

# Accepted Manuscript

Optimal Runge-Kutta Schemes for Pseudo Time-Stepping with High-Order Unstructured Methods

B.C. Vermeire, N.A. Loppi, P.E. Vincent

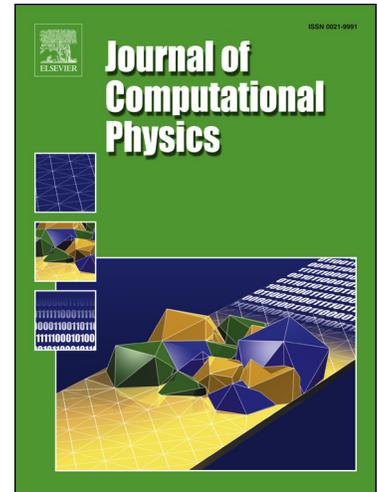
PII: S0021-9991(19)30018-X  
DOI: <https://doi.org/10.1016/j.jcp.2019.01.003>  
Reference: YJCPH 8434

To appear in: *Journal of Computational Physics*

Received date: 27 February 2018  
Revised date: 22 November 2018  
Accepted date: 3 January 2019

Please cite this article in press as: B.C. Vermeire et al., Optimal Runge-Kutta Schemes for Pseudo Time-Stepping with High-Order Unstructured Methods, *J. Comput. Phys.* (2019), <https://doi.org/10.1016/j.jcp.2019.01.003>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



## Highlights

- We generate optimal Runge-Kutta schemes for the artificial compressibility method.
- These are optimized for high-order unstructured spatial discretizations.
- Significant speedup factors are observed relative to classical Runge-Kutta schemes.
- We demonstrate that these scheme are suitable for incompressible turbulent flows.

# Optimal Runge-Kutta Schemes for Pseudo Time-Stepping with High-Order Unstructured Methods

B. C. Vermeire\*

*Department of Mechanical, Industrial, and Aerospace Engineering, Concordia University*

N. A. Loppi and P. E. Vincent

*Department of Aeronautics, Imperial College London*

November 22, 2018

## Abstract

In this study we generate optimal Runge-Kutta (RK) schemes for converging the Artificial Compressibility Method (ACM) using dual time-stepping with high-order unstructured spatial discretizations. We present optimal RK schemes with between  $s = 2$  and  $s = 7$  stages for Spectral Difference (SD) and Discontinuous Galerkin (DG) discretizations obtained using the Flux Reconstruction (FR) approach with solution polynomial degrees of  $k = 1$  to  $k = 8$ . These schemes are optimal in the context of linear advection with predicted speedup factors in excess of  $1.80\times$  relative to a classical  $RK_{4,4}$  scheme. Speedup factors of between  $1.89\times$  and  $2.11\times$  are then observed for incompressible Implicit Large Eddy Simulation (ILES) of turbulent flow over an SD7003 airfoil. Finally, we demonstrate the utility of the schemes for incompressible ILES of a turbulent jet, achieving good agreement with experimental data. The results demonstrate that the optimized RK schemes are suitable for simulating turbulent flows and can achieve significant speedup factors when converging the ACM using dual time-stepping with high-order unstructured spatial discretizations.

---

\*Corresponding author; e-mail [brian.vermeire@concordia.ca](mailto:brian.vermeire@concordia.ca)

## 1 Introduction

The Artificial Compressibility Method (ACM) is an alternative to pressure correction based methods for solving the incompressible Navier-Stokes equations [1]. In the ACM, pressure is solved together with other variables from a transformed hyperbolic steady-state system instead of via an elliptic Poisson equation. Combined with dual time stepping [2], the ACM can be applied to incompressible unsteady flows by approximately solving the steady-state problem in fictitious pseudo-time at each physical time step. In practice, the computational cost of this approach is dominated by the rate of convergence of the solution in pseudo-time. There are several popular approaches to performing pseudo-iterations, such as implicit pseudo-time stepping with the Lower-Upper Symmetric Gauss Seidel (LU-SGS) method [3, 4], or using explicit schemes such as Runge-Kutta (RK) methods [2].

Recent research has led to the development of high-order unstructured spatial discretizations suitable for the ACM, such as the Discontinuous Galerkin (DG) [5, 6], Spectral Difference (SD) [7, 8], and Flux Reconstruction (FR) approaches [9]. It has been demonstrated previously that these methods are particularly appealing for scale-resolving simulations of unsteady turbulent flows via Large Eddy Simulation (LES), Implicit LES (ILES), and Direct Numerical Simulation (DNS) [10, 11, 12]. They are able to provide accurate solutions on mixed-element unstructured grids, and have been shown to be more accurate than industry standard second-order tools per degree of freedom [13]. While significant contributions have been made to the application of high-order unstructured methods to incompressible flows [4, 14, 15, 16], the majority of previous work has focused on their application to the compressible Navier-Stokes equations. Recently, it has been demonstrated that the FR approach can be extended to solve incompressible flow problems using the ACM and dual time stepping. Furthermore, it has been demonstrated that FR with explicit time stepping can be scaled to tens of thousands of GPU compute nodes achieving  $> 50\%$  of peak performance on modern many-core hardware architectures [17].

The main advantages of using the ACM with explicit pseudo-time stepping are its scale invariance and low memory footprint, making it well-suited for modern many-core hardware platforms. Although explicit schemes are relatively inexpensive per time step, they are restricted by their conditional stability limit and, hence, a greater total number of pseudo time steps must be taken. This issue can be addressed by adopting implicit schemes, allowing significantly larger pseudo-time steps. However, the trade-off with implicit schemes is that they require the computation and storage of Jacobian matrices. This becomes prohibitively expensive for higher-order spatial discretizations and is typically memory bandwidth bound. In addition, many methods for solving the resulting linear system, such as Lower-Upper Symmetric Gauss-Seidel (LU-SGS) or Newton-Krylov methods, are not scale invariant and

increasing the amount of parallelism also reduces the numerical effectiveness of the preconditioner. Hence, in the context of high-order spatial discretization using the ACM on modern hardware architectures, explicit approaches remain relevant even though a relatively large number of explicit iterations are usually required to converge the pseudo time system to pseudo-steady-state [18].

The maximum stable time-step for an explicit discretization is a function of both the temporal scheme, such as the particular explicit RK method being used, and the spatial discretization [19]. In this study we develop optimal explicit RK methods in order to reduce the total number of pseudo time steps and hence total computational cost of an explicit high-order unstructured ACM implementation. The methods are optimal in that they allow the largest possible pseudo time step to be taken for a given number of RK stages. The rest of the manuscript is organized as follows. In Chapter 2 we discuss the general ACM formulation and its solution using dual time stepping. In Chapter 3 we summarize general RK methods and in Chapter 4 present the FR approach. We then derive optimal RK schemes for pseudo time stepping, and demonstrate their application to incompressible turbulent flow over an SD7003 airfoil and an incompressible turbulent jet in Chapter 5. Finally, we present conclusions in Chapter 6.

## 2 The Artificial Compressibility Method

### 2.1 Formulation

The ACM was originally introduced by Chorin [1] in the late 1960s for solving steady incompressible fluid flow problems. Rather than relying on a Poisson equation based projection, a coupling pressure term is introduced to the continuity equation to drive the system towards a solenoidal velocity field in the limit of steady state. The original formulation preserves the hyperbolic nature of the system, but destroys time accuracy. The most common procedure for recovering time accuracy is the dual time stepping approach of Jameson [2].

The artificial compressibility system can be written in a conservation form as

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0, \quad (1)$$

where

$$u = \begin{pmatrix} p \\ v_x \\ v_y \\ v_z \end{pmatrix}, \quad (2)$$

and

$$\mathbf{f} = \begin{Bmatrix} f_{e_x} - f_{v_x} \\ f_{e_y} - f_{v_y} \\ f_{e_z} - f_{v_z} \end{Bmatrix}, \quad (3)$$

with

$$f_{e_x} = \begin{Bmatrix} \zeta v_x \\ v_x^2 + p \\ v_x v_y \\ v_x v_z \end{Bmatrix}, \quad f_{e_y} = \begin{Bmatrix} \zeta v_y \\ v_y v_x \\ v_y^2 + p \\ v_y v_z \end{Bmatrix}, \quad f_{e_z} = \begin{Bmatrix} \zeta v_z \\ v_z v_x \\ v_z v_y \\ v_z^2 + p \end{Bmatrix}, \quad (4)$$

$$f_{v_x} = \nu \begin{Bmatrix} 0 \\ \frac{\partial v_x}{\partial x} \\ \frac{\partial v_y}{\partial x} \\ \frac{\partial v_z}{\partial x} \end{Bmatrix}, \quad f_{v_y} = \nu \begin{Bmatrix} 0 \\ \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial y} \\ \frac{\partial v_z}{\partial y} \end{Bmatrix}, \quad f_{v_z} = \nu \begin{Bmatrix} 0 \\ \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial z} \end{Bmatrix}, \quad (5)$$

where  $\zeta$  is the artificial compressibility relaxation factor,  $\nu$  is the kinematic viscosity,  $p$  is the pressure, and  $v_i$  is a velocity component for  $i \in \{x, y, z\}$ . The ACM formulation has a hyperbolic nature in pseudo time, similar to the linear advection equation, since artificial pressure waves travelling at a finite speed are introduced. The eigenvalues of the inviscid flux Jacobian matrices

$$\mathbf{J}_i = \frac{\partial f_{e_i}}{\partial u}, \quad (6)$$

are

$$\lambda_i = \{v_i - c_i, v_i, v_i, v_i + c_i\}, \quad (7)$$

where  $c_i = \sqrt{v_i^2 + \zeta}$  is the pseudo speed of sound. The pseudo speed of sound is proportional to the artificial compressibility relaxation factor  $\zeta$ , which is adjusted to downscale the wave speed to reduce system stiffness.

## 2.2 Dual Time Stepping

Dual time stepping can be considered as consecutive pseudo steady-state simulations [2]. In this approach, the real time is discretised with a backward difference (BDF) scheme, whose solution is found by marching the governing equation in pseudo time. A conservation law with an additional pseudo time derivative can be marched towards a pseudo steady state following

$$\frac{\partial u}{\partial \tau} + \mathbf{I}_c \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0. \quad (8)$$

A cancellation matrix  $\mathbf{I}_c = \text{diag}\{0 \ 1 \ 1 \ 1\}$  needs to be added as a coefficient for the real time derivative to eliminate it from the continuity equation. This ensures that the solution is driven towards the divergence free state.

As the solution is integrated with respect to pseudo time, the real time derivative can be considered as a source term for the right-hand side that

is updated at every real time step. Defining  $\mathcal{R} = R - S_t$ , where  $R = -\nabla \cdot \mathbf{f}$  and  $S_t = \mathbf{I}_c \frac{\partial u}{\partial t}$ , Equation 8 reads

$$\frac{\partial u}{\partial \tau} = \mathcal{R} = R - S_t. \quad (9)$$

Discretizing the real time derivative with an  $M$ -stage BDF [19] scheme and the pseudo time derivative with a  $s_{\max}$ -stage explicit RK scheme [19], we can write a single pseudo iteration as

$$\begin{aligned} u^{n+1,m+1,0} &= u^{n+1,m} \\ u^{n+1,m+1,s} &= u^{n+1,m+1,0} + \alpha_s \Delta \tau \mathcal{R}^{n+1,m+1,s-1}, \text{ for } s = 1 \dots s_{\max} \\ u^{n+1,m+1} &= u^{n+1,m+1,s_{\max}}, \end{aligned} \quad (10)$$

where the superscripts  $n$  and  $m$  denote real and pseudo time levels,  $s$  is the stage index, and  $\alpha_s$  is the stage coefficient. Different real time levels are needed for the BDF expansion in the right hand side computation according to

$$\begin{aligned} \mathcal{R}^{n+1,m+1,s-1} &= \frac{1}{\alpha_{\text{PI}}} \left[ R^{n+1,m+1,s-1} - \mathbf{I}_c S_t^{n+1,m+1,0} \right] \\ &= \frac{1}{\alpha_{\text{PI}}} \left[ R^{n+1,m+1,s-1} - \frac{\mathbf{I}_c}{\Delta t} \left( B_0 u^{n+1,m+1,0} + \sum_{\sigma=0}^M B_{\sigma+1} u^{n-\sigma} \right) \right], \end{aligned} \quad (11)$$

in which  $B_\sigma$  are the coefficients of the BDF schemes listed in Table 1. In this study, the leading term in the BDF is expressed at the first RK stage, which is often referred to as point-implicit source term treatment. The treatment also introduces a scaling coefficient  $\alpha_{\text{PI}} = 1 + \alpha_s \Delta \tau B_0 / \Delta t$  which effectively limits  $\Delta \tau$  if  $\Delta \tau \gg \Delta t$ . In this instance, however, best performance was achieved with  $\alpha_{\text{PI}} = 1$ .

Table 1. Coefficients for the backward difference formulas

	$B_0$	$B_1$	$B_2$	$B_3$
Backward-Euler	1	-1	-	-
BDF2	$\frac{3}{2}$	-2	$\frac{1}{2}$	-
BDF3	$\frac{11}{6}$	-3	$\frac{1}{2}$	$-\frac{1}{3}$

The sub-iteration procedure according to Equation 9 is repeated until the solution has reached a specified convergence tolerance. Subsequently, the pseudo steady-state solution is stored as the real solution and assigned as the initial value for the next time level  $u^{n+1} = u^{n+2,0}$ . Furthermore, the rest of the terms in  $S_t$  are updated accordingly,  $u^{n-(\sigma+1)} = u^{n-\sigma}$  for  $\sigma = 1 \dots M$ . Algorithm 1 illustrates the dual time stepping methodology with BDF2.

**Algorithm 1:** Dual Time Stepping with BDF2.

```

while  $t < t_{max}$  do
  while  $\frac{\partial u}{\partial \tau} > \epsilon$  do
    | Solve Equation 9
  end
  Return  $u^{n+1}$ 
   $u^{n-1} = u^n$ 
   $u^n = u^{n+1}$ 
   $t = t + \Delta t$ 
end

```

### 3 Runge-Kutta Schemes

#### 3.1 Explicit Schemes

A general  $s$ -stage explicit RK scheme can be represented in matrix-vector form by the Butcher tableau [19]

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b} \end{array} \quad (12)$$

where

$$c_i = \sum_{j=1}^s a_{ij}, \quad (13)$$

$$\sum_{i=1}^s b_i = 1, \quad (14)$$

and

$$a_{ij} = 0, \quad j \geq i, \quad (15)$$

which ensures  $\mathbf{A}$  is strictly lower triangular.

When applied to ODE's of the form  $du/dt = \omega u$ , where  $\omega \in \mathbb{C}$ , a  $p$ th-order explicit RK method can always be expressed as [20]

$$u^{n+1} = P_{s,p}(z)u^n, \quad (16)$$

where  $P_{s,p}(z)$  is the schemes stability polynomial and  $z = \omega \Delta t$ . The resulting scheme will be linearly stable provided  $|P_{s,p}| \leq 1$ , where  $|\cdot|$  is the complex modulus. The stability polynomial can be determined directly from the Butcher tableau according to

$$P_{s,p}(z) = 1 + z\mathbf{b}^T (\mathbf{I} - z\mathbf{A})^{-1} \mathbf{e} = \frac{|\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^T|}{|\mathbf{I} - z\mathbf{A}|}. \quad (17)$$

where  $\mathbf{e}$  is a vector of ones. The region of absolute stability  $S$  of the RK method is the set in the complex plane where  $|P_{s,p}| \leq 1$  holds, that is [20],

$$S = \{z \in \mathbb{C} : |P_{s,p}| \leq 1\}, \quad (18)$$

and the linear stability condition is

$$\Delta t \omega \subseteq S. \quad (19)$$

The eigenvalues  $\omega^\delta$  of a general spatial discretization can be obtained via von Neumann analysis for each permissible wavenumber  $\theta^\delta$ . Therefore, to ensure linear stability of any fully-discrete scheme consisting of a spatial discretization and RK pair we require

$$|P_{s,p}(\Delta t \omega^\delta)| \leq 1, \quad \forall \omega^\delta. \quad (20)$$

### 3.2 Optimizing Runge-Kutta Methods

In general, the stability polynomial  $P_{s,p}$  of an explicit RK method is a polynomial of degree  $s$  defined by polynomial coefficients  $\gamma_j$  according to [20]

$$P_{s,p}(z) = \sum_{j=0}^s \gamma_j z^j. \quad (21)$$

For the method to be accurate to order  $p$ , we require the stability polynomial coefficients to be identical to the exponential function up to terms of at least order  $p$ , that is

$$\gamma_j = \frac{1}{j!}, \quad 0 \leq j \leq p. \quad (22)$$

For any explicit scheme we can determine the largest time-step  $\Delta t$  for which Equation 20 remains satisfied. Our objective is to determine the set of free parameters  $\{\gamma_0, \gamma_1, \dots, \gamma_s\}$  that yield the largest possible value of  $\Delta t$  for a given spatial discretization to accelerate the convergence of Equation 9. This can be cast as an optimization problem [20]

$$\begin{aligned} & \underset{\gamma_{p+1}, \gamma_{p+2}, \dots, \gamma_s}{\text{maximize}} && \Delta t \\ & \text{subject to} && |P_{s,p}(\Delta t \omega^\delta)| - 1 \leq 0, \quad \forall \omega^\delta. \end{aligned} \quad (23)$$

We use  $\Delta t_{opt}$  to denote the solution to this optimization problem (the optimal time-step size), and  $P_{opt}$  to denote the corresponding optimal stability polynomial. Finding the global optimum in Equation 23 can be particularly challenging, as it is nonconvex for  $s > 2$  and there are typically several sub-optimal local minima.

Ketcheson and Ahmadi [20] proposed an alternate optimization approach, which seeks to first determine how small the maximum modulus of  $P_{s,p}(\Delta t \omega^\delta)$  can be for a given  $\Delta t$ , which can be cast as

$$\underset{\gamma_{p+1}, \gamma_{p+2}, \dots, \gamma_s}{\text{minimize}} \quad \max_{\omega^\delta \in \boldsymbol{\omega}^\delta} \left( |P_{s,p}(\Delta t \omega^\delta)| - 1 \right). \quad (24)$$

The solution to Equation 24 is denoted  $r_{s,p}(\Delta t, \boldsymbol{\omega}^\delta)$  and we note that  $|P_{s,p}(z)|$  is linear in  $\gamma_j$ . Therefore, Equation 24 is convex [20]. Ketcheson and Ahmadi [20] then reformulated the problem defined by Equation 23 as

$$\begin{aligned} & \underset{\gamma_{p+1}, \gamma_{p+2}, \dots, \gamma_s}{\text{maximize}} && \Delta t \\ & \text{subject to} && r_{s,p}(\Delta t, \boldsymbol{\omega}^\delta) \leq 0. \end{aligned} \quad (25)$$

This is an optimization problem in a single variable, and can be solved using the bisection method outlined in Algorithm 2. The solution of Algorithm 2 yields the maximum possible linearly stable time-step  $\Delta t_{opt}$ , as well as the corresponding optimal stability polynomial  $P_{opt}$ .

**Algorithm 2:** Finding  $\Delta t_{opt}$ .

```

Select  $\Delta t_{max}$ 
 $\Delta t_{min} = 0$ 
while  $\Delta t_{max} - \Delta t_{min} > \epsilon$  do
     $\Delta t = \frac{\Delta t_{min} + \Delta t_{max}}{2}$ 
    Solve Equation 24
    if  $r_{s,p}(\Delta t, \boldsymbol{\omega}^\delta) \leq 0$  then
        |  $\Delta t_{min} = \Delta t$ 
    else
        |  $\Delta t_{max} = \Delta t$ 
    end
end
return  $\Delta t_{opt} = \Delta t_{min}$ 

```

Finally, to generate Butcher tableaus based on the optimal stability polynomial  $P_{opt}$  we used RK-Opt, a package for optimization of Runge-Kutta methods [21, 22]. RK-opt has also been used in related work, such as Parsani et al. [22] who found optimal RK schemes for the SD method, and Kubatko et al. [23] who used it to determine Strong Stability Preserving (SSP) RK schemes for DG methods. Our FRDG schemes differ from those of Kubatko et al. [23] in that we relax the order of accuracy of the identified RK schemes to first-order, since we do not require accuracy in pseudo time. Also, due to the smooth nature of incompressible flows, we focus on optimizing linear stability limits rather than SSP properties. Applying these accuracy constraints to

the stability polynomial we obtain

$$P_{s,1}(z) = 1 + z + \sum_{j=2}^s \gamma_j z^j, \quad (26)$$

which has  $s - 1$  free parameters  $\{\gamma_2, \gamma_3, \dots, \gamma_s\}$ . In the current study we converge the bisection method in Algorithm 2 to  $10^{-12}$  using Matlab 2016a [24]. To solve the convex minimization problem in Equation 24 we used CVX, which is a Matlab-based package for specifying and solving convex programs, using the “best” precision setting [25, 26].

## 4 The Flux Reconstruction Approach

In the current study we use the FR approach for spatial discretization, noting that the RK optimization procedure is also applicable to other general spatial discretizations such as the finite difference, finite volume, finite element, or DG methods. The FR approach, first introduced by Huynh [9], later described using our notation in [27, 28, 29] and summarized here for completeness, is used to spatially discretize a 1D hyperbolic conservation law of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad (27)$$

in a domain  $\Omega$ , where  $u = u(x, t)$  is the solution with a given initial distribution  $u(x, 0) = u_0$ ,  $t$  is time,  $f = f(u)$  is the flux, and  $x$  is the spatial coordinate.

We start by splitting the domain  $\Omega$  into a mesh composed of elements

$$\Omega = \bigcup_{n=1}^{N_e} \Omega_n, \quad \bigcap_{n=1}^{N_e} \Omega_n = \emptyset, \quad (28)$$

where  $\Omega_n$  is one of  $N_e$  elements in the domain. For simplicity, each element  $\Omega_n$  is mapped to a reference element  $I$  via the linear mapping

$$\xi = \Gamma_n(x) = 2 \left( \frac{x - x_n}{x_{n+1} - x_n} \right) - 1, \quad (29)$$

where  $x_n$  is the mesh node corresponding to the left face of  $\Omega_n$ ,  $\Gamma_n$  is the mapping function, and  $\xi$  is the location in reference space. This mapping can be inverted according to

$$x = \Gamma_n^{-1}(\xi) = \left( \frac{1 - \xi}{2} \right) x_n + \left( \frac{1 + \xi}{2} \right) x_{n+1}, \quad (30)$$

which is also linear. This allows us to construct the transformed solution in computational space according to

$$\hat{u}_n^\delta = \hat{u}_n^\delta(\xi, t) = u_n^\delta(\Gamma_n^{-1}(\xi), t), \quad (31)$$

and a transformed flux in computational space according to

$$\hat{f}_n^\delta = \hat{f}_n^\delta(\xi, t) = \frac{f_n^\delta(\Gamma_n^{-1}(\xi), t)}{h_n}. \quad (32)$$

The system of equations in computational space can then be written as

$$\frac{\partial \hat{u}_n^\delta}{\partial t} + \frac{\partial \hat{f}_n^\delta}{\partial \xi} = 0. \quad (33)$$

Following Huynh [9], the solution within each element is represented by a degree  $k$  polynomial, which is allowed to be discontinuous at the interface between elements. This polynomial is supported by nodal basis functions generated at  $k+1$  solution points. Therefore, the solution within each element in computational space can be approximated as

$$\hat{u}_n^\delta = \sum_{l=0}^k \hat{u}_{n,l}^\delta \hat{\phi}_l, \quad (34)$$

where  $\hat{u}_n^\delta = \hat{u}_n^\delta(\xi, t)$  is the polynomial representation of the solution within an element,  $\hat{u}_{n,l}^\delta = \hat{u}_{n,l}^\delta(t)$  is the value of the solution at solution point  $l$ , and  $\hat{\phi}_l = \hat{\phi}_l(\xi)$  is its corresponding nodal basis function in reference space. For the one-dimensional case these basis functions are the Lagrange polynomials

$$\hat{\phi}_l(\xi) = \prod_{m=0, m \neq l}^k \frac{\xi - \xi_m}{\xi_l - \xi_m}. \quad (35)$$

A polynomial representation of the discontinuous flux function  $\hat{f}_n^{\delta D} = \hat{f}_n^{\delta D}(x, t)$  can be constructed using the same polynomial basis as the solution according to

$$\hat{f}_n^{\delta D} = \sum_{l=0}^k \hat{f}_{n,l}^\delta \hat{\phi}_l, \quad (36)$$

where the superscript  $D$  denotes that this flux, like the solution, is allowed to be discontinuous at the interface between elements and  $\hat{f}_{n,l}^\delta = \hat{f}_{n,l}^\delta(t)$  is the flux evaluated at the solution points. This discontinuous flux is then corrected to be  $C_0$  continuous by

$$\hat{f}_n^{\delta C} = \left( \hat{f}_n^{CL} - \hat{f}_{n,L}^{\delta D} \right) g_L + \left( \hat{f}_n^{CR} - \hat{f}_{n,R}^{\delta D} \right) g_R, \quad (37)$$

where  $g_L = g_L(\xi)$  and  $g_R = g_R(\xi)$  are correction functions,  $\hat{f}_{n,L}^{\delta D} = \hat{f}_n^{\delta D}(-1, t)$ , and  $\hat{f}_{n,R}^{\delta D} = \hat{f}_n^{\delta D}(1, t)$ . The terms  $\hat{f}_n^{CL} = \hat{f}_n^{CL}(u_L^-, u_L^+)$  and  $\hat{f}_n^{CR} = \hat{f}_n^{CR}(u_R^-, u_R^+)$  are common interface fluxes computed at the flux points between elements by an appropriate Riemann solver using the interpolated values  $u_L^-, u_L^+, u_R^-,$

and  $u_R^+$  of the solution from the adjoining elements at the left ( $L$ ) and right ( $R$ ) interfaces. The correction functions are degree  $k + 1$  polynomials with the constraints

$$g_L(-1) = 1, \quad g_L(1) = 0, \quad (38)$$

$$g_R(-1) = 0, \quad g_R(1) = 1. \quad (39)$$

The gradient of the continuous flux function can then be found according to

$$\frac{\partial \hat{f}_n^\delta}{\partial \xi} = \frac{\partial \hat{f}_n^{\delta D}}{\partial \xi} + \frac{\partial \hat{f}_n^{\delta C}}{\partial \xi} = \sum_{l=0}^k \hat{f}_{n,l}^\delta \frac{\partial \hat{\phi}_l}{\partial \xi} + (\hat{f}_n^{CL} - \hat{f}_{n,L}^{\delta D}) \frac{\partial g_L}{\partial \xi} + (\hat{f}_n^{CR} - \hat{f}_{n,R}^{\delta D}) \frac{\partial g_R}{\partial \xi}, \quad (40)$$

which is of degree  $k$  and in the same polynomial space as  $\hat{u}_n^\delta$ .

The behaviour of an FR scheme depends on the choice of correction functions  $g_L$  and  $g_R$  for the left and right hand boundaries, respectively. Huynh [9] originally introduced several correction functions including one that recovered a collocation-based nodal DG scheme (FRDG), one that recovered an energy-stable SD scheme (FRSD), and one that recovered a so-called  $g_2$  scheme. Vincent et al. [27, 30] later introduced a single-parameter family of energy-stable flux reconstruction schemes, whose semi-discrete dispersion and dissipation behaviour was studied.

#### 4.1 Von Neumann Analysis

To determine the stability constraints of an FR scheme we perform Von Neumann analysis following Huynh [9], Vincent et al [27], and Vermeire et al. [28, 29]. For this study, we consider the linear advection equation with fully upwind fluxes at element interfaces, since the ACM requires the solution of a non-linear advection problem.

We follow the notation of Vincent et al. [27] for Von Neumann analysis of a general linear advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, \quad (41)$$

which admits plane wave solutions of the form

$$u = e^{I(\theta x - \omega t)}, \quad (42)$$

provided the temporal frequency  $\omega = \omega(\theta)$  satisfies the dispersion relation where  $\theta$  is the wave number and  $I = \sqrt{-1}$ . Specifically, we consider a mesh where all elements have constant width  $h = 1$ . The flux reconstruction scheme can then be cast in matrix-vector form for any element as

$$\frac{\partial \hat{\mathbf{u}}^\delta}{\partial t} = -2\mathbf{D}\hat{\mathbf{u}}^\delta - (\hat{f}^{CL} - 2\mathbf{l}^T \hat{\mathbf{u}}^\delta) \mathbf{g}_{\xi L}, \quad (43)$$

where  $\hat{\mathbf{u}}^\delta$  is the vector of solution point values,  $\mathbf{g}_{\xi L}$  is the gradient of the correction function evaluated at the solution points,

$$\mathbf{l}[k] = \hat{\phi}_{j,k}(-1), \quad (44)$$

is the boundary extrapolation operator, and  $\mathbf{D}$  is the nodal gradient operator.

We seek Bloch wave type solution to Equation 43 of the form

$$\hat{\mathbf{u}}^\delta = e^{I(n\bar{\theta}^\delta - \omega^\delta t)} \hat{\mathbf{v}}^\delta, \quad (45)$$

where  $\bar{\theta}^\delta$  is a prescribed baseline wavenumber within the range  $-\pi \leq \bar{\theta} \leq \pi$  and  $\omega^\delta$  is the resulting temporal frequency of the scheme. The upwind interface flux can then be written as

$$\hat{f}^{CL} = 2\mathbf{r}^T e^{I(n\bar{\theta}^\delta - \bar{\theta}^\delta - \omega^\delta t)} \hat{\mathbf{v}}^\delta, \quad (46)$$

where

$$\mathbf{r}[l] = \hat{\phi}_{n,l}(1). \quad (47)$$

By substituting Equation (45) and Equation (46) into Equation (43) we obtain

$$\mathbf{Q} \hat{\mathbf{v}}^\delta = \omega^\delta \hat{\mathbf{v}}^\delta, \quad (48)$$

where

$$\mathbf{Q} = -2I \left[ \mathbf{D} + \mathbf{g}_{\xi L} \left( \mathbf{r}^T e^{-I\bar{\theta}^\delta} - \mathbf{l}^T \right) \right]. \quad (49)$$

Equation (48) is a classical eigenvalue problem, where  $\hat{\mathbf{v}}$  is one of  $k + 1$  valid eigenvectors with an associated eigenvalue  $\omega^\delta$ . The operator  $\mathbf{Q}$  is a function of  $\bar{\theta}^\delta$ , and it follows that so too are  $\hat{\mathbf{v}}^\delta$  and  $\omega^\delta$ . In order for a fully discrete scheme to remain stable, all of the eigenvalues  $\omega^\delta$  from the spatial discretization must be contained within the region of absolute stability of the temporal discretization for all permissible values of  $\bar{\theta}^\delta$  for a given  $\Delta t$ .

## 4.2 Optimal Methods for DG

By using the FRDG correction function described by Huynh [9] and Vincent et al. [30] in Equation 40 we recover the DG method. Values of  $\Delta t_{opt}$  obtained using Algorithm 2 are summarized in Table 2 for FRDG schemes with polynomial degrees  $k = 1$  to  $k = 8$  using between  $s = 2$  and  $s = 7$  RK stages. These results are also plotted in Figure 1. Tabulated stability polynomial coefficients and corresponding Butcher tableau coefficients are provided as Electronic Supplementary Material with this article. We observe that, for a given polynomial degree, optimal RK schemes with more stages have a larger stability limit. Also, for a given number of stages, optimal RK schemes for higher degree FRDG schemes have a smaller stability limit.

Although optimal RK schemes with more stages permit larger time-steps, they also require an additional residual evaluation per stage. Therefore, the effective computational cost is obtained by normalizing  $\Delta t_{opt}$  of a given scheme by the total number of stages as  $\Delta t_{opt}/s$  as shown in Figure 1. In Table 3 we present the effective speedup of the optimal RK schemes obtained in this study relative to the classical  $RK_{4,4}$  scheme. We observe speedups in excess of  $1.80\times$  depending on the polynomial degree, suggesting that the optimal RK schemes developed in the current study could be nearly twice as fast as conventional methods for converging the ACM in pseudo time when using FRDG. Furthermore, as shown in Figure 1, the benefit of increasing the number of stages beyond  $s = 7$  is only marginal.

We are also interested in the computational cost of converging the pseudo time problem for a fixed spatial resolution. This is proportional to  $(k + 1)\Delta t_{opt}/s$ , which normalizes  $\Delta t_{opt}$  by both the number of stages and the number of degrees of freedom per element. Referred to here as the effective resolution, this measure accounts for the fact that for a fixed spatial resolution fewer relatively large high-order elements can be used. Results for this, shown in Figure 1, demonstrate that we can expect higher-order FRDG schemes to be more expensive than lower-order schemes per degree of freedom. Interestingly, the additional cost of moving to subsequently higher-order schemes reduces as the order is increased. This suggests that the additional computational cost of using, for example, a  $k = 8$  FRDG scheme compared to a  $k = 7$  scheme with an equivalent number of degrees of freedom could be marginal when using the current optimized RK schemes.

Finally, we have plotted a sampling of the eigenspectra for the  $k = 4$  FRDG scheme scaled by the maximum stable time step ( $\omega^\delta \Delta t_{opt}$ ) alongside the region of absolute stability ( $S$ ) and contours of  $|P_{s,p}(z)|$  in Figure 2. It can be seen that large portions of the region of absolute stability for optimal schemes with a low number of stages are effectively unused. For example, for  $s = 2$  there are large stable regions above and below the eigenspectra that are not utilized when the maximum stability condition  $\Delta t_{opt}$  is met. However, as the number of stages, and hence degrees of freedom available to be optimized are increased, the scaled eigenspectra and boundary of the region of absolute stability are nearly identical. This suggests that, given an RK scheme with enough stages, the optimizer will generate a scheme whose stability boundary mimics the shape of the spatial discretization's eigenspectra.

### 4.3 Optimal Methods for SD

It should be noted that the optimization strategy used in this study is not limited to the FRDG correction function, and can be generalized to any valid FR correction function or to other types of spatial discretizations. For example, by using the FRSD correction function described by Huynh [9] and Vincent et al. [30] in Equation 40 we recover a SD scheme. We can

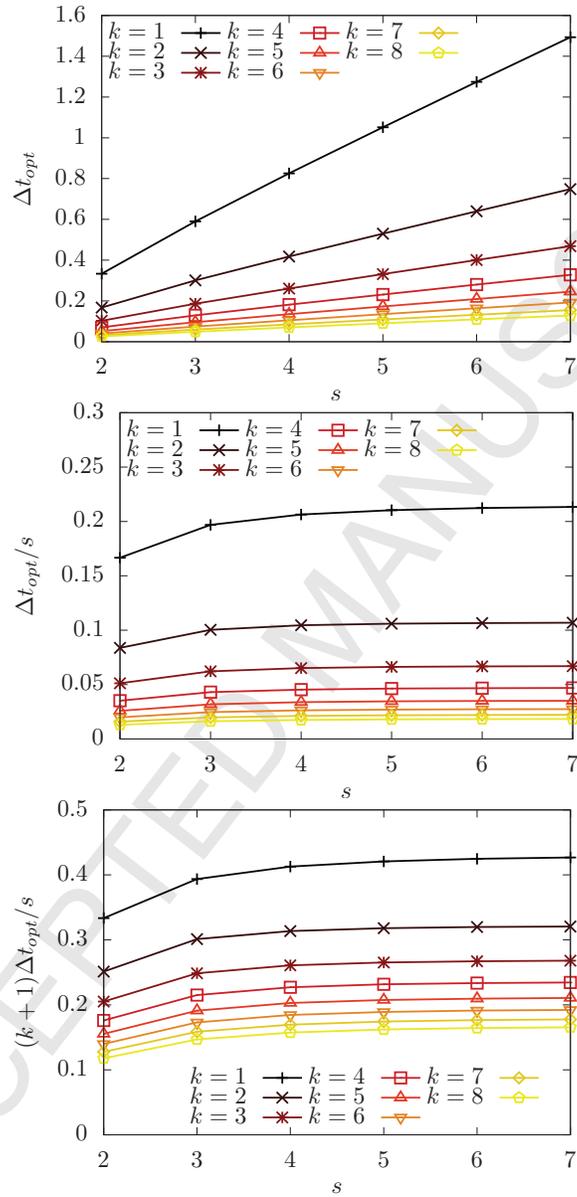


Figure 1. Optimal time-step  $\Delta t_{opt}$  (top), time-step per stage (middle), and time-step per stage normalized by effective resolution (bottom) for optimal RK schemes using FRDG.

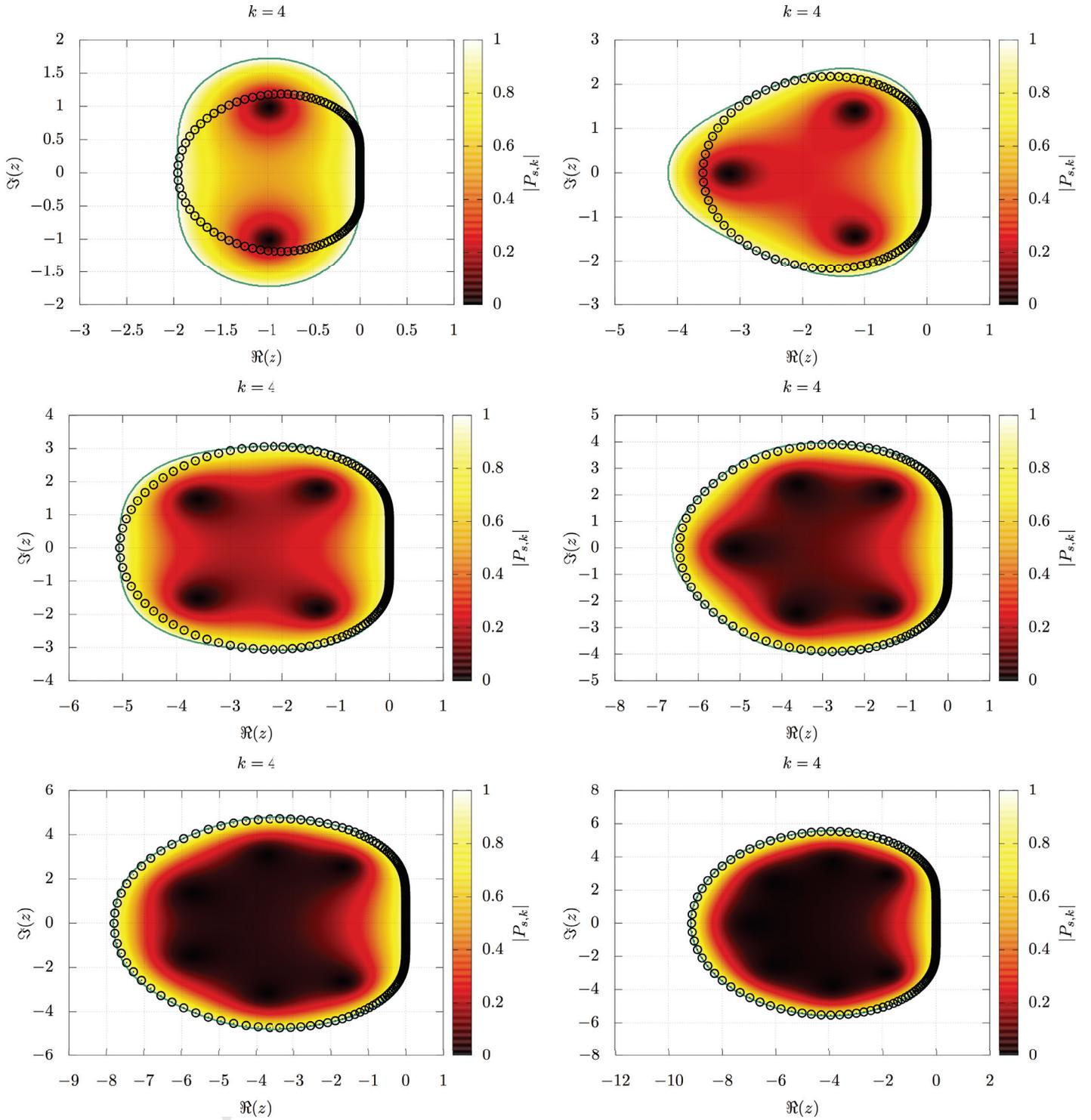


Figure 2. Contours of stability polynomials for optimal RK schemes for  $k = 4$  FRDG with the stability boundary (blue line) and eigenvalues from von Nuemann analysis (black circles) scaled by  $\Delta t_{opt}$ .

Table 2. Values of  $\Delta t_{opt}$  for optimal RK schemes of order  $k = 1$  for FRDG.

Stages	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
2	0.333	0.167	0.102	0.070	0.051	0.039	0.031	0.026
3	0.590	0.301	0.186	0.129	0.095	0.074	0.059	0.048
4	0.825	0.418	0.260	0.181	0.135	0.105	0.084	0.069
5	1.051	0.529	0.331	0.231	0.172	0.135	0.108	0.090
6	1.274	0.639	0.400	0.280	0.209	0.163	0.132	0.109
7	1.493	0.748	0.468	0.328	0.245	0.192	0.155	0.128

Table 3. Speedup factor in terms of maximum effective time-step size  $\Delta t_{opt}/s$  relative to  $RK_{4,4}$  for FRDG.

Stages	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
2	1.43×	1.42×	1.41×	1.40×	1.40×	1.40×	1.40×	1.41×
3	1.69×	1.70×	1.70×	1.71×	1.73×	1.74×	1.75×	1.75×
4	1.77×	1.77×	1.79×	1.81×	1.83×	1.85×	1.87×	1.88×
5	1.81×	1.80×	1.82×	1.85×	1.87×	1.90×	1.92×	1.94×
6	1.82×	1.81×	1.83×	1.86×	1.89×	1.92×	1.94×	1.96×
7	1.83×	1.81×	1.84×	1.87×	1.90×	1.93×	1.95×	1.98×

then generate optimal RK schemes, with values of  $\Delta t_{opt}$  obtained using Algorithm 2 summarized in Table 4 for FRSD schemes with polynomial degrees  $k = 1$  to  $k = 8$  using between  $s = 2$  and  $s = 7$  RK stages. These results are also plotted in Figure 3. Tabulated stability polynomial coefficients and corresponding Butcher tableau coefficients are provided as Electronic Supplementary Material. We observe that, for a given polynomial degree, optimal RK schemes with more stages have a larger stability limit. Also, for a given number of stages, optimal RK schemes for higher degree FRSD schemes have a smaller stability limit. Furthermore, we note the similarity of the results with those obtained for FRDG schemes of the same polynomial degree. This is because  $c_{SD} \rightarrow 0 = c_{DG}$  for large values of  $k$  where  $c_{SD}$  and  $c_{DG}$  are the coefficients that recover the FRSD and FRDG schemes, respectively, following Vincent et al. [30]. Therefore, the higher-order FRSD schemes have similar eigenspectra to their FRDG counterparts and recover similar optimal RK schemes. Also similar to the previous results using the FRDG scheme, we observe speedup factors in excess of 1.80× for all solution polynomial degrees. This suggests that the optimal RK schemes developed in the current study could be nearly twice as fast as conventional methods for converging the ACM in pseudo time when using FRSD. Furthermore, we have demonstrated that optimal RK schemes can be readily generated for various FR correction functions.

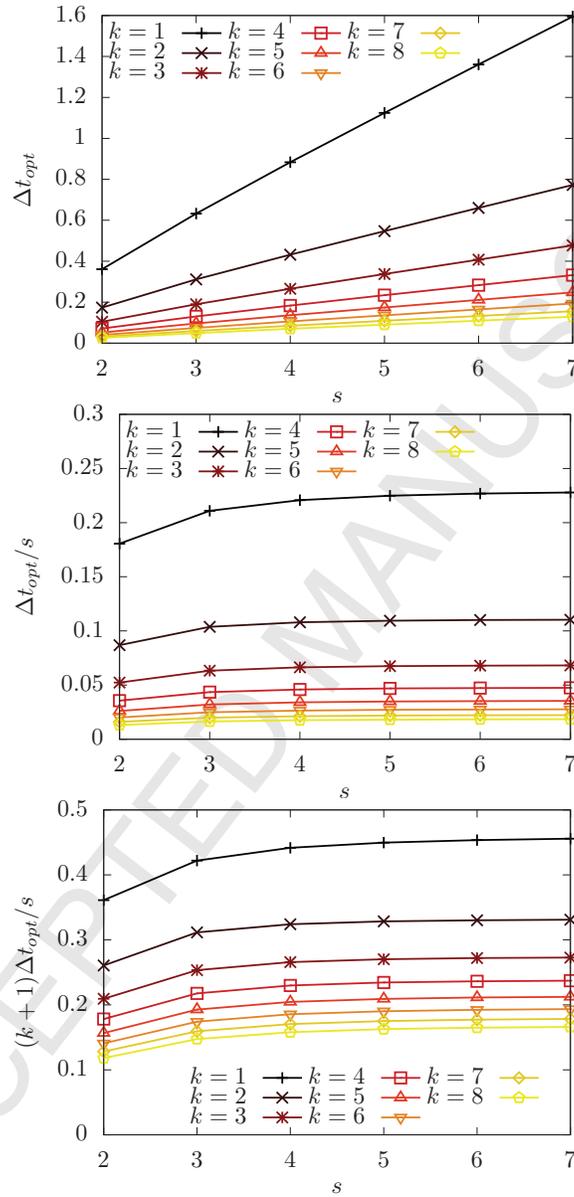


Figure 3. Optimal time-step  $\Delta t_{opt}$  (top), time-step per stage (middle), and time-step per stage normalized by effective resolution (bottom) for optimal RK schemes using FRSD.

Table 4. Values of  $\Delta t_{opt}$  for optimal RK schemes of order  $k = 1$  for FRSD.

Stages	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
2	0.361	0.173	0.104	0.071	0.052	0.040	0.032	0.026
3	0.632	0.311	0.190	0.130	0.096	0.074	0.059	0.049
4	0.883	0.432	0.265	0.183	0.136	0.105	0.085	0.070
5	1.124	0.547	0.337	0.234	0.174	0.135	0.109	0.090
6	1.360	0.660	0.407	0.283	0.211	0.164	0.132	0.109
7	1.594	0.772	0.477	0.332	0.247	0.193	0.156	0.129

Table 5. Speedup factor in terms of maximum effective time-step size  $\Delta t_{opt}/s$  relative to  $RK_{4,4}$  for FRSD.

Stages	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
2	1.43×	1.42×	1.40×	1.40×	1.40×	1.40×	1.40×	1.40×
3	1.67×	1.70×	1.70×	1.71×	1.72×	1.74×	1.75×	1.75×
4	1.75×	1.76×	1.79×	1.81×	1.83×	1.85×	1.87×	1.88×
5	1.78×	1.79×	1.81×	1.84×	1.87×	1.90×	1.92×	1.94×
6	1.80×	1.80×	1.83×	1.86×	1.89×	1.92×	1.94×	1.96×
7	1.81×	1.80×	1.83×	1.87×	1.90×	1.93×	1.95×	1.98×

## 5 Numerical Test Cases

### 5.1 SD7003

We investigate transitional and turbulent flow over an SD7003 airfoil [31] to assess the performance benefits of the optimal RK schemes for ILES of the incompressible Navier-Stokes equations. We use  $k = 1, 2$ , and 3 FRDG schemes with their correspondingly optimal  $RK_{7,1}$  schemes and the classical  $RK_{4,4}$  scheme to solve the pseudo time problem in the ACM. Each simulation is run using a modified version of PyFR 1.6.0 [32] with full details supplied as Electronic Supplementary Material. Each case is run at an angle of attack  $\alpha = 8^\circ$  and Reynolds number  $Re = \frac{V_\infty c}{\nu} = 60,000$ , where  $V_\infty$  is the freestream velocity and  $c$  is the chord length. We used the same quadratically curved hexahedral mesh as Vermeire et al. [13] that has 138,024 elements and a span of  $0.2c$ . This test case is commonly used to examine the suitability of numerical schemes for predicting separation, transition, and turbulent flow. It has been studied previously by, for example, Visbal and collaborators including Visbal et al. [33], Garmann et al. [34], and Beck et al. [35] using a DG Spectral Element Method (DGSEM). The characteristic features of the flow include laminar separation on the upper surface of the airfoil, which then reattaches further downstream forming a laminar separation bubble. The flow transitions to turbulence partway along this separation bubble, creating a turbulent wake behind the airfoil.

We start each simulation using solution polynomials of degree  $k = 1$ . These are run for a total of  $30t_c$ , where  $t_c = \frac{c}{V_\infty}$  is a flow convective time. After  $30t_c$  each set of simulations is then restarted and run for an additional  $30t_c$  using  $k = 2$  and, in the case of the optimal RK scheme simulation, switched to the correspondingly optimal  $RK_{7,1}$  scheme for that polynomial degree. This process is then repeated for each polynomial degree until all simulations have been run for a total of  $30t_c$ . Statistics are collected over the final  $20t_c$  of each simulation at each polynomial degree, allowing any transient effects from startup or increasing the polynomial degree to be damped. We use BDF2 for physical time integration with a fixed time-step of  $\Delta t = 10^{-3}t_c$  and the velocity residuals of the pseudo time problem are converged to  $5 \times 10^{-4}$  at each physical time step. Each scheme was run using its maximum possible stable pseudo time-step, which was acquired via bisection with the constraint that the velocity residuals did not diverge.

Results in Figure 4 show isosurfaces of q-criterion for each solution polynomial degree using  $RK_{4,4}$  and their corresponding optimal  $RK_{7,1}$  schemes in pseudo time. As expected, as the solution polynomial degree is increased with a fixed number of elements the qualitative fidelity of the scale-resolving ILES simulations increases. All three polynomial degrees exhibit a short laminar separation bubble on the leading edge of the airfoil, transition to turbulent flow, and a turbulent wake that propagates downstream. This behaviour is consistent with previous studies using high-order compressible solvers at low Mach numbers. Qualitatively, the results obtained using  $RK_{4,4}$  and the optimal  $RK_{7,1}$  schemes are similar. The measured lift and drag coefficients for each simulation using  $RK_{4,4}$  and the optimal  $RK_{7,1}$  schemes are shown in Table 6 for each solution polynomial degree. These values are generally consistent with those available from similar studies using low Mach number compressible solvers [33, 34, 35]. Importantly, we note that the measured lift and drag coefficients are not sensitive to our choice of pseudo time stepping scheme, with only minor variations observed between the  $RK_{4,4}$  and optimal  $RK_{7,1}$  scheme results. These variations are within the margin of error expected using the the finite  $20t_c$  averaging period.

To assess the relative performance of each scheme we have plotted samples of their convergence history in Figure 5 towards the end of each simulation. It is clear from these figures that the  $RK_{7,1}$  schemes converge significantly faster per iteration than the  $RK_{4,4}$  scheme. In fact, the optimal schemes can complete four to five time-steps in the same number of iterations that it takes the conventional scheme to complete one. Furthermore, we note that simulations using the  $RK_{7,1}$  schemes were between  $1.89\times$  and  $2.11\times$  faster in terms of wall-clock time than those using the classical  $RK_{4,4}$  scheme as shown in Table 7. These results are consistent with results of our theoretical analysis presented in Table 3 for the FRDG scheme. Therefore, we observe that the proposed  $RK_{7,1}$  schemes are able to achieve significant speedup over conventional explicit pseudo time integrators, without any adverse impact on

simulation accuracy.

Table 6. Lift and drag coefficients for the SD7003 airfoil test case.

		$k = 1$	$k = 2$	$k = 3$
$C_D$	RK <sub>4,4</sub>	0.051	0.049	0.049
	RK <sub>7,1</sub>	0.051	0.049	0.049
$C_L$	RK <sub>4,4</sub>	0.945	0.924	0.915
	RK <sub>7,1</sub>	0.947	0.925	0.912

Table 7. Speedup for the SD7003 airfoil test cases using optimal RK schemes with  $s = 7$  relative to the classical RK<sub>4,4</sub> scheme.

	$k = 1$	$k = 2$	$k = 3$
Speedup	1.89×	1.82×	2.11×

## 5.2 Turbulent Jet

To further demonstrate the suitability of the optimal RK schemes for ILES we consider an incompressible turbulent round jet at Reynolds number  $Re = \frac{V_0 d_{jet}}{\nu} = 10,000$  where  $d_{jet}$  is the jet diameter, and  $V_0$  is the midline velocity of the inflow. Experimental data for this test case is summarized by Lipari et al. [36] and previous numerical studies include DNS at  $Re = 5,000$  by Boersma [37] and explicitly filtered LES at  $Re = 11,000$  by Bogey and Bailly [38] performed with compressible solvers at Mach numbers  $Ma = 0.6$  and  $0.9$ , respectively.

Figure 6 depicts a portion of the mesh and the computational domain. A circular 2D mesh of diameter  $D = 48d_{jet}$  was extruded to  $L_x = 100d_{jet}$  in 250 equal increments. The element count of the cylindrical 3D mesh is 247,250 hexahedra and 596,500 prisms. Additionally, in Figure 6,  $x_0$  refers to the streamwise location of the virtual origin which is the starting point of the self-similar region, associated with a linear velocity decay and spreading rate [36]. The self-similar region is defined in terms of the self-similarity coordinate

$$\eta = \frac{r}{x - x_0}, \quad (50)$$

where  $r = \sqrt{y^2 + z^2}$ . The inflow profile of the jet is imposed via

$$\frac{V_{jet}(r)}{V_0} = \frac{1}{2} - \frac{1}{2} \tanh(\delta(r - r_{jet})), \quad (51)$$

where  $r_{jet} = 0.5d_{jet}$  and  $\delta = 40d_{jet}$ . Additionally, a sponge layer given by a source term

$$S_i = -(u_i - u_i^{out}) \left[ \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{1}{2} (x - (L_x - L_{spg})) \right) \right], \quad (52)$$

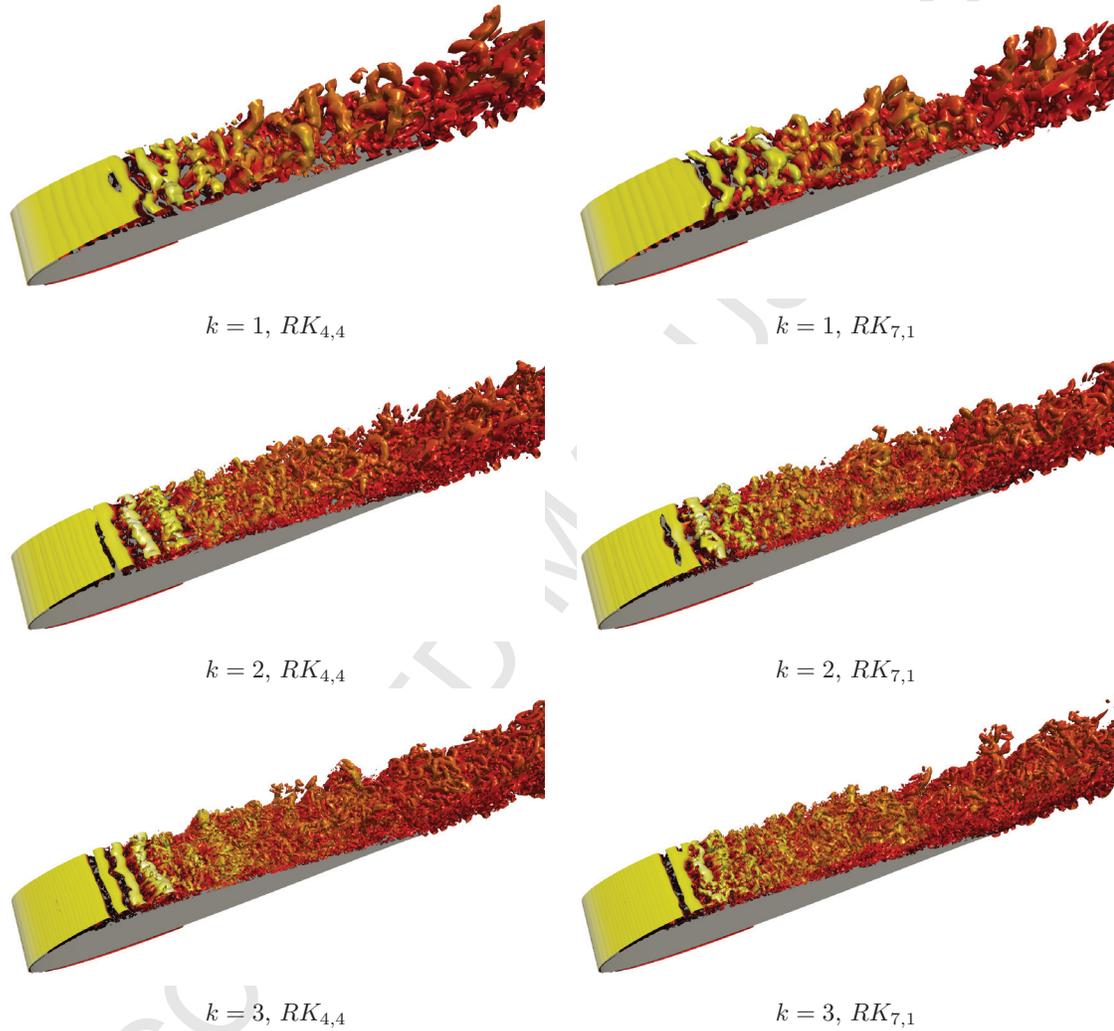


Figure 4. Isosurfaces of Q-criterion coloured by velocity magnitude for the SD7003 test case with solution polynomial degrees of  $k = 1, 2,$  and  $3$  using both  $RK_{4,4}$  and  $RK_{7,1}$  schemes.

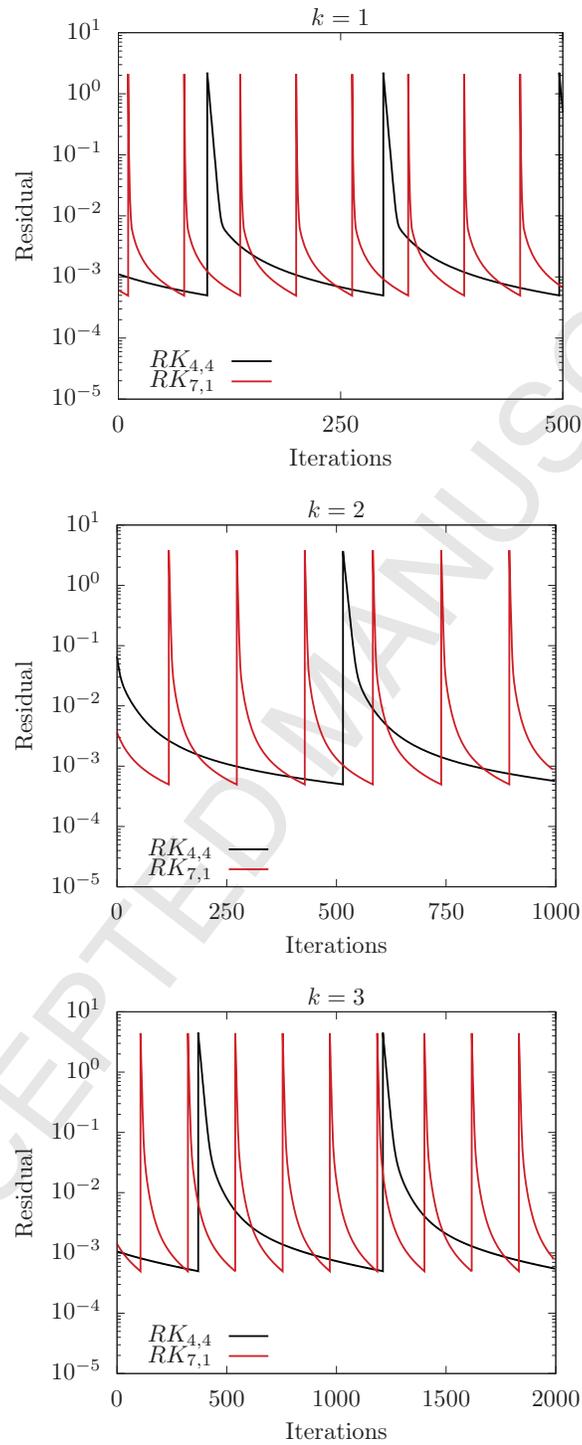


Figure 5. A sample of the convergence history for the SD7003 test case with solution polynomial degrees of  $k = 1, 2$ , and  $3$  using both  $RK_{4,4}$  and  $RK_{7,1}$  schemes. Spikes in the convergence history correspond to the start of a new physical time step.

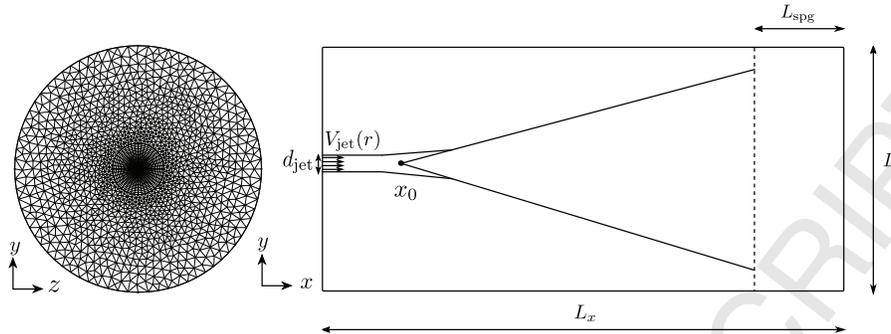


Figure 6. Computational grid in the  $y - z$  plane at  $x = 0$  together with a schematic of the simulation setup in the  $y - x$  plane at  $z = 0$ . The virtual origin is located at  $(x_0, 0, 0)$ .

is added to  $\mathcal{R}$  to dissipate transient waves before the jet reaches the outlet boundary. The distance from the outlet to the point of inflexion of the hyperbolic tangent that gradually dissipates the flow is  $L_{\text{spg}} = 10d_{\text{jet}}$ , and the target outflow state variables are  $u^{\text{out}} = \{17 \ 0 \ 0 \ 0\}^T$ . A no-slip boundary condition is imposed at the vertical front interface outside the jet inflow zone and a pressure outlet boundary condition is used for the outlet. The lateral far-field wall is specified as a non-entraining slip wall boundary. Hence, the jet receives entrainment by drawing fluid from the edges of the domain, which is evident by a small backflow at large  $r$ .

A single simulation was performed with a solution polynomial order  $k = 4$ , and  $\zeta = 2.5$ . The BDF2 scheme was used for physical time and the optimal  $RK_{7,1}$  scheme for  $k = 4$  was used in pseudo-time. Constant time steps of  $\Delta t = 0.008t_c$  and  $\Delta \tau = 0.0054t_c$ , with  $t_c = \frac{d_{\text{jet}}}{V_0}$ , were used throughout the simulation. The jet was initially developed up to  $1,040t_c$  to damp initial transients using 15 pseudo-iterations within each physical time step. The simulation was then restarted with a convergence criterion of  $5 \times 10^{-5}$  for the velocity residuals and statistics were collected up to  $2440t_c$ . The real time steps converged within 18 to 24 pseudo-iterations, resulting in the  $L^2$ -norm of divergence  $\nabla \cdot u = \frac{1}{\zeta} \frac{\partial p}{\partial \tau}$  being approximately  $6 \times 10^{-4}$ .

Figure 7 shows a volume rendering of the instantaneous velocity field to visualise the shape of the jet. The experimental study by Panchapakesan and Lumley [39] at  $Ma \approx 0.01 - 0.02$  and  $Re = 11,000$  are used as a reference for all flow statistics. To find the location of the virtual origin, the time-averaged midline axial velocity decay was shifted in  $x$  to fit a linear constant decay rate through the origin. The midline velocity decay shifted by  $x_0 = 3.2d_{\text{jet}}$  is shown in Figure 8a together with the experimental rate observed by Panchapakesan and Lumley [39]. We observe the our predicted



Figure 7. Volume rendering of the instantaneous volumetric velocity field at  $t = 1220$ .

linear velocity decay region matches with the reference before the sponge regions starts gradually dissipating the velocity. Figure 8b shows the average axial velocity with respect to the self-similarity coordinate  $\eta$ . In addition to averaging in time, the results were spatially averaged along conical surfaces defined by  $\eta(R, x) = \frac{R}{x-x_0}$  in the self-similar region  $24d_{\text{jet}} \leq x \leq 60d_{\text{jet}}$  and normalised by the mean midline velocity  $v_c$ . From Figure 8b, we can see that our mean axial velocities agree with the reference data across the entire self-similarity region. Figures 9a and 9b show the mean axial and radial velocity fluctuations in the self similar region. Both graphs indicate that the simulation is able to accurately capture velocity fluctuations in agreement with the experimental data. These results demonstrate that the optimal RK schemes combined with a BDF scheme are suitable for simulating turbulent flows, and that this approach can correctly predict mean flow and turbulent quantities.

## 6 Conclusions

In this study we have generated optimal RK schemes for convergence of the ACM using dual time-stepping. Noting that our approach is generalizable

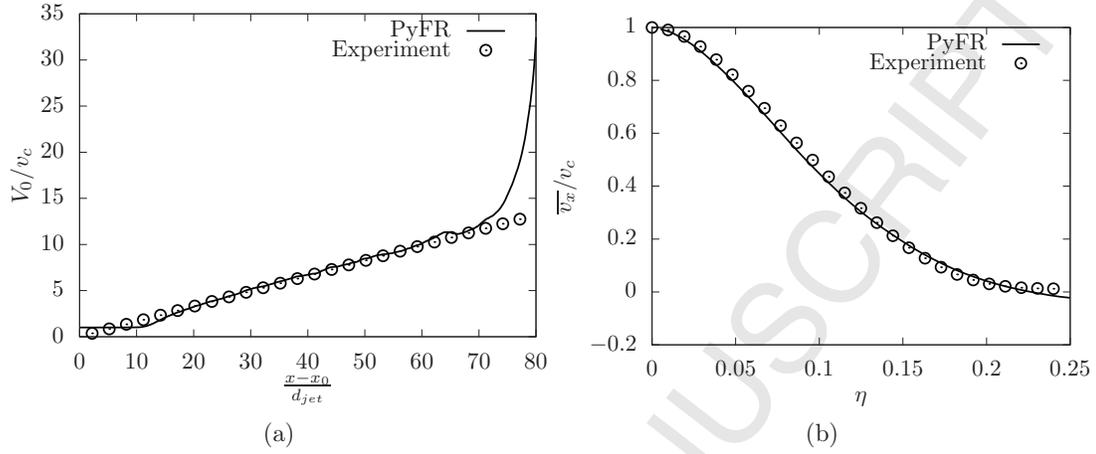


Figure 8. (a) The mean axial mid-line velocity decay rate. (b) The mean axial velocity along the self-similarity coordinate.

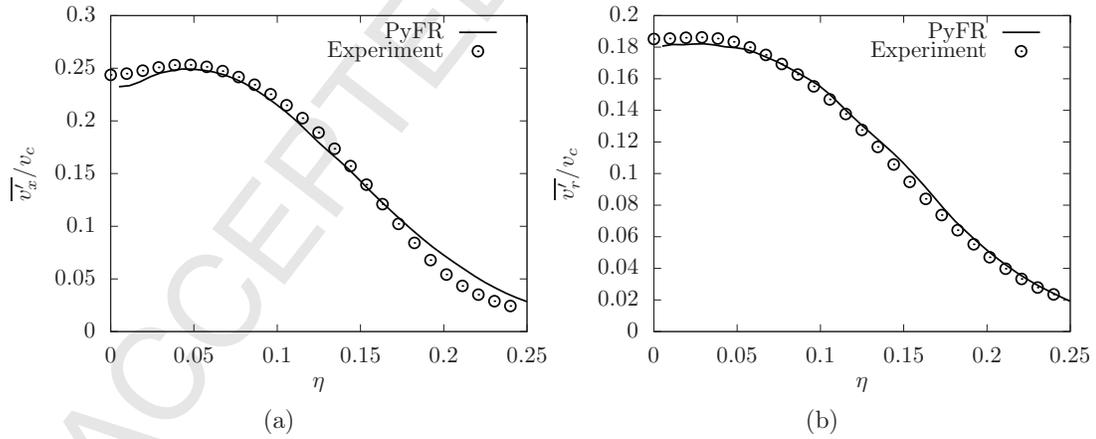


Figure 9. (a) Self-similar mean axial root-mean-square velocity fluctuations. (b) Self-similar mean radial root-mean-square velocity fluctuations.

to a variety of different spatial discretizations, we have demonstrated its utility for the FR approach. Specifically, we have generated optimal schemes for the FRDG and FRSD correction functions, recovering the DG and SD methods respectively, for spatial polynomial degrees of  $k = 1$  to  $k = 8$  and for RK schemes with  $s = 2$  to  $s = 7$  stages. These schemes were optimized in the context of linear advection, predicting speedup factors in excess of  $1.80\times$  relative to the classical  $RK_{4,4}$  scheme. In practice, we were able to demonstrate speedup factors for incompressible ILES of flow over an SD7003 airfoil between  $1.89\times$  and  $2.11\times$ . Furthermore, we have demonstrated the utility of the schemes for incompressible ILES of a turbulent jet, achieving good agreement with experimental data. In summary, these new schemes yield a significant speedup when solving the ACM via dual time-stepping. Furthermore, they are easy to implement and only require modification of a small number of Butcher tableau coefficients for the pseudo time solver. Future work should investigate the utility of these optimized schemes in the context of multigrid and implicit preconditioning strategies for convergence acceleration. Furthermore, the influence of diffusion, boundary conditions, and mesh topology on the optimization procedure could be explored.

## Acknowledgements

BCV would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [RGPAS-2017-507988, RGPIN-2017-06773]. This research was enabled in part by support provided by West-Grid ([www.westgrid.ca](http://www.westgrid.ca)), and Compute Canada ([www.computecanada.ca](http://www.computecanada.ca)) via a Resources for Research Groups allocation. NAL and PEV would like to thank the Engineering and Physical Sciences Research Council and BAE Systems for their support via Industrial CASE doctoral training grant and an Early Career Fellowship (EP/K027379/1) and the Swiss National Supercomputing Centre for access to the Piz Daint graphical processing unit cluster.

## Data Statement

Data Statement: Data relating to the results in this manuscript can be downloaded as Electronic Supplementary Material under a CC-BY-NC-ND 4.0 license.

## References

- [1] A.J. Chorin. A numerical methods for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135:118 – 125, 1997.
- [2] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, 1991. American Institute of Aeronautics and Astronautics. AIAA Paper 1991-1596.
- [3] A. Jameson and S. Yoon. Lower-upper implicit schemes with multiple grids for the Euler equations. *AIAA Journal*, 25(7):929 – 935, 1987.
- [4] Christopher Cox, Chunlei Liang, and Michael W. Plesniak. A high-order solver for unsteady incompressible navier–stokes equations using the flux reconstruction method on unstructured grids with implicit dual time stepping. *Journal of Computational Physics*, 314:414 – 435, 2016.
- [5] B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. general framework. *Mathematics of Computation*, 52(186):411–435, 1989.
- [6] B. Cockburn, S. Hou, and C. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. the multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

- [7] Y. Liu, M. Vinokur, and Z. J. Wang. Discontinuous spectral difference method for conservation laws on unstructured grids. In C. Groth and D. W. Zingg, editors, *Computational Fluid Dynamics 2004*, pages 449–454. Springer Berlin Heidelberg, January 2006.
- [8] D. A. Kopriva. A conservative staggered-grid Chebyshev multidomain method for compressible flows. II: a semi-structured method. Technical report, March 1996.
- [9] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, June 2007. American Institute of Aeronautics and Astronautics. AIAA Paper 2007-4079.
- [10] B. C. Vermeire, J. S. Cagnone, and S. Nadarajah. ILES using the correction procedure via reconstruction scheme. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, TX, January 2013. American Institute of Aeronautics and Astronautics. AIAA Paper 2013-1001.
- [11] B. C. Vermeire, S. Nadarajah, and P. G. Tucker. Canonical test cases for high-order unstructured implicit large eddy simulation. In *AIAA Science and Technology Forum and Exposition (SciTech 2014)*, National Harbor, Maryland, January 2014. American Institute of Aeronautics and Astronautics. AIAA Paper 2014-0935.
- [12] B. C. Vermeire, S. Nadarajah, and P. G. Tucker. Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme. *International Journal for Numerical Methods in Fluids*, 82(5):231–260, 2016. doi:10.1002/flid.4214.
- [13] B.C. Vermeire, F.D. Witherden, and P.E. Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 334:497 – 521, 2017.
- [14] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible navier–stokes equations. *Journal of Computational Physics*, 218(2):794 – 815, 2006.
- [15] Khosro Shahbazi, Paul F. Fischer, and C. Ross Ethier. A high-order discontinuous Galerkin method for the unsteady incompressible navier–stokes equations. *Journal of Computational Physics*, 222(1):391 – 407, 2007.

- [16] B. Cockburn, N. Nguyen, J. Peraire. A hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. In *48th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Orlando, FL, 2010. American Institute of Aeronautics and Astronautics. AIAA Paper 2010-362.
- [17] P. Vincent, F. Witherden, B. Vermeire, J.S. Park, and A. Iyer. Towards green aviation with python at petascale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '16*, pages 1:1–1:11, Piscataway, NJ, USA, 2016. IEEE Press.
- [18] N.A. Loppi, F.D. Witherden, A. Jameson, and P.E. Vincent. A high-order cross-platform incompressible Navier–Stokes solver via artificial compressibility with application to a turbulent jet. *Computer Physics Communications*, 233:193–205, 2018.
- [19] E. Hairer and H. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer, 2 edition, 1996.
- [20] D.I. Ketcheson and A.J. Ahmadi. Optimal stability polynomials for numerical integration of initial value problems. *Communications in Applied Mathematics and Computational Science*, 7(2):247 – 271, 2012.
- [21] D.I. Ketcheson, M. Parsani, and A.J. Ahmadi. Rk-opt: Software for the design of Runge-Kutta methods, Apr 2013.
- [22] M. Parsani, D.I. Ketcheson, and W. Deconinck. Optimized explicit Runge-Kutta schemes for the spectral difference method applied to wave propagation problems. *SIAM Journal on Scientific Computing*, 35(2):957 – 986, 2013.
- [23] E.J. Kubatko, B.A. Yeager, and D.I. Ketcheson. Optimal strong-stability-preserving runge–kutta time discretizations for discontinuous Galerkin methods. *Journal of Scientific Computing*, 60(2):313–344, Aug 2014.
- [24] Matlab 2016a, 2016. The MathWorks, Natick, MA, USA.
- [25] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [26] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).

- [27] P. E. Vincent, P. Castonguay, and A. Jameson. Insights from von Neumann analysis of high-order flux reconstruction schemes. *Journal of Computational Physics*, 230(22):8134–8154, September 2011.
- [28] B.C. Vermeire and P.E. Vincent. On the behaviour of fully-discrete flux reconstruction schemes. *Computer Methods in Applied Mechanics and Engineering*, 315:1053 – 1079, 2017.
- [29] B.C. Vermeire and P.E. Vincent. On the properties of energy stable flux reconstruction schemes for implicit large eddy simulation. *Journal of Computational Physics*, 327:368 – 388, 2016.
- [30] P. E. Vincent, P. Castonguay, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, April 2011.
- [31] M. S. Selig, J. F. Donovan, and D. B. Fraser. *Airfoils at Low Speed*. Stokely, 1989.
- [32] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An Open Source Framework for Solving Advection-Diffusion Type Problems on Streaming Architectures using the Flux Reconstruction Approach. *Computer Physics Communications*, 185(11):3028–3040, November 2014.
- [33] M. R. Visbal, R. E. Gordnier, and M. C. Galbraith. High-fidelity simulations of moving and flexible airfoils at low reynolds numbers. *Experiments in Fluids*, 46(5):903–922, May 2009.
- [34] D. J. Garmann, M. R. Visbal, and P. D. Orkwis. Comparative study of implicit and subgrid-scale model large-eddy simulation techniques for low-reynolds number airfoil applications. *International Journal for Numerical Methods in Fluids*, 71(12):1546–1565, 2013.
- [35] A.D. Beck, T. Bolemann, D. Flad, H. Frank, G.J. Gassner, F. Hindenlang, and C.D. Munz. High-order discontinuous Galerkin spectral element methods for transitional and turbulent flow simulations. *International Journal for Numerical Methods in Fluids*, 76(8):522–548, 2014.
- [36] G. Lipari and P. K. Stansby. Review of experimental data on incompressible turbulent round jets. *Flow, Turbulence and Combustion*, 87(1):79–114, 2011.
- [37] B. J. Boersma. Numerical simulation of the noise generated by a low Mach number, low Reynolds number jet. *Fluid Dynamics Research*, 35(6):425–447, 2004.
- [38] C. Bogey and C. Bailly. Turbulence and energy budget in a self-preserving round jet: direct evaluation using large eddy simulation. *Journal of Fluid Mechanics*, 627:129, 2009.

- [39] N. R. Panchapakesan and J. L. Lumley. Turbulence measurements in axisymmetric jets of air and helium. I - Air jet. II - Helium jet. *Journal of Fluid Mechanics*, 246(1993):197 – 223, 1993.