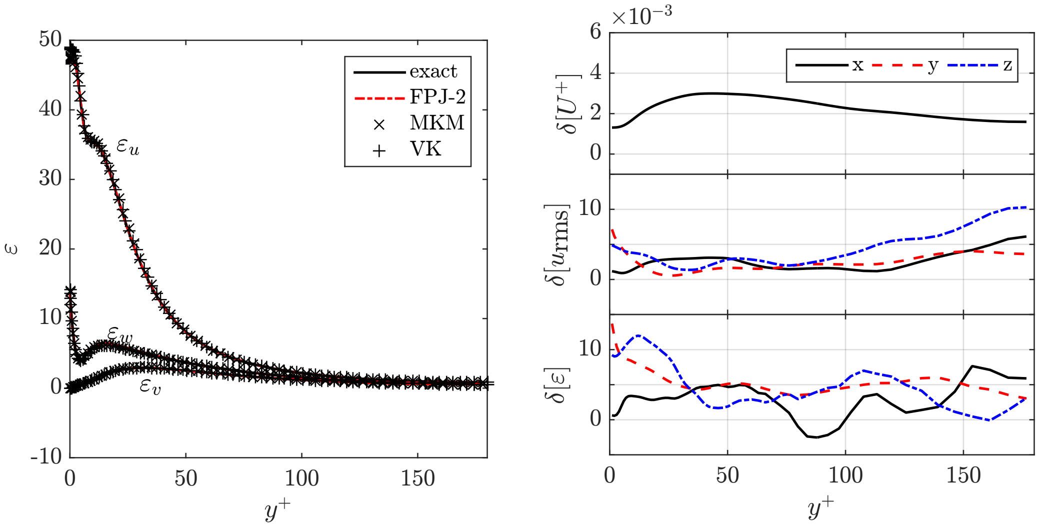

**Figure 8.** On the **left**, comparison of the kinetic energy dissipation rate components for the channel flow with published data at $\mathrm{Re}_\tau = 180$. On the **right**, relative difference of the results obtained with FPJ-2 and exact methods for the computed statistics.

## 4. Conclusions

In this paper, we proposed a cost-effective fractional-step algorithm for the incompressible Navier–Stokes equations based on arbitrary explicit RK time-integration schemes. The approach, called the FPJ method, is able to decrease the number of PPEs to be solved within a time step from $s$ (the number of stages of the RK method) to only one, therefore significantly alleviating the required computational cost. The FPJ method consists of projecting the velocity field at the intermediate sub-stages by means of a suitable approximation of the pseudo-pressure, rather than by solving the corresponding PPE. In this work, we explored a general family of linear interpolations for the pseudo-pressure based on its values at previous time steps. Of note, earlier versions of FPJ methods proposed in the literature are found to be particular members of this family. We assessed the temporal accuracy of the velocity field for various methods obtained using different interpolation formulas.

A systematic characterization of the numerical errors was carried out for two-dimensional Taylor–Green vortices, for which an analytical solution is known. A careful analysis of the results showed that, only for a specific interpolation of the pseudo-pressure (called FPJ-2), the method provides third-order-accurate velocity fields, which are generally only second-order accurate for the other methods. Similarly, a lid-driven cavity test demonstrated that the FPJ-2 method can provide a significant reduction of the numerical error when compared to a reference method where $s$ Poisson solves are performed. Finally, we implemented the FPJ-2 method in an existing NS solver, Incompact3D, to perform a turbulent channel flow simulation. Again, the FPJ-2 scheme showed very little relative error when compared to the traditional method ($s$ PPE solves), and both provided results in very good agreement with reference datasets.

Future work includes a more in-depth investigation of the family of interpolation formulas providing third-order accuracy as well as a careful inspection of the energy-conserving properties of the FPJ algorithms.

**Author Contributions:** Conceptualization, G.C., F.C., and C.D.M.; Investigation, C.D.M. and G.C.; Software, C.D.M.; Supervision, G.C. and F.C.; Writing—original draft, G.C., F.C., and C.D.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** Part of this work was supported by a grant of HPC time from CINECA under the ISCRA project *Fast-P*.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2000.
2. Capuano, F.; Palumbo, A.; de Luca, L. Comparative study of spectral-element and finite-volume solvers for direct numerical simulation of synthetic jets. *Comp. Fluids* **2019**, *179*, 228–237. [CrossRef]
3. Vermeire, B.C.; Witherden, F.D.; Vincent, P.E. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *J. Comp. Phys.* **2017**, *334*, 497–521. [CrossRef]
4. Capuano, F.; Coppola, G.; Balarac, G.; de Luca, L. Energy preserving turbulent simulations at a reduced computational cost. *J. Comp. Phys.* **2015**, *298*, 480–494. [CrossRef]
5. Sanderse, B.; Koren, B. Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier-Stokes equations. *J. Comp. Phys.* **2012**, *231*, 3041–3063. [CrossRef]
6. Chorin, A. Numerical solution of the Navier-Stokes equations. *Math. Comput.* **1968**, *22*, 745–762. [CrossRef]
7. Kim, J.; Moin, P. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.* **1985**, *59*, 308–323. [CrossRef]
8. Verzicco, R.; Orlandi, P. A Finite-Difference Scheme for Three-Dimensional Incompressible Flows in Cylindrical Coordinates. *J. Comp. Phys.* **1996**, *123*, 402–414. [CrossRef]
9. Le, H.; Moin, P. An Improvement of Fractional Step Methods for the Incompressible Navier-Stokes Equations. *J. Comp. Phys.* **1991**, *92*, 369–379. [CrossRef]
10. Capuano, F.; Coppola, G.; Chiatto, M.; de Luca, L. Approximate Projection Method for the Incompressible Navier-Stokes Equations. *AIAA J.* **2016**, *54*, 2179–2182. [CrossRef]
11. Karam, M.; Sutherland, J.C.; Hansen, M.; Saad, T. A Framework for Analyzing the Temporal Accuracy of Pressure Projection Methods. In Proceedings of the AIAA Aviation 2019 Forum, Dallas, TX, USA, 17–21 June 2019; p. 3634. [CrossRef]
12. Hairer, E.; Lubich, C.; Roche, M. *The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods*; Springer: Berlin/Heidelberg, Germany, 1989.
13. Aithal, A.B.; Ferrante, A. A fast pressure-correction method for incompressible flows over curved walls. *J. Comp. Phys.* **2020**, *421*, 109693. [CrossRef]
14. Coppola, G.; Capuano, F.; de Luca, L. Energy-preserving discretizations of the Navier-Stokes equations. classical and modern approaches. In Proceedings of the 23rd Conference of the Italian Association of Theoretical and Applied Mechanics (AIMETA 2017), Salerno, Italy, 4–7 September 2017; Volume 3, pp. 2284–2310.
15. Coppola, G.; Capuano, F.; de Luca, L. Discrete Energy-Conservation Properties in the Numerical Simulation of the Navier–Stokes Equations. *Appl. Mech. Rev.* **2019**, *71*, 010803. [CrossRef]

16. Capuano, F.; Sanderse, B.; De Angelis, E.M.; Coppola, G. A minimum-dissipation time-integration strategy for large-eddy simulation of incompressible turbulent flows. In Proceedings of the 23rd Conference of the Italian Association of Theoretical and Applied Mechanics (AIMETA 2017), Salerno, Italy, 4–7 September 2017; Volume 3, pp. 2311–2323.
17. Capuano, F.; Coppola, G.; Rández, L.; de Luca, L. Explicit Runge-Kutta schemes for incompressible flow with improved energy-conservation properties. *J. Comp. Phys.* **2017**, *328*, 86–94. [CrossRef]
18. Ghia, U.; Ghia, K.N.; Shin, C.T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comp. Phys.* **1982**, *48*, 387–411. [CrossRef]
19. Botella, O.; Peyret, R. Benchmark spectral results on the lid-driven cavity flow. *Comput. Fluids* **1998**, *27*, 421–433. [CrossRef]
20. Moser, R.D.; Kim, J.; Mansour, N.N. Direct numerical simulation of turbulent channel flow up to $\text{Re}_\tau = 590$. *Phys. Fluids* **1999**, *11*, 943–945. [CrossRef]
21. Laizet, S.; Lamballais, E. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *J. Comp. Phys.* **2009**, *228*, 5989–6015. [CrossRef]
22. Laizet, S.; Li, N. Incompact3d: A powerful tool to tackle turbulence problems with up to $O(10^5)$ computational cores. *Int. J. Numer. Methods Fluids* **2011**, *67*, 1735–1757. [CrossRef]
23. Vreman, A.W.; Kuerten, J. Comparison of direct numerical simulation databases of turbulent channel flow at $\text{Re}_\tau = 180$. *Phys Fluids* **2014**, *26*, 015102. [CrossRef]
24. Abe, H.; Kawamura, H.; Matsuo, Y. Direct Numerical Simulation of a Fully Developed Turbulent Channel Flow With Respect to the Reynolds Number Dependence. *J. Fluids Eng.* **2001**, *123*, 382–393. [CrossRef]
25. Juan, C.; del Álamo, J.; Jiménez, J. Spectra of the very large anisotropic scales in turbulent channels. *Phys. Fluids* **2003**, *15*, L41.
26. Kozuka, M.; Seki, Y.; Kawamura, H. DNS of turbulent heat transfer in a channel flow with a high spatial resolution. *Int. J. Heat Fluid Flow* **2009**, *30*, 514–524. [CrossRef]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.