Oral presentation | Higher order methods

Higher order methods-I

Mon. Jul 15, 2024 4:30 PM - 6:30 PM Room C

[3-C-04] High-order Accurate Implicit Scheme Based on Temporal Reconstruction for Solving Compressible Navier-Stokes Equations

*Hanyu Zhou[1], Yu-Xin Ren[1] （1. Tsinghua University）

Keywords: High-order methods, Temporal reconstruction, Implicit methods, Time marching methods

Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024

High-order Accurate Implicit Scheme Based on Temporal Reconstruction for Solving Compressible Navier-Stokes Equations

Hanyu Zhou*, Yu-Xin Ren*,**
* Department of Engineering Mechanics, Tsinghua University, Beijing, 100084, China.
**Corresponding author: ryx@tsinghua.edu.cn

Abstract: A family of new implicit time marching schemes based on Direct Integration and Temporal Reconstruction (DITR) is developed and tested in this paper. These schemes can be used to solve ordinary differential equations (ODEs) directly or solve ODEs arising from semi-discrete schemes solving partial differential equations (PDEs), including the compressible Navier-Stokes equations. The most significant advantages of these methods are that temporally third- and fourth-order schemes can be constructed straightforwardly, and require fewer stages than some popular implicit Runge-Kutta schemes. As an ODE solver, the DITR method are found to be A-stable, with two variants of them being L-stable. A matrix-free iteration method for solving the DITR equations is described for practical implementation of DITR. Numerical results show that DITR methods achieve high order of accuracy with comparatively lower computational cost. When reaching the same error, some DITR methods are able to save significant amount of time compared with the implicit ESDIRK4 method.

Keywords: High-order Methods, Time Marching, Temporal Reconstruction, Unsteady Flows.

1 Introduction

In computational fluid dynamics (CFD), high-order numerical methods have the capability of resolving complex flows effectively and efficiently, which have been attracting great attention recently. Popular high-order CFD methods, including discontinuous Galerkin (DG) methods [1, 2, 3, 4, 5], spectral volume [6] and spectral difference [7] methods, PnPm procedures [8], FR/CPR methods [9, 10, 11, 12], and finite volume (FV) methods [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23], generally adopted the method of line, in which the governing equations are spatially discretized at first resulting in a system of ODEs. The system of ODEs is then solved with ODE integrators such as the popular strong stability preserving Runge-Kutta (SSPRK) methods [24].

Although explicit ODE integrators are simple and efficient for a wide range of CFD problems, the Courant-Friedrichs-Lewy (CFL) condition that limits physical time step in explicit methods could make them inefficient in notably inhomogeneous or anisotropic transient flows, such as wall-bounded turbulence. Such inefficiency could be overcome by applying implicit ODE solvers which usually have better stability attributes. Due to Dahlquist's second barrier [25], only second and first order linear multistep ODE methods could achieve A-stability. Therefore, the L-stable second-order backward differentiation formula (BDF2) is extensively adopted in solving transient CFD problems. Different from multistep methods, the single-step implicit Runge-Kutta (IRK) methods with multiple internal stages can achieve higher order of accuracy while preserving stability [26]. Among the IRK methods, fully coupled IRK methods could achieve optimal order for given number of stages, but they require the solution of a nonlinear system with its dimension several times larger than that of the ODE. The enlarged algebraic system could be especially troublesome for its implementation in CFD solvers. Pazner and Persson [27] made effort in efficiently solving fully IRK methods with DG, and Jameson [28] discussed how to adopt dual time stepping into fully IRKs. Due to the difficulties in solving fully IRK methods, singly diagonally implicit Runge-Kutta (SDIRK) methods are more commonly used in CFD, for example in [15]. SDIRK methods have lower-triangular butcher tableau, enabling the stage values to be solved in a sequence. As a special case of SDIRK, ESDIRK methods are SDIRK with an explicit first stage, which are constructed to have second stage order while being L-stable [29, 30]. High-order ESDIRK schemes and BDF2 have been tested in [31, 32] to solve flow problems, whose results illustrate better accuracy and higher efficiency of high-order ESDIRK methods compared to second order BDF2.

Apart from fully IRK and SDIRK methods, another class of implicit RK methods, mono implicit RK methods (MIRK) [33] with explicit internal stages and an implicit final stage, allow the implicit system to have the same number of dimensions as the original ODE. Cash and Signhal derived examples of A-stable and L-stable high-order MIRK methods in [34, 35]. Kulikov and Shindin presented a similar type of method called nested implicit RK (NIRK) [36]. Discussion was made on symmetry, stiff accuracy and other advantageous properties of a series of Gauss type NIRK methods. [37]. The major drawback of NIRK or MIRK schemes is that the Jacobian matrices of the nonlinear algebraic problems for each NIRK or MIRK step are polynomials of the Jacobian of the ODE's right-hand-side. Cash and Singhal proposed to find MIRK methods whose Jacobian could be approximately factorized, so that the Newton iteration could be realized approximately by solving a series of successive linear problems [35]. Kulikov and Shindin found certain factorization approximation could sabotage the stability of the method in [37], and analyzed how to choose the approximation of Jacobian in [38]. MIRK and NIRK methods are more attractive than SDIRK methods for they require fewer inner stages. Typical fourth order stiffly accurate ESDIRK requires 5 implicit internal stages to be solved, while MIRK and NIRK could solve only one implicit stage.

MIRK and NIRK methods have less implicit stages than SDIRK methods, which gives them potential to gain better efficiency in large scale problems. However, current literature has seldom explored the application of MIRK or NIRK methods in solving PDEs. The fully IRK methods, on the other hand, have been practiced in finite volume [28] and DG [27] as mentioned before, but their implementations in CFD methods are significantly more complex compared to SDIRK type methods.

Different from previous approaches, the current paper aims to construct implicit time marching schemes based on temporal reconstruction. By combining polynomial approximation and numerical quadrature rules in the direction of time, stable, high-order accurate and efficient implicit ODE methods termed as DITR could be obtained straightforwardly. The basic idea of the DITR scheme will be presented in Section 3. Briefly speaking, the construction of the DITR schemes consists of the following two steps. The first step is the direct integration of the system of first-order ODEs. When the right-hand side is integrated using the three point Gauss-Lobatto rule, the resulting DITR schemes have two stages and are at most temporally fourth-order accurate. The actual order of accuracy is then determined by the second step which is the temporal reconstruction. When the temporal reconstruction uses only the information of n and $n+1$ steps, the DITR schemes are single step schemes similar to the MIRK or NIRK schemes. When the temporal reconstruction uses the information of $n-1$, n and $n+1$ steps, the resulting DITR schemes are multistep methods that are different from the common implicit linear multistep methods such as the Adams-Moulton and BDF schemes [39]. The stability of the DITR schemes is mainly determined by the temporal reconstruction step. When adopting proper quadrature rules and temporal reconstructions in these two steps, single-step and multistep methods with arbitrary orders of temporal accuracy can be constructed. Therefore, the DITR approaches are new methodologies for designing implicit time marching schemes that are different from the IRK and linear multistep methods. It will be shown that the new DITR schemes can be implemented in a matrix-free style, guaranteeing their portability. The new implicit time marching schemes are further applied to high-order compact finite volume method to test their capabilities in flow problems. Numerical results illustrate that the DITR schemes are high-order accurate and competitive with the 4th order ESDIRK method in efficiency.

The rest of the paper is organized as follows. High-order compact finite volume schemes are introduced in Section 2 as the spatial discretization. The construction and analyses of DITR methods are presented in Section 3. Section 4 illustrates some numerical results of DITR solving the Euler and N-S equations, combined with spatial discretization from Section 2. Last, Section 5 gives the concluding remarks.

2 High-order compact finite volume method

The new time-marching method developed in the current paper can be applied to any spatial discretizations when solving time-dependent PDEs such as the compressible Euler and Navier-Stokes equations. However, the time marching schemes' actual performance will be affected by the spatial discretizations. Therefore, this section will provide a brief description of finite volume spatial discretization, and additional details of the high-order finite volume scheme used in numerical tests are specified.

2.1 Governing equations

The compressible Navier-Stokes equations have the conservative form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F} - \mathbf{F}_v) = 0 \tag{1}$$

where $\mathbf{U}$ is vector of conservative variables and $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$, $\mathbf{F}_v = [\mathbf{F}_{v,1}, \mathbf{F}_{v,2}, \mathbf{F}_{v,3}]$ are inviscid and viscous tensors of their flux. In Cartesian coordinates $x_k, k = 1, 2, 3$, the components are

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{bmatrix}, \quad \mathbf{F}_j = \begin{bmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ (E+p)u_j \end{bmatrix}, \quad \mathbf{F}_{v,j} = \begin{bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \\ \sum_{k=1}^{3} u_k \tau_{kj} - K_j \end{bmatrix} \tag{2}$$

where ρ is density, $u_i, i = 1, 2, 3$ are velocity components, p is pressure, E is total energy per unit volume, $\tau_{ij}, i, j = 1, 2, 3$ are viscous stress tensor components and $K_i, i = 1, 2, 3$ are heat flux components. δ_{ij} is Kronecker delta. With ideal gas equation of state, Newtonian viscosity and Fourier heat conduction, additional relations

$$\begin{aligned} E &= \frac{p}{\gamma - 1} + \frac{1}{2}\rho \sum_{k=1}^{3} (u_k u_k), \\ p &= \rho R_g T, \\ \tau_{ij} &= \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}\mu \delta_{ij} \sum_{k=1}^{3} \frac{\partial u_k}{\partial x_k}, \\ K_i &= -\kappa \frac{\partial T}{\partial x_i} \end{aligned} \tag{3}$$

are used to close the equations with T being temperature, γ being specific heat ratio, R_g being gas constant, μ being dynamic viscosity and κ being thermal conductivity. Specific heat ratio γ is fixed to 1.4 in this paper. The current paper only considers using simple gas property with $\kappa = \mu c_p / Pr$, and $\mu = \mu_\infty$ being a constant, while c_p is special heat capacity at constant pressure and Prandtl number Pr is fixed to 0.71 in this paper. When $\mu = 0$, $\mathbf{F}_v = 0$ and equation (1) becomes Euler equation. The equations discussed above are in 3D form, and assuming constant distribution of values over x_3 yields the 2D version of NS and Euler equations.

2.2 High-order finite volume spatial discretization

This subsection provides a general framework of high-order finite volume discretization. The computational domain $\mathbf{\Omega}$ is divided into N_{cell} cells $\mathbf{\Omega}_i, i = 1, 2, ...N_{cell}$ which are non-overlapping, forming a mesh. An averaging of conservative quantities over each cell is

$$\overline{\mathbf{U}}_i = \frac{1}{\overline{\mathbf{\Omega}}_i} \int_{\mathbf{\Omega}_i} \mathbf{U}(\mathbf{x}) \mathrm{d}\Omega \tag{4}$$

where $\overline{\mathbf{\Omega}}_i$ is the volume of $\mathbf{\Omega}_i$.

Next, a degree k piecewise polynomial reconstruction is conducted to approximate the distribution of quantities

$$\mathbf{U}_i(\mathbf{x}) = \overline{\mathbf{U}}_i + \sum_{l=1}^{\mathrm{NDOF}(k)} \mathbf{U}_i^l \varphi_{i,l}(\mathbf{x}) \tag{5}$$

in which $\mathbf{U}_j(\mathbf{x})$ is the local polynomial distribution on cell $\mathbf{\Omega}_j$, and $\varphi_{j,i}(\mathbf{x})$ are polynomial basis functions. Eq.(5) can also be considered as a scalar field

$$u_i(\mathbf{x}) = \overline{u}_i + \sum_{l=1}^{\mathrm{NDOF}(k)} u_i^l \varphi_{i,l}(\mathbf{x}) \tag{6}$$

where $u_i(\mathbf{x})$ is one of $\rho(\mathbf{x}), \rho u_1(\mathbf{x}) \ldots$ and should not be confused with the velocity components u_1, u_2, u_3

which was used in the governing equations.

There are NDOF(k) polynomial bases for reconstruction on each cell. The current paper uses zero-mean Taylor basis for reconstruction [15]. Given a specific reconstruction method, the reconstruction coefficients $\mathbf{U}_i^l$ (or u_i^l for its components) can be determined using the distribution of mean value $\overline{\mathbf{U}}_i, i = 1, 2...N_{cell}$ and boundary conditions.

With the piecewise polynomial approximation, the PDEs (1) can therefore become ODEs with cell averaging applied

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t} + \sum_{j \in S_i, j \neq i} \left(\int_{f_{i,j}} [\tilde{\mathbf{F}}(\mathbf{U}_i, \mathbf{U}_j) - \tilde{\mathbf{F}}_v(\mathbf{U}_i, \mathbf{U}_j, \nabla \mathbf{U}_i, \nabla \mathbf{U}_j)] \cdot \mathbf{n} \mathrm{d}A \right) = 0 \tag{7}$$

where $\tilde{\mathbf{F}}(\mathbf{U}_i, \mathbf{U}_j)$ and $\tilde{\mathbf{F}}_v(\mathbf{U}_i, \mathbf{U}_j, \nabla \mathbf{U}_i, \nabla \mathbf{U}_j)$ are approximations of exact fluxes on the cell interfaces, and $f_{i,j} = \mathbf{\Omega}_i \cap \mathbf{\Omega}_j$ is the interface between i, j cells. Set S_i denotes the compact stencil of cell $\mathbf{\Omega}_i$, consisting of cell $\mathbf{\Omega}_i$ and its direct face neighboring cells. The piecewise polynomial approximation (5) does not guarantee a continuous distribution on interfaces, thus the numerical fluxes $\tilde{\mathbf{F}}(\mathbf{U}_i, \mathbf{U}_j)$, $\tilde{\mathbf{F}}_v(\mathbf{U}_i, \mathbf{U}_j, \nabla \mathbf{U}_i, \nabla \mathbf{U}_j)$ are functions of the approximate field on both sides. Inviscid numerical flux $\tilde{\mathbf{F}}(\mathbf{U}_i, \mathbf{U}_j)$ is typically computed with an approximate Riemann solver which will be specified for each numerical test in the paper. Numerical viscous flux $\tilde{\mathbf{F}}_v(\mathbf{U}_i, \mathbf{U}_j, \nabla \mathbf{U}_i, \nabla \mathbf{U}_j)$ in this paper follows the practice of Wang [15]. The integration of flux terms on the cell interfaces are conducted using numerical quadrature rules with enough algebraic precision.

As the spatial derivatives are approximated in (7), it is referred to as the semi-discretized form of finite volume method. The inviscid term has a truncation error of $O(h^{k+1})$ for smooth problems, with h being the size of mesh. As the approximate fields $\mathbf{U}_i$ are functions of average values $\overline{\mathbf{U}}_i$, (7) can be rearranged into the assembled form

$$\frac{d\overline{\mathbf{U}}}{dt} = \mathbf{R}(t, \overline{\mathbf{U}}) \tag{8}$$

with

$$\overline{\mathbf{U}} = \begin{bmatrix} \overline{\mathbf{U}}_1 \\ \dots \\ \overline{\mathbf{U}}_{N_{cell}} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} -\sum_{j \in S_1, j \neq 1} \left(\int_{f_{1,j}} [\tilde{\mathbf{F}} - \tilde{\mathbf{F}}_v] \cdot \mathbf{n} dA \right) \\ \dots \\ -\sum_{j \in S_{N_{cell}}, j \neq N_{cell}} \left(\int_{f_{N_{cell},j}} [\tilde{\mathbf{F}} - \tilde{\mathbf{F}}_v] \cdot \mathbf{n} dA \right) \end{bmatrix}. \tag{9}$$

The arguments in the numerical fluxes are omitted. The right-hand-side vector $\mathbf{R}$ is also a function of t because boundary conditions or additional source terms could depend on t generally speaking.

2.3 Variational reconstruction

In order to determine the coefficients of polynomial bases $\mathbf{U}_i^l$ (or u_i^l) in Eq.(5), a reconstruction method needs to be specified. Traditional second-order FV methods for unstructured grid needs only to reconstruct a $k = 1$ polynomial, namely linear distribution on each cell. The variational reconstruction [15] is a compact high-order reconstruction scheme, which features high-order accuracy achieved on a compact stencil. The current section will explain the variational reconstruction and specify relevant details.

The current paper uses local zero-mean Taylor basis, similar to [15]. Taking 2 dimensional polynomials as an example:

$$\varphi_{i,l} = \left(\frac{x - x_{c,i}}{\Delta x_i} \right)^{p_l} \left(\frac{y - y_{c,i}}{\Delta y_i} \right)^{q_l} - \overline{\left(\frac{x - x_{c,i}}{\Delta x_i} \right)^{p_l} \left(\frac{y - y_{c,i}}{\Delta y_i} \right)^{q_l}} \tag{10}$$

where $[x_{c,j}, y_{c,j}]^{\mathrm{T}} = \mathbf{x}_{c,j}$ are barycenters of cell $\mathbf{\Omega}_j$, $\Delta x_i, \Delta y_i$ are the cell's length scale, and p_l, q_l are powers of the bases. For $k = 3$ basis used in two-dimensional cases, $p_l = [1, 0, 2, 1, 0, 3, 2, 1, 0]_l$, and $q_l = [0, 1, 0, 1, 2, 0, 1, 2, 3]_l$. The mean value term lets the basis have zero mean values, which is calculated with:

$$\overline{\left(\frac{x - x_{c,i}}{\Delta x_i} \right)^{p_l} \left(\frac{y - y_{c,i}}{\Delta y_i} \right)^{q_l}} = \frac{1}{\overline{\mathbf{\Omega}}_j} \int_{\mathbf{\Omega}_j} \left(\frac{x - x_{c,i}}{\Delta x_i} \right)^{p_l} \left(\frac{y - y_{c,i}}{\Delta y_i} \right)^{q_l} \mathrm{d}\Omega. \tag{11}$$

The length scales are chosen to be the largest distance of nodes from the barycenter, which has the from

$$\Delta x_j = \Delta y_j = \max_k (\|\mathbf{x}_{j,k} - \mathbf{x}_{c,j}\|_2) \tag{12}$$

where $\mathbf{x}_{j,k}$ are coordinates of nodes belonging to cell $\mathbf{\Omega}_j$.

The variational reconstruction defines the solution of reconstruction coefficients to be the minimum point of a globally defined functional I

$$I = \sum I_f \tag{13}$$

where I_f are interface jump integrations (IJI) defined on each cell interface f. Hereafter, we use $f = f_{ij}$ to denote the interface of cell $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_j$ by default. Using the two-dimensional case as an example, the current paper uses IJI in the form of

$$I_f = \omega_f^G \int_{\mathbf{\Omega}_i \cap \mathbf{\Omega}_j} \sum_{p+q=0}^{p+q \le k} \left[\omega_f^D(p,q) d_{ij}^{p+q} \left(\frac{\partial^{p+q} u_i(x,y)}{\partial x^p \partial y^q} - \frac{\partial^{p+q} u_j(x,y)}{\partial x^p \partial y^q} \right) \right]^2 \mathrm{d}f \tag{14}$$

where ω_f^G is geometric weight and ω_f^D is derivative weight. The facial length scale d_{ij} is taken as

$$d_{ij} = \|\mathbf{x}_{c,i} - \mathbf{x}_{c,j}\|_2 \tag{15}$$

which is the distance between the barycenters of the cells. The IJI represents facial discontinuities of reconstruction polynomials using function values and their various partial derivatives on the interface. For 2-D simulation and cubic (with $k = 3$) reconstruction, the current paper uses

$$\begin{aligned}
&\omega_f^D(0,0) = 1 \times \omega_D(0) \\
&\omega_f^D(1,0) = \omega_f^D(0,1) = 1 \times \omega_D(1) \\
&\omega_f^D(2,0) = \omega_f^D(0,2) = 1 \times \omega_D(2),\ \omega_f^D(1,1) = \sqrt{2} \times \omega_D(2) \\
&\omega_f^D(3,0) = \omega_f^D(0,3) = 1 \times \omega_D(3),\ \omega_f^D(1,2) = \omega_f^D(2,1) = \sqrt{3} \times \omega_D(3)
\end{aligned} \tag{16}$$

with

$$\omega_D(0) = 1, \omega_D(1) = 0.5925, \omega_D(2) = \omega_D(3) = 0.2117 \tag{17}$$

as the derivative weight. This form of derivative weight in Eq.(16) is rotational invariant and isotropic. The coefficients in Eq.(17) are taken from [40], which were obtained through the optimization of dissipation and dispersion relations.

The geometric weight uses the form

$$\omega_f^G = \sqrt{S^{\frac{1}{d-1}} d_{ij}^{-1}} \tag{18}$$

where d is the number of spatial dimension and S is the area of the interface. This form of geometric weight is also taken from [40].

The minimization problem

$$[u_i^l] = \arg\min(I) \tag{19}$$

defines the solution to the variational reconstruction system. It is observed that I is a quadratic function of $u_i^l, i = 1...N_{cell}$ and $I \ge 0$, therefore I is a positive semi-definite quadratic form about the global vector of reconstruction coefficients. Consequently, the minimization problem converts to a linear equation

$$\frac{\partial I}{\partial u_i^l} = 0, l = 1, 2, \ldots \mathrm{NDOF}(k), \quad i = 1, 2, \ldots N_{cell}. \tag{20}$$

This linear equation can be arranged in a cell-block form:

$$\sum_{f=f_{ij}, j \in S_i, j \ne i} \mathbf{A}_{ij} \mathbf{u}_i = \sum_{f=f_{ij}, j \in S_i, j \ne i} \mathbf{B}_{ij} \mathbf{u}_j + \mathbf{b}_{ij} (\overline{u}_j - \overline{u}_i) \tag{21}$$

where S_i is the compact stencil of i composed of cell $\mathbf{\Omega}_i$ and its face neighbors. Local reconstruction coefficient vector is $\mathbf{u}_i = [u_i^1, u_i^2 \dots]^{\mathrm{T}}$. The reconstruction coefficient matrices and vectors in Eq.(21) are

$$
\begin{aligned}
\mathbf{A}_{ij} &= \left[\left(\omega_f^G \int_{\mathbf{\Omega}_i \cap \mathbf{\Omega}_j} \sum_{p+q=0}^{p+q \le k} \omega_f^D(p,q)^2 d_{ij}^{2(p+q)} \frac{\partial^{p+q}\varphi_{i,r}}{\partial x^p \partial y^q} \frac{\partial^{p+q}\varphi_{i,l}}{\partial x^p \partial y^q} \mathrm{d}f \right)_{lr} \right]_{\mathrm{NDOF}(k)\times\mathrm{NDOF}(k)}, \\
\mathbf{B}_{ij} &= \left[\left(\omega_f^G \int_{\mathbf{\Omega}_i \cap \mathbf{\Omega}_j} \sum_{p+q=0}^{p+q \le k} \omega_f^D(p,q)^2 d_{ij}^{2(p+q)} \frac{\partial^{p+q}\varphi_{j,r}}{\partial x^p \partial y^q} \frac{\partial^{p+q}\varphi_{i,l}}{\partial x^p \partial y^q} \mathrm{d}f \right)_{lr} \right]_{\mathrm{NDOF}(k)\times\mathrm{NDOF}(k)}, \\
\mathbf{b}_{ij} &= \left[\left(\omega_f^G \omega_f^D(0,0)^2 \int_{\mathbf{\Omega}_i \cap \mathbf{\Omega}_j} \varphi_{i,l} \mathrm{d}f \right)_{l} \right]_{\mathrm{NDOF}(k)\times 1}.
\end{aligned}
\tag{22}
$$

The local coefficients matrices and vectors Eq.(22) can be calculated at the start of computation. The linear reconstruction system Eq.(21) is solved with block-SOR method. The details of linear solving of variational reconstruction and the treatments of boundary conditions can be found in [15]. The fact to be emphasized here is that the reconstruction is implicit, and the reconstruction coefficients must be obtained through a series of iterations. To ensure the efficiency of the numerical method, the variational reconstruction is used together with temporally implicit discretization of the governing equations. During an implicit solving procedure, the updating of mean values and reconstruction coefficients are decoupled and executed alternately, with the block-SOR iteration carried out only once for every iteration of the implicit marching scheme. As each block-SOR updating is compact in the sense of data dependency, the variational reconstruction is able to have better data locality and less communication during the iteration process.

3 Direct integration with temporal reconstruction (DITR) methods for high order time marching

3.1 General ideas

Considering the first order ODE equation (8) arising from high-order finite volume method with $t \in [0, \infty)$ and $\overline{\mathbf{U}} \in \mathbb{R}^N$:

$$
\frac{\mathrm{d}\overline{\mathbf{U}}}{\mathrm{d}t} = \mathbf{R}(t, \overline{\mathbf{U}})
$$

which can be considered a more general first order ODE with $\overline{\mathbf{U}}$ being a general solution vector. For example, in finite difference schemes, $\overline{\mathbf{U}}$ represents the vector of grid point values. And for finite volume schemes, $\overline{\mathbf{U}}$ is the vector of all cell averages. A direct integration of equation (8) leads to a time-marching relation

$$
\overline{\mathbf{U}}^{n+1} = \overline{\mathbf{U}}^{n} + \int_{t^n}^{t^{n+1}} \mathbf{R}(t, \overline{\mathbf{U}}) \mathrm{d}t \tag{23}
$$

In order to acquire the integration result in equation (23), a numeric quadrature rule in the interval $[t^n, t^{n+1}]$ is used. For example, the current paper uses quadrature rule based on three-point polynomial interpolation, and the direct integration has the form:

$$
\begin{aligned}
\frac{\overline{\mathbf{U}}^{n+1} - \overline{\mathbf{U}}^{n}}{\Delta t^n} =& \frac{1}{\Delta t^n} \int_{t^n}^{t^{n+1}} \mathbf{R}(t, \overline{\mathbf{U}}) \mathrm{d}t \\
\approx& b_1 \mathbf{R}(t^n, \overline{\mathbf{U}}(t^n)) + b_2 \mathbf{R}(t^{n+c_2}, \overline{\mathbf{U}}(t^{n+c_2})) + b_3 \mathbf{R}(t^{n+1}, \overline{\mathbf{U}}(t^{n+1}))
\end{aligned}
\tag{24}
$$

where $t^{n+c_2} = t^n + c_2(t^{n+1} - t^n)$ and $c_2 \in (0, 1)$. $\Delta t^n = t^{n+1} - t^n$ is the time step size. The parameter c_2 represents the relative place of the second abscissa in the quadrature rule, where the first and third fixed at t^n and t^{n+1}. Using quadratic polynomial interpolation, the weights of the quadrature rule will be:

$$\begin{aligned} b_1 &= \frac{1}{2} - \frac{1}{6c_2}, \\ b_2 &= \frac{1}{6c_2(1-c_2)}, \\ b_3 &= \frac{1}{2} - \frac{1}{6(1-c_2)}. \end{aligned} \tag{25}$$

The quadrature rule defined with Eq.(24) and Eq.(25) has algebraic precision of degree 2. When $c_2 = 1/2$, the quadrature rule has algebraic precision of degree 3 and becomes the three point Gauss-Lobatto rule. The numeric integration process used in equation (24) is referred to as a direct integration process because it is directly derived from the ODE.

Besides $\overline{\mathbf{U}}^{n+1}$, $\overline{\mathbf{U}}(t^{n+c_2})$ is also unknown. To make Eq.(25) solvable, the temporal reconstruction is introduced in the present paper. The temporal reconstruction is to reconstruct $\overline{\mathbf{U}}(t)$ from known $\overline{\mathbf{U}}$ at previous time steps and the desired solution $\overline{\mathbf{U}}^{n+1}$ of the numerical schemes. Since $\frac{\mathrm{d}\overline{\mathbf{U}}}{\mathrm{d}t} = \mathbf{R}$ can be computed from $\overline{\mathbf{U}}$, R at previous time steps as well as $\mathbf{R}^{n+1}$ can be also used in the temporal reconstruction. The inclusion of $\overline{\mathbf{U}}^{n+1}$ and $\mathbf{R}^{n+1}$ in the reconstruction makes the scheme implicit. Using the reconstructed $\overline{\mathbf{U}}(t)$, $\overline{\mathbf{U}}(t^{n+c_2})$ can be evaluated, making Eq.(25) solvable. The current paper has only considered using a subset of $\overline{\mathbf{U}}(t), \mathbf{R}(t, \overline{\mathbf{U}}(t))$ at $t \in \{t^{n+1}, t^n, t^{n-1}\}$ as conditions of the polynomial reconstruction. Generally, the polynomial interpolation of $\overline{\mathbf{U}}(t)$ could be expressed as

$$\begin{aligned} \overline{\mathbf{U}}(t) \approx\, & A_0^n(t)\overline{\mathbf{U}}^{n-1} + A_1^n(t)\overline{\mathbf{U}}^{n} + A_2^n(t)\overline{\mathbf{U}}^{n+1} \\ & + \Delta t^n D_0^n(t)\mathbf{R}^{n-1} + \Delta t^n D_1^n(t)\mathbf{R}^{n} + \Delta t^n D_2^n(t)\mathbf{R}^{n+1} \end{aligned} \tag{26}$$

where $A_i^n(t), D_i^n(t), i = 0, 1, 2$ are polynomial base functions. Due to the precision of quadrature rule, the polynomial reconstruction is only expected to reach 3rd order constructing a 4th order accurate scheme. Consequently, at most 4 out of the 6 conditions would be used at the same time. The polynomial interpolation in the direction of time in Eq.(26) is referred to as temporal reconstruction, for a continuous distribution of $\overline{\mathbf{U}}$ is reconstructed with point values, similar to the finite volume reconstruction.

Combining the direct integration in Eq.(24) and a temporal reconstruction in Eq.(26), a Direct Integration with Temporal Reconstruction (DITR) method is determined. The current paper sticks to the same formula of direct integration in Eq.(24), while experimenting on different forms of temporal reconstruction in Eq.(26).

As a remark, the Adams-Moulton scheme which is one of the implicit linear multistep methods also use the direct integration approach. However, it relies on the reconstruction of $\mathbf{R}(t)$ to make the numerical quadrature similar to the right-hand side of Eq.(25) computable. For the third order scheme, quadratic reconstruction in terms of $\mathbf{R}^{n+1}, \mathbf{R}^n, \mathbf{R}^{n-1}$ is used. It is well known that the high-order Adams-Moulton schemes have rather poor stability property. The present DITR scheme reconstructs U rather than R. According to Eq.(26), more information can be used in the reconstruction, making the present method more compact and flexible. The present approach can achieve high-order of accuracy and better stability property when the reconstruction is designed properly.

If the quadrature rule in direct integration is replaced with midpoint rule, and the temporal reconstruction uses linear reconstruction, the implicit midpoint method for ODE can be derived, which is 2nd order accurate. A high order accurate DITR method could only be obtained with a high order accurate quadrature rule and high degree temporal reconstruction.

Assuming the quadrature rule has algebraic precision of degree m, and the polynomial interpolation is of degree n, a straightforward analysis on local truncation error could be conducted. Approximation (24) yields a truncation error of $O((\Delta t)^{m+2})$ expressed in equation (27)

$$\begin{aligned} \overline{\mathbf{U}}^{n+1} - \overline{\mathbf{U}}^{n} =& \int_{t^n}^{t^{n+1}} \mathbf{R}(t, \overline{\mathbf{U}})\mathrm{d}t \\ =& \Delta t^n \left[b_1 \mathbf{R}(t^n, \overline{\mathbf{U}}(t^n)) + b_2 \mathbf{R}(t^{n+c_2}, \overline{\mathbf{U}}(t^{n+c_2})) + b_3 \mathbf{R}(t^{n+1}, \overline{\mathbf{U}}(t^{n+1}))\right] \\ & + O((\Delta t^n)^{m+2}) \end{aligned} \tag{27}$$

due to the precision degree of quadrature rule. Approximation Eq.(26) has a truncation error of

$O((\Delta t)^{n+1})$ expressed in Eq.(28)

$$\begin{aligned}\overline{\mathbf{U}}(t^{n+c_2}) = & A_0^n(t^{n+c_2})\overline{\mathbf{U}}^{n-1} + A_1^n(t^{n+c_2})\overline{\mathbf{U}}^{n} + A_2^n(t^{n+c_2})\overline{\mathbf{U}}^{n+1} \\ & + \Delta t^n D_0^n(t^{n+c_2})\mathbf{R}^{n-1} + \Delta t^n D_1^n(t^{n+c_2})\mathbf{R}^{n} + \Delta t^n D_2^n(t^{n+c_2})\mathbf{R}^{n+1} \\ & + O((\Delta t^n)^{n+1})\end{aligned} \tag{28}$$

as a result of polynomial degree. Substituting Eq.(28) into the t^{n+c_2} stage of Eq.(27), the truncation error of the entire scheme becomes $O((\Delta t)^{n+2}) + O((\Delta t)^{m+2})$:

$$\begin{aligned}\overline{\mathbf{U}}^{n+1} = & \overline{\mathbf{U}}^{n} + \Delta t^n \Big[b_1\mathbf{R}(t^n, \overline{\mathbf{U}}^{n}) + b_2\mathbf{R}(t^{n+c_2}, \overline{\mathbf{U}}^{n+c_2} + O((\Delta t^n)^{n+1})) + b_3\mathbf{R}(t^{n+1}, \overline{\mathbf{U}}^{n+1})\Big] \\ & + O((\Delta t^n)^{m+2}) \\ = & \overline{\mathbf{U}}^{n} + \Delta t^n \Big[b_1\mathbf{R}(t^n, \overline{\mathbf{U}}^{n}) + b_2\mathbf{R}(t^{n+c_2}, \overline{\mathbf{U}}^{n+c_2}) + b_3\mathbf{R}(t^{n+1}, \overline{\mathbf{U}}^{n+1})\Big] \\ & + O((\Delta t^n)^{m+2} + O((\Delta t^n)^{n+2}))\end{aligned} \tag{29}$$

where $\overline{\mathbf{U}}^{n+1}, \overline{\mathbf{U}}^{n}$ are step values assumed to be accurate here, and $\overline{\mathbf{U}}^{n+c_2}$ is the approximate c_2 stage value determined by the temporal reconstruction. Equation (29) assumes $\mathbf{R}(t, \overline{\mathbf{U}})$ to be sufficiently regular and therefore does not change the order of error. Therefore, for smooth problems the order of accuracy of a DITR method is theoretically $\min(m, n) + 1$.

The following sections will illustrate some practical DITR methods based on equations (24) and (26).

3.2 Variants of DITR methods

3.2.1 The DITR U2R2 method

To make the method single-step and 4th order accurate, we choose $\overline{\mathbf{U}}^{n}, \overline{\mathbf{U}}^{n+1}$, $\mathbf{R}^n, \mathbf{R}^{n+1}$ as the interpolation conditions, making the interpolation basically cubic Hermite interpolation. Therefore, the interpolation of $\overline{\mathbf{U}}(t^{n+c_2})$ is:

$$\begin{aligned}\overline{\mathbf{U}}^{n+c_2} & = a_{1,U2R2}\overline{\mathbf{U}}^{n} + a_{2,U2R2}\overline{\mathbf{U}}^{n+1} \\ & + \Delta t^n d_{1,U2R2}\mathbf{R}^n + \Delta t^n d_{2,U2R2}\mathbf{R}^{n+1}\end{aligned} \tag{30}$$

with $\overline{\mathbf{U}}^{n+c_2}$ being the numerical approximation of $\overline{\mathbf{U}}(t^{n+c_2})$ and the interpolation bases at c_2 node being:

$$\begin{aligned}a_{1,U2R2} & = 1 - (3{c_2}^2 - 2{c_2}^3), \\ a_{2,U2R2} & = 3{c_2}^2 - 2{c_2}^3, \\ d_{1,U2R2} & = c_2 - 2{c_2}^2 + {c_2}^3, \\ d_{2,U2R2} & = -{c_2}^2 + {c_2}^3.\end{aligned} \tag{31}$$

The temporal reconstruction of Eq.(30) combined with direct integration equation (24) forms the DITR U2R2 method. Here U2R2 denotes 2 $\overline{\mathbf{U}}$ and 2 $\mathbf{R}$ step values at $n, n+1$ used in the reconstruction. Following analyses indicate that using the most recent $\overline{\mathbf{U}}$ and $\mathbf{R}$ will yield a scheme with better stability. Therefore, the current paper does not include the time step indices in the naming of DITR methods. This naming convention is continued in the following methods.

When $c_2 = 1/2$, quadrature rule in equation (24) has precision of degree 3, making the DITR U2R2 method 4th order accurate, and the stage value $\overline{\mathbf{U}}^{n+c_2}$ has a precision of degree 3. When $c_2 \neq 1/2$, DITR U2R2 becomes 3rd order accurate.

In order to further examine the accuracy order, equations (31), (24) can be reformulated into a standard IRK method, yielding a Butcher tableau shown in Table 1.

0	0	0	0
c_2	$d_1 + a_2 b_1$	$a_2 b_2$	$d_2 + a_2 b_3$
1	b_1	b_2	b_3
	b_1	b_2	b_3

Table 1: Butcher tableau of DITR U2R2

According to table 1 with the coefficients determined by (31) and (25), one can find that the 4th order accurate DITR U2R2 method with $c_2 = 1/2$ is actually the Lobatto IIIA method of order 4 [39]. The classic order and stage order of DITR U2R2 could also be verified using Table 1 via the simplifying assumptions, which is a trivial procedure given the formulae provided in [39].

Following standard analysis based on Dahlquist's equation $\frac{dy}{dt} = \lambda y$ [39], the stability function giving by $y^1 = R(h\lambda)y^0$ applied to DITR U2R2 is in the form:

$$R_{U2R2}(z) = -\frac{4\,z - 2\,c_2\,z - c_2\,z^2 + z^2 + 6}{2\,z + 2\,c_2\,z - c_2\,z^2 - 6} \tag{32}$$

which becomes the (2,2)-Padé approximation when $c_2 = 1/2$ and DITR U2R2 becomes Lobatto IIIA. Analysis on equation (32) would show that $c_2 \in [1/2, 1)$ is a sufficient and necessary condition of DITR U2R2 being A-stable given $c_2 \in (0, 1)$. The limit at infinity

$$\lim_{z\to\infty} R_{U2R2}(z) = \frac{1 - c_2}{c_2} \tag{33}$$

confirm that DITR U2R2 is unable to achieve L-stability by adjusting c_2.

DITR U2R2 ($c_2 = 1/2$) or Lobatto IIIA method is symmetric, which is a preferable property when integrating reversible systems including orbital motion and particle systems. However, the symmetry in this ODE method could be considered harmful in CFD application. Most CFD systems of interest are physically dissipative, while for a symmetric method $R(z) \to 1$ when $z \to \infty$, which means the method is more likely to preserve spurious modes arising from spatial discretization. Although DITR U2R2 cannot achieve L-stability, using a value of $c_2 > 1/2$ would still produce $\lim_{z\to\infty} R(z) \in (0, 1)$, which would be a useful property in the simulation of dissipative systems. With $c_2 > 1/2$, stiff modes could vanish faster over the time steps, while $c_2 = 1/2$ tends to preserve them.

3.2.2 The DITR U2R1 method

Giving up one interpolation condition in DITR U2R2 forces the scheme to have 3rd order accuracy. The current paper removes $\mathbf{R}^n$ from U2R2, namely using $\overline{\mathbf{U}}^n$, $\overline{\mathbf{U}}^{n+1}$, $\mathbf{R}^{n+1}$ for interpolation, which is able to produce an L-stable DITR scheme.

Similar with U2R2, DITR U2R1 has the interpolation written as:

$$\begin{aligned} \overline{\mathbf{U}}^{n+c_2} &= a_{1,U2R1}\overline{\mathbf{U}}^n + a_{2,U2R1}\overline{\mathbf{U}}^{n+1} \\ &+ \Delta t^n d_{2,U2R1}\mathbf{R}^{n+1} \end{aligned} \tag{34}$$

with $\overline{\mathbf{U}}^{n+c_2}$ being the numerical approximation of $\overline{\mathbf{U}}(t^{n+c_2})$ and the interpolation bases at c_2 node being:

$$\begin{aligned} a_{1,U2R1} &= 1 - (2c_2 - {c_2}^2), \\ a_{2,U2R1} &= 2c_2 - {c_2}^2, \\ d_{2,U2R1} &= {c_2}^2 - c_2. \end{aligned} \tag{35}$$

The temporal reconstruction equation (34) combined with direct integration equation (24) forms the DITR U2R1 method.

The interpolation bases shown in (35) are quadratic. Therefore, the scheme yields 3rd order accuracy and the choice of c_2 does not affect the order of accuracy. Similar to U2R2, DITR U2R1's order of accuracy can be examined using standard procedures for Runge-Kutta methods[39].

The linear stability function for DITR U2R1 is

$$R_{U2R1}(z) = \frac{2\,z + 6}{z^2 - 4\,z + 6}, \tag{36}$$

which is (1,2)-Padé approximation and not affected by c_2. It can be found $|R_{U2R1}(z)| < 1, \forall \mathrm{Re}(z) < 0$, and obviously $\lim_{z\to\infty} R_{U2R1}(z) = 0$. Therefore, DITR U2R1 method is L-stable.

The stability function also shows that when $\mathbf{R}$ is a linear function of $\overline{\mathbf{U}}$, the solution is not affected by c_2. However, for nonlinear $\mathbf{R}$, c_2 changes the behavior of U2R1.

3.2.3 The DITR U3R1 method

To exploit the information when multiple previous steps are available, using conditions from t^{n-1} would be preferable. The current paper chooses $\overline{\mathbf{U}}^{n-1}$,$\overline{\mathbf{U}}^{n}$, $\overline{\mathbf{U}}^{n+1}$ and $\mathbf{R}^{n+1}$ as U3R1's interpolation conditions, as other choices do not produce sufficient linear stability. The interpolation at c_2 node becomes:

$$\begin{aligned}\overline{\mathbf{U}}^{n+c_2} &= a_{0,U3R1}\overline{\mathbf{U}}^{n} + a_{1,U3R1}\overline{\mathbf{U}}^{n} + a_{2,U3R1}\overline{\mathbf{U}}^{n+1} \\ &+ \Delta t^n d_{2,U3R1}\mathbf{R}^{n+1}\end{aligned} \tag{37}$$

with $\overline{\mathbf{U}}^{n+c_2}$ being the numerical approximation of $\overline{\mathbf{U}}(t^{n+c_2})$ and the interpolation bases at c_2 node being:

$$\begin{aligned}a_{0,U3R1} &= -\frac{c_2\left(c_2-1\right)^2}{\Theta\left(\Theta+1\right)^2}, \\ a_{1,U3R1} &= \frac{\left(\Theta+c_2\right)\,\left(c_2-1\right)^2}{\Theta}, \\ a_{2,U3R1} &= \frac{c_2\,\left(-\Theta^2\,c_2+2\,\Theta^2-\Theta\,{c_2}^2+3\,\Theta-2\,{c_2}^2+3\,c_2\right)}{\left(\Theta+1\right)^2}, \\ d_{2,U3R1} &= \frac{c_2\,\left(\Theta+c_2\right)\,\left(c_2-1\right)}{\Theta+1}.\end{aligned} \tag{38}$$

where $\Theta = \Delta t^{n-1}/\Delta t^n$.

The temporal reconstruction equation (37) combined with direct integration equation (24) forms the DITR U3R1 method.

The method is 4th order when $c_2 = 1/2$, when both the interpolation and integration has precision of degree 3. For the special case of $c_2 = 1/2$, simplified coefficients are given:

$$\begin{aligned}a_{0,U3R1} &= -\frac{1}{8\,\Theta\left(\Theta+1\right)^2}, \\ a_{1,U3R1} &= \frac{\Theta+\frac{1}{2}}{4\,\Theta}, \\ a_{2,U3R1} &= \frac{6\,\Theta^2+11\,\Theta+4}{8\left(\Theta+1\right)^2}, \\ d_{2,U3R1} &= -\frac{\Theta+\frac{1}{2}}{4\,\left(\Theta+1\right)}.\end{aligned} \tag{39}$$

Linear stability is analyzed when $\Theta = 1$ and $c_2 = 1/2$. The solution to the test problem produces two solutions:

$$\begin{pmatrix} R^{(1)}_{U3R1}(z) \\ R^{(2)}_{U3R1}(z) \end{pmatrix} = \begin{pmatrix} \frac{10\,z-\sqrt{-6\,z^3+129\,z^2+432\,z+576}+24}{6\,z^2-29\,z+48} \\ \frac{10\,z+\sqrt{-6\,z^3+129\,z^2+432\,z+576}+24}{6\,z^2-29\,z+48} \end{pmatrix}. \tag{40}$$

It is obvious $R^{(1)}_{U3R1}(z) \to 0, R^{(2)}_{U3R1}(z) \to 0$ when $z \to \infty$. When observed numerically, it is found both $|R^{(1)}_{U3R1}(z)|$ and $|R^{(2)}_{U3R1}(z)|$ are less than 1 in the left half plane of z. The current paper therefore believes the DITR U3R1 method is indeed L-stable. Note that DITR U3R1 is not a linear multistep method, and the root locus curve analysis used in those methods can not be directly applied here. To the author's knowledge, no such implicit time marching method has been reported in the literature.

With 4th order accuracy and observed L-stability, the DITR U3R1 method is potentially more favorable than U2R2 and U2R1.

3.2.4 Summary of DITR methods

The U2R2, U2R1 and U3R1 variants of the DITR method can be written is a unified form. The first equation is the direct integration:

$$\overline{\mathbf{U}}^{n+1} = \overline{\mathbf{U}}^{n} + \Delta t^n\left(b_1\mathbf{R}^n + b_2\mathbf{R}^{n+c_2} + b_3\mathbf{R}^{n+1}\right) \tag{41}$$

with weights decided by (25), and also listed Table 2. For simplicity, in numerical expressions, notations such as $\mathbf{R}^{n+c_2} = \mathbf{R}(t^{n+c_2}, \overline{\mathbf{U}}^{n+c_2})$ are used from now on.

b_1	b_2	b_3
$\frac{1}{2} - \frac{1}{6c_2}$	$\frac{1}{6c_2(1-c_2)}$	$\frac{1}{2} - \frac{1}{6(1-c_2)}$

Table 2: Butcher tableau of DITR U2R2

The second equation is the temporal reconstruction:

$$\overline{\mathbf{U}}^{n+c_2} = a_0^n \overline{\mathbf{U}}^{n-1} + a_1^n \overline{\mathbf{U}}^{n} + a_2^n \overline{\mathbf{U}}^{n+1} + \Delta t^n \left(d_1^n \mathbf{R}^n + d_2^n \mathbf{R}^{n+1} \right) \tag{42}$$

where the coefficients vary in different schemes according to Table 3. The time step ratio for the two-step U3R1 method is $\Theta = \Delta t^{n-1}/\Delta t^n$.

DITR Method	U2R2	U2R1	U3R1
a_0^n	0	0	$-\frac{c_2\,(c_2-1)^2}{\Theta\,(\Theta+1)^2}$
a_1^n	$1-(3{c_2}^2-2{c_2}^3)$	$1-(2c_2-{c_2}^2)$	$\frac{(\Theta+c_2)\,(c_2-1)^2}{\Theta}$
a_2^n	$3{c_2}^2-2{c_2}^3$	$2c_2-{c_2}^2$	$\frac{c_2\,\left(-\Theta^2\,c_2+2\,\Theta^2-\Theta\,{c_2}^2+3\,\Theta-2\,{c_2}^2+3\,c_2\right)}{(\Theta+1)^2}$
d_1^n	$c_2-2{c_2}^2+{c_2}^3$	0	0
d_2^n	$-{c_2}^2+{c_2}^3$	${c_2}^2-c_2$	$\frac{c_2\,(\Theta+c_2)\,(c_2-1)}{\Theta+1}$

Table 3: Interpolation coefficients for different DITR methods

In practice, the current research chooses $c_2 = 0.5, 0.55$ for U2R2. U2R2 $c_2 = 0.5$ has optimal order of accuracy (4th order), but it is symmetric. U2R2 $c_2 = 0.55$ is 3rd order accurate but breaks symmetry and gains some stability.

For U2R1, c_2 does not affect order of accuracy and stability. We empirically choose $c_2 = 0.25$ for U2R1, the reason roots in the convergence of solving PDEs using dual time stepping .

For U3R1, since any c_2 gives L-stability, so $c_2 = 0.5$ which offers optimal 4th order of accuracy is always used.

3.3 Iteration method for solving DITR

In practice, the implicit equations Eq.(41) and Eq.(42) are solved iteratively. In order to utilize the matrix-free linear solvers implemented for BDF and ESDIRK methods, the current section will introduce a iteration method. The equations are reformulated into

$$\begin{bmatrix} \mathcal{F}^{n+c_2} \\ \mathcal{F}^{n+1} \end{bmatrix} = 0 \tag{43}$$

where

$$\begin{bmatrix} \mathcal{F}^{n+c_2} \\ \mathcal{F}^{n+1} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathcal{G}^{n+c_2} \\ \mathcal{G}^{n+1} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \frac{a_0^n \overline{\mathbf{U}}^{n-1} + a_1^n \overline{\mathbf{U}}^{n} + a_2^n \overline{\mathbf{U}}^{n+1} - \overline{\mathbf{U}}^{n+c_2}}{\Delta t^n} + d_1^n \mathbf{R}^n + d_2^n \mathbf{R}^{n+1} \\ \frac{\overline{\mathbf{U}}^{n} - \overline{\mathbf{U}}^{n+1}}{\Delta t^n} + b_1 \mathbf{R}^n + b_2 \mathbf{R}^{n+c_2} + b_3 \mathbf{R}^{n+1} \end{bmatrix} \tag{44}$$

where $\mathcal{G}$ are the un-preconditioned residual functions and $\mathbf{P}$ is a preconditioning matrix:

$$\mathbf{P} = \begin{bmatrix} \mathrm{P}_{11}\mathbf{I} & \mathrm{P}_{12}\mathbf{I} \\ \mathrm{P}_{21}\mathbf{I} & \mathrm{P}_{22}\mathbf{I} \end{bmatrix} \tag{45}$$

where P_{ij} entries are scalar real values. Using the dual time stepping technique [41], and a linearized backward euler method for updating [42], the current paper proposes a iteration process with two sub-steps in each iteration:

$$\begin{aligned} \left(\frac{\mathbf{I}}{\Delta\tau} - \frac{\partial \mathcal{F}^{n+c_2}(\overline{\mathbf{U}}^{n+c_2}, \overline{\mathbf{U}}^{n+1})}{\partial \overline{\mathbf{U}}^{n+c_2}} \right) \Delta \overline{\mathbf{U}}^{n+c_2,m} &= \mathcal{F}^{n+c_2}(\overline{\mathbf{U}}^{n+c_2,m}, \overline{\mathbf{U}}^{n+1,m}) \\ \left(\frac{\mathbf{I}}{\Delta\tau} - \frac{\partial \mathcal{F}^{n+1}(\overline{\mathbf{U}}^{n+c_2}, \overline{\mathbf{U}}^{n+1})}{\partial \overline{\mathbf{U}}^{n+1}} \right) \Delta \overline{\mathbf{U}}^{n+1,m} &= \mathcal{F}^{n+1}(\overline{\mathbf{U}}^{n+c_2,m+1}, \overline{\mathbf{U}}^{n+1,m}) \end{aligned} \tag{46}$$

where $\Delta\overline{\mathbf{U}}^{n+c_2,m} = \overline{\mathbf{U}}^{n+c_2,m+1} - \overline{\mathbf{U}}^{n+c_2,m}$ and $\Delta\overline{\mathbf{U}}^{n+1,m} = \overline{\mathbf{U}}^{n+1,m+1} - \overline{\mathbf{U}}^{n+1,m}$. In the first updating formula in Eq.(46), $\overline{\mathbf{U}}^{n+1}$ is considered static and the linearizing is only performed on $\overline{\mathbf{U}}^{n+c_2}$. The second updating formula then makes $\overline{\mathbf{U}}^{n+c_2}$ static, and performs linearizing on $\overline{\mathbf{U}}^{n+1}$ only. Note that the right side $\mathcal{F}$ functions and Jacobian terms $\frac{\partial\mathcal{F}}{\partial\overline{\mathbf{U}}}$ always uses the latest versions of $\overline{\mathbf{U}}^{n+c_2}$ and $\overline{\mathbf{U}}^{n+1}$. Both $\overline{\mathbf{U}}^{n+c_2}$ and $\overline{\mathbf{U}}^{n+1}$ are updated immediately after each increment values are obtained. The linear systems are in the same form of those in BDF and ESDIRK methods, and they can be solved using matrix-free linear solvers like LU-SGS solvers.

The current paper chooses preconditioning to be

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \beta\mathbf{I} \\ 0 & \mathbf{I} \end{bmatrix} \tag{47}$$

and β values are chosen for each DITR method. For U2R2 $c = 0.5$, we empirically choose $\beta = 1$. For U2R2 $c = 0.55$, we empirically choose $\beta = 1.333$. For U2R2 $c = 0.25$, we empirically choose $\beta = 1$. For U3R1, we empirically use $\beta = 1.333$ for all Θ.

4 Numerical tests

During numerical tests, BDF2 and ESDIRK4 methods taken from [29, 31] are chosen to be baseline time marching methods. For DITR methods, instances of U2R2 $c_2 = 0.5$, U2R2 $c_2 = 0.55$, U2R1 and U3R1 are tested.

The isentropic vortex, two dimensional vortex shedding and double Mach reflection problems use P^3 variational reconstruction finite volume method declared in section 2 as spatial discretization. Iterative solution of the implicit reconstruction is conducted before each right-hand-side evaluation, which consists of 1 block-Jacobi iteration by default. Pseudo time iteration on mean values are solved using 5 times of block-Jacobi iteration by default, which is found both stable and efficient enough for VFV solving transient problems. The inviscid numerical flux for the isentropic vortex and vortex shedding problems are approximate Riemann solvers of Roe [43]. For the double Mach reflection problem, the local Lax-Friedrichs flux is used.

4.1 Isentropic vortex

The isentropic vortex problem is a classic accuracy testing problem for Euler equations. The settings can be found in [22]. The free-stream flow is $(\rho, u, v, p) = (1, 1, 1, 1)$, and a perturbation at initial time:

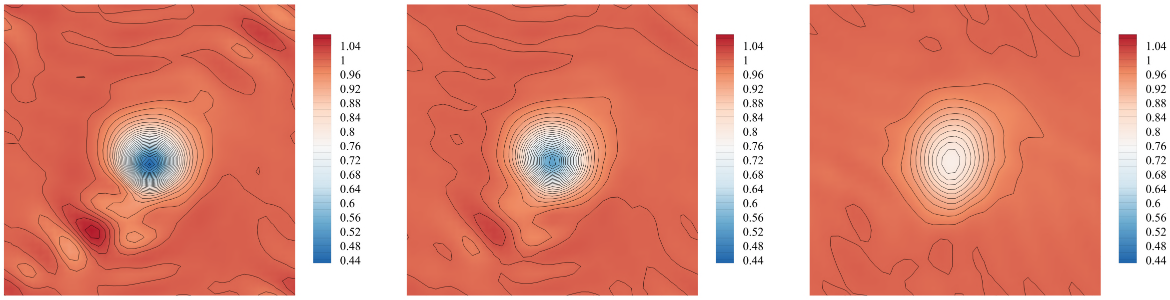
$$\begin{cases} (\delta u, \delta v) & = \frac{\epsilon}{2\pi}\exp\left(\frac{1-[(x-x_c)^2+(y-y_c)^2]}{2}\right)(-y+y_c, x-x_c) \\ \delta T & = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}\exp\left(1-[(x-x_c)^2+(y-y_c)^2]\right) \\ \delta S & = 0 \end{cases} \tag{48}$$

with ideal gas setting of $T = p/\rho, S = p/\rho^\gamma, \gamma = 1.4$. Initial vortex center is chosen $(x_c, y_c) = (5, 5)$, and vortex strength is $\epsilon = 5$. The analytic solution to the isentropic vortex problem is a translation of initial field with speed $(1, 1)$. The computational domain is $[0, 10] \times [0, 10]$, using periodic boundary conditions.

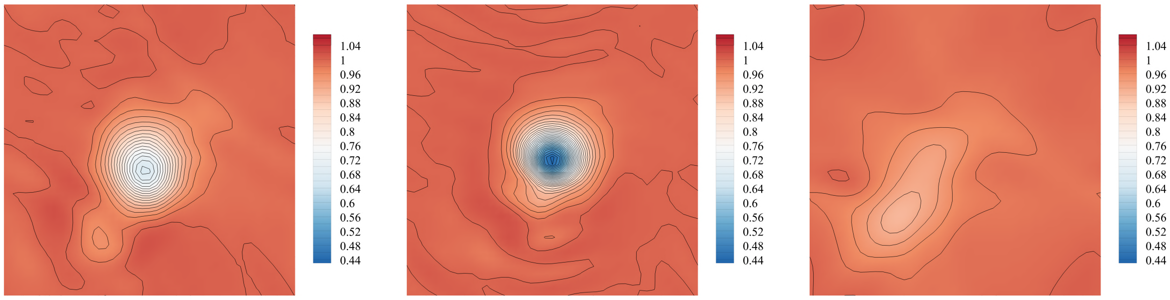
First, the implicit ODE integrators are tested with aggressively large time steps. The mesh is 40×40 square grid, and solution is calculated until $t = 10$ with $\Delta t = 1$. The CFL number based on $\Delta t, \Delta x$ is roughly 11, making the propagation of the vortex hard to simulate.

Results of large time step testing are shown in figure 1. Clearly, from figure 1f, BDF2 almost completely smears the initial vortex with only 10 steps for one period. The higher order methods somehow preserve the characteristics of a vortex. DITR U2R2 $c_2 = 0.5$ produces significant numerical oscillation along the propagation direction as shown in figure 1a, while the DITR U2R2 $c_2 = 0.55$ inhibits them better in figure 1b. The peak value in the vortex center produced by DITR U2R2 $c_2 = 0.5$ is comparable with ESDIRK4, while DITR U2R2 $c_2 = 0.55$ gives a flatter density peak. Both U2R1 and U3R1 methods are L-stable like ESDIRK4, and they are indeed better at suppressing non-physical oscillations. Note that it seems ESDIRK4 outperforms all the DITR methods in this case, but DITR only needs 2 internal stages to be solved while ESDIRK4 needs 5.

Next, precision and efficiency of different ODE methods are qualitatively evaluated with isentropic vortex solved on a 160×160 grid until $t = 2$. The fine mesh makes spatial discretization error negligible



(a) DITR U2R2 $c_2 = 0.5$ (b) DITR U2R2 $c_2 = 0.55$ (c) DITR U2R1



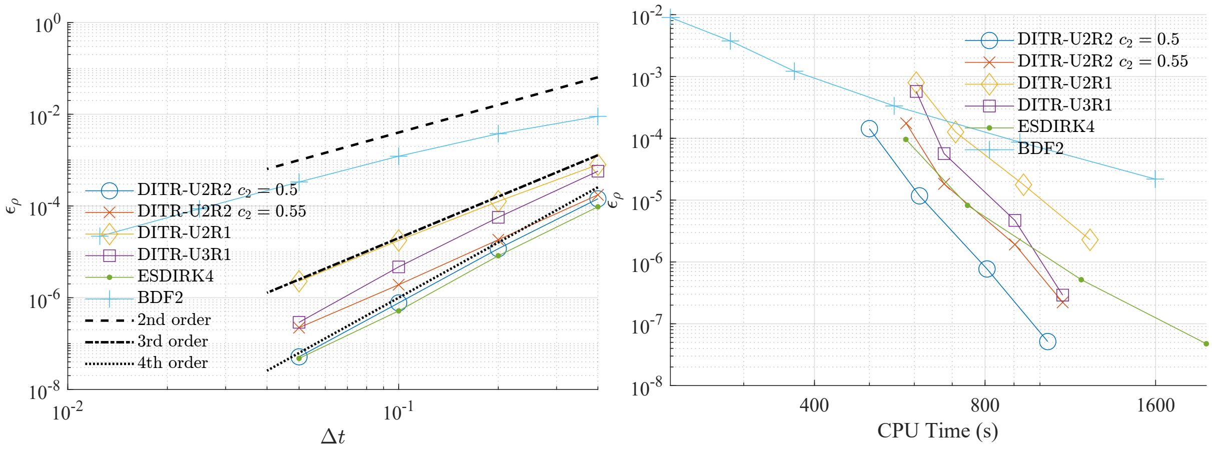
(d) DITR U3R1 (e) ESDIRK4 (f) BDF2

Figure 1: Density of isentropic vortex problem, with aggressively large time step $\Delta t = 1$ at $t = 10$

compared with time marching error. The density error is defined as an $L1$ norm in the form of

$$\epsilon_\rho = \frac{\int |\rho - \rho_a| \mathrm{d}x\mathrm{d}y}{100} \tag{49}$$

with ρ the numeric result of density and ρ_a the analytic result.



(a) Density error vs. time step size (b) Density error vs. total CPU time consumption

Figure 2: Convergence and efficiency test with isentropic vortex problem

Results of convergence and efficiency study with isentropic vortex is shown in figure 2.

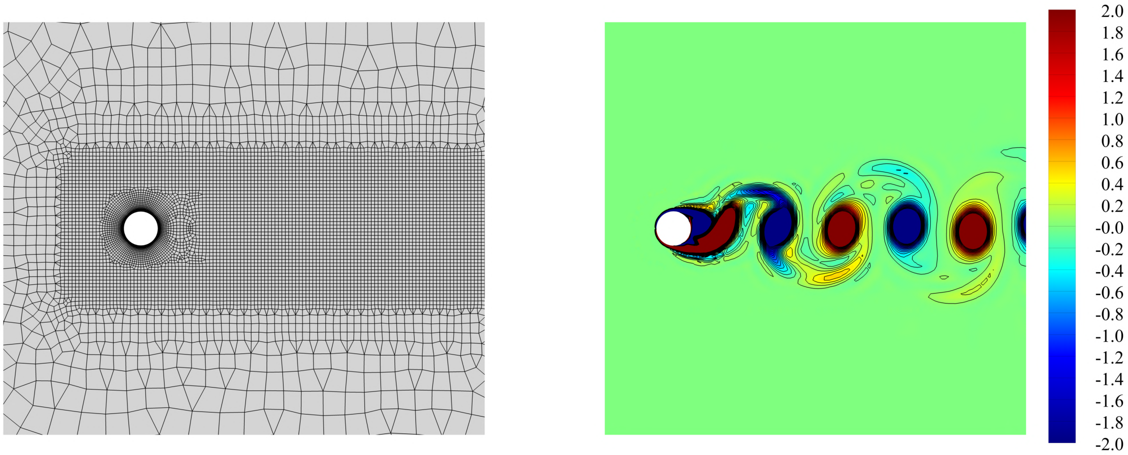
Figure 2a shows that with the same time step size, ESDIRK4 has the best accuracy, while DITR U2R2 $c_2 = 0.5$ is close to ESDIRK4 when time step is refined. DITR U2R2 $c_2 = 0.55$ is less accurate than DITR U2R2 $c_2 = 0.5$, but it performs almost as well as DITR U2R2 $c_2 = 0.5$ when time step is large. DITR U3R1 and U2R1 are less accurate than U2R2 methods with the same time step.

As of order of error, in Figure 2a, error of ESDIRK4 is barely able to reach 4th order of convergence, while DITR U2R2 $c_2 = 0.5$ and U3R1 methods are also able to reach 4th order convergence with smaller time steps. U2R1 and U2R2 $c_2 = 0.55$ methods are closer to the 3rd order slope, which conforms with their theoretical order.

Figure 2b implies that when consuming the same computational resource, DITR U2R2 methods have the best accuracy and efficiency. All the high-order time marching methods have better efficiency than BDF2, while DITR U2R2 methods have better efficiency than ESDIRK4. The symmetric DITR U2R2 $c_2 = 0.5$ has better efficiency compared with more stable DITR U2R2 $c_2 = 0.55$.

4.2 Two dimensional vortex shedding

Vortex shedding from a circular cylinder and forming a vortex street is a classic test problem for transient fluid simulation. Due to the refined mesh near solid wall, such cases usually prefer implicit time marching over explicit ones whose time steps are bounded by CFL condition. The current paper studies the 2D laminar case, where Reynolds number $Re_d = \rho_\infty u_\infty d/\mu_\infty$ is 1200, with Mach number being $Ma = 0.1$. Parameters are normalized so that freestream speed, density and diameter of the cylinder are unit values. Small Mach number makes the flow more incompressible, and the speed of sound makes time steps in explicit time marching restricted. Implicit time marching schemes can automatically omit the restrictions of the speed of sound, thus being potentially more favorable.



(a) Mesh

(b) Vorticity distribution

Figure 3: Mesh and a instance of z-vorticity distribution in $Re = 1200$ vortex shedding problem

Figure 3a demonstrates the unstructured grid used in the $Re = 1200$ vortex street calculation, and Figure 3b demonstrates z-vorticity distribution in the 2-D vortex street after it is fully developed.

In order to quantitatively compare different time marching schemes, the time marching error is compared. Using the same mesh as in Figure 3a and the same compact FV spatial discretization, a numeric reference solution is calculated with ESDIRK4 using very fine time step $\Delta t = 0.00125$. In the reference solution, restart information at $t = 200$ is stored, in which the vortex street has fully developed. Next, starting from the $t = 200$ flow field, combined with different time marching schemes and time step sizes varying from 0.04 to 0.1, transient flow is simulated until $t = 210$. CPU consumption and error values are evaluated. The transient errors are defined with

$$\begin{aligned}\|\epsilon_{My}\|_{t,2}^2 =& \frac{1}{N_t}\sum_{i=1}^{N_t}\left(M_{y,i} - M_{y,ref}(t_i)\right)^2 \\ \|\epsilon_{Mx}\|_{t,2}^2 =& \frac{1}{N_t}\sum_{i=1}^{N_t}\left(M_{x,i} - M_{x,ref}(t_i)\right)^2\end{aligned} \tag{50}$$

where N_t is the number of time steps for $t \in (200, 210]$, t_i is the time on time steps, and $M_{y,i}$ and $M_{x,i}$

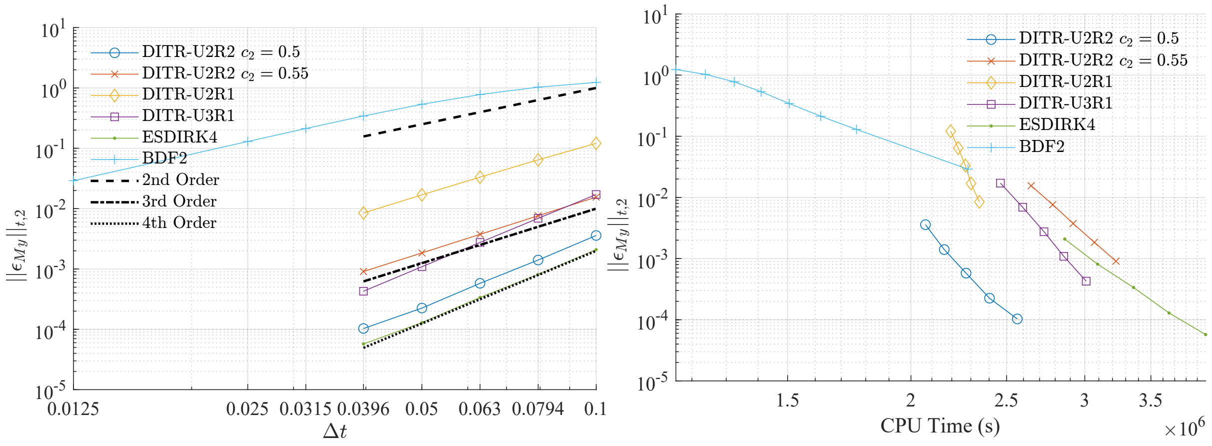
are the norms of time derivatives of y and x momentum:

$$M_y = \int_\Omega \left| \frac{\partial \rho u_y}{\partial t} \right| \mathrm{d}\Omega, \quad M_x = \int_\Omega \left| \frac{\partial \rho u_x}{\partial t} \right| \mathrm{d}\Omega \tag{51}$$

Meanwhile, $M_{y,ref}(t_i)$ and $M_{x,ref}(t_i)$ are those values obtained in the reference solution.

This manner of calculating transient error avoids the difficulty of preserving all the transient solutions at each time step of the very fine reference solution. As the reference solution uses significantly smaller time step, its temporal error is considered to be negligible. In other words, the reference solution is a good enough approximation of the exact solution of the semi-discretized FV equations Eq.(8). Compared with the reference solution, the solution using regular time steps induces major temporal discretization error, which can be illustrated with the errors constructed in Eq.(50).

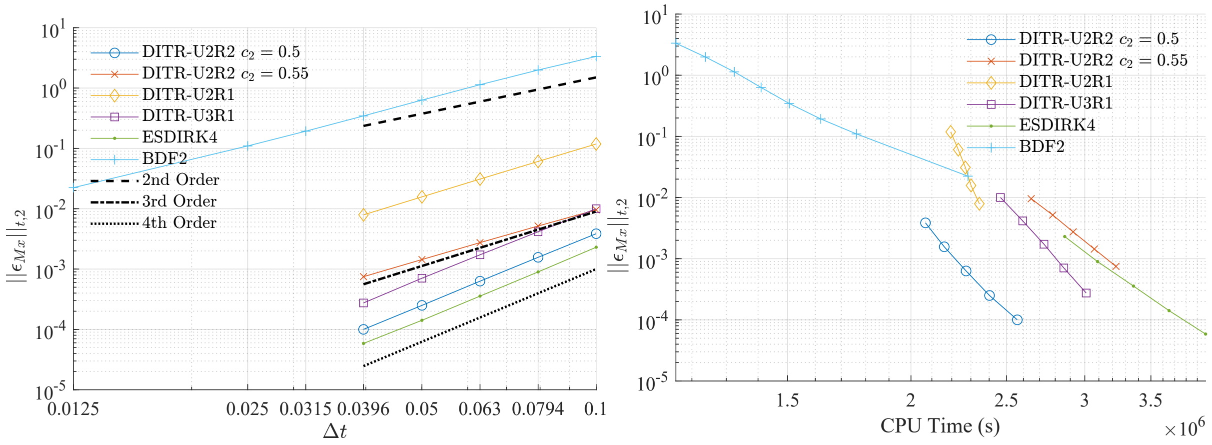
In order to mitigate the influence of the error induced by implicit dual time stepping in each step, each dual time stepping are terminated after the residual is smaller than 10^{-7} of the starting value.



(a) Error vs. time step size

(b) Error vs. CPU Time

Figure 4: Convergence and efficiency analysis with ϵ_{My} in $Re = 1200$ vortex shedding problem



(a) Error vs. time step size

(b) Error vs. CPU Time

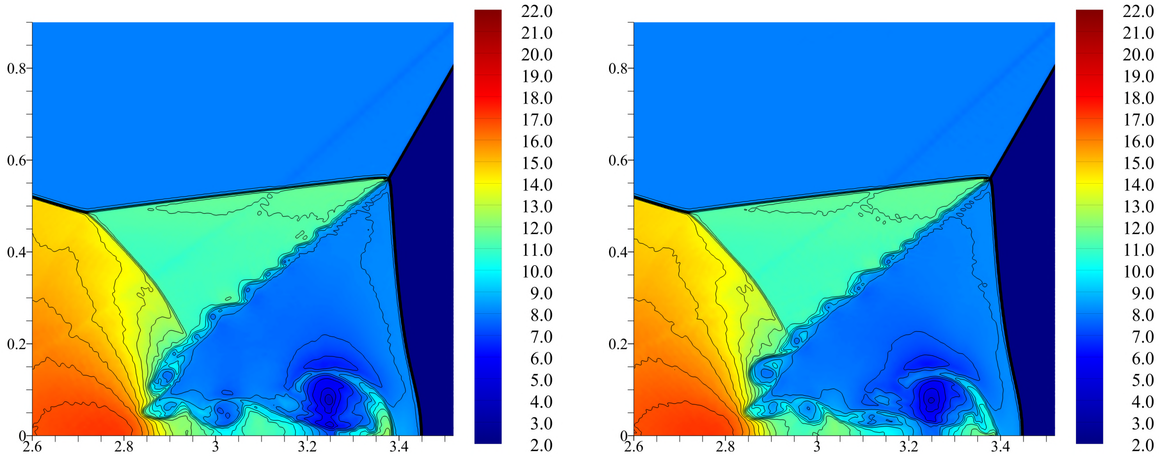
Figure 5: Convergence and efficiency analysis with ϵ_{Mx} in $Re = 1200$ vortex shedding problem

Results of the errors versus time step size and CPU time are illustrated in Figure 4 and 5. Figure 4a and 5a illustrates the results of convergence analysis, where with the refinement of time step, the temporal discretization error is reduced. The ESDIRK4, DITR U2R2 $c_2 = 0.5$, DITR U3R1 methods approximately 4th order convergence in Figure 4a and 5a, while U2R1 and U2R2 $c_2 = 0.55$ are 3rd order. BDF2 is indeed 2nd order accurate, and is only able to be comparable with high-order methods when Δt is very small. Among the DITR methods, U2R2 $c_2 = 0.5$ has the smallest error. U3R1 and U2R2 $c_2 = 0.55$ are close but U3R1 has higher order of convergence. Although better than BDF2, U2R1 has the worst error of vortex street simulation among DITR methods.

Figure 4b and 5b use CPU Time as the horizontal axis, therefore comparison of efficiency is illustrated. In Figure 4b and 5b, in order to achieve an error with magnitude of 10^{-3}, the high-order methods are significantly more economic than the 2nd order BDF2. Among the high order methods, DITR U2R2 $c_2 = 0.55$ is close to ESDIRK4 in efficiency, while U2R2 $c_2 = 0.5$ and U3R1 are more efficient than ESDIRK4. The most efficient U2R2 $c_2 = 0.5$ takes less than 70% of the time used in ESDIRK4 when the error is 10^-4.

4.3 Double Mach reflection

The double Mach reflection problem [44] is tested to compare the resolution capabilities of different time marching schemes. The double Mach reflection computes inviscid ideal gas in $[0, 4] \times [0, 1]$, initialized by a Ma 10 moving shock located at $x = 1/6 + \cot(60°)y$. The boundary of $y = 0, x \in [1/6, 4]$ is inviscid wall, and all other boundaries conforms with the Ma 10 moving shock. Details about the initial and boundary settings may be found in [44]. The compact FV scheme is additionally equipped with an accuracy preserving CWBAP limiter [45], which grants the ability to capture spatial discontinuities. To handle strong discontinuities, the local Lax-Friedrichs flux is used here. The computations are conducted on a uniform quadrilateral mesh with mesh size $h = 1/480$. Physical time step is set to a relatively large $\Delta t = 2 \times 10^{-4}$, and the solutions at $t = 0.25$ are compared.



(a) DITR U2R2 $c_2 = 0.5$

(b) DITR U2R2 $c_2 = 0.55$

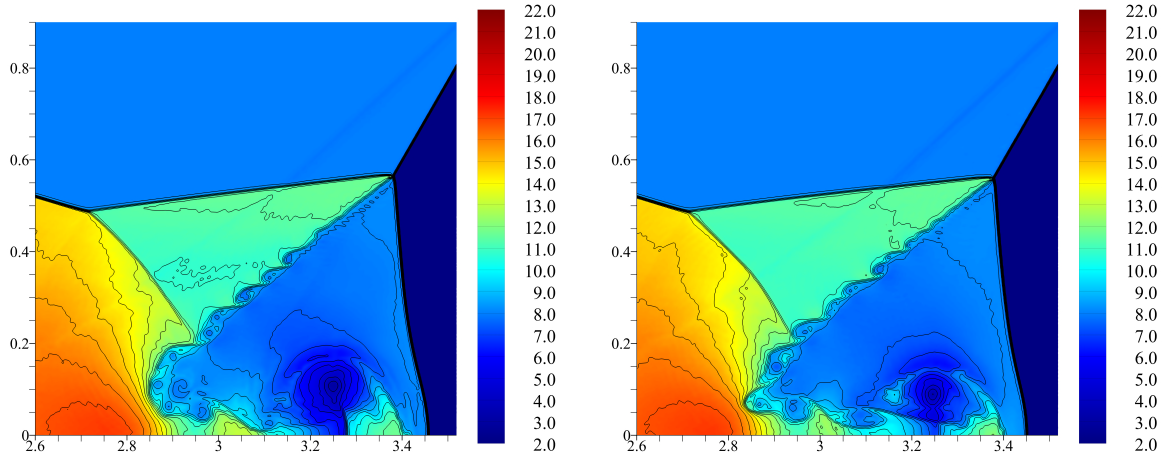
Figure 6: Density in double Mach reflection problem, DITR U2R2

Figure 6, 7 and 8 illustrate a zoomed view of density distribution. From Figure 8, it is observed BDF2 produces a deformed Mach stem, which, according to further tests, can be corrected by using smaller Δt. Using the same $\Delta t = 2 \times 10^{-4}$, all high-order methods produce a normal Mach stem. Meanwhile, the second order BDF2 completely smears the K-H instability in the shear layer, despite that the spatial discretization is 4th order accurate. All high-order time march methods methods can successfully simulate K-H instability and resolve the small structures induced. The results of DITR U2R2 $c_2 = 0.5$, U2R2 $c_2 = 0.55$ and U3R1 are very similar with ESDIRK4, while U2R1 appears to produce less vortices in the shear layer.

Like the results from isentropic vortex and vortex shedding simulation, using the same time step, DITR methods use less time than ESDIRK4 due to the reduction of stage numbers. The U2R2 $c_2 = 0.5$ method consumes around 63% of ESDIRK4's time, and U3R1 uses 79%.

5 Conclusion

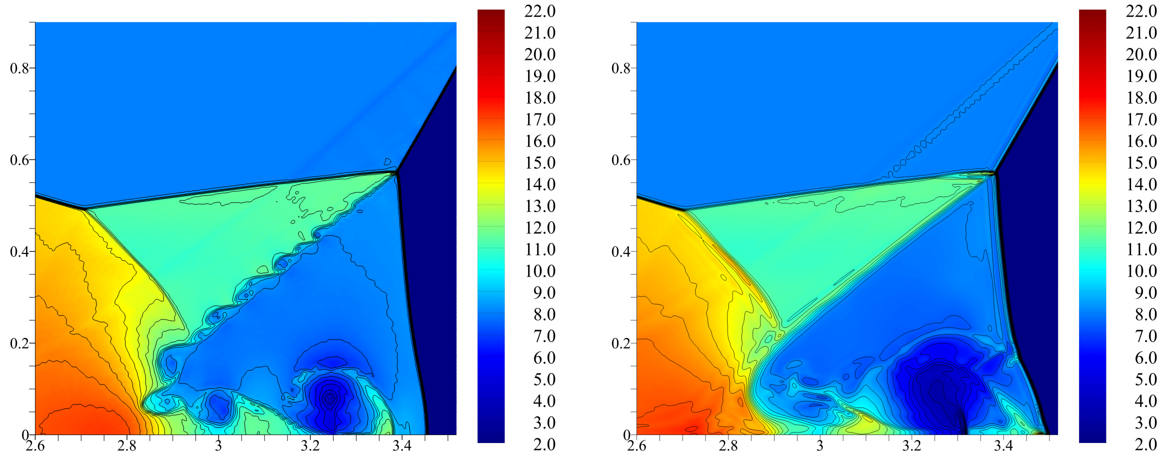
The current paper has described a method of developing time marching schemes using a direct integration and a temporal reconstruction. Using one form of quadrature rule and several forms of temporal reconstruction, a series of specific DITR methods are discovered and analyzed. The DITR U2R2 method



(a) DITR U2R1

(b) DITR U3R1

Figure 7: Density in double Mach reflection problem, DITR U2R1 and U3R1



(a) ESDIRK4

(b) BDF2

Figure 8: Density in double Mach reflection problem, baseline methods

is A-stable, and with $c_2 = 0.5$, U2R2 is 4th order accurate. The DITR U2R1 method is L-stable with 3rd order accuracy, and the DITR U3R1 method is L-stable with 4th order accuracy.

After analyzing the results of numerical tests, it is confirmed all the DITR methods can exhibit their theoretical order of accuracy. With the same time step, DITR methods are much more accurate than BDF2 and comparable with ESDIRK4. Due to having only 2 stages, DITR methods takes less time than ESDIRK4 each step. When reaching the same accuracy, some DITR methods are distinctively faster than ESDIRK4. In summary, the DITR methods are steady implicit time marching methods that are easy to implement and more efficient.

References

[1] William H Reed and Thomas R Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[2] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *Journal of Computational Physics*, 138(2):251–285, 1997.

[3] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[4] Bernardo Cockburn and Chi-Wang 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.

[5] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16:173–261, 2001.

[6] Z.J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.

[7] Yen Liu, Marcel Vinokur, and Z.J. Wang. Spectral difference method for unstructured grids i: Basic formulation. *Journal of Computational Physics*, 216(2):780–801, 2006.

[8] Michael Dumbser, Dinshaw S. Balsara, Eleuterio F. Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, 2008.

[9] Hung T Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. In *18th AIAA computational fluid dynamics conference*, page 4079, 2007.

[10] Hung T Huynh. A reconstruction approach to high-order schemnes including discontinuous galerkin for diffusion. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 403, 2009.

[11] Peter E Vincent, Patrice Castonguay, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47:50–72, 2011.

[12] Zhi Jian Wang and Haiyang Gao. A unifying lifting collocation penalty formulation including the discontinuous galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, 2009.

[13] Qian Wang, Yu-Xin Ren, and Wanai Li. Compact high order finite volume method on unstructured grids i: Basic formulations and one-dimensional schemes. *Journal of Computational Physics*, 314:863–882, 2016.

[14] Qian Wang, Yu-Xin Ren, and Wanai Li. Compact high order finite volume method on unstructured grids ii: Extension to two-dimensional euler equations. *Journal of Computational Physics*, 314:883–908, 2016.

[15] Qian Wang, Yu-Xin Ren, Jianhua Pan, and Wanai Li. Compact high order finite volume method on unstructured grids iii: Variational reconstruction. *Journal of Computational physics*, 337:1–26, 2017.

[16] Jianhua PAN, Qian WANG, Yusi Zhang, and Yuxin REN. High-order compact finite volume methods on unstructured grids with adaptive mesh refinement for solving inviscid and viscous flows. *Chinese Journal of Aeronautics*, 31(9):1829–1841, 2018.

[17] Yu-Si Zhang, Yu-Xin Ren, and Qian Wang. Compact high order finite volume method on unstructured grids iv: Explicit multi-step reconstruction schemes on compact stencil. *Journal of Computational Physics*, 396:161–192, 2019.

[18] Timothy Barth and Paul Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. In *28th aerospace sciences meeting*, page 13, 1990.

[19] Michel Delanaye and Yen Liu. Quadratic reconstruction finite volume schemes on 3d arbitrary

unstructured polyhedral grids. In *14th Computational Fluid Dynamics Conference*, page 3259, 1999.

[20] Carl F Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.

[21] Oliver Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of computational physics*, 144(1):194–212, 1998.

[22] Changqing Hu and Chi-Wang Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[23] Michael Dumbser, Martin Käser, Vladimir A Titarev, and Eleuterio F Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226(1):204–243, 2007.

[24] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

[25] Germund G Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, 1963.

[26] John Charles Butcher. *Numerical methods for ordinary differential equations.* John Wiley & Sons, 2016.

[27] Will Pazner and Per-Olof Persson. Stage-parallel fully implicit runge–kutta solvers for discontinuous galerkin fluid simulations. *Journal of Computational Physics*, 335:700–717, 2017.

[28] Antony Jameson. Evaluation of fully implicit runge kutta schemes for unsteady flow calculations. *Journal of Scientific Computing*, 73(2-3):819–852, 2017.

[29] Christopher A Kennedy and Mark H Carpenter. Additive runge–kutta schemes for convection–diffusion–reaction equations. *Applied numerical mathematics*, 44(1-2):139–181, 2003.

[30] Anne Kværnø. Singly diagonally implicit runge–kutta methods with an explicit first stage. *BIT Numerical Mathematics*, 44:489–502, 2004.

[31] Hester Bijl, Mark H Carpenter, Veer N Vatsa, and Christopher A Kennedy. Implicit time integration schemes for the unsteady compressible navier–stokes equations: laminar flow. *Journal of Computational Physics*, 179(1):313–329, 2002.

[32] Li Wang and Dimitri J Mavriplis. Implicit solution of the unsteady euler equations for high-order accurate discontinuous galerkin discretizations. *Journal of Computational Physics*, 225(2):1994–2015, 2007.

[33] Jeff R Cash. A class of implicit runge-kutta methods for the numerical integration of stiff ordinary differential equations. *Journal of the ACM (JACM)*, 22(4):504–511, 1975.

[34] JR Cash. On a class of implicit runge-kutta procedures. *IMA Journal of Applied Mathematics*, 19(4):455–470, 1977.

[35] JR Cash and A Singhal. Mono-implicit runge—kutta formulae for the numerical integration of stiff differential systems. *IMA Journal of Numerical Analysis*, 2(2):211–227, 1982.

[36] Gennady Yu Kulikov and Sergey K Shindin. On a family of cheap symmetric one-step methods of order four. In *International Conference on Computational Science*, pages 781–785. Springer, 2006.

[37] G Yu Kulikov and SK Shindin. Adaptive nested implicit runge–kutta formulas of gauss type. *Applied numerical mathematics*, 59(3-4):707–722, 2009.

[38] G Yu Kulikov, AI Merkulov, and SK Shindin. Asymptotic error estimate for general newton-type methods and its application to differential equations. 2007.

[39] Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*, volume 375. Springer Berlin Heidelberg New York, 1996.

[40] Qian-Min Huang, Yu-Xin Ren, Qian Wang, and Jian-Hua Pan. High-order compact finite volume schemes for solving the reynolds averaged navier-stokes equations on the unstructured mixed grids with a large aspect ratio. *Journal of Computational Physics*, 467:111458, 2022.

[41] Antony Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational fluid dynamics conference*, page 1596, 1991.

[42] Hong Luo, Joseph D Baum, and Rainald Löhner. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics*, 146(2):664–690, 1998.

[43] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[44] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of computational physics*, 54(1):115–173, 1984.

[45] Zhuohang Wu and Yu-xin Ren. The compact and accuracy preserving limiter for high-order finite volume schemes solving compressible flows. *Journal of Scientific Computing*, 96(3):77, 2023.