Temporal accuracy of FastRK3

Mira Tipirneni

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Aeronautics and Astronautics

University of Washington

2021

Reading Committee:
Antonino Ferrante, Chair
Uri Shumlak

Program Authorized to Offer Degree:
Aeronautics and Astronautics

University of Washington

Abstract

Temporal accuracy of FastRK3

Mira Tipirneni

Chair of the Supervisory Committee:
Associate Professor Antonino Ferrante
Aeronautics and Astronautics

Standard third-order Runge-Kutta (RK3) [4] can solve the incompressible Navier-Stokes equations with third-order accuracy, but does so by solving the Poisson equation for pressure at each of its three sub-steps. FastRK3 [1] is a computational method for solving the incompressible Navier-Stokes equations in curvilinear coordinates over an orthogonal computational grid and is build upon the standard RK3 method. It solves the Poisson equation for pressure only once per time step. This implies that FastRK3 saves 66.6% of computational time standard RK3 spends solving the Poisson equation. The present thesis provides an analytical proof for the order of accuracy of FastRK3 and proves that it maintains third-order accuracy for free shear flows. The analytical proof is then validated numerically by performing the temporal accuracy analysis of FastRK3 for the Taylor-Green vortex flow.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I'm very grateful for the patient guidance and support of my advisor, Professor Antonino Ferrante. Not long ago I had only just learned the importance of using double precision data in post-processing. Now here we are, with a completed piece of work. Many thanks to Professor Ferrante for all his help in getting me to this point. I would also like to thank Professor Uri Shumlak for serving on my committee and for his helpful feedback on this work.

The work presented here has been developed in collaboration with Abhiram Aithal. I would like to thank Abhiram for his indispensable insight and counsel throughout the project, and for his many contributions which include developing FastRK3 and much of its proof of accuracy.

I imagine writing a thesis is never easy, but writing one a year into a pandemic and quarantine had unique challenges. Marie, Darren, Aubin, Iden, (and Duke), I am so grateful for the good company, good food, and many laughs during an otherwise trying time. I owe y'all my (relative) sanity.

Last but not least, thank you to my family for all the love and support, without which, this would not have been possible.

This work was facilitated though the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer at the University of Washington.

DEDICATION

to my parents; thank you for life, love, and for believing in me.

Chapter 1
INTRODUCTION

1.1 Background

Incompressible flows, flows in which the density of a fluid element remains constant as it is advected through space, are ubiquitous in engineering applications and natural phenomena. Any flow with a Mach number below 0.3 can be approximated as incompressible, and this category of flow is relevant in many applications (low-speed aerodynamics, hydrodynamics, meteorology, biology, etc). Accurate simulation of such flows can provide insight into fundamental flow physics, as well as advance understanding of flows specific to these various applications.

Temporal order of accuracy is important when simulating flows. Non-linear effects can only be resolved by solvers with at least second-order accuracy, and the higher the order of accuracy of a solver, the greater the reduction in error when decreasing the simulation time step. Generally, methods with second-order temporal accuracy are sufficient for resolving non-linearities in incompressible flows. However, there are cases in which a third-order formulation is beneficial, or even necessary. For example, the order reduction associated with simulating wall-bounded flows means second-order algorithms such as Adams-Bashforth (AB) yield first order solutions in these cases [1]. This motivates the need for a third-order time integration scheme.

Runge-Kutta methods have various properties which are beneficial when solving the Navier-Stokes (NS) equations. These methods are self-starting and tend to have better stability characteristics than other solvers. The stability regions of explicit second and third-order RK methods are larger than the region for AB [1]. The RK3 method is also conditionally stable for pure convection whereas the stability of second-order schemes AB and two-step

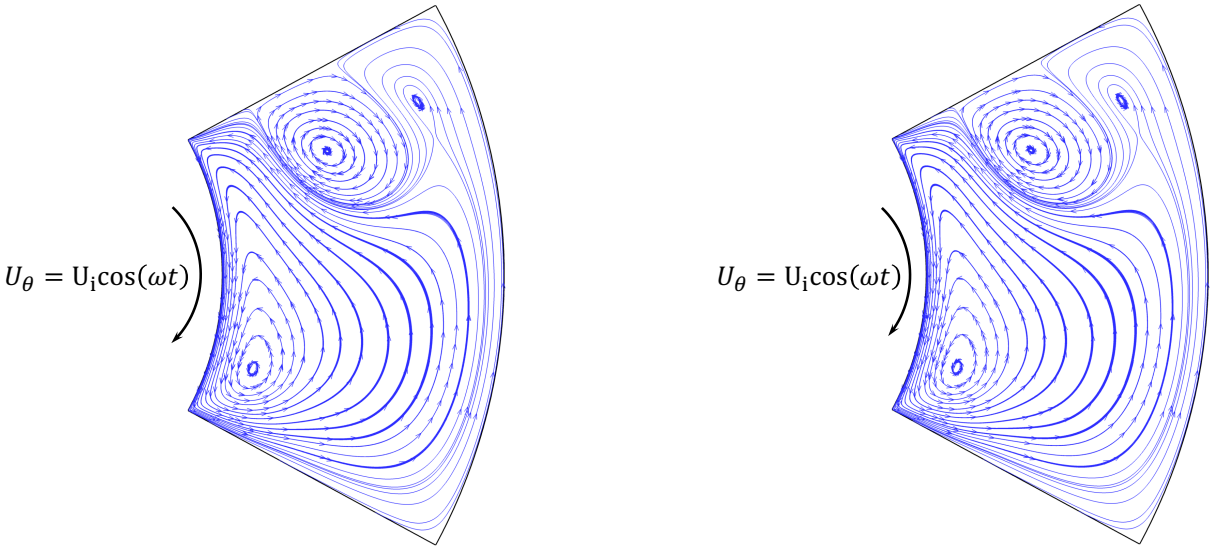
Runge-Kutta (RK2) depends on the viscous term and Reynolds number. RK methods are also configurable; algorithm coefficients can be chosen to achieve specific orders of accuracy for the solution.

Sanderse and Koren (SK) [4] developed a framework for solving the incompressible NS equations using explicit RK methods and the pressure-projection method. This formulation necessitates that the Poisson equation for pressure is solved at every sub-step of the RK method. For RK3, this means that the Poisson equation is solved three times for every time step. Iterative methods are needed to solve this equation and despite improvements to the convergence speed of these methods (by use of FFTs, decompositions, etc.), solving the Poisson equation is by far the most time-intensive step in simulating incompressible flows. Thus, it is expeditious to modify the algorithm and minimize the number of Poisson solves necessary to maintain third-order accuracy.

1.2 The case for pressure extrapolation

Aithal and Ferrante [1] developed FastRK3, a modification of the standard RK3 method [4] which only solves the Poisson equation once per time step. Rather than solving the Poisson equation for pressure in the first two sub-steps of RK3, the values for pressure at these steps are linearly extrapolated in time from values of pressure at previous time steps. Figs. 1.1, 1.2, and 1.3 show results from Aithal and Ferrante [1] which indicate that the numerical solution of the unsteady circular lid-driven cavity for the standard RK3 method and FastRK3 method are nearly identical for both velocity and pressure. The temporal accuracy of FastRK3 was not studied by Aithal and Ferrante.

Capuano et al. [7] tested a linear pressure extrapolation and a pressure copy method with a third-order RK scheme and found that the pressure extrapolation method was able to maintain third-order accuracy. Most recently, Michele et al. [5] performed a parametric study, testing various linear extrapolations with different RK methods. In this study, extrapolations are constructed as functions of parameters α and β. The authors found through numerical tests that for the classical fourth-order RK scheme (RK4) there is a family of



(a) Streamlines of the numerical solution for $\mathrm{Re}_i = 280$ from standard RK3 at $t_{\max} = 40$

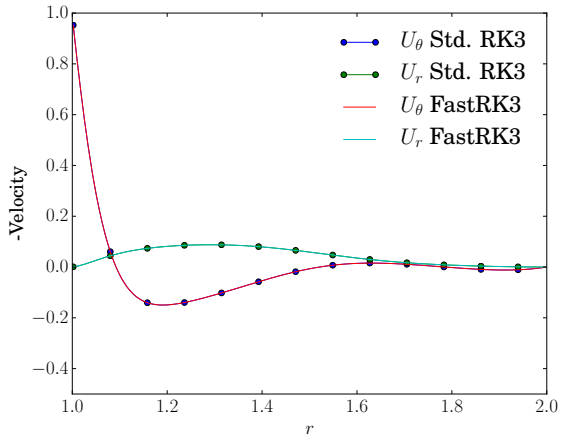
(b) Streamlines of the numerical solution for $\mathrm{Re}_i = 280$ from FastRK3 at $t_{\max} = 40$

Figure 1.1: Lid-driven polar cavity flow after four rotations of the inner wall. [1]

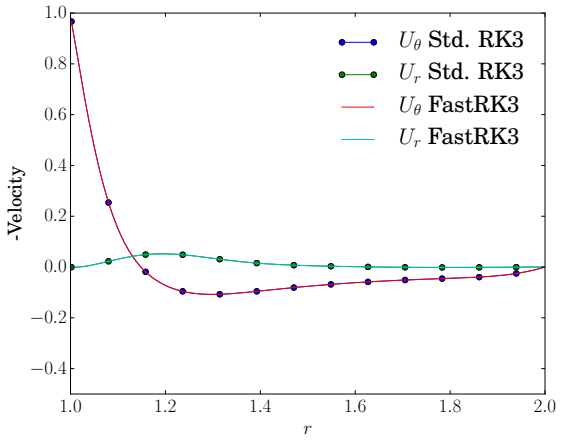
extrapolations which follow the relationship:

$$\beta = -\frac{2}{3}\alpha + \frac{5}{6} \tag{1.1}$$

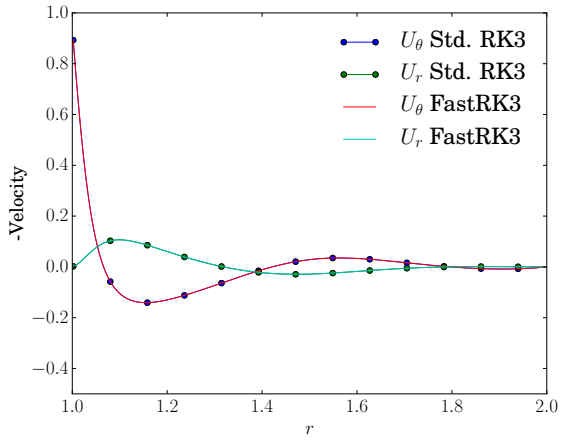
and give a third-order accurate result. Fig. 1.4, which shows the velocity errors for the tests, indicates that error is minimized for extrapolations which fall on the line in Eq. 1.1 and increase as distance from this line increases. From these works it is clear that achieving third-order accuracy is possible without solving the Poisson equation at every sub-step. However, an analytical proof that Eq. 1.1 makes the scheme third-order accurate in time has not been provided.



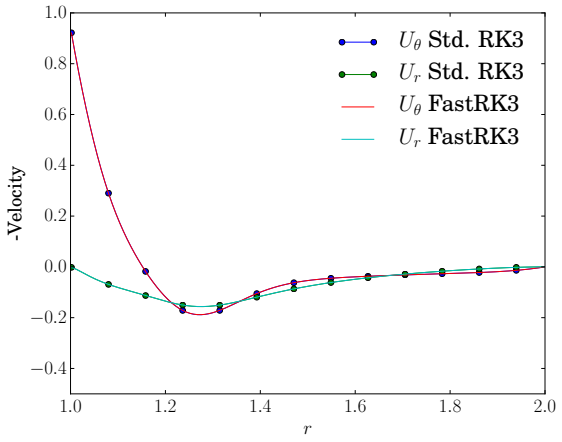
(a) $\theta = \pi/18$, $\mathrm{Re}_i = 280$



(b) $\theta = -\pi/18$, $\mathrm{Re}_i = 280$

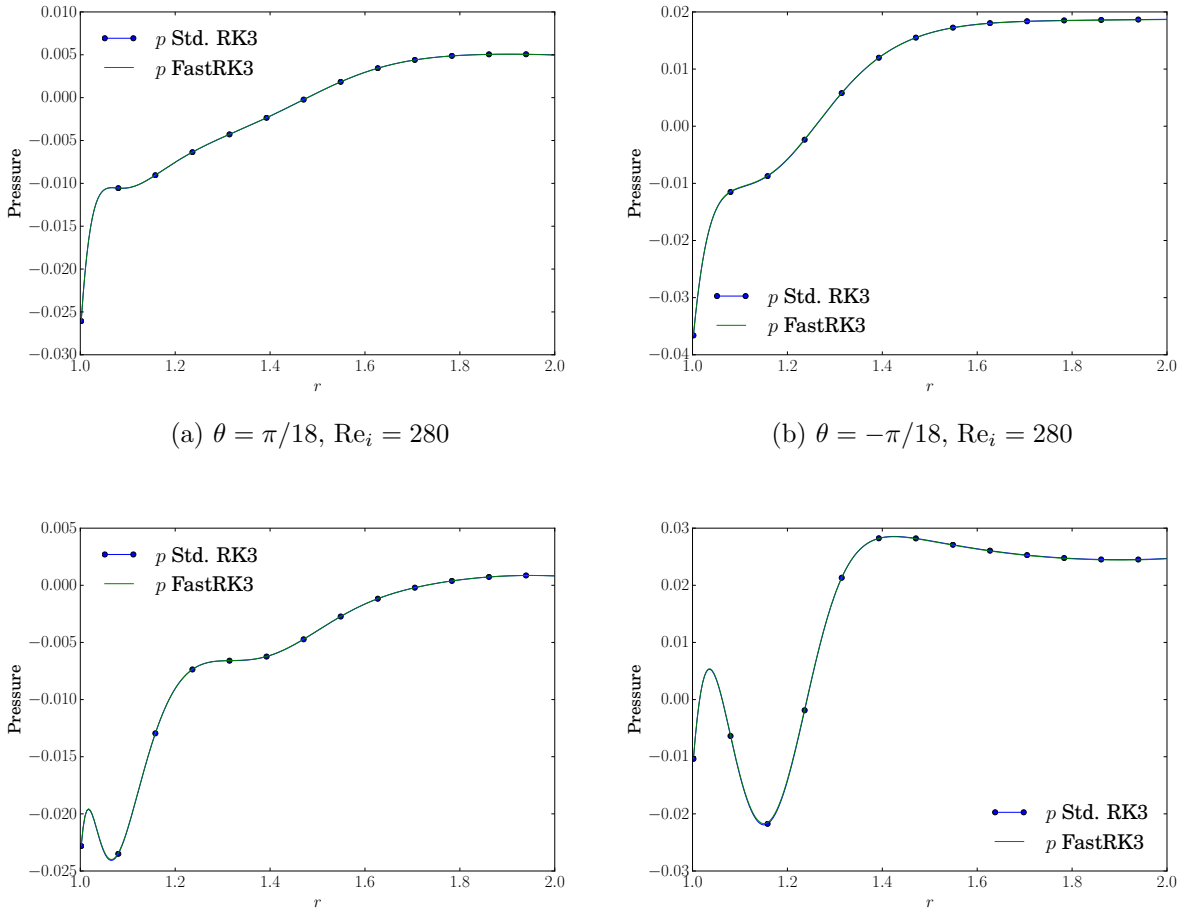


(c) $\theta = \pi/9$, $\mathrm{Re}_i = 280$



(d) $-\theta = \pi/9$, $\mathrm{Re}_i = 280$

Figure 1.2: Comparison of FastRK3 and standard RK3 solutions for U_θ and U_r along radial lines for the lid-driven polar cavity at $t_{\max} = 40$ (after four rotations of the inner wall). [1]



(a) $\theta = \pi/18$, $\mathrm{Re}_i = 280$

(b) $\theta = -\pi/18$, $\mathrm{Re}_i = 280$

(c) $\theta = \pi/9$, $\mathrm{Re}_i = 280$

(d) $-\theta = \pi/9$, $\mathrm{Re}_i = 280$

Figure 1.3: Comparison of FastRK3 and standard RK3 solutions for p along radial lines for the lid-driven polar cavity at $t_{\max} = 40$ (after four rotations of the inner wall). [1]

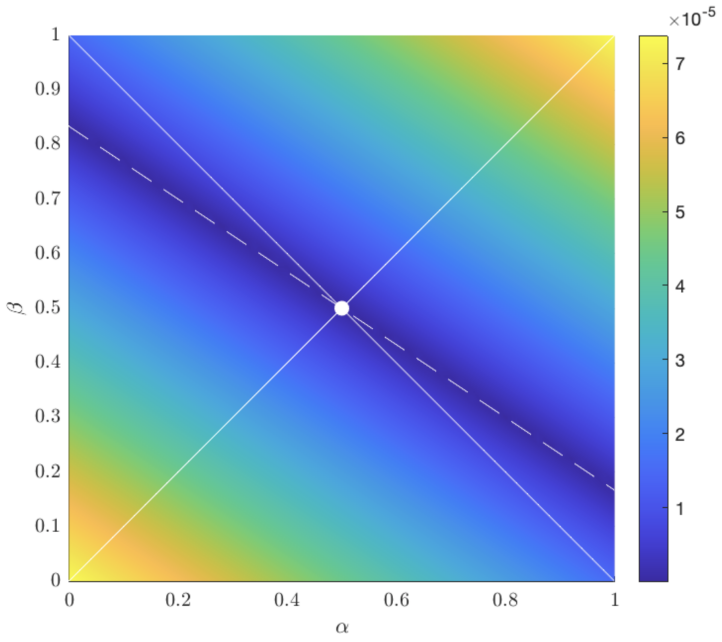


Figure 1.4: Magnitude of the L_∞ norm of the error in velocity in the $\alpha - \beta$ plane using the RK4 method. The dashed line, representing $\beta = -2/3\alpha + 5/6$, describes the set of most accurate methods. [5]

1.3 Goals

The objectives of this thesis are:

1. to provide an analytical proof for Eq. (1.1) in order for the extrapolation of pressure of a FastRK3 method to be third-order accurate in time,

2. to prove numerically that FastRK3 is third-order accurate in time when the extrapolation is performed while satisfying the constraint of Eq. (1.1).

Chapter 2

MATHEMATICAL DESCRIPTIONS

2.1 RK methods

Runge-Kutta (RK) methods are multi-step methods which use quadrature to numerically integrate differential equations. For the differential equation:

$$y' = f(y) \quad \text{where} \quad y = g(x), \tag{2.1}$$

the general RK form is:

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij} Y'_j \quad \text{for} \quad i = 1, \ldots, s, \tag{2.2}$$

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i Y'_i, \tag{2.3}$$

where

$$Y'_i = f(Y_i). \tag{2.4}$$

In this formulation h is the step-size in x, a_{ij} and b_i are RK coefficients, s is the number of internal stages, y_n is the approximated solution at x_n, and Y_i and Y'_i are approximations to y and y' at $x_n + c_i h$.

RK methods can be made to be explicit if

$$a_{ij} = 0 \quad \text{for} \quad i \leq j. \tag{2.5}$$

In this case, the method simplifies to

$$\check{Y}_i = y_n + h \sum_{j=1}^{i} \check{a}_{ij} \check{Y}'_{j-1} \quad \text{for} \quad i = 1, \ldots, s, \tag{2.6}$$

$$y_{n+1} = \check{Y}_s, \tag{2.7}$$

where $\check{Y}_i$ and $\check{Y}_i'$ are approximations to y and y' at $x_n + \check{c}_i h$ and

$$\check{\mathbf{A}} = \begin{bmatrix} a_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{s1} & \cdots & a_{s,s-1} & 0 \\ b_1 & \cdots & b_{s-1} & b_s \end{bmatrix} \quad \check{\mathbf{c}} = \begin{bmatrix} c_2 \\ \vdots \\ c_s \\ 1 \end{bmatrix}.$$

Components $\check{a}_{ij}$ and $\check{c}_i$ are the shifted RK coefficients.

An extensive analysis of the accuracy of these methods when applied to differential-algebraic systems of equations (e.g. the spatially-discretized incompressible Navier-Stokes equations) is presented in [6] and [10]. The details are beyond the scope of this work, but the main principle is that algorithm coefficients and number of steps must satisfy certain constraints in order to achieve desired orders of accuracy.

2.2 RK3 method for incompressible flows

2.2.1 Incompressible NS equations

We begin with the conservative form of the incompressible Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.8}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u}. \tag{2.9}$$

The variables are flow velocity $\mathbf{u}$, pressure p, fluid density ρ, and fluid kinematic viscosity ν. These equations are then discretized in space, in this case using a second-order central difference scheme. The discretized equations are:

$$DU + b(t) = 0, \tag{2.10}$$

$$\frac{dU}{dt} = F(U) - \frac{1}{\rho} Gp, \tag{2.11}$$

where D, F, and G are the discrete divergence, flux, and gradient operators, respectively, $U(t)$ is the discretized velocity field, $p(t)$ is the discretized pressure, and $b(t)$ accounts for

unsteady boundary conditions. The flux term is defined as:

$$F(U) = -C(U) + VU, \tag{2.12}$$

where C is the discrete convective operator and V is the discrete viscous operator.

The main distinguishing characteristic of incompressible flows is that the continuity equation simplifies to a divergence free constraint on velocity. This condition is typically satisfied by first ignoring the pressure term in the velocity evolution equation (Eq. 2.11) and calculating an intermediate velocity, U^*, only based on the contribution to velocity evolution from fluxes. Then the velocity is projected to a divergence free field by solving the Poisson equation for pressure:

$$Lp_{n+1} = \frac{\rho}{\Delta t} DU^*, \tag{2.13}$$

where L is the discrete Laplacian operator, p_{n+1} is the pressure at t_{n+1}, and Δt is the solver time step. This solution for pressure results in a divergence free velocity field U_{n+1} at t_{n+1}:

$$U_{n+1} = U^* - \frac{\Delta t}{\rho} Gp_{n+1}. \tag{2.14}$$

This procedure is called the projection method [3].

Sanderse and Koren (SK) [4] developed a method for solving the incompressible Navier-Stokes equations using explicit RK methods and the projection method. The formulation begins by applying the projection to the discretized NS equations (Eqs. 2.10 and 2.11) and reducing these equations to a single equation. The projection is handled as follows. Differentiating Eq. 2.10 and substituting into the divergence of Eq. 2.11:

$$\frac{1}{\rho} Lp = DF + \dot{b}(t). \tag{2.15}$$

In this step it is assumed that D is not a function of time (i.e. the mesh is not time-varying). Note the similarity between Eq. 2.13 and Eq. 2.15; this is the Poisson equation for pressure. Solving this equation for p and substituting into Eq. 2.11 gives:

$$\frac{dU}{dt} = PF - \rho G L^{-1} \dot{b}(t), \tag{2.16}$$

where

$$P = I - \rho G L^{-1} D \tag{2.17}$$

is the projection operator which projects the velocity onto a divergence-free field. Now that the NS equations have been combined into a single vector evolution equation (Eq. 2.16), we can apply the RK formulation to solve it.

2.2.2 Applying the RK formulation

Applying the RK formulation to Eq. 2.16 yields

$$U_i = u_n + \Delta t \sum_{j=1}^{i} \check{a}_{ij} P F_{j-1} - G L^{-1}(b(t_i) - b(t_n)), \quad i = 1, \ldots, s, \tag{2.18}$$

where the subscript i denotes the stage of the method and s is the total number of stages. The inverse Laplace operator adds complication in solving for U_i so instead of using Eq. 2.18 in implementation, the expression for P (Eq. 2.17) is substituted:

$$U_i = u_n + \Delta t \sum_{j=1}^{i} \check{a}_{ij} F_{j-1} - G L^{-1} \left(\Delta t \sum_{j=1}^{i} \check{a}_{ij} D F_{j-1} + (b(t_i) - b(t_n)) \right). \tag{2.19}$$

A pseudo-pressure ϕ_i is introduced and the expression simplifies to:

$$U_i = u_n + \Delta t \sum_{j=1}^{i} \check{a}_{ij} F_{j-1} - \check{c}_i \Delta t G \phi_i, \tag{2.20}$$

where

$$L\phi_i = \sum_{j=1}^{i} \frac{1}{\check{c}_i} \check{a}_{ij} D F_{j-1} + \frac{b(t_i) - b(t_n)}{\check{c}_i \Delta t}. \tag{2.21}$$

This equation for ϕ_i is solved using a Poisson solver, e.g., the FastPoc method of Aithal and Ferrante [1]. When comparing Eq. 2.21 with the exact solution for p_i:

$$L p_i = D F_i + \dot{b}(t_i), \tag{2.22}$$

it is clear by inspection that

$$\phi_i = p_i + \mathcal{O}(\Delta t). \tag{2.23}$$

This is because

$$\dot{b}(t_i) = \frac{b(t_i) - b(t_n)}{\check{c}_i \Delta t} + \mathcal{O}(\Delta t), \tag{2.24}$$

and the summation term in Eq. 2.21 is simply an approximation of the mean value of DF over the interval $[t_n, t_i]$, which is by definition a first-order approximation of DF_i (in Eq. 2.22). Additionally, an exact solution for $\phi(t_i)$ can be constructed as the mean value of pressure over the interval $[t_n, t_i]$:

$$\phi(t_i) = \frac{1}{\check{c}_i \Delta t} \int_{t_n}^{t_i} p(t) dt. \tag{2.25}$$

Sanderse and Koren [4] devise a method to achieve a second-order accurate approximation for pressure, using the RK coefficients $\check{\mathbf{A}}$ and $\check{\mathbf{c}}$ to construct a linear combination of ϕ_i:

$$\boldsymbol{\psi} = (\check{\mathbf{A}})^{-1} \mathrm{diag}(\check{c}_1, \ldots, \check{c}_s) \boldsymbol{\phi}, \tag{2.26}$$

$$p_{n+1} = \psi_3. \tag{2.27}$$

They then derive the order conditions necessary to reach third-order accuracy in velocity and second-order accuracy in pressure with a three-step method using the formulation presented in Eqs. 2.20, 2.21, 2.26, and 2.27. These conditions are satisfied by a single set of coefficients:

$$\check{\mathbf{A}} = \begin{bmatrix} 1/3 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & 3/4 & 1/4 \end{bmatrix} \quad \check{\mathbf{c}} = \begin{bmatrix} 1/3 \\ 1 \\ 1 \end{bmatrix}.$$

These coefficients, along with Eqs. 2.20, 2.21, 2.26, and 2.27 comprise standard RK3.

Sanderse and Koren [4] also show that with the coefficients given above, ϕ_i will be first-order approximations of the exact expression for $\phi(t_i)$:

$$\phi_i = \phi(t_i) + \mathcal{O}(\Delta t), \tag{2.28}$$

in addition to being first-order approximations of p_i. By the principle of the midpoint method, however, ϕ_i is a second-order approximation to both $\phi(t)$ and $p(t)$ at the midpoints in time, $t_n + \check{c}_i \frac{\Delta t}{2}$.

2.3 FastRK3

FastRK3 [1] and the fast-projection methods of [7] and [5] modify this formulation by calculating ϕ_i for $i = 1, \ldots, s-1$ by linear extrapolation rather than solving the Poisson equation, which is typically the most time-consuming component of solving an incompressible flow. Linear extrapolation takes the form:

$$\phi(t) \approx \phi(\hat{t}_{n-1}) + \frac{t - \hat{t}_{n-1}}{\hat{t}_n - \hat{t}_{n-1}} (\phi(\hat{t}_n) - \phi(\hat{t}_{n-1})), \tag{2.29}$$

where $\phi(t)$ is the exact solution for ϕ at some time t and $\phi(\hat{t}_n)$ and $\phi(\hat{t}_{n-1})$ are values of $\phi(t)$ from two previous points in time. Linear extrapolation is by nature second-order accurate so Eq. 2.29 is a second-order approximation of $\phi(t)$. In general, we do not have the exact solutions for $\phi(\hat{t}_n)$ and $\phi(\hat{t}_{n-1})$ but rather have the approximations ϕ_n and ϕ_{n-1}. As previously noted, ϕ_i are second-order approximations for the exact $\phi(t)$ at $t_n + \check{c}_i \frac{\Delta t}{2}$. In order to maintain this level of accuracy, we set

$$\hat{t}_n = t_n - \frac{\Delta t}{2}, \tag{2.30}$$

$$\hat{t}_{n-1} = t_{n-1} - \frac{\Delta t}{2}, \tag{2.31}$$

where ϕ_n and ϕ_{n-1} are second-order approximations of the exact $\phi(t)$, and extrapolate to

$$t = t_n + \check{c}_i \frac{\Delta t}{2}. \tag{2.32}$$

Thus, the extrapolations take the form:

$$\phi_i = \left(1 + \frac{\check{c}_i + 1}{2}\right) \phi_n - \left(\frac{\check{c}_i + 1}{2}\right) \phi_{n-1}, \tag{2.33}$$

and ϕ_i maintain second-order accuracy with respect to the exact $\phi(t)$ at the midpoints in time. Note that this extrapolation uses information from a previous point in time and is therefore not self-starting.

2.3.1 The α, β relationship

Theoretically, maintaining the accuracy of the ϕ_i variables by using the extrapolation in Eq. 2.33 is necessary to attain third-order accuracy. However, it's been shown that multiple sets of extrapolation coefficients can produce third-order results. To explain this, we adopt the parameterized form of the extrapolation from [5]:

$$\phi_i = (1 + \beta + \alpha\check{c}_i)\phi_n - (\beta + \alpha\check{c}_i)\phi_{n-1}. \tag{2.34}$$

The parameters α and β dictate where ϕ_i are placed in time as follows:

$$\hat{t}_i = t_n + \alpha\check{c}_i\Delta t, \quad i = 1, \ldots, s-1, \tag{2.35}$$

$$\hat{t}_n = t_n - \beta\Delta t. \tag{2.36}$$

The general extrapolation given in Eq. 2.34 has error:

$$\epsilon_i = \phi_i - \phi(t_i) = \left[\left(\beta - \frac{1}{2}\right) + \check{c}_i\left(\alpha - \frac{1}{2}\right)\right]\phi_n - \left[\left(\beta - \frac{1}{2}\right) + \check{c}_i\left(\alpha - \frac{1}{2}\right)\right]\phi_{n-1} + \mathcal{O}(\Delta t^2) \tag{2.37}$$

Imposing the constraint $\sum_{i=1}^{s-1}\epsilon_i = \mathcal{O}(\Delta t^2)$ gives:

$$(s-1)\left(\beta - \frac{1}{2}\right) + \left(\alpha - \frac{1}{2}\right)\sum_{i=1}^{s-1}\check{c}_i = 0 \tag{2.38}$$

For the RK4 method used in [5] this simplifies to:

$$\beta = -2/3\alpha + 5/6 \tag{2.39}$$

which is Eq. 1.1, the relationship given in [5]. Note that the relationship is the same for the RK3 method with coefficients given by [4].

2.3.2 Proof of accuracy

To understand how a second-order cumulative error of the extrapolations leads to third-order accuracy, we consider the mechanics of the RK3 method. For this investigation we

consider FastRK3 to be an approximation to standard RK3. To differentiate between the two algorithms, standard RK3 variables will be denoted with a tilde ($\tilde{}$).

In proceeding, we assume that all analysis done on the ϕ_i extrapolations is applicable to extrapolations of $G\phi_i$, as G is simply a linear operator and should not affect the accuracy of extrapolations. We also assume that ϕ_1 and ϕ_2 calculated via Poisson solve can be approximated to second-order by some extrapolation which satisfies 1.1. (See Appendix A for numerical verification of this claim.) This assumption means that the errors of $\tilde{\phi}_1$ and $\tilde{\phi}_2$ with respect to the analytical $\phi(t)$ will also sum to second-order and leads to the following:

$$G\tilde{\phi}_1 - G\phi_1 = c_\phi \Delta t + \mathcal{O}(\Delta t^2), \tag{2.40}$$

$$G\tilde{\phi}_2 - G\phi_2 = -c_\phi \Delta t + \mathcal{O}(\Delta t^2). \tag{2.41}$$

First stage of RK3

The first stage ($i = 1$) of standard RK3 is given by

$$\begin{aligned} \frac{\tilde{U}_1^* - \tilde{U}_n}{\Delta t} &= \check{a}_{11}\tilde{F}_n, \\ \frac{\tilde{U}_1 - \tilde{U}_1^*}{\Delta t} &= -\check{c}_1 \frac{G\tilde{\phi}_1}{\rho}, \end{aligned} \tag{2.42}$$

where, $\tilde{U}_n$ and $\tilde{U}_1$ are the arrays of discretized velocity components at t_n and $t_1 = t_n + \check{c}_1 \Delta t$, respectively, and $\tilde{U}_1^*$ is the array of the approximate velocity at t_1. $\tilde{F}_n$ represents the sum of discretized convective and diffusive terms at t_n, and, $\tilde{\phi}_1$ is the array of the discretized pressure-like scalar at $t = t_n + \frac{\check{c}_1}{2}\Delta t$. Note that $\tilde{U}_1$ differs from the exact velocity, $\bar{U}_1$, at t_1 by $\mathcal{O}(\Delta t^2)$ [9]:

$$\tilde{U}_1 = \bar{U}_1 + \mathcal{O}(\Delta t^2). \tag{2.43}$$

Summing the two equations of Eq. (2.42) gives

$$\frac{\tilde{U}_1 - \tilde{U}_n}{\Delta t} = \check{a}_{11}\tilde{F}_n - \check{c}_1 \frac{G\tilde{\phi}_1}{\rho}. \tag{2.44}$$

In standard RK3, in order to solve Eq. (2.44) for $\tilde{U}_1$, the pressure $\tilde{\phi}_1$ is obtained by solving the Poisson equation

$$DG\tilde{\phi}_1 = \frac{\rho}{\check{c}_1 \Delta t}\left(D\tilde{U}_1^* + b_1\right), \tag{2.45}$$

where $b_1 = b(t_1)$. Instead, in our method, FastRK3, in order to save computational time, we use an approximate array for the pressure $\tilde{\phi}_1$ at $t = t_n + \frac{\check{c}_1}{2}\Delta t$ (i.e., $\phi_1 = \phi_1(\tilde{\phi}_n, \tilde{\phi}_{n-1})$), such that we compute the velocity array U_1 at time t_1, by solving

$$\frac{U_1 - \tilde{U}_n}{\Delta t} = \check{a}_{11}\tilde{F}_n - \check{c}_1 \frac{G\phi_1}{\rho}. \tag{2.46}$$

Subtracting Eq. (2.44) from Eq. (2.46), we get

$$\frac{U_1 - \tilde{U}_1}{\Delta t} = \frac{\check{c}_1}{\rho}\left(G\tilde{\phi}_1 - G\phi_1\right). \tag{2.47}$$

Therefore, the instantaneous error $(U_1 - \tilde{U}_1)$ depends directly on how well ϕ_1 is approximating $\tilde{\phi}_1$. Substituting Eq. (2.40) in the above equation, we get

$$U_1 - \tilde{U}_1 = \frac{\check{c}_1}{\rho}\left[c_\phi \Delta t^2 + \mathcal{O}(\Delta t^3)\right] \tag{2.48}$$

Eq. (2.48) shows that the velocity U_1 is a second-order approximation of $\tilde{U}_1$ which, given Eq. (2.43), implies that U_1 is still a second-order approximation of the exact velocity $\bar{U}_1$. We now show that the discretized divergence of velocity DU_1 is a third-order approximation of $D\tilde{U}_1$. Applying the discrete divergence operator D to Eq. (2.44) gives

$$\frac{D\tilde{U}_1 - D\tilde{U}_n}{\Delta t} = \check{a}_{11} D\tilde{F}_n - \check{c}_1 \frac{DG\tilde{\phi}_1}{\rho}, \tag{2.49}$$

where, DG is the discrete Laplacian operator. Applying the discrete divergence operator D to Eq. (2.46) gives

$$\frac{DU_1 - D\tilde{U}_n}{\Delta t} = \check{a}_{11} D\tilde{F}_n - \check{c}_1 \frac{DG\phi_1}{\rho}. \tag{2.50}$$

Subtracting Eq. (2.49) from Eq. (2.50) and using Eq. (2.40) gives

$$\begin{aligned}
\frac{DU_1 - D\tilde{U}_1}{\Delta t} &= \frac{\check{c}_1}{\rho}\left(DG\tilde{\phi}_1 - DG\phi_1\right), \\
\frac{DU_1 - D\tilde{U}_1}{\Delta t} &= \frac{\check{c}_1}{\rho}\left(D\left[G\tilde{\phi}_1 - G\phi_1\right]\right), \\
\frac{DU_1 - D\tilde{U}_1}{\Delta t} &= \frac{\check{c}_1}{\rho}\left(D\left[c_\phi \Delta t + \mathcal{O}(\Delta t^2)\right]\right),
\end{aligned} \tag{2.51}$$

and, thus,

$$DU_1 - D\tilde{U}_1 = \mathcal{O}(\Delta t^2), \tag{2.52}$$

Eq. (2.52) shows that DU_1 is an approximation of $D\tilde{U}_1$ with second order accuracy. Next, we compare the fluxes computed using standard RK3 and FastRK3 as

$$F_1 - \tilde{F}_1 = [C(\tilde{U}_1) - C(U_1)] + VU_1 - V\tilde{U}_1. \tag{2.53}$$

Since $C(U) \sim U^2$, the leading order Δt term is in the diffusive term. Substituting Eq. (2.40) in Eq. (2.53), we get

$$F_1 - \tilde{F}_1 = V\frac{\check{c}_1}{\rho}\left[c_\phi \Delta t^2 + \mathcal{O}(\Delta t^3)\right] + \left[C(\tilde{U}_1) - C(U_1)\right]. \tag{2.54}$$

Second stage of RK3

Next, the second stage ($i = 2$) of standard RK3 is given by

$$\frac{\tilde{U}_2 - \tilde{U}_n}{\Delta t} = \check{a}_{21}\tilde{F}_n + \check{a}_{22}\tilde{F}_1 - \check{c}_2\frac{G\tilde{\phi}_2}{\rho}, \tag{2.55}$$

where $\tilde{U}_2$ is the discretized velocity at $t_2 = t_n + \check{c}_2\Delta t$. $\tilde{F}_1$ represents the combined discretized convective and diffusive terms at t_1 computed using $\tilde{U}_1$, and $\tilde{\phi}_2$ is the array of discretized pressure at $t = t_n + \frac{\check{c}_2}{2}\Delta t$. Note that $\tilde{U}_2$ differs from the exact velocity $\bar{U}_2$ at t_2 by $\mathcal{O}(\Delta t^3)$ [9]:

$$\tilde{U}_2 = \bar{U}_2 + \mathcal{O}(\Delta t^3). \tag{2.56}$$

Similarly to the first stage of standard RK3, the second stage would also require solving the Poisson equation for pressure to obtain $\tilde{\phi}_2$. Instead we use an approximate discretized array for ϕ_2 at $t = t_n + \frac{\check{c}_2}{2}\Delta t$ (i.e., $\phi_2 = \phi_2(\tilde{\phi}_n, \tilde{\phi}_{n-1})$), such that we obtain the velocity array U_2 at time t_2, as

$$\frac{U_2 - \tilde{U}_n}{\Delta t} = \check{a}_{21}\tilde{F}_n + \check{a}_{22}\tilde{F}_1 - \check{c}_2\frac{G\phi_2}{\rho}, \tag{2.57}$$

where, F_1 represents the combined discretized convective and diffusive terms computed using U_1. Subtracting Eq. (2.55) from Eq. (2.57) results in

$$\frac{U_2 - \tilde{U}_2}{\Delta t} = \check{a}_{22}\left(F_1 - \tilde{F}_1\right) + \frac{\check{c}_2}{\rho}\left(G\tilde{\phi}_2 - G\phi_2\right). \tag{2.58}$$

Hence, the instantaneous error of the discretized velocity U_2 at t_2 is given by substituting Eqs. (2.54) and (2.41) in Eq. (2.58) as

$$U_2 - \tilde{U}_2 = \frac{\check{c}_2}{\rho}\left[-c_\phi \Delta t^2 + \mathcal{O}(\Delta t^3)\right] + \check{a}_{22} V \frac{\check{c}_1}{\rho}\left[c_\phi \Delta t^3 + \mathcal{O}(\Delta t^4)\right] + \check{a}_{22}\Delta t\left[C(\tilde{U}_1) - C(U_1)\right] \tag{2.59}$$

and, thus,

$$U_2 - \tilde{U}_2 = \mathcal{O}(\Delta t^2). \tag{2.60}$$

Further, similarly to the first stage of FastRK3 above, it can be shown that the accuracy of the discretized divergence of velocity U_2 is given by

$$DU_2 - D\tilde{U}_2 = \mathcal{O}(\Delta t^2). \tag{2.61}$$

Next, we compare the fluxes computed using standard RK3 and FastRK3 as

$$F_2 - \tilde{F}_2 = V(U_2 - \tilde{U}_2) + \left[C(\tilde{U}_2) - C(U_2)\right], \tag{2.62}$$

and substituting Eq. (2.59) into Eq. (2.62), we get

$$\begin{aligned} F_2 - \tilde{F}_2 = V\frac{\check{c}_2}{\rho}\left[-c_\phi \Delta t^2 + \mathcal{O}(\Delta t^3)\right] + VV\frac{\check{a}_{22}\check{c}_1}{\rho}\left[c_\phi \Delta t^3 + \mathcal{O}(\Delta t^4)\right] \\ + \left[C(\tilde{U}_2) - C(U_2)\right] + V\check{a}_{22}\Delta t\left[C(\tilde{U}_1) - C(U_1)\right]. \end{aligned} \tag{2.63}$$

Third stage of RK3

Finally, the third stage ($i = 3$) of standard RK3 is given by

$$\frac{\tilde{U}_{n+1} - \tilde{U}_n}{\Delta t} = \check{a}_{32}\tilde{F}_1 + \check{a}_{33}\tilde{F}_2 - \frac{G\tilde{\phi}_{n+1}}{\rho}, \tag{2.64}$$

where $\tilde{U}_{n+1}$ is the discretized velocity at $t_{n+1} = t_n + \Delta t$, and $\tilde{\phi}_{n+1}$ is the array of discretized pressure at $\hat{t}_{n+1} = t_n + \Delta t/2$. Note that the discretized velocity at the third stage of RK3, $\tilde{U}_{n+1}$, differs from the exact velocity, $\bar{U}_{n+1}$, at t_{n+1} by $\mathcal{O}(\Delta t^4)$ [9]:

$$\tilde{U}_{n+1} = \bar{U}_{n+1} + \mathcal{O}(\Delta t^4). \tag{2.65}$$

The third stage of FastRK3 is given by

$$\frac{U_{n+1} - \tilde{U}_n}{\Delta t} = \breve{a}_{32} F_1 + \breve{a}_{33} F_2 - \frac{G\phi_{n+1}}{\rho}, \tag{2.66}$$

where, F_2 represents the combined discrete convective and diffusive terms computed using U_2. It should be noted that we *do* solve the Poisson equation for pressure only once and for this final stage of FastRK3. The solution to the Poisson equation, Eq. (2.70), for pressure ϕ_{n+1}, varies from the solution $\tilde{\phi}_{n+1}$ of Eq. (2.69) in the interior grid points, while satisfying the prescribed boundary conditions. This is because the terms appearing in the Poisson equation, Eq. (2.70), are F_1 and F_2 rather than $\tilde{F}_1$ and $\tilde{F}_2$ as in Eq. (2.69). In order to compute the accuracy of the third stage of FastRK3, we first determine the instantaneous error $(\tilde{\phi}_{n+1} - \phi_{n+1})$ as shown below. Applying the discrete divergence operator D to Eqs. (2.64) and (2.66) gives, respectively,

$$\begin{aligned} \frac{D\tilde{U}_{n+1} - D\tilde{U}_n}{\Delta t} &= D\left(\breve{a}_{32}\tilde{F}_1 + \breve{a}_{33}\tilde{F}_2\right) - \frac{DG\tilde{\phi}_{n+1}}{\rho}, \\ \frac{DU_{n+1} - D\tilde{U}_n}{\Delta t} &= D\left(\breve{a}_{32}F_1 + \breve{a}_{33}F_2\right) - \frac{DG\phi_{n+1}}{\rho}, \end{aligned} \tag{2.67}$$

and, thus,

$$\begin{aligned} -\frac{b_{n+1} - b_n}{\Delta t} &= D\left(\breve{a}_{32}\tilde{F}_1 + \breve{a}_{33}\tilde{F}_2\right) - \frac{DG\tilde{\phi}_{n+1}}{\rho}, \\ -\frac{b_{n+1} - b_n}{\Delta t} &= D\left(\breve{a}_{32}F_1 + \breve{a}_{33}F_2\right) - \frac{DG\phi_{n+1}}{\rho}, \end{aligned} \tag{2.68}$$

where we have used the discretized continuity equation (2.10): $D\tilde{U}_n + b_n = 0$; $D\tilde{U}_{n+1} + b_{n+1} = 0$; $DU_{n+1} + b_{n+1} = 0$. Rearranging the terms in Eq. (2.68) gives the following Poisson equations for $\tilde{\phi}_{n+1}$ and ϕ_{n+1},

$$\frac{DG\tilde{\phi}_{n+1}}{\rho} = D\left(\breve{a}_{32}\tilde{F}_1 + \breve{a}_{33}\tilde{F}_2\right) + \frac{b_{n+1} - b_n}{\Delta t}. \tag{2.69}$$

$$\frac{DG\phi_{n+1}}{\rho} = D\left(\breve{a}_{32}F_1 + \breve{a}_{33}F_2\right) + \frac{b_{n+1} - b_n}{\Delta t}, \tag{2.70}$$

Subtracting Eq. (2.70) from Eq. (2.69), we get

$$\frac{DG\tilde{\phi}_{n+1}}{\rho} - \frac{DG\phi_{n+1}}{\rho} = D\left[\breve{a}_{32}(\tilde{F}_1 - F_1) + \breve{a}_{33}(\tilde{F}_2 - F_2)\right]. \tag{2.71}$$

Substituting Eqs. (2.54) and (2.63) into Eq. (2.73), we obtain

$$\frac{DG\tilde{\phi}_{n+1}}{\rho} - \frac{DG\phi_{n+1}}{\rho} = -D\left\{V\frac{\check{a}_{32}\check{c}_1}{\rho}\left[c_\phi\Delta t^2 + \mathcal{O}(\Delta t^3)\right] + V\frac{\check{a}_{33}\check{c}_2}{\rho}\left[-c_\phi\Delta t^2 + \mathcal{O}(\Delta t^3)\right]\right.$$
$$+VV\frac{\check{a}_{33}\check{a}_{22}\check{c}_1}{\rho}\left[c_\phi\Delta t^3 + \mathcal{O}(\Delta t^4)\right] + V\check{a}_{33}\check{a}_{22}\Delta t\left[C(\tilde{U}_1) - C(U_1)\right]$$
$$\left.+\check{a}_{32}\left[C(\tilde{U}_1) - C(U_1)\right] + \check{a}_{33}\left[C(\tilde{U}_2) - C(U_2)\right]\right\}, \tag{2.72}$$

$$\frac{DG\tilde{\phi}_{n+1}}{\rho} - \frac{DG\phi_{n+1}}{\rho} = -\frac{DVc_\phi}{\rho}\left[\Delta t^2\left(\check{a}_{32}\check{c}_1 - \check{a}_{33}\check{c}_2\right) + \mathcal{O}(\Delta t^3)\right]. \tag{2.73}$$

This result is generally second-order but goes to third-order under the constraint:

$$\check{a}_{32}\check{c}_1 = \check{a}_{33}\check{c}_2, \tag{2.74}$$

which holds for the RK3 coefficients given by SK [4]. Here, the condition that the first-order error introduced by linear extrapolation should sum to zero has become relevant. Therefore, imposing the constraint given in Eq. (2.74) we get

$$D\left(G\tilde{\phi}_{n+1} - G\phi_{n+1}\right) = \mathcal{O}(\Delta t^3). \tag{2.75}$$

Equation (2.75) shows that the error on pressure gradient introduced by FastRK3 (by computing the Poisson equation for pressure and using the linear extrapolation for stage 1 and 2 for ϕ_i) versus the standard RK3 (for which the Poisson equation is solved at each stage of RK3) is of third order.

The instantaneous error $(\tilde{U}_{n+1} - U_{n+1})$ is computed by subtracting Eq. (2.64) from Eq. (2.66) as

$$\frac{U_{n+1} - \tilde{U}_{n+1}}{\Delta t} = \check{a}_{32}\left(F_1 - \tilde{F}_1\right) + \check{a}_{33}\left(F_2 - \tilde{F}_2\right) + \frac{\left(G\tilde{\phi}_{n+1} - G\phi_{n+1}\right)}{\rho}, \tag{2.76}$$

and, then, by using Eqs. (2.54), (2.63), (2.74)and (2.75) in Eq. (2.76) gives

$$\frac{U_{n+1} - \tilde{U}_{n+1}}{\Delta t} = \mathcal{O}(\Delta t^3), \tag{2.77}$$

to finally give

$$U_{n+1} - \tilde{U}_{n+1} = \mathcal{O}(\Delta t^4). \tag{2.78}$$

Eq. (2.78) shows that U_{n+1} is a fourth-order accurate approximation of $\tilde{U}_{n+1}$, which given Eq. (2.65), implies that U_{n+1} is still a fourth-order accurate approximation of the exact velocity $\bar{U}_{n+1}$.

Thus far, we have derived the error of FastRK3 to advance the numerical solution of one time-step, and we now determine its global error at time T. In order to integrate the governing equations from time $t = 0$ to $t = T$, we need to perform time integration via FastRK3 $N = T/\Delta t$ times. Therefore, the resulting global error of FastRK3 at time T is the sum of the one time-step error for all N timesteps:

$$U_N - \bar{U}_N = \frac{T}{\Delta t}\, \mathcal{O}(\Delta t^4) = \mathcal{O}(\Delta t^3). \tag{2.79}$$

Therefore, we have shown that FastRK3 is third-order accurate in the global temporal error for velocity.

Building from the work of [5] we have shown that the set of extrapolations which have a cumulative second-order error will produce third-order accurate solutions for velocity, even if the extrapolations themselves are first order accurate. Thus, for any RK method with at least three stages there is some linear function $\beta(\alpha)$ for which third-order accuracy is achieved.

2.4 Temporal accuracy analysis

Due to the discretization in time of the partial differential equation (PDE) to be solved numerically, there is a local truncation error (LTE) of the numerical solution that is a function of Δt, the time step. The LTE can be found by taking the difference of the modified (PDE) (the result of discretization) from the Taylor-expansion of the exact solution. The form of the temporal LTE at each point in space within the computational domain is thus a power series in Δt where the leading order term determines the order of accuracy of the solution.

This can be summarized by saying that the error of our solution (solved with time step Δt) with respect to the analytical or exact solution takes the following form:

$$\boldsymbol{\epsilon}_{\Delta t} = A\Delta t^r + C, \tag{2.80}$$

where $\boldsymbol{\epsilon}$ is a matrix which contains the error values at every point in the domain, A is a matrix which contains weighting coefficients which depend on location within the domain, C is a matrix which contains higher order Δt terms, and r is the order of accuracy. The order r can then be calculated by comparing the error $\boldsymbol{\epsilon}$ for simulations run with varying time steps. In general, there is not a known analytical solution for a given flow so $\boldsymbol{\epsilon}$ cannot be directly calculated. Rather, solutions are compared against each other to calculate r. There are a few methods for which this strategy can work. Both of the methods described below will be used in analyzing the results in Ch. 3.

2.4.1 Method 1: comparing successive numerical solutions

In this method solutions are found for successively increasing the time steps by some constant factor; in this work we use a factor of 2. The smallest time step is denoted as Δt. We define $E_{m\Delta t,n\Delta t}$ as the L2 norm of the difference between solutions with time steps $m\Delta t$ and $n\Delta t$. This can also be written as

$$E_{m\Delta t,n\Delta t} = ||\boldsymbol{\epsilon}_{m\Delta t} - \boldsymbol{\epsilon}_{n\Delta t}||_2 \,, \tag{2.81}$$

where $\boldsymbol{\epsilon}_{m\Delta t}$ and $\boldsymbol{\epsilon}_{n\Delta t}$ take the form of Eq. 2.80. Note that $\boldsymbol{\epsilon}_{m\Delta t}$ and $\boldsymbol{\epsilon}_{n\Delta t}$ are not calculated directly but that the difference of the solutions is equivalent to the difference of errors $\boldsymbol{\epsilon}$. As C contains higher order terms of Δt, we can assume its contribution to the error is negligible for sufficiently small Δt. Furthermore, A is not a function of time step, and Δt, m, n, and r are all constant spatially. Thus, Eq. 2.81 reduces to

$$E_{m\Delta t,n\Delta t} = ||A||_2 \, \Delta t^r (m^r - n^r). \tag{2.82}$$

In order to isolate r, we introduce the ratio

$$\frac{E_{2m\Delta t,2n\Delta t}}{E_{m\Delta t,n\Delta t}} = 2^r. \tag{2.83}$$

Thus

$$r = \log\left(\frac{E_{2m\Delta t,2n\Delta t}}{E_{m\Delta t,n\Delta t}}\right) / \log(2), \tag{2.84}$$

where the ratio in Eq. 2.83 is directly calculated from the velocity solutions corresponding to simulation time steps $2m\Delta t$, $2n\Delta t$, $m\Delta t$,and $n\Delta t$. As previously mentioned, in this work we successively double the time step. In this case, we set $2n = m$ and order is calculated using time steps $4\Delta t$, $2\Delta t$, and Δt.

2.4.2 Method 2: comparing with a reference solution

In this method, we compare the numerical solution with a reference solution. Introducing the reference solution error:

$$\boldsymbol{\epsilon}^*_{\Delta t} = A^*\Delta t^{r^*} + C^*, \tag{2.85}$$

we redefine the parameter:

$$E_{m\Delta t,n\Delta t} = \left|\epsilon'_{m\Delta t} - \epsilon'_{n\Delta t}\right|, \tag{2.86}$$

where

$$\epsilon'_{\Delta t} = ||\boldsymbol{\epsilon}_{\Delta t} - \boldsymbol{\epsilon}^*_{\Delta t}||_2 \tag{2.87}$$

is the L2 norm of the test solution and reference solution:

$$\epsilon'_{\Delta t} = \left|\left|A\Delta t^r + C - A^*\Delta t^{r^*} - C^*\right|\right|_2. \tag{2.88}$$

As long as $r \leq r^*$ we can again impose the argument that C and C^* are negligible. If the reference solution does not have at least the same order of accuracy as the test solution then C^* is not negligible and this calculation is not a valid means of calculating order of accuracy.

If $r < r^*$ then $A^* \Delta t^{r^*}$ is also a higher order term and can be neglected. In this case Eq. 2.88 becomes:

$$\epsilon'_{\Delta t} = ||A||_2 \, \Delta t^r \quad \text{for} \quad r < r^*. \tag{2.89}$$

Otherwise,

$$\epsilon'_{\Delta t} = ||(A - A*)||_2 \, \Delta t^r \quad \text{for} \quad r = r^*. \tag{2.90}$$

In either case, we end with the same expression for r as in Eq. 2.84.

Chapter 3

RESULTS

3.1 Test setup

We set up a test to confirm numerically the following:

1. The α, β relationship (Eq. 1.1) gives the set of extrapolations for which the net error is second-order.

2. ϕ_1 and ϕ_2 calculated by solving the Poisson equation in standard RK3 can be approximated (to second-order) by a special case of the α, β extrapolation. This is the assumption made in Section 2.3.2.

3. FastRK3 is third-order accurate as long as the extrapolation follows the α, β relationship.

To do so, we will run simulations of the Taylor-Green vortex flow, for which an analytical solution exists. We will run these tests using standard RK3 and two different extrapolations for FastRK3; one which follows the α, β relationship, and one which does not.

3.1.1 Test extrapolations

In order to probe the details of the extrapolation we test two different formulations. The first method, herein called the endpoint extrapolation method (EP), sets $\alpha = 1$, $\beta = 0$. In this method ϕ_1, ϕ_2, and ϕ_{n+1} are chosen to be defined at $t = t_n + \check{c}_i \Delta t$, or the times at which

U_i are defined. The extrapolation is

$$\phi_1 = \frac{4}{3}\phi_n - \frac{1}{3}\phi_{n-1}, \tag{3.1}$$

$$\phi_2 = 2\phi_n - \phi_{n-1}. \tag{3.2}$$

This method does not satisfy the α, β relationship 1.1.

The second method sets $\alpha = \beta = \frac{1}{2}$, assuming that ϕ_1, ϕ_2, and ϕ_{n+1} are at $t = t_n + \frac{\check{c}_i}{2}\Delta t$. Thus, we call this the midpoint extrapolation method (MP). This was the extrapolation used by Capuano et al. [7] and it was one of the extrapolations studied by Michele et al. [5]. The extrapolation is

$$\phi_1 = \frac{5}{3}\phi_n - \frac{2}{3}\phi_{n-1}, \tag{3.3}$$

$$\phi_2 = 2\phi_n - \phi_{n-1}. \tag{3.4}$$

The MP extrapolation satisfies the α, β constraint so we expect that these tests will yield a third-order accurate solution for velocity.

3.1.2 Test case: Taylor-Green vortex

We will use the two-dimensional Taylor-Green vortex (TGV) test case [8]. This flow has an analytical solution for velocity and pressure:

$$\bar{u}(x, y, t) = -\omega_0 \frac{k_y}{k^2} \cos(k_x x) \sin(k_y y) \exp(-\nu k^2 t), \tag{3.5}$$

$$\bar{v}(x, y, t) = \omega_0 \frac{k_x}{k^2} \sin(k_x x) \cos(k_y y) \exp(-\nu k^2 t), \tag{3.6}$$

$$\bar{p}(x, y, t) = -\frac{\omega_0^2}{4k^2} \cos(2k_x x) \sin(2k_y y) \exp(-2\nu k^2 t), \tag{3.7}$$

where ω_0 is the initial maximum vorticity, k_x and k_y are the wave numbers in the x and y directions, $k^2 = k_x^2 + k_y^2$, and ν is the kinematic viscosity. This analytical solution will be useful in calculating the accuracy of the simulation results.

In the simulations, x and y range from $[0, 1]$, $k_x = k_y = 2\pi$, and $\omega_0 = 100$. The flow is simulated on a computational mesh of 128^2 grid points with Re = 200. The test

	dt	N_t	CFL (approx)
$32\Delta t$	$4.8e-4$	3646	0.5
$16\Delta t$	$2.4e-4$	7292	0.25
$8\Delta t$	$1.2e-4$	14,584	0.12
$4\Delta t$	$6.0e-5$	29,168	0.061
$2\Delta t$	$3.0e-5$	58,336	0.031
Δt	$1.5e-5$	116,672	0.015
$N_{x,y} = 128$, $Re = 200$, $t_{\max} = 1.75$			

Table 3.1: Temporal convergence test parameters.

parameters are detailed in Table 3.1. Simulations are run to $t_{\max} = 1.75$, at which point the maximum vorticity has decayed to about half its initial value. The smallest time step used is $\Delta t_{\min} = 1.5 \times 10^{-5}$, which corresponds to $N = 116,672$ time steps. For every algorithm, we test six time steps, successively increasing by a factor of two. This allows for calculation of convergence rates at different points in the stability region. For FastRK3 simulations, we run standard RK3 for the first time step and then switch to FastRK3. This is necessary as FastRK3 is not self-starting and needs ϕ_{n-1} to proceed.

3.2 Standard RK3 temporal accuracy

First, we establish the accuracy of the standard RK3 method [4]. Fig. 3.1 shows the stage-wise velocity accuracy for the standard RK3 algorithm, calculated by comparing with the analytical solution. As expected, the accuracy increases with each sub-step and the final velocity is third-order accurate.

In Section 2.3.2 we assumed that there is some extrapolation which follows the relationship given in 1.1 and approximates standard RK3 $G\tilde{\phi}_1$ and $G\tilde{\phi}_2$ to second order. In order to confirm this, we consider the evolution of $G\tilde{\phi}_1$, $G\tilde{\phi}_2$, and $G\tilde{\phi}_{n+1}$ at some point within

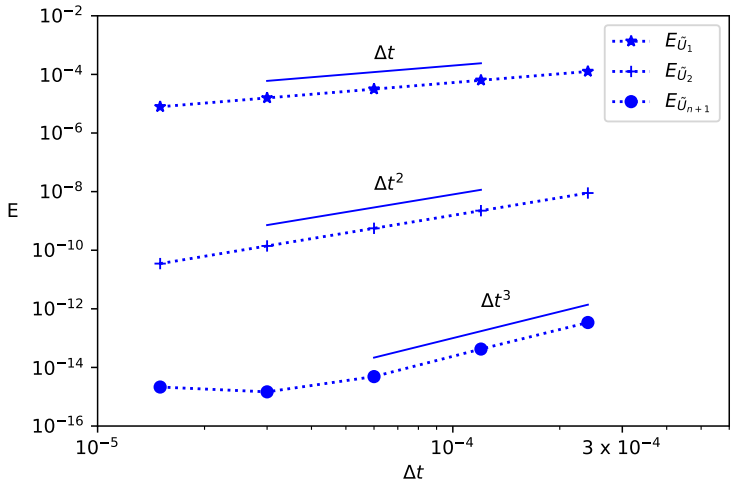


Figure 3.1: Temporal convergence of standard RK3 first, second, and third stage velocities.

the domain, shown in Fig. 3.2a. Notice that $G\tilde{\phi}_1$ and $G\tilde{\phi}_2$ have some constant offset from the curve for $G\tilde{\phi}_{n+1}$. One way this can be interpreted is to think of $G\tilde{\phi}_1$ and $G\tilde{\phi}_2$ as approximations to $G\tilde{\phi}_{n+1}$ at some $\hat{t}_1 = t_n + c_1'\Delta t$ and $\hat{t}_2 = t_n + c_2'\Delta t$, respectively. We can also generalize the placement in time of $G\tilde{\phi}_{n+1}$ at some $\hat{t}_{n+1} = t_n + c_3'\Delta t$. Adhering to the framework given in [5], c_i' are all functions of $\tilde{\alpha}$, $\tilde{\beta}$, and $\check{c}_i$.

Writing out these functions explicitly we have:

$$\hat{t}_1 = t_n + c_1'\Delta t = t_n + \tilde{\alpha}\check{c}_1\Delta t, \tag{3.8}$$

$$\hat{t}_2 = t_n + c_2'\Delta t = t_n + \tilde{\alpha}\check{c}_2\Delta t, \tag{3.9}$$

$$\hat{t}_{n+1} = t_n + c_3'\Delta t = t_{n+1} - \tilde{\beta}\Delta t, \tag{3.10}$$

where $\tilde{\alpha}$ and $\tilde{\beta}$ designate the specific α and β values that produce equivalent (to second-order) extrapolations to the $\tilde{\phi}_i$ found by Poisson solve for standard RK3. We can then define the offsets between $\hat{t}_1$ and $\hat{t}_2$ and $\hat{t}_n$:

$$\hat{dt}_1 = \hat{t}_1 - \hat{t}_n = (\tilde{\beta} + \tilde{\alpha}\check{c}_1), \tag{3.11}$$

$$\hat{dt}_2 = \hat{t}_2 - \hat{t}_n = (\tilde{\beta} + \tilde{\alpha}\check{c}_2). \tag{3.12}$$

These offsets are found numerically by shifting $G\tilde{\phi}_1$ and $G\tilde{\phi}_2$ in time and noting where the error with $G\tilde{\phi}_{n+1}$ is minimized. From Fig. 3.2b we can determine that

$$\hat{dt}_1 = \frac{1}{2}\Delta t, \tag{3.13}$$

$$\hat{dt}_2 = \frac{7}{6}\Delta t. \tag{3.14}$$

Substituting Eqs. 3.13 and 3.14 into Eqs. 3.11 and 3.12 and noting that $\check{c}_1 = \frac{1}{3}$ and $\check{c}_2 = 1$ we find that

$$\tilde{\alpha} = 1, \tag{3.15}$$

$$\tilde{\beta} = \frac{1}{6}, \tag{3.16}$$

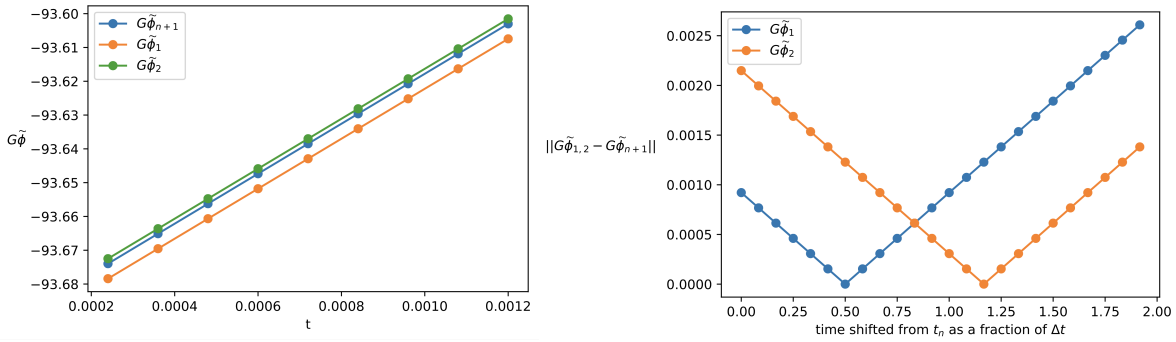
which do fall on the line described by Eq. 1.1. These values for α and β correspond to the following extrapolations:

$$\phi_1 = \frac{3}{2}\phi_n - \frac{1}{2}\phi_{n-1}, \tag{3.17}$$

$$\phi_2 = \frac{13}{6}\phi_n - \frac{7}{6}\phi_{n-1}, \tag{3.18}$$

which we will call the sweet spot (SS) extrapolations. Numerical verification that the SS extrapolations are second-order accurate relative to the standard RK3 solutions is provided in Appendix A.

Using these results we can visualize how the various ways of calculating $G\phi_1$ and $G\phi_2$ relate to each other. Fig. 3.3 shows the results of a Poisson solve and three different extrapolations along with the exact solution for $G\phi(t)$, calculated using Eqs. 2.25 and 3.7, at some point within the domain. The extrapolations are calculated using standard RK3 values, $G\tilde{\phi}_n$ and $G\tilde{\phi}_{n-1}$, and are representative of the values for $G\phi_1$ and $G\phi_2$ for the first time step after the switch from standard RK3 to FastRK3. Note how the MP extrapolations lie on the exact solution curves. This implies that both MP extrapolations are equivalent to the exact solution to first-order. The error of the MP extrapolation relative to the exact solution will thus sum to second-order as the errors of the extrapolations individually are second-order.



(a) Evolution of ϕ_1, ϕ_2, and ϕ_{n+1}.

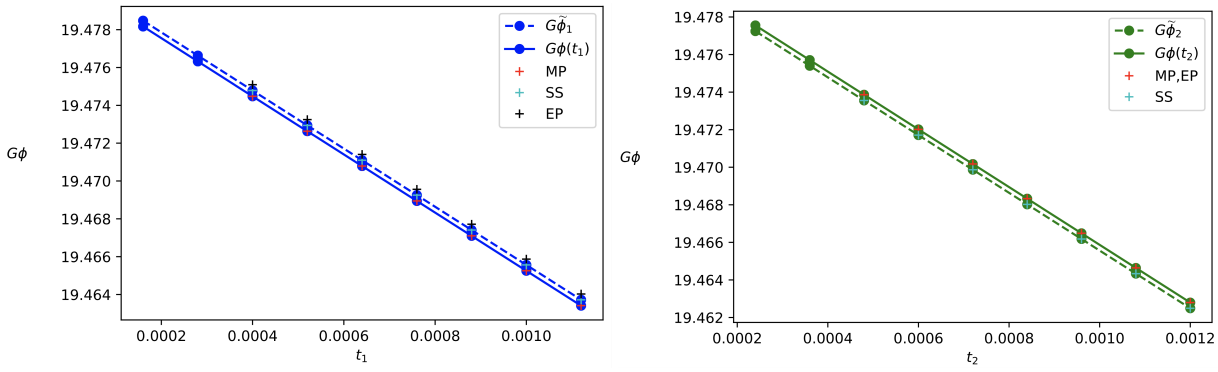
(b) Offsets in time

Figure 3.2: The relative positions of $G\tilde{\phi}_1$, $G\tilde{\phi}_2$, and $G\tilde{\phi}_{n+1}$ in time at a point within the computational domain.

Similarly, the SS extrapolations lie on the standard RK3 solution curves, supporting the hypothesis that SS extrapolations are equivalent to $G\tilde{\phi}_1$ and $G\tilde{\phi}_2$ to first-order. Furthermore, the offset between the exact/MP solutions and the standard RK3/SS solutions in the first stage seems to be equal and opposite to the offset in the second stage, implying that the first-order error between the MP extrapolations and the standard RK3 solutions should sum to zero. The same can be said of the errors between the SS extrapolations/standard RK3 solutions and the exact solution. The EP extrapolation has some first-order error relative to the exact solution for the first stage and coincides with the exact solution for the second stage. Thus, we can confirm there is some net first-order error across the two extrapolations.

3.3 FastRK3 temporal accuracy

We can see the effect of this first-order error as it propagates through the algorithm by comparing the stage-wise results of EP and MP. Fig. 3.4 shows the errors of the EP and MP methods when compared with standard RK3 as a reference solution. The extrapolations for EP and MP are both first-order. However, $G\phi_{n+1}$ for MP reaches third order while it is only second order for EP. This is the result of the first-order net error in the EP extrapolation,

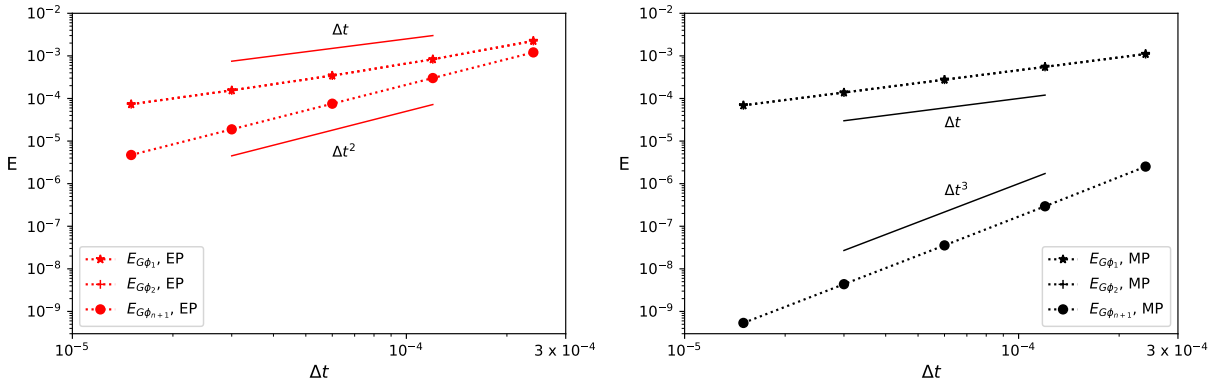


(a) ϕ_1 (b) ϕ_2

Figure 3.3: $G\phi$ evolution for standard RK3 and three different extrapolation methods, compared with the exact solution for G$\phi(t)$.

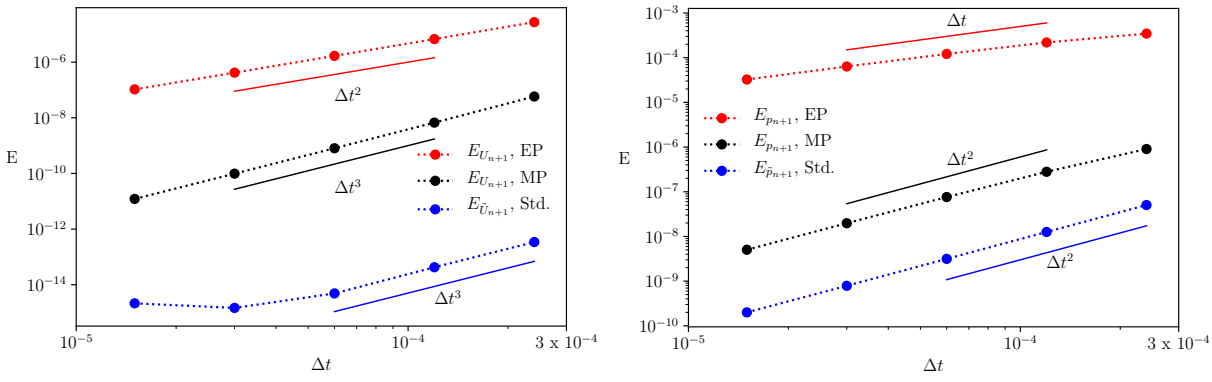
which limits the accuracy of $G\phi_{n+1}$.

Next, we consider the final result for velocity (shown in Fig. 3.5a). These errors are calculated by comparing with the analytical solution. From this plot, we can see that MP is third-order accurate while the EP is second-order. This is a direct result of the different $G\phi_{n+1}$ accuracies for these methods. Ultimately the net first-order error in the EP extrapolation prevents the solver from achieving third-order accuracy. We can also see the effect of the first-order error in EP extrapolation when considering the p_{n+1} results, calculated using Eqs. 2.26 and 2.27, and shown in Fig. 3.5b. These errors are calculated by comparing successive solutions. MP produces a second-order approximation for pressure while the first-order error in the EP extrapolation limits the accuracy of the pressure solution to first-order.



(a) Endpoint method (b) Midpoint method

Figure 3.4: Stage-wise $G\phi$ temporal accuracy for FastRK3 relative to standard RK3.



(a) velocity (b) pressure

Figure 3.5: Final stage temporal accuracy for FastRK3 EP, FastRK3 MP, and standard RK3.

Chapter 4

SUMMARY

In this work, we have provided an analytical analysis for the temporal accuracy of FastRK3, a numerical method for solving incompressible flows. In the process, we analytically derived the set of extrapolations (Eq. 1.1) that will yield third-order accurate results, which had previously only been found numerically [5]. We also generalized this derivation to any RK method with a number of stages greater than two and generic RK coefficients. The basis of this derivation is that ultimately, the accuracy of the method depends on the order of net error introduced by extrapolation rather than the order of error for each individual extrapolation. Thus, there are multiple extrapolation schemes which will achieve third-order accuracy for FastRK3. We then verified numerically that a second-order net error in extrapolation is the determining criteria for FastRK3 to yield a third-order accurate solution. From this analytical and numerical analysis, we confirmed that FastRK3 achieves third-order temporal accuracy for free shear flows. We expect a loss of accuracy for wall-bounded flows such that the solution is second-order [1, 2].

BIBLIOGRAPHY

[1] A. B. Aithal , A. Ferrante. A fast pressure-correction method for incompressible flows over curved walls. *Journal of Computational Physics*, 421:109693, 2020.

[2] A. B. Aithal, M. Tipirneni, A. Ferrante. Temporal accuracy of FastRK3. In prep.

[3] A. J. Chorin. A Numerical Method for Solving Incompressible Viscous Flow Problems. *Journal of Computational Physics*, 135(2):118–125, 1997.

[4] B. Sanderse, B. Koren. Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 231(8):3041–3063, 2012.

[5] C. De Michele, F. Capuano, G. Coppola. Fast-Projection Methods for the Incompressible Navier-Stokes Equations. *Fluids (Basel)*, 5(4):222, 2020.

[6] E. Hairer, C. Lubich, M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer, 1989.

[7] F. Capuano, G. Coppola, M. Chiatto, L. de Luca. Approximate Projection Method for the Incompressible Navier-Stokes Equations. *AIAA Journal*, 54(7):2179–2182, 2016.

[8] G. I. Taylor. On the Decay of Vortices in a Viscous Fluid. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 46(274):671–674, 1923.

[9] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[10] V. Brasey, E. Hairer. Half-Explicit Runge–Kutta Methods for Differential-Algebraic Systems of Index 2. *SIAM Journal on Numerical Analysis*, 30(2):538–552, 1993.

Appendix A

ACCURACY OF THE SS EXTRAPOLATION

Using the same simulation setup described in Section 3.1.2 we test FastRK3 with the SS extrapolation formulation (Eqs. 3.17 and 3.18). The resulting stage-wise pseudo-pressure gradient errors relative to standard RK3 are shown in Fig. A.1. The extrapolated $G\phi_1$ and $G\phi_2$ are indeed second-order approximations of the standard RK3 solutions, as previously theorized. Also, as there is no net first-order error in the extrapolations, $G\phi_{n+1}$ is a third-order approximation of $G\tilde{\phi}_{n+1}$.

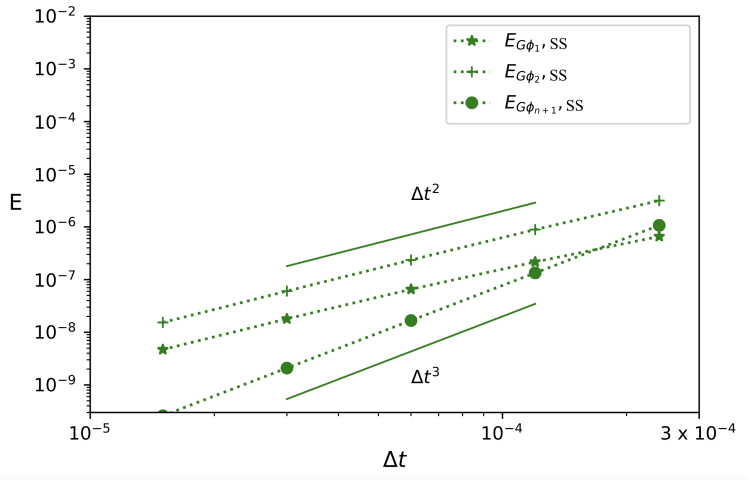


Figure A.1: Stage-wise $G\phi$ temporal accuracy for FastRK3 (SS extrapolation) relative to standard RK3.