AIAA SciTech Forum
11–15 & 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-0156



# High-Order Space-Time Discontinuous Galerkin Discretization Using Tensor-Product Formulations

Sung-Hwan Yoon[1] and Dimitri J. Mavriplis[2]
*Department of Mechanical Engineering, University of Wyoming, Laramie, Wyoming, 82071-3295*



## Abstract

We investigate the use of higher order time integration schemes for discontinuous Galerkin (DG) spatial discretizations using tensor product formulations. A space-time DG formulation is used to develop DG in time temporal discretizations, which are then compared with fully implicit Runge Kutta time integration methods. These temporal discretizations are solved at each time step using an explicit pseudo-time stepping approach, which can be accelerated using a p-multigrid solver in pseudo-time. For the cases tested herein, the various temporal discretizations achieve or exceed their design orders of accuracy. The pseudo time stepping approach is shown to be feasible for all schemes, provided the FIRK schemes are written in a form similar to the DGT schemes.

## I. Introduction

We are interested in the development of implicit time-stepping approaches for CFD turbulence-scale resolving methods such as large eddy simulations (LES). In previous work we have developed a high-order discontinuous Galerkin (DG) method which incorporates dynamic mesh adaptation through the p4est library [1] operating on an overset off-body mesh for high resolution capturing of vortical and wake flows for rotorcraft and wind turbine applications [2, 3]. The use of very high order spatial discretizations is motivated by the superior accuracy and low cost of these discretizations, particularly when constructed using a tensor product formulation [2, 4, 5]. However, the explicit time step limit of the current approach becomes very restrictive as the order of the discretization is raised, as well as in regions where the dynamic AMR results in small mesh elements, leading to the requirement of considering implicit time-stepping methods.

In the construction of suitable implicit time-stepping methods for high-order discretizations, there are various factors that must be taken into account. Firstly, a suitable temporal discretization must be chosen, most notably one which exhibits attractive accuracy and stability properties. Additionally, the evaluation and solution of the resulting implicit systems must be efficient enough that they do not compromise the efficiency of the tensor-product formulation which makes the use of very high order discretizations feasible in the first place.

For a general finite-element discretization, using basis functions constructed from polynomials of order p, the number of degrees of freedom within an element scales as $N=(p+1)^d$, where d denotes the number of dimensions, and the cost of evaluating the residual vector scales as $N^2$ or $(p+1)^{2d}$. However, using basis functions constructed as tensor products of one-dimensional polynomials, the cost of a residual evaluation can be reduced to $(p+1)^{d+1}$, making the use of very high order discretization more feasible. In order to capitalize on this favorable scaling, one approach to the development of time-implicit methods is to use a space-time discretization, where the time dimension is added to the formulation thus increasing d from d=3 to d=4 in the tensor product scaling. However, this scaling only applies to the evaluation of solution values or residuals and not to the solution of the non-linear systems that arise in the context of each implicit time step, where the Jacobian matrices within each element becomes dense. One approach to solving these implicit systems is to use a Newton-Krylov method with preconditioners that can be

[1] Assistant Research Scientist; email: sung.hwan.yoon@uwyo.edu
[2] Professor, AIAA Associate Fellow; email: mavripl@uwyo.edu

evaluated using the tensor-product formulation [5, 6]. An alternate approach consists of using explicit time-stepping in pseudo-time, to drive the implicit system to a pseudo steady-state [7]. This approach can be further accelerated through the use of a p-multigrid method which employs explicit pseudo-time stepping on each level, where coarser levels are constructed by reducing the p-order discretization [8–10].

Although the DG in time (DGT) schemes that result from the space-time DG formulations have been used successfully with tensor-product formulations [5, 6], there are good reasons to investigate other implicit temporal discretizations. For example, it is well known that fully-implicit Runge-Kutta schemes (FIRK) (as opposed to diagonally implicit RK (DIRK) schemes) are closely related and in some cases equivalent to DGT schemes [11]. This means that FIRK schemes can be implemented with the same efficiency as DGT schemes within a tensor-product approach. The advantage of FIRK schemes is that they have been well studied and can be constructed to produce desirable accuracy and stability properties [12, 13]. For example, a Gauss FIRK scheme with s stages has a temporal design accuracy of 2$s$, whereas the Radau 2A schemes have accuracy of 2$s$-1. Here the number of stages is equivalent to the number of degrees of freedom in the time dimension, which corresponds to $s=p+1$ for DGT schemes. This suggests that FIRK schemes may be capable of much higher accuracy than equivalent DGT schemes, with some evidence to support this claim shown in references [14, 15]. However, in practice this may not always hold, since DGT schemes may themselves exhibit superconvergence for well resolved problems, and in the case of stiff problems, the FIRK schemes suffer from order-reduction, where the observed temporal order reduces to the stage order [12, 13]. For the aforementioned Gauss and Radau FIRK schemes, the stage order is equal to the number of stages [13], meaning that the expected order of accuracy for stiff problems is $s$, which is equivalent to the nominal order of accuracy of $p+1$ for DGT schemes.

Perhaps more importantly, the Gauss FIRK schemes are A-stable but not L-stable, whereas the Radau 2A FIRK schemes are known to be both A-stable and L-stable. Published numerical results comparing these schemes provide evidence that indeed the Radau schemes appear to be more effective for computational fluid dynamics problems, albeit on a limited set of problems [14, 15]. Given that DGT schemes using Gauss-Legendre points correspond most closely to the Gauss FIRK schemes, this may imply that the straight-forward use of DGT schemes based on the same basis functions and quadrature points as their spatial counterparts runs the risk of producing less than optimal implicit-time stepping schemes. Furthermore, nonlinear stability of FIRK schemes has also been studied in detail [12, 13], which may provide an additional tool for the development of nonlinearly stable time-implicit CFD codes.

Since we are interested in driving the solution of the resulting implicit systems with explicit pseudo-time stepping, the stability of this approach for the various temporal discretizations is also of importance. In reference [14], it was shown that various FIRK schemes are in fact unstable for explicit pseudo-time stepping in their original form, and a preconditioned form of these temporal schemes is derived, which is shown to be stable in pseudo-time. Remarkably, the preconditioned FIRK formulation derived in reference [14] corresponds to the natural formulation of DGT schemes, and we show in this paper that the corresponding DGT scheme is indeed stable in pseudo-time in its original form.

Finally, we also consider stability of these various temporal discertizations in pseudo time for small physical time steps. Although this is an area which has not been studied in detail for high-order temporal schemes, it is well known that dual time-stepping schemes for BDF2 temporal discretizations become unstable for small physical time steps unless certain pseudo-time terms are treated implicitly [16]. This is important because our goal is the design of a single implicit time-stepping scheme which can be use within an adaptive h and p refinement strategy with widely varying resolution. Additionally, our design criteria for the scheme include the ability to converge regions with small physical time steps, which may be close to the explicit time-step limit, at a cost that is not excessively larger than the use of explicit time stepping itself, thus avoiding the need to switch between implicit and explicit time steps in different regions of the simulation, along with the associated difficulties such strategies entail.

In the following section we first derive the semi-discrete formulation for the DG approach, where only the spatial terms are discretized. In section III, we derive the various temporal discretizations that are considered in this paper and show the equivalence between the DGT and FIRK schemes. In section IV, we study the stability of these schemes for explicit advancement in pseudo-time. Here we also formulate an implicit approach in pseudo-time that is expected to improve stability and performance for small physical time steps. In section V, we describe the p-multigrid strategy employed to solve some of these implicit temporal systems. The results section contains numerical results illustrating the spatial and temporal accuracy of these schemes, as well as the convergence effectiveness of the pseudo-time approaches used in this work. A summary and concluding remarks are given in section VII.

## II. Semi-Discrete Formulation for Discontinuous Galerkin Discretization

Let us consider the multi-dimensional hyperbolic conservation laws as follows.

$$\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = 0, \tag{1}$$

where **Q** is the state variable vector and **F** is the flux function vector. When we multiply by a set of test functions $\phi(\mathbf{x})$ and integrate over a space element, the weak statement for the governing equation can be expressed as:

$$\int_{\Omega_E} \left[\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q})\right] \phi(\mathbf{x})\, d\Omega_E = 0. \tag{2}$$

Integrating by parts and applying Green's theorem yields:

$$\int_{\Omega_E} \frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t} \phi(\mathbf{x}) d\Omega_E - \int_{\Omega_E} (\mathbf{F} \cdot \nabla)\phi(\mathbf{x}) d\Omega_E + \int_{\partial\Omega_E} (\mathbf{F} \cdot \mathbf{n})\, \phi(\mathbf{x})\, d(\partial\Omega_E) = 0. \tag{3}$$

The state vector is written as an expansion in terms of the basis functions and the solution degrees of freedom in space as:

$$\mathbf{Q} = \sum_{ij} \mathbf{Q}_{ij} \phi_i(x) \phi_j(y)$$

The first term is temporal derivative integral, and it can be simplified as

$$\int_{\Omega_E} \frac{\partial \mathbf{Q}(\mathbf{x}, t)}{\partial t} \phi(\mathbf{x}) d\Omega_E = \mathbf{M} \frac{\partial \mathbf{Q}_{ij}}{\partial t} \tag{4}$$

where **M** is the mass matrix defined as $\mathbf{M} = \int \phi_i\, \phi_j d\Omega_E$ for two-dimensional space. The second and third term of Eq.(4) are volume and surface integral, respectively. We can define a spatial residual, $\mathbf{R}_{ij}(\mathbf{Q})$, into which the volume and the surface integral are combined as follows.

$$\mathbf{R}_{ij}(\mathbf{Q}) = -\int_{\Omega_E} (\mathbf{F} \cdot \nabla)\phi(\mathbf{x}) d\Omega_E + \int_{\partial\Omega_E} (\mathbf{F} \cdot \mathbf{n})\, \phi(\mathbf{x})\, d(\partial\Omega_E) \tag{5}$$

Then, we can invoke isoparametric mapping and expand the solution in order to obtain a system of algebraic equations to be solved. The details of the expansion and integration processes for the tensor-product form of basis functions are explained in [17]. Substituting Eqs. (4) and (5) into Eq. (3) and omitting the *ij*-index for simplicity, a semi-discrete formulation can be written as:

$$\mathbf{M} \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{Q}) = 0 \tag{6}$$

In the present paper, we focus on two-dimensional space for simplicity. Nevertheless, the derived formulations can be extended to three-dimensional space in a straight-forward straightforward.

## III. Temporal Discretization

### A. Backward Difference Formula (BDF)

Starting from the set of ordinary differential equations given by Eq. (6), the formulations for the 1st and 2nd order accurate backward difference formulae (BDF1 and BDF2) are given respectively as:

$$\frac{\mathbf{M}}{\Delta t}[\mathbf{Q}^{n+1} - \mathbf{Q}^{n}] + \mathbf{R}(\mathbf{Q}^{n+1}) = 0 \tag{7}$$

$$\frac{\mathbf{M}}{\Delta t}\left[\frac{3}{2}\mathbf{Q}^{n+1} - 2\mathbf{Q}^{n} + \frac{1}{2}\mathbf{Q}^{n-1}\right] + \mathbf{R}(\mathbf{Q}^{n+1}) = 0 \tag{8}$$

Herein, let us define the left-hand-side of the Eq.(3) as $\mathbf{L}(\mathbf{Q})$, an unsteady residual that includes the temporal discretization as well as spatial discretization, for the corresponding BDF1 and BDF2 schemes as:

$$\text{BDF1:} \quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{\mathbf{M}}{\Delta t}[\mathbf{Q}^{n+1} - \mathbf{Q}^{n}] + \mathbf{R}(\mathbf{Q}^{n+1}) \tag{9}$$

$$\text{BDF2:} \quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{\mathbf{M}}{\Delta t}\left[\frac{3}{2}\mathbf{Q}^{n+1} - 2\mathbf{Q}^{n} + \frac{1}{2}\mathbf{Q}^{n-1}\right] + \mathbf{R}(\mathbf{Q}^{n+1}) \tag{10}$$

Then, Eqs. (9) and (10) are implicit time-accurate equations for the time advancement of the hyperbolic conservation laws given in the formulation of Eq. (1). The solution of these equations can be achieved by solving $\mathbf{L}(\mathbf{Q}^{n+1}) = 0$ at each time step. BDF schemes have a severe limitation when used in conjunction with a high-order spatial discretization. BDF schemes can be A-stable only up to second order (the Dahlquist barrier) [18]. Thus, we are interested in higher-order A-stable (and even L-stable) schemes including DGT and FIRK.

### B. Space-Time Formulation of Discontinuous Galerkin Discretization

In order to introduce the differences in the formulations between space DG and space-time DG, rather than a compact way described in [19], we derive the space-time DG formulation by multiplying the space DG formulation given as Eq. (3) by an additional basis function for the time dimension $\psi(t)$ and integrating it over a time interval $I^n = [t^n, t^{n+1}]$, where $t^{n+1} = t^n + \Delta t$.

$$\int_{I^n}\left[\int_{\Omega_E}\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t}\phi(\mathbf{x})d\Omega_E - \int_{\Omega_E}(\mathbf{F}\cdot\nabla)\phi(\mathbf{x})d\Omega_E + \int_{\partial\Omega_E}(\mathbf{F}\cdot\mathbf{n})\,\phi(\mathbf{x})\,d(\partial\Omega_E)\right]\psi(t)dt = 0 \tag{11}$$

Applying Eq.(5) yields

$$\int_{I^n}\int_{\Omega_E}\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t}\phi(\mathbf{x})\psi(t)d\Omega_E dt + \int_{I^n}\mathbf{R}(\mathbf{Q})\psi(t)dt = 0. \tag{12}$$

The first term can be written as:

$$\int_{I^n}\int_{\Omega_E}\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t}\phi(\mathbf{x},t)d\Omega_E dt = \int_{I^n}\int_{\Omega_E}\frac{\partial(\mathbf{Q}\phi)}{\partial t}d\Omega_E dt - \int_{I^n}\int_{\Omega_E}\mathbf{Q}\frac{\partial\phi}{\partial t}d\Omega_E dt \tag{13}$$

$$= \int_{\Omega_E}[\mathbf{Q}(\mathbf{x},t_-^{n+1})\phi(t_-^{n+1}) - \mathbf{Q}(\mathbf{x},t_-^{n})\phi(t_+^{n})]\,d\Omega_E - \int_{I^n}\int_{\Omega_E}\mathbf{Q}\frac{\partial\phi}{\partial t}d\Omega_E dt \tag{14}$$

where $\phi(\mathbf{x},t) = \phi(\mathbf{x})\psi(t)$ and the temporal basis function is assumed to be of the same form as the spatial basis function: $\psi(t) = \phi(t)$. Then, $\mathbf{L}(\mathbf{Q})$ can be defined as follows.

$$\text{DG in time:} \quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{1}{\Delta t}\left[\int_{\Omega_E}[\mathbf{Q}(\mathbf{x},t_-^{n+1})\phi(t_-^{n+1}) - \mathbf{Q}(\mathbf{x},t_-^{n})\phi(t_+^{n})]\,d\Omega_E - \int_{I^n}\int_{\Omega_E}\mathbf{Q}^{n+1}\frac{\partial\phi}{\partial t}d\Omega_E dt\right] + \frac{1}{\Delta t}\int_{I^n}\mathbf{R}(\mathbf{Q}^{n+1})\phi(t)dt \tag{15}$$

It should be noted that the solution vector in the space DG formulation of Eq. (3) is defined within a space element, while that in the space-time DG formulation of Eq. (11) is defined within a space-time element. For example, in two-dimensional space, the solution vectors can be expanded in terms of tensor-product basis functions respectively as follows:

$$\text{space DG:} \quad \mathbf{Q}(\mathbf{x},t) = \sum_{i,j}^{N} Q_{ij}(t)\phi_i(x)\phi_j(y) \tag{16}$$

$$\text{space-time DG:} \quad \mathbf{Q}(\mathbf{x},t) = \sum_{i,j,k}^{N} Q_{ijk}\phi_i(x)\phi_j(y)\phi_k(t) \tag{17}$$

Herein, let us define a sub-vector that consists of quadrature point values for spatial DG: $\mathbf{q} = Q_{ij}$. When the spatial $p$-order is one ($p_s = 1$), for instance, the sub-vector $\mathbf{q}$ can be written as:

$$\mathbf{q} = Q_{ij} = \begin{bmatrix} Q_{11} \\ Q_{21} \\ Q_{12} \\ Q_{22} \end{bmatrix}. \tag{18}$$

Then, we can also express the quadrature point values in a space-time element as $\mathbf{q}_k = Q_{ijk}$. For the temporal $p$-order $p_t$, it becomes

$$\mathbf{q}_k = Q_{ijk} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \vdots \\ \mathbf{q}_{N_\mathrm{t}} \end{bmatrix}, \tag{19}$$

where $N_\mathrm{t} = p_t + 1$. Similarly, we can define $\mathbf{r}_k$ for the residual vector. While the quadrature point values for the time interval $[t^n, t^{n+1}]$ are included in $\mathbf{q}_k^{n+1}$, boundary point values (solutions at $t^n$ and $t^{n+1}$) are not included since Gauss-Legendre quadrature points are defined only inside of the time element. The boundary values are calculated by solution expansion of $\mathbf{q}_k$ as follows.

$$\widehat{\mathbf{q}}^{n+1} = \sum_{k=1}^{N_\mathrm{t}} \mathbf{q}_k^{n+1}\, \phi_k(t_-^{n+1}) \tag{20}$$

where $\widehat{\mathbf{q}}^{n+1}$ is solution value at $t^{n+1}$. Upwind differencing is used at the time element boundaries and thus $\widehat{\mathbf{q}}^{n}$ is calculated by expanding $\mathbf{q}_k^n$ in the same way within the previous time element $[t^{n-1}, t^n]$ .

In this work, DGT temporal discretizations using a temporal order which is independent of the spatial order up to $p_t$=8 have been implemented and tested. In the following, we describe in more detail the temporal discretization for $p_t$=1, using a tensor product basis in space and time, with Gauss-Legendre quadrature points. This specific discretization is then compared with the corresponding Gauss FIRK scheme. First, the right-hand-side of Eq. (15) is expanded to obtain a system of algebraic equations. We begin by expanding the solution of $\mathbf{Q}(\mathbf{x}, t_-^{n+1})$ as follows:

$$\mathbf{Q}(\mathbf{x}, t_-^{n+1}) = \sum_{m,n,l=1}^{N} Q_{mnl}^{n+1}\phi_m(x)\phi_n(y)\phi_l(t_-^{n+1}) \tag{21}$$

$$= \sum_{m,n=1}^{N} \left\{ \sum_{l=1}^{N_\mathrm{t}} Q_{mnl}^{n+1}\phi_l(t_-^{n+1}) \right\} \phi_m(x)\phi_n(y) \tag{22}$$

$$= \sum_{m,n=1}^{N} \hat{Q}_{mn}^{n+1} \phi_m(x) \phi_n(y) \tag{23}$$

Then,

$$\int_{\Omega_E} \mathbf{Q}(\mathbf{x}, t_-^{n+1}) \phi(t_-^{n+1}) \, d\Omega_E = \phi_k(t_-^{n+1}) \int_{\Omega} \left\{ \sum_{m,n=1}^{N} \hat{Q}_{mn}^{n+1} \phi_m(x) \phi_n(y) \right\} \phi_i(x) \phi_j(y) dx dy \tag{24}$$

$$\approx \phi_k(t_-^{n+1}) \sum_{\lambda,\mu=1}^{N} \left\{ \sum_{m,n=1}^{N} \hat{Q}_{mn}^{n+1} \phi_m(x_\lambda) \phi_n(y_\mu) \right\} \phi_i(x_\lambda) \phi_j(y_\mu) \{ J \omega_\lambda \omega_\mu \} \tag{25}$$

$$= \phi_k(t_-^{n+1}) \hat{Q}_{ij}^{n+1} \{ J \omega_i \omega_j \} \tag{26}$$

$$= \mathbf{M} \hat{\mathbf{q}}^{n+1} \phi_k(t_-^{n+1}) \tag{27}$$

Here $J$ is the element Jacobian, and $\omega_i$ are the quadrature weights.
Similarly, $\int_{\Omega_E} \mathbf{Q}(\mathbf{x}, t_-^n) \phi(t_+^n) d\Omega_E = \mathbf{M} \hat{\mathbf{q}}^n \phi_k(t_+^n)$. Thus, the first integral of Eq. (15) can be written as:

$$\frac{1}{\Delta t} \int_{\Omega_E} [\mathbf{Q}(\mathbf{x}, t_-^{n+1}) \phi(t_-^{n+1}) - \mathbf{Q}(\mathbf{x}, t_-^n) \phi(t_+^n)] \, d\Omega_E = \frac{\mathbf{M}}{\Delta t} [\hat{\mathbf{q}}^{n+1} \phi_k(t_-^{n+1}) - \hat{\mathbf{q}}^n \phi_k(t_+^n)] \tag{28}$$

For the case of $p_t = 1$, by given in Eq. (20),

$$\hat{\mathbf{q}}^{n+1} = \sum_{k=1}^{N_t=2} \mathbf{q}_k^{n+1} \phi_k(t_-^{n+1}) = \mathbf{q}_1^{n+1} \phi_1(t_-^{n+1}) + \mathbf{q}_2^{n+1} \phi_2(t_-^{n+1}) \tag{29}$$

Thus, Eq. (28) can be expressed in a matrix form as:

$$\frac{1}{\Delta t} \int_{\Omega_E} [\mathbf{Q}(\mathbf{x}, t_-^{n+1}) \phi(t_-^{n+1}) - \mathbf{Q}(\mathbf{x}, t_-^n) \phi(t_+^n)] \, d\Omega_E = \frac{\mathbf{M}}{\Delta t} \left\{ \mathbf{\Phi}^{n+1} \begin{bmatrix} \mathbf{q}_1^{n+1} \\ \mathbf{q}_2^{n+1} \end{bmatrix} - \hat{\mathbf{\Psi}}^n \begin{bmatrix} \hat{\mathbf{q}}^n \\ \hat{\mathbf{q}}^n \end{bmatrix} \right\} \tag{30}$$

where

$$\mathbf{\Phi}^{n+1} = \begin{bmatrix} \phi_1(t_-^{n+1}) \phi_1(t_-^{n+1}) & \phi_1(t_-^{n+1}) \phi_2(t_-^{n+1}) \\ \phi_2(t_-^{n+1}) \phi_1(t_-^{n+1}) & \phi_2(t_-^{n+1}) \phi_2(t_-^{n+1}) \end{bmatrix}, \tag{31}$$

$$\hat{\mathbf{\Psi}}^n = \begin{bmatrix} \phi_1(t_+^n) & 0 \\ 0 & \phi_2(t_+^n) \end{bmatrix}. \tag{32}$$

When we apply the same expansion process, the second integral of Eq. (15) becomes:

$$-\frac{1}{\Delta t} \int_{I^n} \int_{\Omega_E} \mathbf{Q}^{n+1} \frac{\partial \phi}{\partial t} d\Omega_E dt = -\frac{1}{\Delta t} J \omega_i \omega_j \sum_{\nu=1}^{N_t=2} Q_{ij\nu}^{n+1} D_{k\nu} \omega_\nu \tag{33}$$

$$= -\frac{\mathbf{M}}{\Delta t} \sum_{\nu=1}^{N_t=2} \mathbf{q}_\nu^{n+1} D_{k\nu} \omega_\nu \tag{34}$$

$$= -\frac{\mathbf{M}}{\Delta t} \{ \mathbf{q}_1^{n+1} D_{k1} \omega_1 + \mathbf{q}_2^{n+1} D_{k2} \omega_2 \} \tag{35}$$

where $D_{k\nu} = \frac{\partial \phi_k(t_\nu)}{\partial t}$. Eq. (35) can be also expressed in a matrix form as:

$$-\frac{1}{\Delta t}\int_{I^n}\int_{\Omega_E} \mathbf{Q}^{n+1}\frac{\partial \phi}{\partial t} d\Omega_E dt = -\frac{\mathbf{M}}{\Delta t}\boldsymbol{D}_{\boldsymbol{\omega}}\begin{bmatrix}\mathbf{q}_1^{n+1}\\ \mathbf{q}_2^{n+1}\end{bmatrix} \tag{36}$$

where $\boldsymbol{D}_{\boldsymbol{\omega}} = \begin{bmatrix} D_{11}\omega_1 & D_{12}\omega_2 \\ D_{21}\omega_1 & D_{22}\omega_2\end{bmatrix}$.

The third integral of Eq. (15) is the spatial residual part, which can be written as

$$\frac{1}{\Delta t}\int_{I^n} \mathbf{R}(\mathbf{Q}^{n+1})\phi(t)dt = \frac{1}{\Delta t}\int_{I^n}\left\{\sum_{l=1}^{N}\mathbf{r}_l^{n+1}\phi_l(t)\right\}\phi_k(t)dt \tag{37}$$

$$\approx \frac{1}{\Delta t}\sum_{\nu=1}^{N}\left\{\sum_{l=1}^{N}\mathbf{r}_l^{n+1}\phi_l(t_\nu)\right\}\phi_k(t_\nu)\,(J_t\omega_\nu) \tag{38}$$

$$= \mathbf{r}_k^{n+1}\left(\frac{\omega_k}{2}\right) \tag{39}$$

where $J_t = \frac{dt}{dt'} = \frac{\Delta t}{2}$, t' being the non-dimensional time coordinate in the isoparametric mapped element. The corresponding matrix form is as follows:

$$\frac{1}{\Delta t}\int_{I^n} \mathbf{R}(\mathbf{Q}^{n+1})\phi(t)dt = \frac{1}{2}\boldsymbol{\Omega}\begin{bmatrix}\mathbf{r}_1^{n+1}\\ \mathbf{r}_2^{n+1}\end{bmatrix} \tag{40}$$

where $\boldsymbol{\Omega} = \begin{bmatrix}\omega_1 & 0\\ 0 & \omega_2\end{bmatrix}$. Combining Eqs. (30), (36), and (40) into Eq. (15) yields a matrix form of $\mathbf{L}(\mathbf{Q})$ as:

$$\text{DG in time:}\quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{\mathbf{M}}{\Delta t}\left\{(\boldsymbol{\Phi}^{n+1} - \boldsymbol{D}_{\boldsymbol{\omega}})\begin{bmatrix}\mathbf{q}_1^{n+1}\\ \mathbf{q}_2^{n+1}\end{bmatrix} - \widehat{\boldsymbol{\Psi}}^n\begin{bmatrix}\widehat{\mathbf{q}}^n\\ \widehat{\mathbf{q}}^n\end{bmatrix}\right\} + \frac{\boldsymbol{\Omega}}{2}\begin{bmatrix}\mathbf{r}_1^{n+1}\\ \mathbf{r}_2^{n+1}\end{bmatrix} \tag{41}$$

### C. Fully Implicit Runge-Kutta Scheme (FIRK)

Two and three stage Gauss schemes and Radau 2A schemes are investigated in Refs. [14,15]. Gauss schemes use Gauss-Legendre quadrature points for the multi-stages and thus the formulations are similar to the DG in time schemes resulting from our space-time formulation. For simplicity, let us consider two stage Gauss scheme. The two stage Gauss scheme takes the form as:

$$\text{FIRK:}\quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{\mathbf{M}}{\Delta t}\left\{\begin{bmatrix}\mathbf{q}_1^{n+1}\\ \mathbf{q}_2^{n+1}\end{bmatrix} - \begin{bmatrix}\widehat{\mathbf{q}}^n\\ \widehat{\mathbf{q}}^n\end{bmatrix}\right\} + \boldsymbol{A}\begin{bmatrix}\mathbf{r}_1^{n+1}\\ \mathbf{r}_2^{n+1}\end{bmatrix} \tag{42}$$

where the matrix A of coefficients is

$$\boldsymbol{A} = \begin{bmatrix}\frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6}\\ \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4}\end{bmatrix} \tag{43}$$

In Ref. [14], a preconditioned version (p-FIRK) was also studied, and it can be expressed as:

$$\text{p-FIRK:} \quad \mathbf{L}(\mathbf{Q}^{n+1}) = \frac{\mathbf{M}}{\Delta t}\left\{\boldsymbol{A}^{-1}\begin{bmatrix}\mathbf{q}_1^{n+1}\\ \mathbf{q}_2^{n+1}\end{bmatrix} - \boldsymbol{A}^{-1}\begin{bmatrix}\widehat{\mathbf{q}}^{n}\\ \widehat{\mathbf{q}}^{n}\end{bmatrix}\right\} + \begin{bmatrix}\mathbf{r}_1^{n+1}\\ \mathbf{r}_2^{n+1}\end{bmatrix} \tag{44}$$

As can be seen, although the original formulation of the FIRK scheme is substantially different than that of the DGT scheme above, the preconditioned form is very close to the DGT scheme, especially when considering that the $\boldsymbol{\Omega}$ matrix is diagonal and that the second term in the first set of brackets represents a constant term in both cases.

## IV. Pseudo-Time Stepping

Once we have the formulation of $\mathbf{L}(\mathbf{Q}^{n+1})$ for the various temporal discretizations, we are interested in solving $\mathbf{L}(\mathbf{Q}^{n+1}) = 0$ at every time step to advance in time. In order to solve this non-linear set of equations, pseudo-time stepping, also known as dual-time stepping, is an often used approach which seeks to mimic the physical time evolution of the solution by adding a pseudo-time term $\mathbf{M}\frac{\partial \mathbf{Q}}{\partial \tau}$ as [7]:

$$\mathbf{M}\frac{\partial \mathbf{Q}}{\partial \tau} + \mathbf{L}(\mathbf{Q}^{n+1}) = 0 \tag{45}$$

and integrating in pseudo-time $\tau$ until steady-state is achieved.

### A. Convergence Characteristics: FIRK vs. DGT

Before we start describing how to solve Eqn. (45), we need to make sure that the pseudo-time stepping approach converges to a steady-state solution. Following Ref. [14], we consider the application these temporal discretizations to the scalar equation:

$$\frac{du}{dt} = au$$

where $a$ is a complex coefficient lying in the left-half plane. A dual time stepping scheme applied for each temporal discretization to the above equation as:

$$\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\tau} = -\mathbf{M}^{-1}\mathrm{L}(\mathbf{Q})$$

results in a linear constant coefficient inhomogeneous system of the form:

$$\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\tau} = \mathbf{B}\mathbf{Q} + \mathbf{c} \tag{46}$$

where the B matrix is given by the corresponding entry in Table 1 below along with the constant c vector for each temporal discretization. The general solution takes the form of:

$$\mathbf{Q} = \sum_m \kappa_m \exp(\lambda_m \tau)\boldsymbol{\nu}_m + \mathrm{S}(\tau) \tag{47}$$

where $\kappa_m$ are arbitrary constants; $\lambda_m$ and $\boldsymbol{\nu}_m$ are eigenvalues and eigenvectors of the matrix $\mathbf{B}$, respectively. $\sum_m \kappa_m \exp(\lambda_m \tau)\boldsymbol{\nu}_m$ is the transient solution and $\mathrm{S}(\tau)$ is the steady-state solution. When $Re(\lambda_m) < 0$, the transient term will decay to zero as $\tau \to \infty$, so that the solution approaches $\mathrm{S}(\tau)$.

Table 1 shows the matrices and the associated eigenvalues for each temporal discretization. As pointed out in Ref. [14], the eigenvalues for FIRK could have a positive real part even when $Re(a) < 0$, while p-FIRK has both roots that have a negative real part whenever $Re(a) < 0$, establishing the feasibility of applying the pseudo-time integration approach to p-FIRK. Interestingly, as seen in Table 1, the eigenvalues of DGT are considerably similar to those of p-FIRK, which also shows the feasibility of applying pseudo-time stepping to DGT.

**Table 1 The matrices B and c in the inhomogeneous system of Eq. (46), and the eigenvalues of B**

| ` | **B** | **c** | $\lambda$ |
|---|---|---|---|
| FIRK | $a\mathbf{A}-\frac{1}{\Delta t}\mathbf{I}$ | $\frac{1}{\Delta t}\begin{bmatrix}\hat{q}^n\\ \hat{q}^n\end{bmatrix}$ | $\frac{1}{4}a-\frac{1}{\Delta t}\pm ia\sqrt{\frac{1}{48}}$ |
| p-FIRK | $a\mathbf{I}-\frac{1}{\Delta t}\mathbf{A}^{-1}$ | $\frac{1}{\Delta t}\mathbf{A}^{-1}\begin{bmatrix}\hat{q}^n\\ \hat{q}^n\end{bmatrix}$ | $a-\frac{3}{\Delta t}\pm i\frac{\sqrt{3}}{\Delta t}$ |
| DG in time | $a\left(\frac{\boldsymbol{\Omega}}{2}\right)-\frac{1}{\Delta t}(\boldsymbol{\Phi}^k-\mathbf{D}_{\boldsymbol{\omega}})$ | $\frac{1}{\Delta t}\widehat{\boldsymbol{\Psi}}^n\begin{bmatrix}\hat{q}^n\\ \hat{q}^n\end{bmatrix}$ | $\frac{a}{2}-\frac{1}{\Delta t}\pm i\frac{\sqrt{2}}{2\Delta t}$ |



### B. Explicit RK1 Pseudo-Time Stepping

This nonlinear system can be solved by Newton's method, or by nonlinear multigrid methods. In the present study, for simplicity, we use the explicit RK1 (or the 1st order accurate forward Euler) time stepping in pseudo-time, which gives

$$\frac{\mathbf{M}}{\Delta\tau}(\mathbf{Q}^{k+1}-\mathbf{Q}^k)+\mathbf{L}(\mathbf{Q}^{n+1})=0 \tag{48}$$

Here, $\mathbf{Q}^k$ is the pseudo-time level and $\mathbf{Q}^n$ is the physical-time level such that

$$\mathbf{Q}^{k+1}\approx\mathbf{Q}^k\rightarrow\ \mathbf{Q}^{n+1}\text{ as }k\rightarrow\infty\text{ and }\mathbf{L}(\mathbf{Q}^k)\rightarrow 0 \tag{49}$$

Thus, we can write Eq. (48) as:

$$\mathbf{Q}^{k+1}=\mathbf{Q}^k-\Delta\tau\mathbf{M}^{-1}\mathbf{L}(\mathbf{Q}^k) \tag{50}$$

This gives an explicit equation to be solved for $\mathbf{Q}^{k+1}$.

#### 1) BDF-RK1

Let us start with BDF schemes to see how we can solve Eq. (50) for each temporal discretization.

$$\text{BDF1:}\quad \mathbf{q}^{k+1}=\mathbf{q}^k-\frac{\Delta\tau}{\Delta t}[\mathbf{q}^k-\mathbf{q}^n]-\Delta\tau\mathbf{M}^{-1}\mathbf{r}^k \tag{51}$$

$$\text{BDF2:}\quad \mathbf{q}^{k+1}=\mathbf{q}^k-\frac{\Delta\tau}{\Delta t}\left[\frac{3}{2}\mathbf{q}^k-2\mathbf{q}^n+\frac{1}{2}\mathbf{q}^{n-1}\right]-\Delta\tau\mathbf{M}^{-1}\mathbf{r}^k \tag{52}$$

It should be noted that the values $\mathbf{q}^n$ and $\mathbf{q}^{n-1}$ are constants throughout the pseudo-time iteration. When $\frac{\Delta\tau}{\Delta t}$ is small, these formulations reproduce the steady-state time-stepping equations, if $\Delta\tau$ is used in the place of $\Delta t$. If $\Delta t\gg 1$, then $\Delta\tau$ explicit stability limit should be the same as the explicit time step for physical time stepping. If $\Delta t\approx 1$ or $\Delta t\ll 1$, then this becomes unstable unless $\Delta\tau$ is reduced substantially.

In the formulations of Eqns. (51) and (52), physical-time derivatives contribute to the right-hand-side of the equations, which means the contributions are treated explicitly. In order to maintain stability for small $\Delta t$ without reducing $\Delta\tau$, we need to treat the term which corresponds to $\mathbf{q}^{n+1}$ implicitly in the BDF formulations [16]. Formulations for the explicit or implicit physical time-derivative contribution (PTDC) can be combined as:

BDF1-RK1: $$\left[1+\varepsilon\frac{\Delta\tau}{\Delta t}\right]\mathbf{q}^{k+1}=\left[1+(\varepsilon-1)\frac{\Delta\tau}{\Delta t}\right]\mathbf{q}^{k}+\frac{\Delta\tau}{\Delta t}\mathbf{q}^{n}-\Delta\tau\mathbf{M}^{-1}\mathbf{r}(\mathbf{q}^{k}) \tag{53}$$

BDF2-RK1: $$\left[1+\varepsilon\frac{3}{2}\frac{\Delta\tau}{\Delta t}\right]\mathbf{q}^{k+1}=\left[1+(\varepsilon-1)\frac{3}{2}\frac{\Delta\tau}{\Delta t}\right]\mathbf{q}^{k}+\frac{\Delta\tau}{\Delta t}\left[2\mathbf{q}^{n}-\frac{1}{2}\mathbf{q}^{n-1}\right]-\Delta\tau\mathbf{M}^{-1}\mathbf{r}(\mathbf{q}^{k}) \tag{54}$$

where $\varepsilon$ is an implicit parameter which determines the physical-time derivative contribution to be explicit ($\varepsilon=0$) or implicit ($\varepsilon=1$). The second term on the right-hand-side is a constant, so for ε=1 the stability of these equations is governed by that of the steady-state problem (i.e. by the spatial residual) and we expect a constant stability limit in pseudo time which is independent of the physical time step size.

**2) DGT-RK1**

Similarly, formulations of the explicit or implicit physical time-derivative contribution (PTDC) for DG in time can be derived as follows.

DGT-RK1:

$$\left[\mathbf{I}+\varepsilon\frac{\Delta\tau}{\Delta t}(\boldsymbol{\Phi}^{k}-\boldsymbol{D}_{\boldsymbol{\omega}})\right]\begin{bmatrix}\mathbf{q}_1^{k+1}\\ \mathbf{q}_2^{k+1}\end{bmatrix}=\left[\mathbf{I}+(\varepsilon-1)\frac{\Delta\tau}{\Delta t}(\boldsymbol{\Phi}^{k}-\boldsymbol{D}_{\boldsymbol{\omega}})\right]\begin{bmatrix}\mathbf{q}_1^{k}\\ \mathbf{q}_2^{k}\end{bmatrix}+\frac{\Delta\tau}{\Delta t}\widehat{\boldsymbol{\Psi}}^{n}\begin{bmatrix}\widehat{\mathbf{q}}^{n}\\ \widehat{\mathbf{q}}^{n}\end{bmatrix}-\Delta\tau\mathbf{M}^{-1}\left(\frac{\boldsymbol{\Omega}}{2}\right)\begin{bmatrix}\mathbf{r}_1^{k}\\ \mathbf{r}_2^{k}\end{bmatrix} \tag{55}$$

It is worthwhile to note that the matrix $\left[\mathbf{I}+\varepsilon\frac{\Delta\tau}{\Delta t}(\boldsymbol{\Phi}^{k}-\boldsymbol{D}_{\boldsymbol{\omega}})\right]$ on the left-hand-side is a block-diagonal matrix which can be pre-calculated. The inversion process is not expensive.

**3) FIRK-RK1**

We also can express the PTDC formulations for FIRK and p-FIRK as:

FIRK-RK1: $$\left[\mathbf{I}+\varepsilon\frac{\Delta\tau}{\Delta t}\right]\begin{bmatrix}\mathbf{q}_1^{k+1}\\ \mathbf{q}_2^{k+1}\end{bmatrix}=\left[\mathbf{I}+(\varepsilon-1)\frac{\Delta\tau}{\Delta t}\right]\begin{bmatrix}\mathbf{q}_1^{k}\\ \mathbf{q}_2^{k}\end{bmatrix}+\frac{\Delta\tau}{\Delta t}\begin{bmatrix}\widehat{\mathbf{q}}^{n}\\ \widehat{\mathbf{q}}^{n}\end{bmatrix}-\Delta\tau\mathbf{M}^{-1}\boldsymbol{A}\begin{bmatrix}\mathbf{r}_1^{k}\\ \mathbf{r}_2^{k}\end{bmatrix} \tag{56}$$

p-FIRK-RK1: $$\left[\mathbf{I}+\varepsilon\frac{\Delta\tau}{\Delta t}\boldsymbol{A}^{-1}\right]\begin{bmatrix}\mathbf{q}_1^{k+1}\\ \mathbf{q}_2^{k+1}\end{bmatrix}=\left[\mathbf{I}+(\varepsilon-1)\frac{\Delta\tau}{\Delta t}\boldsymbol{A}^{-1}\right]\begin{bmatrix}\mathbf{q}_1^{k}\\ \mathbf{q}_2^{k}\end{bmatrix}+\frac{\Delta\tau}{\Delta t}\boldsymbol{A}^{-1}\begin{bmatrix}\widehat{\mathbf{q}}^{n}\\ \widehat{\mathbf{q}}^{n}\end{bmatrix}-\Delta\tau\mathbf{M}^{-1}\begin{bmatrix}\mathbf{r}_1^{k}\\ \mathbf{r}_2^{k}\end{bmatrix} \tag{57}$$

## V. p-Multigrid Strategy

A non-linear multigrid strategy provides an avenue for further accelerating the convergence in pseudo time of the implicit system problem which avoids the explicit consideration of Jacobian matrices. In particular, a p-multigrid method can be constructed where the solution is first computed approximately on the fine grid using a small number of explicit iterations in pseudo time, after which the solution is transferred to a "coarser" mesh with fewer degrees of freedom obtained by lowering the p-order discretization on the same physical or geometric mesh. After the coarse level problem is (partially) solved, the corrections to the coarse level solution can be propagated back to the fine level using a prolongation step. When this procedure is applied recursively on progressively finer meshes, a complete p-multigrid solver is obtained. By using explicit iterations (in pseudo time) on each level, the efficiency of the tensor-product formulation is retained resulting in an efficient solver for very high p-order discretizations.

In addition to performing explicit time steps on each level of the p-multigrid sequence, a consistent approach for restricting the residuals and the solution from fine levels to coarser levels, as well as a prolongation technique for transferring the coarse level solution corrections back to finer levels, must be employed. The approach taken in this work is to use a Galerkin projection from the fine p to coarse p-1 level within an element. This formulation starts with the statement that the solution evaluated at some point x in the element should have the same value expressed in terms of the fine or coarse level basis functions:

$$\sum_{j=1}^{p} b_j \phi_j^{p-1}(x) = \sum_{i=1}^{p+1} a_i \phi_i^{p}(x) \tag{58}$$

Here $a_i$ are the solution coefficients on the fine level and $b_j$ are the corresponding (unknown) coarse level coefficients, with the corresponding basis functions being denoted by the $\phi$'s. Pre-multiplying by a test function in the *p*-1 space and integrating over the element gives:

$$\sum_{j=1}^{p} \int_{\text{element}} \phi_k^{p-1} \phi_j^{p-1} \, b_j = \sum_{i=1}^{p+1} \int_{\text{element}} \phi_k^{p-1} \phi_i^{p} a_i \tag{59}$$

which can be written in matrix form as:

$$M_{kj}^{(p-1)\times(p-1)} b_j = N_{ki}^{(p-1)\times p} a_i \tag{60}$$

Here $\boldsymbol{M}$ is seen to be the usual mass matrix in the *p*-1 space and $\boldsymbol{N}$ is a rectangular matrix. The $b_j$ coefficients are then obtained by inverting the mass matrix as:

$$b = [M^{-1}]_{p-1} [N]_{(p-1)\times p} a \tag{61}$$

In the prolongation step, the procedure is reversed, and the coefficients are obtained in terms of the b coefficients as:

$$a = [M^{-1}]_{p} [N]_{p\times(p-1)} b \tag{62}$$

The construction is such that a prolongation operation followed by restriction results in the identity matrix. Notably, the prolongation operator constructed in this manner does not correspond to the transpose of the restriction operator, as is often used in geometric and algebraic multigrid methods [20].

## VI. Numerical Results

### A. Numerical Tests for the Formal Order of Accuracy

The conservative form of the two-dimensional (in space) Euler equation is considered as governing equations and square Cartesian meshes are used for numerical tests in this section.

#### 1) Spatial Accuracy Study for Ringleb Flow

Ringleb flow is an exact solution for the two-dimensional Euler equations [21]. This exact solution has been used to test the spatial order of accuracy of the discretization [17, 22]. The computational domain is set to $0 \le x \le 1$ and $0 \le y \le 1$. The exact solution is also used for boundary conditions in the x- and y-directions. The flow is initialized with the exact analytic solution and then the simulation is run until the spatial residual converges to machine precision. A grid refinement study is carried and the error is calculated by measuring the difference of density between the exact solution and the computed solution.

Table 2 shows $L_1$, $L_2$ and $L_\infty$ errors for three spatial discretizations (p=2, 4, 9). The average values of the $L_2$ error order are 2.98, 5.08 and 10.68 for p=2, 4 and 9, respectively. $L_1$ and $L_\infty$ error orders are also similar. The designed formal order of accuracy seems to be achieved for all p-orders in space.

**Table 2 Spatial accuracy study for Ringleb flow**

| | Δx | $L_1$ error | $L_1$ order | $L_2$ error | $L_2$ order | $L_\infty$ error | $L_\infty$ order |
|---|---|---|---|---|---|---|---|
| P2 | 0.2000 | 2.8007E-06 | | 3.8289E-06 | | 1.6136E-05 | |
| | 0.1000 | 3.5406E-07 | 2.98 | 4.8820E-07 | 2.97 | 2.9687E-06 | 2.44 |
| | 0.0500 | 4.4524E-08 | 2.99 | 6.1669E-08 | 2.98 | 4.6975E-07 | 2.66 |
| | 0.0250 | 5.5833E-09 | 3.00 | 7.7510E-09 | 2.99 | 6.8463E-08 | 2.78 |
| | Avg. | | 2.99 | | 2.98 | | 2.63 |
| P4 | 0.2000 | 4.7068E-09 | | 7.8441E-09 | | 7.3407E-08 | |
| | 0.1000 | 1.2795E-10 | 5.20 | 2.2032E-10 | 5.15 | 2.3379E-09 | 4.97 |
| | 0.0500 | 3.7273E-12 | 5.10 | 6.5383E-12 | 5.07 | 7.5168E-11 | 4.96 |
| | 0.0250 | 1.2032E-13 | 4.95 | 2.0208E-13 | 5.02 | 2.4372E-12 | 4.95 |
| | Avg. | | 5.09 | | 5.08 | | 4.96 |
| P9 | 0.7500 | 8.5877E-09 | | 2.0036E-08 | | 2.6948E-07 | |
| | 0.5000 | 8.2675E-11 | 11.45 | 5.7677E-10 | 8.75 | 1.1527E-08 | 7.77 |
| | 0.3750 | 5.9244E-12 | 9.16 | 1.7035E-11 | 12.24 | 3.8535E-10 | 11.81 |
| | 0.3000 | 6.1323E-13 | 10.16 | 9.3003E-13 | 13.03 | 1.3772E-11 | 14.93 |
| | 0.2500 | 8.7751E-14 | 10.66 | 1.1466E-13 | 11.48 | 7.9181E-13 | 15.67 |
| | 0.2143 | 2.1695E-14 | 9.07 | 3.3885E-14 | 7.91 | 3.7392E-13 | 4.87 |
| | Avg. | | 10.10 | | 10.68 | | 11.01 |

**2) Temporal Accuracy Study for An Isentropic Vortex Convection**

An isentropic vortex convection problem [19, 23, 24] is considered to assess temporal accuracy. As an initial condition, an isentropic vortex is superposed to the mean flow field. Initial mean flow and perturbation values for the isentropic vortex are given by

$$u_\infty = 0.5, v_\infty = 0, p_\infty = \rho_\infty = T_\infty = 1, (\delta u, \delta v) = \frac{\beta}{2\pi} e^{(1-r^2)/2}(-\bar{y}, \bar{x}), \delta T = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2},$$

where $\beta$ is the vortex intensity set to 4 and $\gamma = 1.4$. Here, $(\bar{x}, \bar{y}) = (x - x_{v0}, y - y_{v0})$, where $x_{v0}$ and $y_{v0}$ are coordinates of the center of initial vortex : $(x_{v0}, y_{v0}) = (0,0)$, and $r^2 = \bar{x}^2 + \bar{y}^2$. The entire flow field is required to be isentropic so, for a perfect gas, $p/\rho^\gamma = 1$.

From the relations, $\rho = \rho_\infty + \delta\rho$, $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T = T_\infty + \delta T$, and the isentropic relation, the resulting conservative variables are given by

$$\rho = T^{1/(\gamma-1)} = (T_\infty + \delta T)^{1/(\gamma-1)} = \left[1 - \frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2}\right]^{1/(\gamma-1)}, \rho u = \rho(u_\infty + \delta u) = \rho\left[1 - \frac{\beta}{2\pi} e^{(1-r^2)/2}\bar{y}\right]$$
$$\rho v = \rho(v_\infty + \delta v) = \rho\left[1 + \frac{\beta}{2\pi} e^{(1-r^2)/2}\bar{x}\right], p = \rho^\gamma, e = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2).$$

**Table 3 Temporal accuracy study for DGT**

| | Δt | $L_1$ error | $L_1$ order | $L_2$ error | $L_2$ order | $L_\infty$ error | $L_\infty$ order |
|---|---|---|---|---|---|---|---|
| P1 | 1.000 | 4.6267E-05 | | 2.4704E-04 | | 4.7856E-03 | |
| | 0.500 | 7.5068E-06 | 2.62 | 3.9115E-05 | 2.66 | 7.6858E-04 | 2.64 |
| | 0.250 | 1.0173E-06 | 2.88 | 4.9508E-06 | 2.98 | 9.2563E-05 | 3.05 |
| | 0.125 | 1.3278E-07 | 2.94 | 6.3772E-07 | 2.96 | 1.1358E-05 | 3.03 |
| | Avg. | | 2.81 | | 2.87 | | 2.91 |
| P2 | 1.000 | 2.7632E-06 | | 1.6165E-05 | | 3.0092E-04 | |
| | 0.500 | 1.3700E-07 | 4.33 | 6.9904E-07 | 4.53 | 1.0947E-05 | 4.78 |
| | 0.250 | 5.1223E-09 | 4.74 | 2.6392E-08 | 4.73 | 3.8318E-07 | 4.84 |
| | 0.125 | 1.6802E-10 | 4.93 | 8.5126E-10 | 4.95 | 1.2780E-08 | 4.91 |
| | Avg. | | 4.67 | | 4.74 | | 4.84 |

**Table 4 Temporal accuracy study for two stage Gauss and Radau 2A scheme**

| | Δt | $L_1$ error | $L_1$ order | $L_2$ error | $L_2$ order | $L_\infty$ error | $L_\infty$ order |
|---|---|---|---|---|---|---|---|
| Gauss | 1.000 | 1.2295E-05 | | 7.3265E-05 | | 1.2441E-03 | |
| | 0.500 | 7.2081E-07 | 4.09 | 3.4346E-06 | 4.41 | 4.4038E-05 | 4.82 |
| | 0.250 | 5.3668E-08 | 3.75 | 2.5884E-07 | 3.73 | 3.8282E-06 | 3.52 |
| | 0.125 | 3.6222E-09 | 3.89 | 1.7224E-08 | 3.91 | 2.6195E-07 | 3.87 |
| | Avg. | | 3.91 | | 4.02 | | 4.07 |
| Radau 2A | 1.000 | 3.7273E-05 | | 1.9873E-04 | | 3.7876E-03 | |
| | 0.500 | 5.6780E-06 | 2.71 | 3.1314E-05 | 2.67 | 6.2362E-04 | 2.60 |
| | 0.250 | 7.4971E-07 | 2.92 | 3.9743E-06 | 2.98 | 8.0173E-05 | 2.96 |
| | 0.125 | 9.5939E-08 | 2.97 | 5.0059E-07 | 2.99 | 9.9944E-06 | 3.00 |
| | Avg. | | 2.87 | | 2.88 | | 2.86 |

The computational domain is set to $-10 \leq x \leq 10$ and $-10 \leq y \leq 10$. Periodic boundary conditions are used in *x*- and *y*-directions. The vortex convects to the right with the freestream ($u_\infty = 0.5$) and due to the periodic boundary conditions, goes back to the initial location every non-dimensional time interval $\Delta t = 20$.

Since the flow field is inviscid, the exact solution is just a passive advection of the initial vortex with a mean velocity. The overall error, however, includes both spatial and temporal errors. In order to eliminate the spatial error and to isolate the temporal error, a reference solution for each temporal *p*-order discretization is obtained first using a time-step of $\Delta t$=0.01. Then, the difference of density between the reference solution and a corresponding computed solution is considered as the temporal error.

Table 3 shows the $L_1$, $L_2$ and $L_\infty$ errors at $t$=2. The average values of the $L_2$ error order for $p$=1 and 2 in time are 2.87 and 4.74, respectively. $L_1$ and $L_\infty$ error orders also show similar values. The chosen temporal discretizations ($p_t$=1, 2) achieve accuracies of approximately 2p+1, within the range of time step sizes tested herein for the given $\Delta x$ and spatial *p*-order. Thus at least for this well resolved case, it appears that the DGT schemes achieve superconvergence compared to the expected accuracy of p+1.

In Table 4, the temporal accuracy of the two stage Gauss and Radau2A schemes are seen to be approximately 4 and 3, respectively. This corresponds to the expected order of accuracy for these FIRK schemes which are 2s and 2s-1, respectively, where s is the number of stages. In terms of the number of stages in the DGT schemes corresponds to p+1, this means that the Gauss FIRK scheme achieves accuracy of 2p+2, while the Radau 2A scheme achieves accuracy of 2p+1. Similar results on an equivalent test problem for these and higher order FIRK schemes are reported in [15].

### B. Comparison of Convergence Performance

In this section, the convergence performances of BDF1-RK1, DGT-RK1 and FIRK-RK1 are investigated. The isentropic vortex convection problem used in the previous section serves as the time-dependent flow problem. Tests were performed using a fixed spatial discretization ($p_s = 5$) on the domain size of $-10 \leq x \leq 10$ and $-10 \leq y \leq 10$.

The convergence performance is compared in terms of the number of explicit pseudo-time steps required to drive the L2-norm of the residual to less than $10^{-12}$. The convergence history for the very first step in physical time is only compared to see the differences between the schemes more closely. It was verified that the convergence behavior for the first physical time step is similar for the rest of the steps, but the plots are not presented here.

As expected, the solution efficiency depends on the physical time step size, as well as the largest pseudo-time step that can be used in the iterative scheme without provoking instabilities. The CFL number is an important parameter in this respect. Basically, we assume that the CFL number in pseudo time must be less than one. For high-order spatial discretizations, however, the actual time step size is severely reduced. We compute the maximum allowable $\Delta t$ based on the formulation as follows.

$$\mathrm{CFL}_t = (|U| + c)\frac{\Delta t}{\min(dx, dy)} \tag{63}$$

$$\mathrm{CFL}_t^* = (|\mathrm{U}| + \mathrm{c})\frac{\Delta \mathrm{t}}{\min(\mathrm{dx}, \mathrm{dy})} \cdot (\mathrm{p}+1)^{1.8} \tag{64}$$

where $\mathrm{CFL}_t$ is a CFL number for physical time step $\Delta t$. $\mathrm{CFL}_t^*$ is a modified CFL number which considers the effect of high-order spatial discretization, which has been derive by empirical numerical evidence [22]. $\mathrm{CFL}_t^* = 1$ means that the $\Delta t$ is the maximum allowable time step for a corresponding high-order spatial discretization. As shown in Eq. (63) and (64), they have the relation as:

$$\frac{\mathrm{CFL}_t^*}{\mathrm{CFL}_t} = (p+1)^{1.8} \tag{65}$$

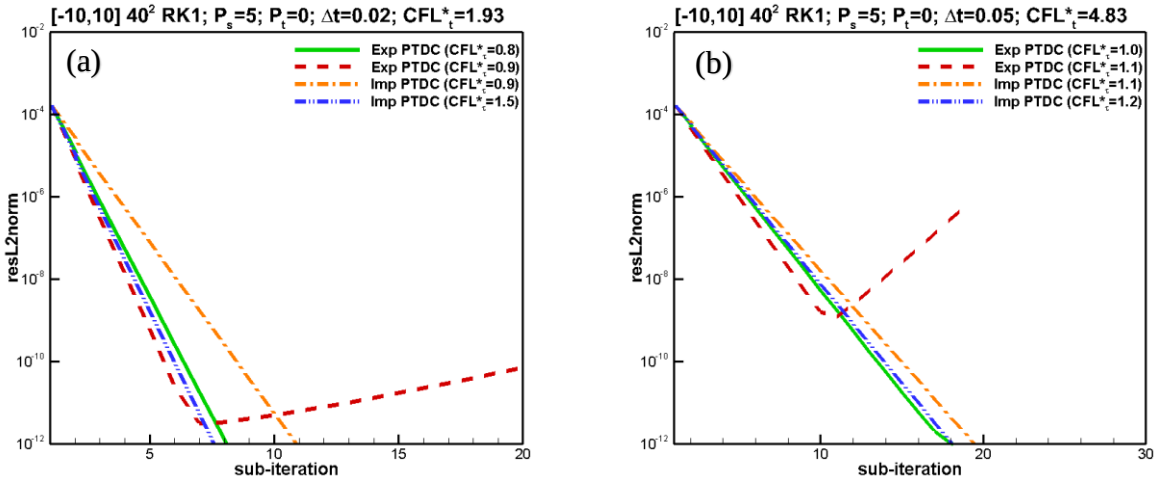
$\mathrm{CFL}_t^*$ has larger values than $\mathrm{CFL}_t$. For example, when p=5, $\mathrm{CFL}_t^*$ is about 25 times larger than $\mathrm{CFL}_t$. Similarly, $\mathrm{CFL}_\tau^*$ for pseudo-time step is defined. $\mathrm{CFL}_t^*$ and $\mathrm{CFL}_\tau^*$ are used as parameters for comparison purposes.

**1) BDF1-RK1**

First, the effect of the implicit PTDC is studied for BDF1-RK1. As mentioned in the previous section, DGT with $p_t = 0$ corresponds to BDF1. Figure 1shows the convergence histories of explicit and implicit PTDC for five different physical time step sizes $\Delta t$=0.02, 0.05, 0.1, and 0.5. The corresponding $\mathrm{CFL}_t^*$ values are 1.9, 4.8, 9.7, and 48, respectively. For each physical-time step size, various pseudo-time step sizes $\Delta\tau$ in terms of $\mathrm{CFL}_\tau^*$ were tested.

In each plot, the green solid line and blue dashed-dot-dot line indicate the best convergence rate for the explicit and implicit PTDC, respectively. When $\mathrm{CFL}_\tau^*$ is increased further, the explicit PTDC eventually becomes unstable. The red dashed line shows the unstable converge history for the corresponding smallest $\mathrm{CFL}_\tau^*$ case. The orange line corresponds to the implicit PTDC case with the same $\mathrm{CFL}_\tau^*$. Overall, as the physical time step size $\Delta t$ increases, the convergence rate becomes slow and the maximum $\mathrm{CFL}_\tau^*$ for stable convergence decreases. In terms of $\mathrm{CFL}_\tau^*$, the implicit PTDC scheme shows much better stability for small physical time steps. In Figure 1(a), for example, we can use $\mathrm{CFL}_\tau^*$ up to 1.5 for implicit PTDC, while the explicit case already diverges for $\mathrm{CFL}_\tau^*$=0.9. On the other hand, for large physical time step $\Delta t$, the explicit and implicit PTDC become identical as shown in Figure 1(d).

It should be noted that the slope of the best convergence curve remains almost the same for the explicit and implicit PTDC, although we can use larger $\Delta\tau$ for the implicit PTDC scheme. For the implicit PTDC, the convergence rate becomes slower than that for the explicit counterpart at the same $\mathrm{CFL}_\tau$ due to the factor $1/\left[1+\varepsilon\frac{\Delta\tau}{\Delta t}\right]$. Figure 1(c) shows a good example. The best performance convergence rate for the explicit PTDC scheme (green line) coincides with the best for the implicit PTDC scheme (blue line). Performance improvement for higher p-order is discussed in the following sections.



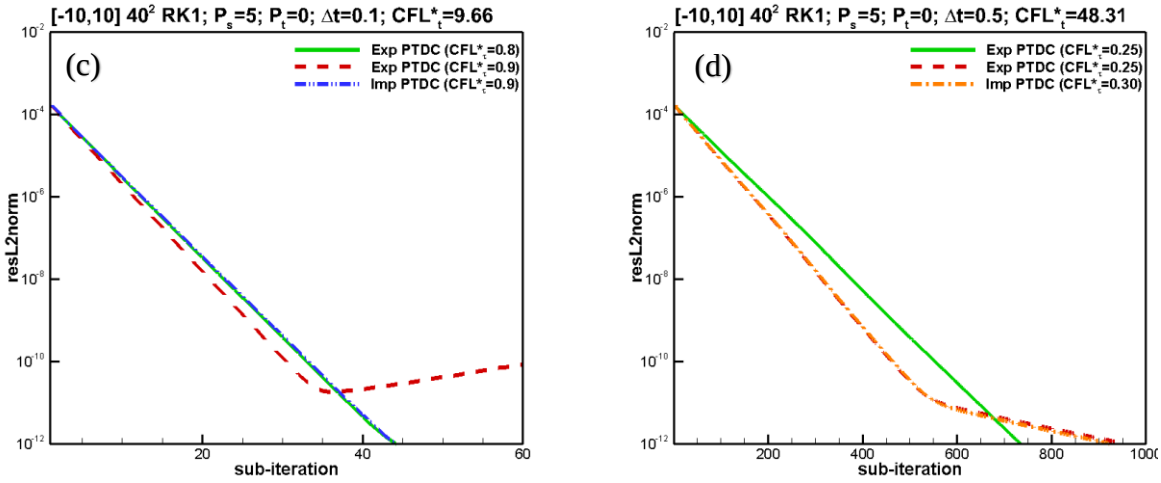



**Figure 1 Convergence histories of explicit/implicit PTDC for BDF1 for $\Delta t$=0.02, 0.05, 0.1, 0.5**

**2) DGT-RK1**

In order to study the effect of high p-order in time on convergence rate, the convergence histories of $p_t = 0$ (BDF1) and $p_t = 5$ are compared. Explicit PTDC is used for both $p_t = 0$ and $p_t = 5$ to isolate the effect of high-order $p_t$. Figure 2 shows the case of the physical time step $\Delta t$=0.1. Like the line colors in Figure 1, the green solid line indicates the best convergence rate and the red dashed line shows the unstable converge history for the smallest $\text{CFL}_\tau^*$. For $p_t = 5$ case, the convergence rate is about four times slower than that for BDF1. While the best performance occurs with $\text{CFL}_t^*$=0.8 for BDF1, the convergence of $p_t = 5$ already becomes slow at $\text{CFL}_t^*$=0.7 indicating the beginning of instability. Both $p_t = 0$ and $p_t = 5$ become fully unstable at $\text{CFL}_t^*$=0.9.

Figure 3 shows the same comparison for the physical time step $\Delta t$=0.26. Interestingly, $p_t = 5$ shows a slightly better convergence rate. BDF1 shows the best performance at $\text{CFL}_t^*$=0.4 (green line). Compared to the case of $\Delta t$=0.1, we can see again that the convergence rate becomes five times slower, as commented in the discussion for Figure 1. On the other hand, $\text{CFL}_t^*$ up to 1.2 works for $p_t = 5$. It should be noted that the convergence performance for $p_t = 5$ remains almost same for both $\Delta t$=0.1 and 0.26.

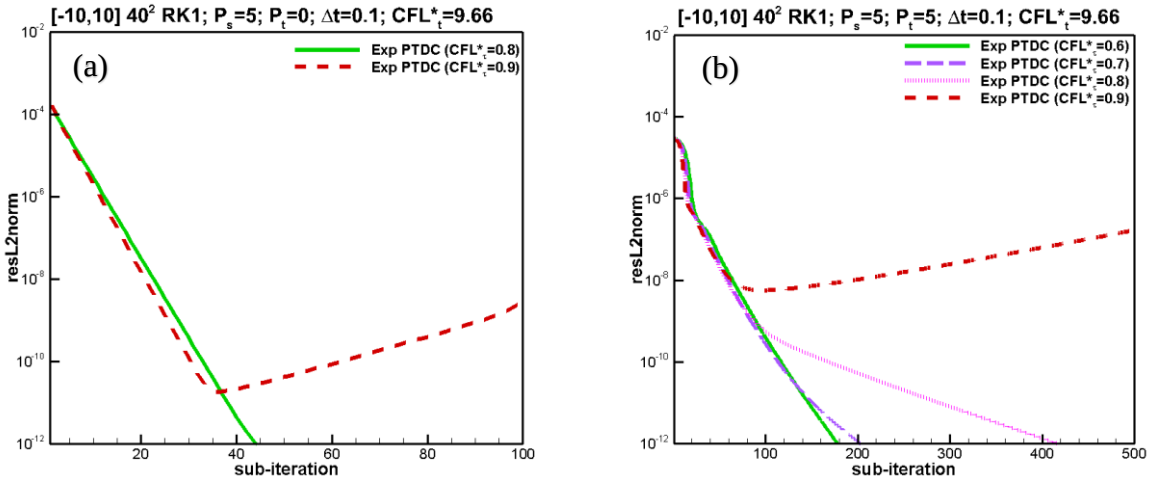



**Figure 2 Effect of high-order $p_t$ on convergence rate for $\Delta t$=0.1 (explicit PTDC): (a) BDF1, (b) $p_t$=5 DGT**

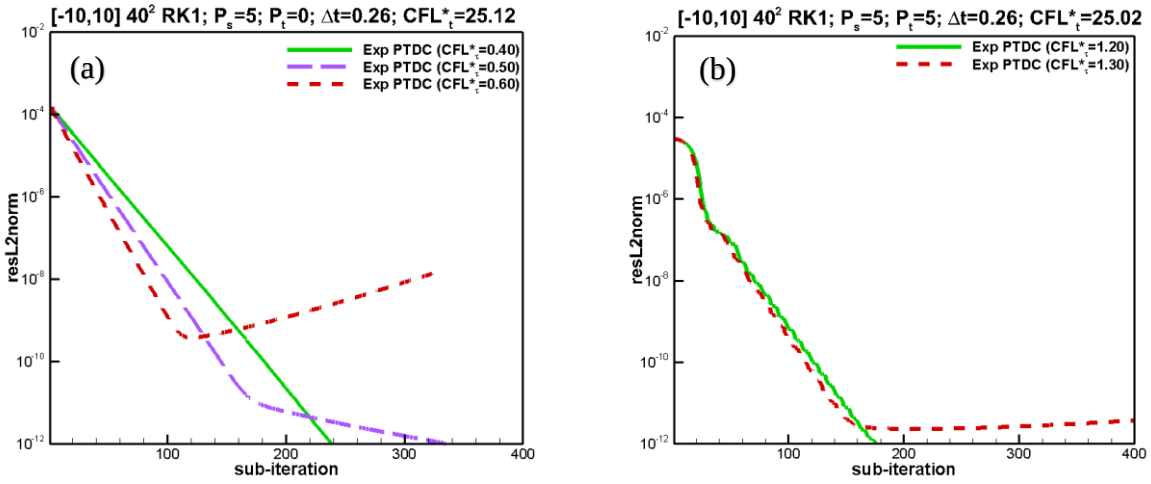
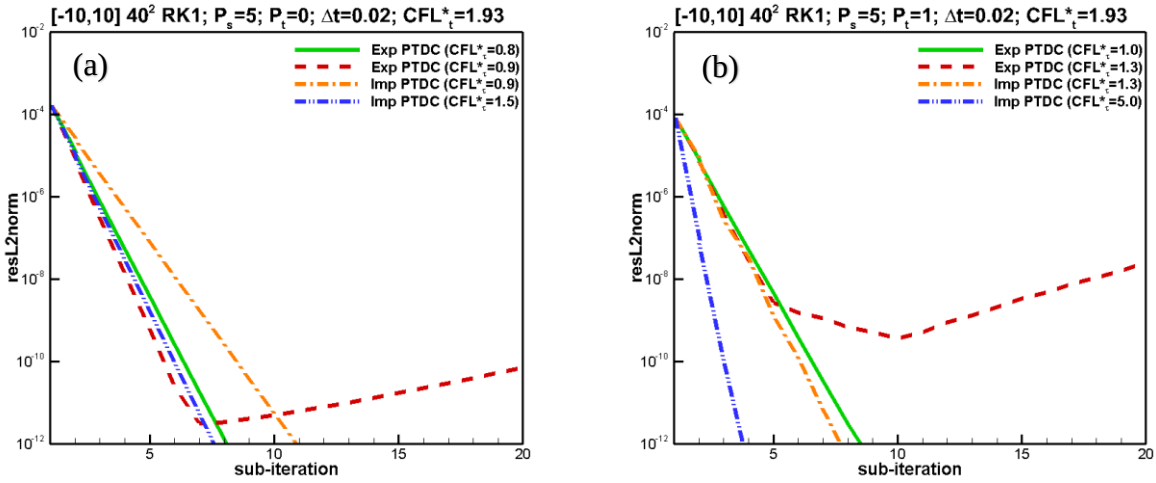



**Figure 3 Effect of high-order $p_t$ on convergence rate for $\Delta t$=0.26 (explicit PTDC): (a) BDF1, (b) $p_t$=5 DGT**



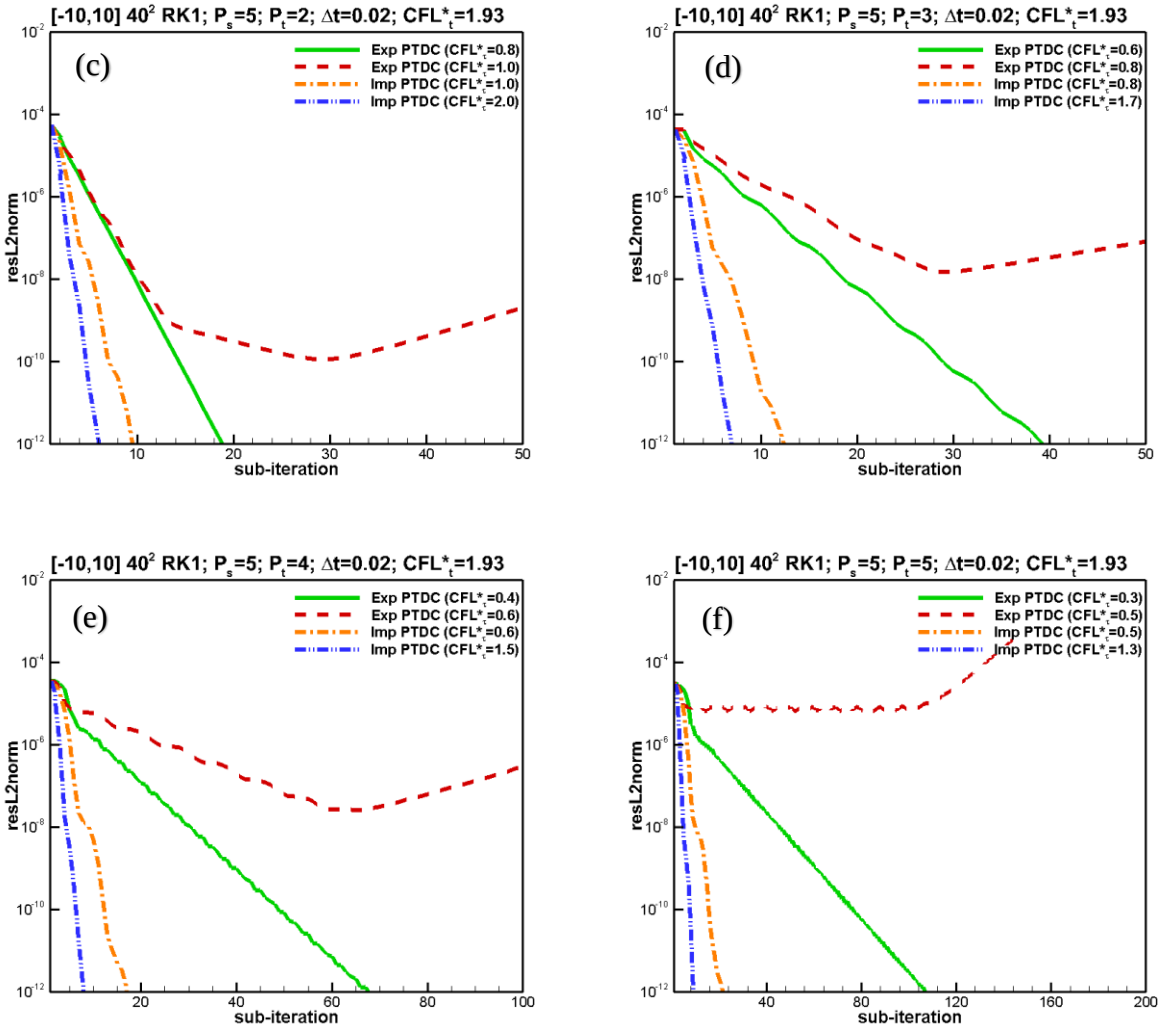



**Figure 4 Convergence histories of explicit/implicit PTDC for DGT with $\Delta t$=0.02: (a) $p_t$=0, (b) $p_t$=1, (c) $p_t$=2, (d) $p_t$=3, (e) $p_t$=4, (f) $p_t$=5**

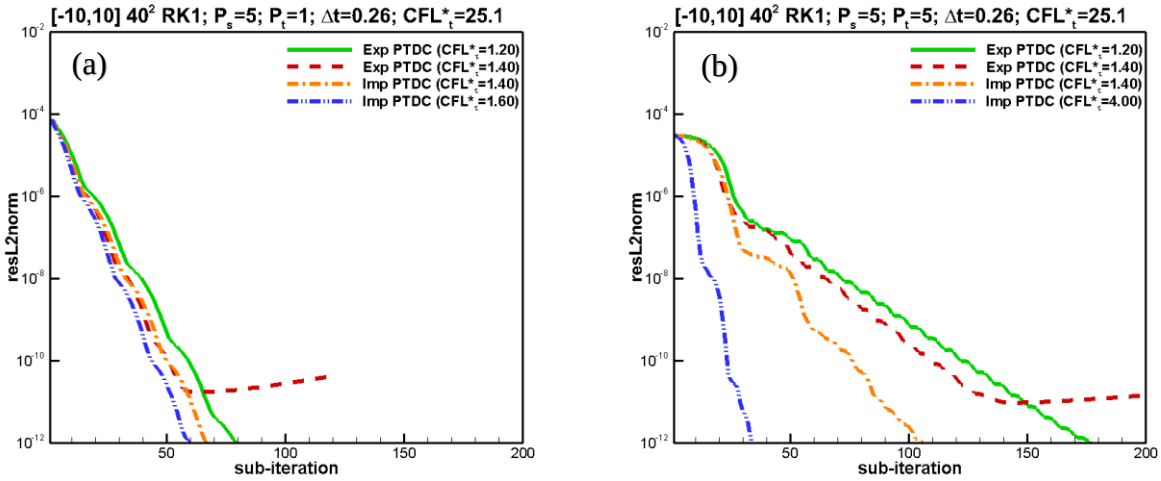



**Figure 5 Convergence histories of explicit/implicit PTDC for DGT with $\Delta t$=0.26: (a) $p_t$=1, (b) $p_t$=5**

Next, the effect of the implicit PTDC scheme for DG in time is investigated. Figure 4 shows the explicit and implicit PTDC schemes for DGT with $p_t$=0~5. A small physical time step size $\Delta t$=0.02 is considered since the difference between the explicit and implicit PTDC schemes becomes clear for small $\Delta t$. The notation of line color follows the same pattern as in the previous figures. Overall, the implicit PTDC scheme shows better convergence performance. For BDF1 ($p_t$=0) in (a), as commented for Figure 1, the difference in the best performing convergence rate is insignificant, although larger $\mathrm{CFL}_t^*$ can be used for the implicit PTDC scheme. As $p_t$ increases, however, the implicit PTDC shows much better convergence performance. For $p_t$ =5 in Figure 4(f), the implicit PTDC provides 10x speed-up.

The number of sub-iterations where the L2-norm of the residual becomes less than $10^{-12}$ are summarized in Table 5. As $p_t$ increases, the number of cycles needed for explicit PTDC increases rapidly from 9 for $p_t$=0 (BDF1) to 107 for $p_t$=5. On the other hand, the number of cycles needed for the implicit PTDC remains same around 10, even for $p_t$=5, which suggests a large benefit of the implicit PTDC when it is used for small physical time step $\Delta t$.

**Table 5 Best performance in terms of the number of sub-iterations for DGT with $\Delta t$=0.02**

| | explicit PTDC ($\varepsilon = 0$) | | implicit PTDC ($\varepsilon = 1$) | |
|---|---|---|---|---|
| $p_t$ | $\mathrm{CFL}_\tau^*$ | # of sub-iteration | $\mathrm{CFL}_\tau^*$ | # of sub-iteration |
| 0 | 0.8 | 9 | 1.5 | 8 |
| 1 | 1.0 | 9 | 5.0 | 4 |
| 2 | 0.8 | 19 | 2.0 | 7 |
| 3 | 0.6 | 40 | 1.7 | 7 |
| 4 | 0.4 | 68 | 1.5 | 8 |
| 5 | 0.3 | 107 | 1.3 | 10 |

**Table 6 Best performance in terms of the number of sub-iterations for DGT with $\Delta t$=0.26**

| | explicit PTDC ($\varepsilon = 0$) | | implicit PTDC ($\varepsilon = 1$) | |
|---|---|---|---|---|
| $p_t$ | $\mathrm{CFL}_\tau^*$ | # of sub-iteration | $\mathrm{CFL}_\tau^*$ | # of sub-iteration |
| 1 | 1.2 | 80 | 1.6 | 61 |
| 5 | 1.2 | 177 | 4.0 | 34 |

Figure 5 and Table 6 show the same comparison for the case of larger physical time step size $\Delta t$=0.26. The implicit PTDC scheme also shows better convergence performance, but the 5x faster speed-up is smaller than the case of $\Delta t$=0.02 since the difference between explicit and implicit PTDC schemes decreases as $\Delta t$ increases.

In summary, for smaller physical time steps $\Delta t$, the implicit PTDC scheme shows much better performance, especially at higher $p_t$ order. For larger $\Delta t$, the explicit and implicit PTDC schemes become more and more identical, but higher $p_t$ order shows better performance than lower $p_t$ orders.

### 3) FIRK-RK1

The effect of implicit PTDC for FIRK is investigated in this subsection. In order to compare DGT with FIRK, the temporal p-order is fixed as $p_t$=1 and $p_s$=5 is used. Two-stage pre-conditioned Radau 2A scheme is also tested. Figure 6 shows comparison of the convergence histories for $\Delta t$ =0.01. Blue solid/red dashed/green dashed-dot/orange dashed-dot-dot lines indicate DGT/FIRK-G/pFIRK-G/pFIRK-R, respectively. FIRK-G and FIRK-R indicate Gauss and Radau 2A scheme, respectively. Each convergence history corresponds to the best performance of convergence for each scheme. Figure 6(a) shows the explicit PTDC cases. For this small $\Delta t$, the original FIRK-G shows the fastest convergence rate, while DGT is the slowest. For the implicit PTDC case in Figure 6(b), however, DGT convergence rate is about two times faster, compared to the explicit PTDC. FIRK-G, pFIRK-G and pFIRK-R

show almost same the convergence rate even for the implicit PTDC scheme. This trend is also seen for $\Delta t$=0.02 in Figure 7.

Similarly, Figure 8 shows the same comparison for $\Delta t$=0.26. The original FIRK-G becomes unstable for this large $\Delta t$ even with very small $\mathrm{CFL}_t^*$ ($\mathrm{CFL}_t^* \ll 0.4$ was also tested, but not presented here). Although DGT, pFIRK-G and pFIRK-R show slight speed-up for the implicit PTDC, the results of all schemes remain mostly same for the explicit and implicit PTDC. In Figure 9, as expected, the explicit or implicit PTDC becomes identical for $\Delta t$=2.6.

It is worthwhile to note that when $\Delta t$ is large, the pre-conditioned Radau 2A (pFIRK-R) acts just like DGT for both the explicit and implicit PTDC. For small $\Delta t$, the convergence history of pFIRK-R is similar to that of DGT for the explicit PTDC, but similar to that of the pre-conditioned Gauss (pFIRK-G) for the implicit PTDC. It can be said that overall, pFIRK-R acts like DGT, but the implicit DGT out-performs the pFIRK-G and pFIRK-R for small $\Delta t$.

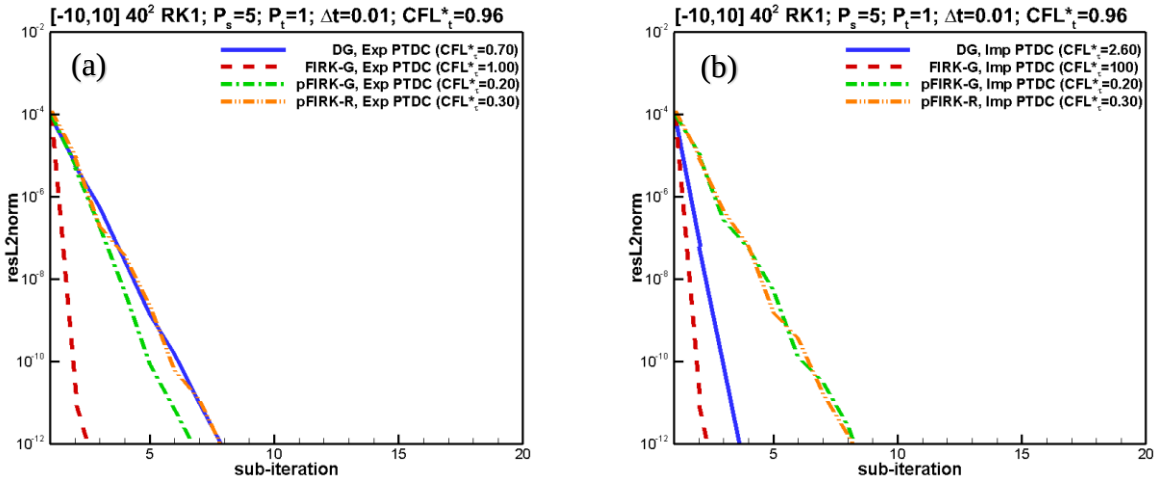



**Figure 6 Comparison of convergence histories for $\Delta t$=0.01: (a) explicit PTDC, (b) implicit PTDC**

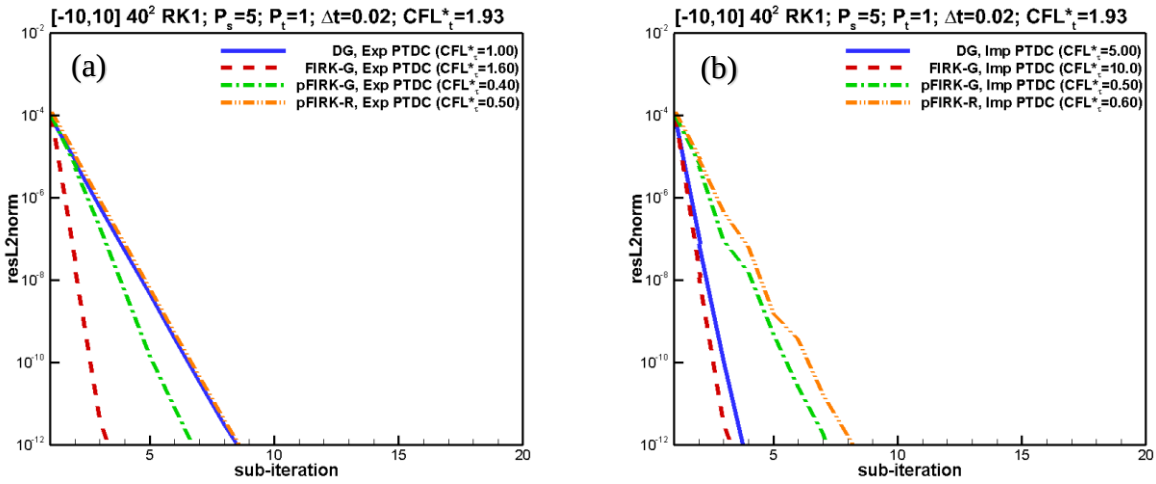



**Figure 7 Comparison of convergence histories for $\Delta t$=0.02: (a) explicit PTDC, (b) implicit PTDC**

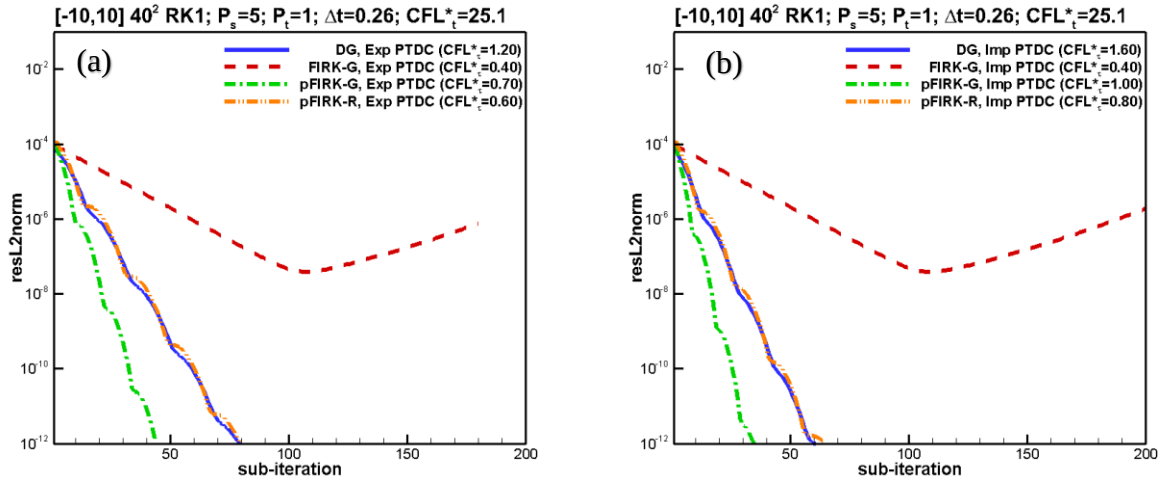



**Figure 8 Comparison of convergence histories for $\Delta t$=0.26: (a) explicit PTDC, (b) implicit PTDC**

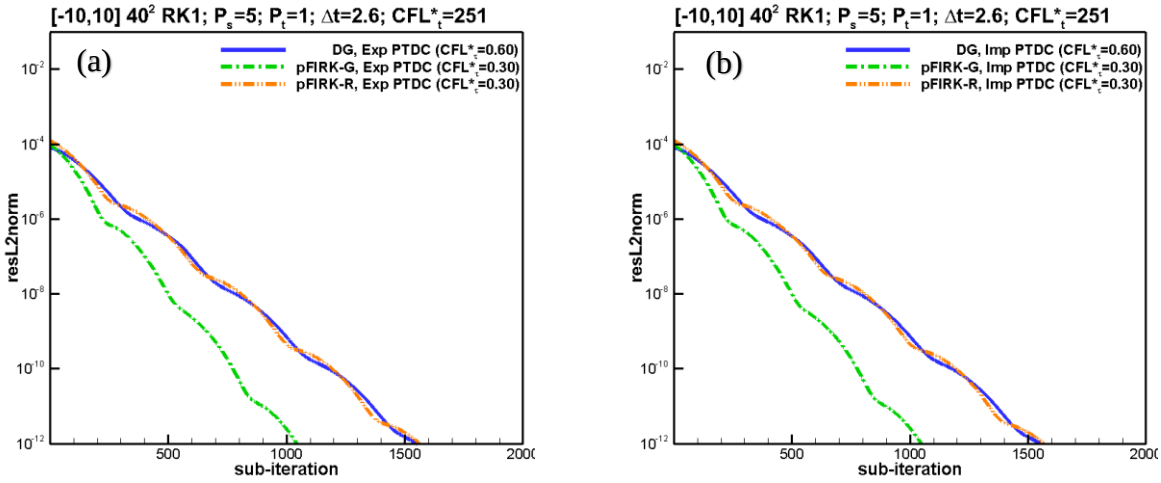



**Figure 9 Comparison of convergence histories for $\Delta t$=2.6: (a) explicit PTDC, (b) implicit PTDC**

### C. Convergence Acceleration Using p-Multigrid Method

In this section we examine the performance of the p-multgrid solver for the DG in time discretizations. The space-time formulation with p-multigrid is also tested using the isentropic vortex problem discussed in the previous section. The space-time DG discretization supports decoupled p-orders in space and time, thus allowing for p-multigrid to be applied in space only, in time only, or in both space and time simultaneously. Furthermore, as $p_t$=0 corresponds to a BDF1, the implementation and solver performance is verified by ensuring that results obtained using the DG in time implementation at $p_t$=0 is consistent with the results obtained using the BDF1 implementation discussed in the previous section.

Figure 10 shows the convergence histories of the space-time DG discretization on a 40x40 grid for the isentropic vortex problem. The p-orders in space and time are $p_s$=5 and 8 (sixth- and nineth-order spatial accuracy), and $p_t$=0 (first-order temporal accuracy), respectively. The results compare a single level approach to 2-level multigrid (MG V2). In terms of p-multigrid cycles, the 2-level multigrid approach is 2x times and 3x times faster than single level for $p_s$=5 and $p_s$=8, respectively.

Next, higher temporal orders of accuracy are tested for DG in time. Figure 11 shows the convergence histories for $p_s$=5 and 8, and $p_t$=8. As the p-order in time increases to $p_t$=8, the convergence rates with respect to the number of multigrid cycles become 4x and 3x slower than $p_t$=0 case, as shown in Figure 11, for the single level $p_s$=5 and 8, respectively. The 2-level multigrid system is 2x times and 3x times faster than single level system for $p_s$=5 and 8, which is consistent with p-multigrid results for DG in space only. Note that for the higher-order temporal

discretizations, p-coarsening in both space and time is done on the coarse multigrid level. Although the comparisons are done here in terms of multigrid cycles, for a two level non-recursive multigrid method, the cpu time to convergence is much larger than that of the single level solver, due to the requirement of solving the coarse level to completion at each multigrid cycle. However, two level multigrid comparisons in terms of multigrid cycles are useful as they provide an estimate of the theoretical optimal speedup that a well formulated recursive multigrid algorithm can provide.

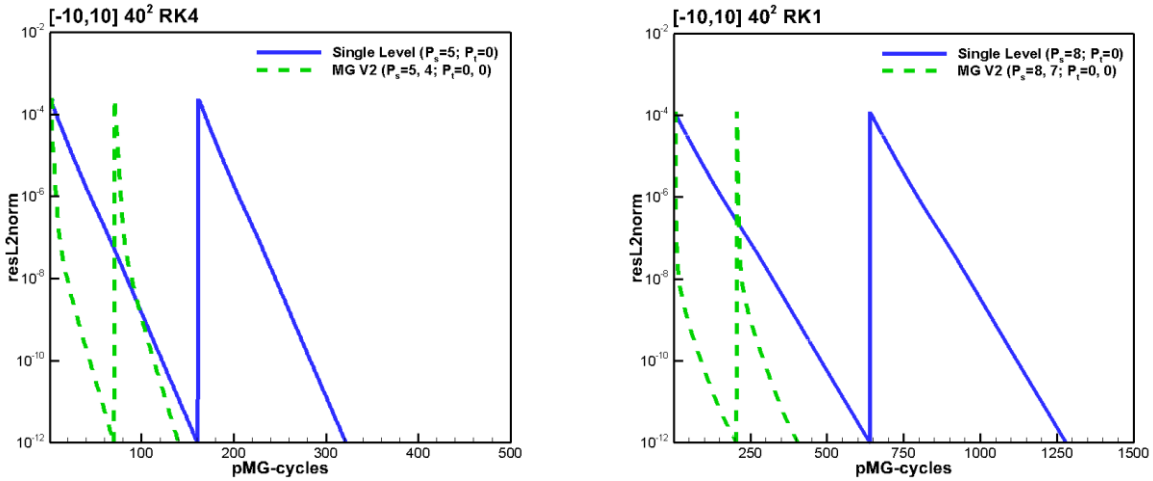



**Figure 10 Convergence histories of p-multigrid for BDF1 ($p_t$=0): (a) $p_s$=5, (b) $p_s$=8**

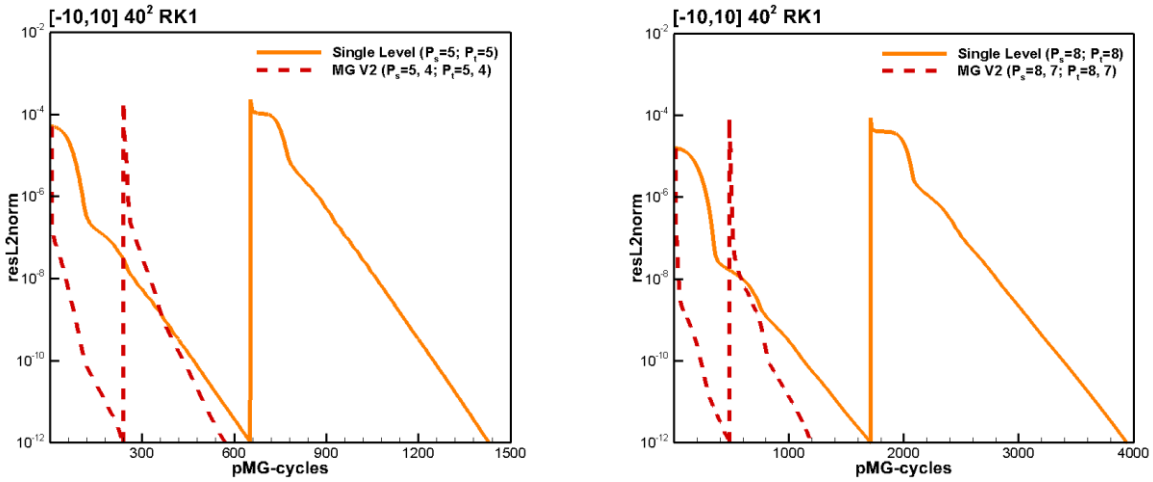



**Figure 11 Convergence histories of p-multigrid for DGT ($p_t$=8): (a) $p_s$=5, (b) $p_s$=8**

## VII. Conclusion

In this work, we present a high-order space-time DG discretization based on a tensor-product formulation. The method allows an arbitrary high-order for both temporal and spatial discretization. This method is compared with multi-stage fully implicit Runge Kutta schemes, which are shown to take on a similar form. For two-dimensional (in space) compressible inviscid flows, spatial and temporal accuracy studies are performed for Ringleb flow and for the convection of an isentropic vortex, respectively. The temporal discretizations are solved at each time step using an explicit pseudo-time stepping approach, which can be accelerated using a p-multigrid solver in pseud-time. For the cases tested herein, the various temporal discretizations achieve or exceed their design orders of accuracy. The

pseudo time stepping approach is shown to be feasible for all schemes, provided the FIRK schemes are written in a form similar to the DGT schemes.

## Acknowledgments

This work was supported by NASA Grant 80NSSC18M0154 under the $T^3$ project.

## References

[1] C. Burstedde, L. C. Wilcox, and O. Ghattas, "*P4est*: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees," *SIAM J Sci Comput*, vol. 33, no. 3, pp. 1103–1133, May 2011, doi: 10.1137/100791634.

[2] A. C. Kirby *et al.*, "Wind farm simulations using an overset hp-adaptive approach with blade-resolved turbine models," *Int. J. High Perform. Comput. Appl.*, vol. 33, no. 5, pp. 897–923, Mar. 2019, doi: 10.1177/1094342019832960.

[3] K. Kara, M. J. Brazell, A. C. Kirby, D. J. Mavriplis, and E. P. Duque, "Hover Predictions Using a High-Order Discontinuous Galerkin Off-Body Discretization," in *AIAA Scitech 2020 Forum*, 0 vols., American Institute of Aeronautics and Astronautics, 2020.

[4] G. Em. Karniadakis and S. J. Sherwin, *Spectral/hp element methods for CFD*. New York (N.Y.) : Oxford university press, 1999.

[5] L. T. Diosady and S. M. Murman, "Tensor-product preconditioners for higher-order space–time discontinuous Galerkin methods," *J. Comput. Phys.*, vol. 330, pp. 296–318, Feb. 2017, doi: 10.1016/j.jcp.2016.11.022.

[6] L. T. Diosady and S. M. Murman, "Scalable tensor-product preconditioners for high-order finite-element methods: Scalar equations," *J. Comput. Phys.*, vol. 394, pp. 759–776, Oct. 2019, doi: 10.1016/j.jcp.2019.04.047.

[7] A. Jameson, "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings," in *10th Computational Fluid Dynamics Conference*, 0 vols., American Institute of Aeronautics and Astronautics, 1991.

[8] C. R. Nastase and D. J. Mavriplis, "High-Order Discontinuous Galerkin Methods Using an Hp-Multigrid Approach," *J Comput Phys*, vol. 213, no. 1, pp. 330–357, Mar. 2006, doi: 10.1016/j.jcp.2005.08.022.

[9] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal, "p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations," *J. Comput. Phys.*, vol. 207, no. 1, pp. 92–113, Jul. 2005, doi: 10.1016/j.jcp.2005.01.005.

[10] M. Franciolini and S. M. Murman, "Multigrid preconditioning for a space-time spectral-element discontinuous-Galerkin solver," in *AIAA Scitech 2020 Forum*, 0 vols., American Institute of Aeronautics and Astronautics, 2020.

[11] N. L. Mundis and D. J. Mavriplis, "Finite-element Time Discretizations for the Unsteady Euler Equations," in *53rd AIAA Aerospace Sciences Meeting*, 0 vols., American Institute of Aeronautics and Astronautics, 2015.

[12] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Wiley, 2008.

[13] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Wiley, 1991.

[14] A. Jameson, "Evaluation of Fully Implicit Runge Kutta Schemes for Unsteady Flow Calculations," *J. Sci. Comput.*, vol. 73, no. 2, pp. 819–852, Dec. 2017, doi: 10.1007/s10915-017-0476-x.

[15] W. Pazner and P.-O. Persson, "Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations," *J. Comput. Phys.*, vol. 335, pp. 700–717, Apr. 2017, doi: 10.1016/j.jcp.2017.01.050.

[16] N. D. Melson, H. L. Atkins, and M. Sanetrik, "Time-accurate Navier-Stokes calculations with multigrid acceleration," presented at the The Sixth Copper Mountain Conference on Multigrid Methods, Part 2, Nov. 1993, [Online]. Available: https://ntrs.nasa.gov/api/citations/19940016996/downloads/19940016996.pdf.

[17] A. Kirby, "Enabling High-Order Methods for Extreme-Scale Simulations," University of Wyoming, 2018.

[18] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numer. Math.*, vol. 3, no. 1, pp. 27–43, Mar. 1963, doi: 10.1007/BF01963532.

[19] D. J. Mavriplis and C. R. Nastase, "On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes," *J. Comput. Phys.*, vol. 230, no. 11, pp. 4285–4300, May 2011, doi: 10.1016/j.jcp.2011.01.022.

[20] F. Bassi, A. Ghidoni, and S. Rebay, "Optimal Runge–Kutta smoothers for the p-multigrid discontinuous Galerkin solution of the 1D Euler equations," *Spec. Issue High Order Methods CFD Probl.*, vol. 230, no. 11, pp. 4153–4175, May 2011, doi: 10.1016/j.jcp.2010.04.030.
[21] F. Ringleb, "Exakte Lösungen der Differentialgleichungen einer adiabatischen Gasströmung," *ZAMM - J. Appl. Math. Mech. Z. Für Angew. Math. Mech.*, vol. 20, no. 4, pp. 185–198, 1940, doi: 10.1002/zamm.19400200402.
[22] M. J. Brazell, A. C. Kirby, and D. Mavriplis, "A high-order discontinuous-Galerkin octree-based AMR solver for overset simulations," doi: 10.2514/6.2017-3944.
[23] C.-W. Shu, "Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws," NASA ICASE Report No. 97-65, Nov. 1997. Accessed: Jun. 06, 2020. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19980007543.
[24] L. Wang and D. J. Mavriplis, "Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations," *J. Comput. Phys.*, vol. 225, no. 2, pp. 1994–2015, Aug. 2007, doi: 10.1016/j.jcp.2007.03.002.