



Development of a Balanced Adaptive Time-Stepping Strategy Based on an Implicit JFNK-DG Compressible Flow Solver

Yu Pan¹ · Zhen-Guo Yan^{1,2} · Joaquim Peiró¹ · Spencer J. Sherwin¹

Received: 7 November 2020 / Revised: 8 March 2021 / Accepted: 14 April 2021 /

Published online: 26 August 2021

© The Author(s) 2021

Abstract

A balanced adaptive time-stepping strategy is implemented in an implicit discontinuous Galerkin solver to guarantee the temporal accuracy of unsteady simulations. A proper relation between the spatial, temporal and iterative errors generated within one time step is constructed. With an estimate of temporal and spatial error using an embedded Runge-Kutta scheme and a higher order spatial discretization, an adaptive time-stepping strategy is proposed based on the idea that the time step should be the maximum without obviously influencing the total error of the discretization. The designed adaptive time-stepping strategy is then tested in various types of problems including isentropic vortex convection, steady-state flow past a flat plate, Taylor-Green vortex and turbulent flow over a circular cylinder at $Re = 3900$. The results indicate that the adaptive time-stepping strategy can maintain that the discretization error is dominated by the spatial error and relatively high efficiency is obtained for unsteady and steady, well-resolved and under-resolved simulations.

Keywords Adaptive time-stepping · Unsteady simulations · High order · Discontinuous Galerkin · Implicit time integration

Mathematics Subject Classification 76N06 · 65M60 · 65L06

✉ Zhen-Guo Yan
zgyan@skla.cardc.cn; yanzhg@mail.ustc.edu.cn

Yu Pan
yu.pan16@imperial.ac.uk; yu.pan0729@qq.com

¹ Department of Aeronautics, Imperial College London, London, UK

² State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang 621000, Sichuan, China

1 Introduction

Implicit time integration schemes have the potential to increase the efficiency of high-Reynolds number simulations by relaxing the challenging time step restriction, and have been successfully adopted in a variety of steady-state and unsteady simulations [2, 21, 22, 27, 29]. However, implicit time integration schemes are generally more complex and introduce many parameters, which usually have large influences on the performance of the solver. Take the widely used implicit time integration schemes based on Runge-Kutta (RK) temporal discretization scheme and Jacobian-free Newton Krylov (JFNK) method as an example, there are parameters such as the order of RK scheme, the time step size, the Newton iteration convergence tolerance, the linear iteration convergence tolerance and possibly other parameters introduced by preconditioners [15, 16, 29]. These parameters are usually highly problem dependent and have large influences on the accuracy, efficiency and robustness in specific simulations, the choices of which actually become a complex multi-objective optimization problem. Therefore, a reliable approach for determining these parameters is essential especially for computational fluid dynamics (CFD) software with a wide range of application areas and for the automatic simulations of massive cases in a design pipeline.

In steady-state simulations, there were many studies on this topic, most of which mainly focus on accelerating the convergence to steady state. In reference [28], an expert system for choosing an efficient CFL was developed, in which the convergence history is manually separated into three different stages and different adaptive strategy is used in each stage. In reference [18], a solution-limiting method is developed to determine the CFL. Bücken et al. compared some of the existing adaptive methods in 2009 and concluded that there is no clear winner [6]. In 2019, an error based adaptive CFL method was developed to accelerate the late stage of convergence but still need to manually divide the convergence history into different stages [12]. Compared with the studies on efficiency, much fewer studies have focused on robustness. Lian et al. [18] addressed this problem with the solution-limiting method and claimed that the robustness is largely improved. Other attempts on this topic are mainly some techniques such as rolling back and recomputing with a smaller time step if the solution is not satisfactory [30]. These studies have highlighted the importance of these parameter choices on different aspects of the solver and on different research areas. Although, significant progress has been made, as is concluded in reference [6] optimal CFL evolution is still an open problem.

For unsteady simulations, there are more parameters and there is an extra trouble to worry about, the temporal accuracy. The idea of using adaptive time-stepping in unsteady simulations is old, which can date back to time adaptation studies of ordinary differential equations (ODEs) [1, 24, 25]. Various methods for choosing the step size (or step size controllers) were proposed and compared in the history as well as other topics such as the requirement of the temporal error estimator and the stability properties in stiff problems, a comprehensive review of which can be found in reference [24, 25]. Birken adopted a similar idea as those in the ODE systems and developed a time step adaptation method in the simulation of compressible Navier-Stokes equations using an embedded scheme to estimate the local temporal error. This method is adopted in the comparison of Rosenbrock methods and explicit first stage singly diagonally implicit Runge-Kutta (ESDIRK) time integration methods [5]. Noventa et al.

developed a similar time step adaptation strategy in unsteady incompressible turbulent flow simulations [21], which is further tested in compressible simulations focusing on the comparison of different step size controllers [22]. The adaptive method based on local temporal error was further developed for goal oriented time step adaptation and compared with a method based on global error estimate [19]. However, most of them such as, the studies in references [4, 5, 21, 22], consider the temporal discretization methods separately as an ODE solver using the method of lines but do not take into account specific physical properties of the problem and specific spatial discretization adopted. These methods rely on a user defined temporal error tolerance, the choice of which is highly problem dependent and needs a deep understanding of the simulation settings. However, a naive choice of temporal error tolerance may lead to a large increase of CPU time without obviously improving the results as has been demonstrated in reference [21].

In this work, an adaptive time-stepping method is proposed to maintain temporal accuracy and high efficiency. The basic idea is to determine the time step based on the relations of different discretization errors. The time step is determined as the largest time step such that temporal errors are smaller than the spatial discretization errors. In this sense, the temporal discretization is believed to be sufficiently accurate since further decreasing the time step will not improve the simulation obviously. As will be shown in numerical tests in Sect. 3, this choice is the relatively efficient choice because it is around the maximum choice to be sufficiently accurate in time in our definition.

The governing equations, numerical methods and numerical error relations are presented in Sect. 2. Moreover, the adaptive strategy proposed in this paper is also presented in Sect. 2. Results of various test cases are presented in Sect. 3 to demonstrate the properties of the adaptive methods in different problems. Finally, conclusions are drawn in Sect. 4.

2 Governing Equations and Numerical Methods

The governing compressible Navier-Stokes equations, representing conservation of mass, momentum and energy, can be written in abridged form as

$$\frac{\partial \mathbf{U}}{\partial t} = L(\mathbf{U}) = - \sum_{i=1}^d \frac{\partial \mathbf{H}_i}{\partial \mathbf{x}_i}; \quad \mathbf{H}_i = \mathbf{F}_i(\mathbf{U}) - \mathbf{G}_i(\mathbf{U}, \nabla \mathbf{U}), \quad (1)$$

where L is the analytical nonlinear spatial operator, d is the number of dimensions of the problem, and $\mathbf{U} = [\rho, \rho u_1, \dots, \rho u_d, E]^T$ is the vector of conservative variables. The inviscid flux, $\mathbf{F} = \mathbf{F}(\mathbf{U})$, and the viscous flux, $\mathbf{G} = \mathbf{G}(\mathbf{U}, \nabla \mathbf{U})$, in the i th direction are given by

$$\mathbf{F}_i = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + p \delta_{1,i} \\ \vdots \\ \rho u_d u_i + p \delta_{d,i} \\ u_i(E + p) \end{bmatrix}, \quad \mathbf{G}_i = \begin{bmatrix} 0 \\ \tau_{i1} \\ \vdots \\ \tau_{id} \\ \sum_{j=1}^d u_j \tau_{ij} - q_i \end{bmatrix}. \tag{2}$$

Here, $E = \rho(C_v T + u_i u_i / 2)$ is the total energy in which C_v is the specific heat capability with constant volume, $p = (\gamma - 1)(E - \rho u_i u_i / 2)$ is the pressure, γ is the specific heat ratio, $\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_i}{\partial x_i} \delta_{ij})$ is the viscous tensor, $q_i = -\kappa \frac{\partial T}{\partial x_i}$ is the heat flux, μ is the dynamic viscosity, and κ is the thermal conductivity.

In what follows we briefly outline the formulation of a compressible flow solver based on a discontinuous Galerkin (DG) spatial discretization [13], an implicit time integration via ESDIRK temporal discretization schemes [17] and a JFNK method [16]. Further details of the solver can be found in reference [29].

2.1 Discretizations and Error Estimate

2.1.1 Spatial Discretization

In DG methods, the computational domain is divided into non-overlapping elements and a polynomial space is defined on each element. Performing integration by parts after multiplying the test function ϕ_p to Eq. (1), we get the weak form of Eq. (1) in element e :

$$\int_{\Omega_e} \frac{\partial \mathbf{U}}{\partial t} \phi_p \, d\Omega_e = \int_{\Omega_e} \nabla \phi_p \cdot \mathbf{H} \, d\Omega_e - \int_{\Gamma_e} \phi_p \mathbf{H}^n \, d\Gamma_e, \tag{3}$$

where Γ_e is the element boundary, and \mathbf{H}^n is the boundary flux on the elemental outward normal direction (\mathbf{n}). Furthermore, the vector of conserved variables \mathbf{U} is expressed in the polynomial space as

$$\mathbf{U}_\delta(\mathbf{x}, t) \approx \sum_{p=0}^{N_p} \phi_p(\mathbf{x}) \mathbf{u}_p(t) \tag{4}$$

on each computational element. Here, \mathbf{u}_p is the coefficient of the p th basis function and $N_p + 1$ is the number of basis functions. To discretize Eq. (3), a quadrature rule is adopted to approximate the integration. The \mathbf{U} on these quadrature points can be calculated by

$$\mathbf{U}_{\delta,i} = \sum_{p=0}^{N_p} \phi_p(\mathbf{x}_i) \mathbf{u}_p(t) = \sum_{p=0}^{N_p} \mathbf{B}_{i,p} \mathbf{u}_p(t), \tag{5}$$

where the subscript i indicates values on the i th quadrature point, and \mathbf{B} is the backward transform matrix. Using the Roe’s Riemann flux [26] and the symmetric interior penalty DG (SIPG) method [10] to approximate the advective numerical flux and the viscous numerical fluxes, respectively, the semi-discrete equation can be expressed as

$$\begin{aligned}
 \frac{d\mathbf{u}}{dt} &= \mathcal{L}_\delta(\mathbf{u}) \\
 &= \mathbf{M}^{-1} \sum_{j=1}^d \mathbf{B}^T \mathbf{D}_j^T \Lambda(wJ) \mathbf{H}_j \\
 &\quad - \mathbf{M}^{-1} (\mathbf{B}^\Gamma \mathbf{M}_c)^\Gamma \Lambda(w^\Gamma J^\Gamma) \hat{\mathbf{H}}^n \\
 &\quad - \mathbf{M}^{-1} \sum_{j=1}^d \mathbf{B}^T \mathbf{D}_j^T \mathbf{J}^T \Lambda(w^\Gamma J^\Gamma) \hat{\mathbf{S}}_j^n,
 \end{aligned} \tag{6}$$

where w is the quadrature weight, J is the grid metric Jacobian, Λ represents a diagonal matrix, $\mathbf{M} = \mathbf{B}^T \Lambda(wJ) \mathbf{B}$ is the mass matrix, \mathbf{D}_j is the derivative matrix in the j th direction, the superscript Γ represents the corresponding matrices or variables on element boundaries, $\hat{\mathbf{S}}_j^n$ is a symmetric flux from the SIPG method, \mathbf{M}_c is the mapping matrix between ϕ^Γ and ϕ , and \mathbf{J} is the interpolation matrix from quadrature points of an element to quadrature points of its element boundaries. Details of the spatial discretization can be found in reference [29].

The corresponding semi-discrete equation for the vector \mathbf{U}_δ can be obtained using relation of Eq. (5) and

$$L_\delta(\mathbf{U}(\mathbf{u})) = \mathbf{B} \mathcal{L}_\delta(\mathbf{u}), \tag{7}$$

which is written as

$$\frac{\partial \mathbf{U}_\delta}{\partial t} = L_\delta(\mathbf{U}_\delta). \tag{8}$$

The spatial truncation error can be expressed as

$$\mathbf{E}_s(\mathbf{U}_\delta) = L_\delta(\mathbf{U}_\delta) - L(\mathbf{U}) = C_s D_s(\mathbf{U}) h^{P+1} + O(h^{P+2}), \tag{9}$$

where C_s is a coefficient related to the discretization scheme but independent of \mathbf{U} , $D_s(\mathbf{U})$ is a spatial derivative term dependent on \mathbf{U} and P is the polynomial order of the basis functions in Eq. (5).

2.1.2 Temporal Discretization

An embedded version of the ESDIRK method [17] is adopted to discretize Eq. (3) in time. The i th stages of the process of ESDIRK are

$$\mathbf{U}^{(0)} = \mathbf{U}^n, \tag{10}$$

$$\mathbf{S}^{(i)} = \mathbf{U}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} L(\mathbf{U}^{(j)}), \quad i = 1, 2, \dots, S, \tag{11}$$

$$\mathbf{U}^{(i)} = \mathbf{S}^{(i)} + \Delta t a_{ii} L(\mathbf{U}^{(i)}), \quad i = 1, 2, \dots, S, \tag{12}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \sum_{i=1}^S b_i L(\mathbf{U}^{(i)}), \tag{13}$$

$$\hat{\mathbf{U}}^{n+1} = \mathbf{U}^n + \Delta t \sum_{i=1}^{S+1} \hat{b}_i L(\mathbf{U}^{(i)}), \tag{14}$$

where a_{ij} , b_i and \hat{b}_i are coefficients of the Butcher tableaus on the i th RK stage which can be found in Appendix A and in reference [17]. \mathbf{U}^{n+1} is the approximation at time step $n + 1$. $\hat{\mathbf{U}}^{n+1}$ is the embedded solution for temporal error estimate at the cost of one extra implicit stage. For the embedded ESDIRK schemes adopted, $\hat{\mathbf{U}}^{n+1}$ is one order higher than \mathbf{U}^{n+1} . Thus, the leading term of the temporal error of \mathbf{U}^{n+1} can be accurately estimated as

$$\begin{aligned} \mathbf{E}_t^{n+1} &= \mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1} \\ &= \Delta t \sum_{i=1}^{S+1} (b_i - \hat{b}_i) L(\mathbf{U}^{(i)}) \\ &= C_t D_t(\mathbf{U}) \Delta t^{N+1} + O(\Delta t^{N+2}), \end{aligned} \tag{15}$$

where b_{S+1} is always zero in the ESDIRK schemes adopted, C_t is a coefficient related to the ESDIRK scheme adopted but independent of \mathbf{U} , $D_t(\mathbf{U})$ is a problem dependent temporal derivative term, and N is the order of accuracy of the ESDIRK scheme. In the derivation of Eq. (15), the analytical equations, Eq. (1), have been adopted to replace the spatial operator with the temporal derivatives, thus spatial error is not included. The estimation also excludes the temporal errors from previous time steps and will be referred to as the local temporal error in this paper.

2.1.3 Discrete Formulation

The discrete formulation of the governing equations is obtained by replacing \mathbf{U} and L in Eqs. (10)–(14) with their discrete counterparts (\mathbf{U}_δ and L_δ). To reduce the size of the non-linear system for implicit stages ($a_{ii} \neq 0$), we choose to rewrite the discrete counterparts of Eqs. (11)–(12) as

$$\mathbf{s}^{(i)} = \mathbf{u}_\delta^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathcal{L}_\delta(\mathbf{u}_\delta^{(j)}), i = 1, 2, \dots, S, \tag{16}$$

$$\mathbf{u}_\delta^{(i)} = \mathbf{s}^{(i)} + \Delta t a_{ii} \mathcal{L}_\delta(\mathbf{u}_\delta^{(i)}), i = 1, 2, \dots, S, \tag{17}$$

$$\mathbf{U}_\delta^{(i)} = \mathbf{B} \mathbf{u}_\delta^{(i)}, \tag{18}$$

since the length of \mathbf{u}_δ is usually smaller than that of \mathbf{U}_δ . In rewriting the above equations, the relations of Eqs. (5) and (7) are used.

2.1.4 JFNK Method

Equation (17) of the implicit stages ($a_{ii} \neq 0$) can be written as the nonlinear system

$$\mathbf{N}(\mathbf{u}_\delta^{(i)}) = \mathbf{u}_\delta^{(i)} - \mathbf{s}^{(i)} - \Delta t a_{ii} \mathcal{L}_\delta(\mathbf{u}_\delta^{(i)}) = \mathbf{0}. \tag{19}$$

This system is iteratively solved by Newton’s method with an initial guess $\mathbf{v}^0 = \mathbf{s}^{(i)}$. The Newton step, $\Delta \mathbf{v} = \mathbf{v}^{k+1} - \mathbf{v}^k$, is updated through the solution of the linear system

$$\frac{\partial \mathbf{N}(\mathbf{v}^k)}{\partial \mathbf{v}} \Delta \mathbf{v} = -\mathbf{N}(\mathbf{v}^k). \tag{20}$$

A preconditioned GMRES iterative solver is adopted here to solve the unsymmetric system of Eq. (20). A Jacobian-free method is further adopted in the GMRES to approximate the Jacobian matrix and a vector inner product operator using a finite difference approximation, that is

$$\frac{\partial \mathbf{N}(\mathbf{v}^k)}{\partial \mathbf{v}} \mathbf{q}_i = \frac{\mathbf{N}(\mathbf{v}^k + \xi \mathbf{q}_i) - \mathbf{N}(\mathbf{v}^k)}{\xi}, \tag{21}$$

where \mathbf{q}_i is a set of orthogonal vectors in the Krylov space, the expression of ξ can be found in references [16, 29]. A low memory block relaxed Jacobi iterative preconditioner is used in the GMRES solver to speedup the simulation [29]. The GMRES tolerance of $0.05 \|\mathbf{N}(\mathbf{v}^k)\|$ adopted here has no influence on the temporal accuracy but affects its efficiency. Other choices of the GMRES tolerance can be found in references [8, 14].

When the Newton residual is smaller than the convergence tolerance of the Newton iterations τ , which is

$$\|\mathbf{N}(\mathbf{v}^k)\| = \|\mathbf{R}(\mathbf{v}^k)\| \leq \tau, \tag{22}$$

$\mathbf{u}_i^{(i)} = \mathbf{v}^k$ is regarded as the approximate solution of the nonlinear system, where \mathbf{R} is the remaining Newton residual vector $\mathbf{N}(\mathbf{u}_i^{(i)})$ after convergence. A proper choice of τ is important because it determines the magnitude of the iterative error introduced by the iterative solver, which may degrade the overall error and the convergence order of accuracy [22]. Assume $\mathbf{u}_\delta^{(i)}$ is the exact solution of the nonlinear system and $\mathbf{u}_i^{(i)}$ is the iterative approximation, we can get $\mathbf{N}(\mathbf{u}_\delta^{(i)}) = \mathbf{0}$ and $\mathbf{N}(\mathbf{u}_i^{(i)}) = \mathbf{R}^{(i)}$. Thus,

$$\mathbf{N}(\mathbf{u}_i^{(i)}) - \mathbf{N}(\mathbf{u}_\delta^{(i)}) = (\mathbf{u}_i^{(i)} - \mathbf{u}_\delta^{(i)}) - \Delta t a_{ii} \left(\mathcal{L}_\delta(\mathbf{u}_i^{(i)}) - \mathcal{L}_\delta(\mathbf{u}_\delta^{(i)}) \right) = \mathbf{R}^{(i)}. \tag{23}$$

The contribution of the iterative error to the total discretization error will be discussed in the following.

2.1.5 Error Estimate

In this section, the local discretization errors introduced at time step $n + 1$ are estimated, which mainly consist of the spatial error, temporal error and iterative error. The accumulation of these errors, which we refer to as global errors, is more difficult to estimate and will not be considered in this paper. Studies considering the global errors can be found in reference [19].

We recall that the numerical solution at the time step $n + 1$ is calculated by

$$\mathbf{U}_{it}^{n+1} = \mathbf{U}^n + \Delta t \sum_{i=1}^S b_i L_\delta(\mathbf{U}_{it}^{(i)}), \tag{24}$$

and the exact solution at $n + 1$, $\mathbf{U}(t^{n+1})$, based on \mathbf{U}^n can be expressed as

$$\begin{aligned} \mathbf{U}(t^{n+1}) &= \hat{\mathbf{U}}^{n+1} + O(\Delta t^{N+2}) \\ &= \mathbf{U}^n + \Delta t \sum_{i=1}^{S+1} \hat{b}_i L(\mathbf{U}^{(i)}) + O(\Delta t^{N+2}), \end{aligned} \tag{25}$$

where $O(\Delta t^{N+2})$ represents all the high-order truncation terms and is not detailed here. Subtracting Eq. (24) by Eq. (25) and ignoring the high-order truncation terms of $O(\Delta t^{N+2})$, the leading terms of the local discretization error can be expressed as

$$\mathbf{U}_{it}^{n+1} - \mathbf{U}(t^{n+1}) \approx \Delta t \sum_{i=1}^S b_i L_\delta(\mathbf{U}_{it}^{(i)}) - \Delta t \sum_{i=1}^{S+1} \hat{b}_i L(\mathbf{U}^{(i)}). \tag{26}$$

If we now add and subtract the summation terms of $\Delta t \sum_{i=1}^{S+1} \hat{b}_i L(\mathbf{U}^{(i)})$ and $\Delta t \sum_{i=1}^S b_i L_\delta(\mathbf{U}_\delta^{(i)})$ to the right-hand side of Eq. (26), we obtain

$$\begin{aligned} \mathbf{U}_{it}^{n+1} - \mathbf{U}(t^{n+1}) &\approx \Delta t \sum_{i=1}^{S+1} (b_i - \hat{b}_i) L(\mathbf{U}^{(i)}) + \Delta t \sum_{i=1}^S b_i (L_\delta(\mathbf{U}_\delta^{(i)}) - L(\mathbf{U}^{(i)})) \\ &\quad + \Delta t \sum_{i=1}^S b_i (L_\delta(\mathbf{U}_{it}^{(i)}) - L_\delta(\mathbf{U}_\delta^{(i)})), \end{aligned} \tag{27}$$

where b_{S+1} is always zero in the ESDIRK schemes adopted. The $L_\delta(\mathbf{U}_{it}^{(i)}) - L_\delta(\mathbf{U}_\delta^{(i)})$ in the last term of Eq. (27) is related to Eq. (23) and can be considered as the iterative error $\mathbf{E}_{it}^{(i)}$. Using the relations of Eqs. (9) and (15) and ignoring high-order terms of $O(h^{P+2})$, we can now rewrite each term as

$$\begin{aligned} \mathbf{U}_{it}^{n+1} - \mathbf{U}(t^{n+1}) &\approx C_t D_t(\mathbf{U})(\Delta t)^{N+1} + \Delta t C_s \bar{D}_s(\mathbf{U})h^{P+1} + \Delta t \bar{\mathbf{E}}_{it} \\ &= \mathbf{E}_t^{n+1} + \Delta t \bar{\mathbf{E}}_s + \Delta t \bar{\mathbf{E}}_{it} \\ &= \mathbf{B}(\mathbf{e}_t^{n+1} + \Delta t \bar{\mathbf{e}}_s + \Delta t \bar{\mathbf{e}}_{it}), \end{aligned} \tag{28}$$

where we have used the overbar to denote summations of the form $\bar{\psi} = \sum_{i=1}^S b_i \psi^{(i)}$, which is a weighted average of ψ because $\sum_{i=1}^S b_i = 1$.

Equation (28) suggests that the error generated in one time step is the sum of the local temporal error, averaged spatial error and averaged iterative error. This indicates that to decrease the total error one should decrease these three errors simultaneously and it will not help much if one of them is decreased much smaller than the largest of the three terms.

Based on this observation, an adaptive time-stepping strategy and an adaptive Newton tolerance are developed as follows in Sects. 2.2 and 2.3, respectively. The basic idea is to choose a time step and a Newton tolerance such that the temporal and iterative errors are smaller than the spatial error. In Sect. 2.2, the iterative error is assumed to be negligible compared with the spatial and temporal errors, which can be achieved as long as Eq. (19) converges to a sufficiently small level. The method to control the iterative error will be discussed in Sect. 2.3.

2.2 Adaptive Time-Stepping Strategy

Based on the error estimate, a time step adaptation method is proposed with the basic idea that the time step should be the maximum value without obviously influencing the total discretization error. With the spatial error fixed in a specific simulation, this implies that

$$\|\mathbf{E}_t^{n+1}\| = \beta \Delta t \|\bar{\mathbf{E}}_s\| \quad (29)$$

with $\beta < 1.0$ and a proper defined norm. When $\|\mathbf{E}_t^{n+1}\|$ is orders of magnitude smaller than $\Delta t \|\bar{\mathbf{E}}_s\|$, the total error will be dominated by the spatial error and further decreasing the time step (thus the temporal error) will not improve the solution much.

To maintain Eq. (29) is satisfied, spatial and temporal errors need to be estimated. The embedded ESDIRK scheme offers an accurate estimate of temporal error \mathbf{E}_t^{n+1} using Eq. (15). As for the spatial truncation error $\bar{\mathbf{E}}_s$, it is estimated using a one order higher approximation at the end of each time step, that is

$$\bar{\mathbf{E}}_s \approx L_\delta^{P+1}(\mathbf{U}_{it}^{n+1}) - L_\delta^P(\mathbf{U}_{it}^{n+1}) = C_s D_s(\mathbf{U}) h^{P+1} + O(h^{P+2}), \quad (30)$$

the leading term of which is the same as that in Eq. (9). Equation (29) implies that the desired time step ($\Delta \hat{t}$) should satisfy that

$$\|\mathbf{E}_t^{n+1}(\Delta \hat{t})\| \approx \|C_t D_t(\mathbf{U}) \Delta \hat{t}^{N+1}\| = \beta \Delta \hat{t} \|\bar{\mathbf{E}}_s\|. \quad (31)$$

Dividing Eq. (31) by Eq. (15), time step $\Delta \hat{t}$ can be calculated by

$$\Delta \hat{t} = \Delta t^n \left(\frac{\beta \Delta t \|\bar{\mathbf{E}}_s\|}{\|\mathbf{E}_t\|} \right)^{1/N}, \quad (32)$$

which can be used as the time step for the next time step. The simplest elementary controller [25] is adopted here in determining the time step and other choices are also possible [25]. In the derivation of Eqs. (30), (31) and (32), the change of $D_t(\mathbf{u})$ is assumed to be small throughout the step.

In the designs of adaptive time-stepping methods in references [5, 22], an L_2 norm of the \mathbf{E}_t^{n+1} for all the variables throughout the flow field is adopted as the norm in their adaptive strategy. However, the norm of the error with this choice may be dominated by a specific variable or the large scale unsteady structures. As a result, the adapted time step may be determined by one of the variables and/or the dominating unsteady structure in the flow field, which could be too large for the small unsteady structures of interest.

To avoid these problems, the norm is defined in each element. The L_2 norms of \mathbf{E}_t^{n+1} and $\bar{\mathbf{E}}_s$ are calculated in element e for variable m and the elemental time step is calculated using

$$\Delta t_{e,m}^{n+1} = \Delta t^n \left(\frac{\beta \Delta t \|\bar{\mathbf{E}}_s\|_{e,m} + 1.5^N \epsilon_m}{\|\mathbf{E}_t\|_{e,m} + \epsilon_m} \right)^{1/N}. \quad (33)$$

Here, the ϵ_m is added to the denominator to avoid division by zero. But it also serves as a threshold of the error magnitude of interest. If the local temporal error magnitudes are much smaller than ϵ_m , the local unsteady structures are either too weak to be of interest, or are already accurately captured, which indicates the current time step choice is already

excessively accurate in that element. Thus, a $1.5^N \epsilon_m$ is added to the numerator to maintain the elemental and the time step will at least grow by 1.5 times. The choice of 1.5, which is not optimized, ensures that the time step will increase but not too fast.

The value of ϵ_m adopted here is

$$\epsilon_m = 10^{-12} \mathbf{U}_{it,m}^{\text{rms}}, \tag{34}$$

where $\mathbf{U}_{it,m}^{\text{rms}}$ denotes the root mean square value over the whole flow field of the mass, momentum magnitude and energy variables. Such a threshold value is based on the rounding error of the m th variable. One purpose of introducing this term is to avoid noisy elemental time steps caused by rounding errors in uniform flow areas.

With elemental time steps $\Delta t_{e,m}^{n+1}$, the time step for variable m can be calculated using different strategies. For example, a minimum of $\Delta t_{e,m}^{n+1}$ would maintain that Eq. (29) is satisfied in every element. Here we choose to calculate it by a weighted average of $\Delta t_{e,m}^{n+1}$ as

$$\Delta t_m^{n+1} = \frac{\sum_{e=1}^{N_e} \|\mathbf{E}_t\|_{e,m}^r \Delta t_{e,m}^{n+1}}{\sum_{e=1}^{N_e} \|\mathbf{E}_t\|_{e,m}^r}. \tag{35}$$

The temporal error $\|\mathbf{E}_t\|_{e,m}^r$ is used as the weights such that elements with larger temporal errors have larger contributions to the global time step. The $r \in [0, \infty)$ can be chosen based on the smallest temporal structure of interest in the simulations, a larger value of which would lead to a larger contribution from elements with large temporal errors. In this paper, $r = 1$ is used. This weighted averaging helps maintain a relatively smooth change of the time step. At last, the time step Δt^{n+1} is calculated by

$$\Delta t^{n+1} = \min(\Delta t_m^{n+1}). \tag{36}$$

2.3 Adaptive Newton Tolerance

A similar idea as the time step adaptation can be adopted here in choosing the Newton tolerance so that the iterative error is smaller than temporal error to avoid affecting the temporal accuracy. In references [5, 21], a Newton tolerance τ in Eq. (22) is proposed by directly relating the Newton tolerance to the temporal error, that has the form

$$\tau = \eta \|\mathbf{e}_t\| \tag{37}$$

with $\eta = 0.1$. However, the relation between the Newton residual vector, \mathbf{R} , and the iteration error, $\Delta t \bar{\mathbf{e}}_{it}$, introduced into Eq. (28) is not trivial. In the following this relation is estimated to illustrate the assumptions made in choosing the Newton tolerance of Eq. (37).

Considering that

$$\Delta t \|\bar{\mathbf{e}}_{it}\| = \Delta t \left\| \sum_{i=1}^S b_i \mathbf{e}_{it}^{(i)} \right\| \leq \Delta t \sum_{i=1}^S b_i \|\mathbf{e}_{it}^{(i)}\|, \tag{38}$$

and $\sum_{i=1}^S b_i = 1$, a sufficient condition for

$$\Delta t \|\bar{\mathbf{e}}_{it}\| \leq \eta \|\mathbf{e}_t\| \tag{39}$$

is that

$$\Delta t \|\mathbf{e}_{it}^{(i)}\| \leq \eta \|\mathbf{e}_t\|, \tag{40}$$

which maintains that the iterative error is smaller than the temporal error in Eq. (28) with $\eta < 1.0$.

The relation between $\Delta t \mathbf{e}_{it}^{(i)}$ and $\mathbf{R}^{(i)}$ is estimated in the following. The $\mathbf{R}^{(i)}$ can be expressed as

$$\mathbf{R}^{(i)} = \mathbf{N}(\mathbf{u}_{it}^{(i)}) - \mathbf{N}(\mathbf{u}_\delta^{(i)}) = \mathbf{N}'(\mathbf{u}_{it}^{(i)}) (\mathbf{u}_{it}^{(i)} - \mathbf{u}_\delta^{(i)}) + O(\Delta \mathbf{u}^2), \tag{41}$$

where $\Delta \mathbf{u} = \mathbf{u}_{it}^{(i)} - \mathbf{u}_\delta^{(i)}$. Therefore, the iterative error of the solution $\mathbf{u}_{it}^{(i)}$ can be approximated as

$$\begin{aligned} \mathbf{u}_{it}^{(i)} - \mathbf{u}_\delta^{(i)} &= (\mathbf{N}')^{-1} (\mathbf{R}^{(i)} - O(\Delta \mathbf{u}^2)) \\ &= (\mathbf{I} - \Delta t a_{ii} \mathcal{L}'_\delta (\mathbf{u}_{it}^{(i)}))^{-1} (\mathbf{R}^{(i)} - O(\Delta \mathbf{u}^2)). \end{aligned} \tag{42}$$

Subtracting Eq. (42) by Eq. (23) and ignoring high-order terms of $O(\Delta \mathbf{u}^2)$, we can further deduce the relation between $\Delta t \mathbf{e}_{it}^{(i)}$ and \mathbf{R} as

$$\begin{aligned} \Delta t \mathbf{e}_{it}^{(i)} &= \Delta t (\mathcal{L}_\delta (\mathbf{u}_{it}^{(i)}) - \mathcal{L}_\delta (\mathbf{u}_\delta^{(i)})) \\ &= \Delta t \mathcal{L}'_\delta (\mathbf{u}_{it}^{(i)}) (\mathbf{N}')^{-1} \mathbf{R}^{(i)}. \end{aligned} \tag{43}$$

Substituting Eq. (43) into Eq. (40), we can get

$$\|\mathcal{L}'_\delta (\mathbf{I}/\Delta t - a_{ii} \mathcal{L}'_\delta)^{-1} \mathbf{R}^{(i)}\| \leq \eta \|\mathbf{e}_t\|. \tag{44}$$

The presence of $\mathcal{L}'_\delta (\mathbf{I}/\Delta t - a_{ii} \mathcal{L}'_\delta)^{-1}$, which is difficult or expensive to estimate accurately, makes the relation between $\Delta t \mathbf{e}_{it}^{(i)}$ and \mathbf{R} problem dependent and difficult to evaluate. As Δt becomes larger $\mathcal{L}'_\delta (\mathbf{I}/\Delta t - a_{ii} \mathcal{L}'_\delta)^{-1}$ becomes closer and closer to a diagonal matrix with diagonal values of $1/a_{ii}$ and at the limit of $\Delta t \rightarrow \infty$, Eq. (44) becomes

$$\|\mathbf{R}^{(i)}\| \leq \eta a_{ii} \|\mathbf{e}_t\| \tag{45}$$

with $0.25 < a_{ii} < 0.45$ in the ESDIRK schemes adopted. When Δt becomes smaller and smaller, the entities of the matrix $\mathcal{L}'_\delta (\mathbf{I}/\Delta t - a_{ii} \mathcal{L}'_\delta)^{-1}$ will tend to zeros. Therefore, an even larger Newton tolerance will maintain that Eq. (44) is satisfied. Overall, Eq. (45) appears to provide a good estimate of the convergence upper bound at least at the limits of $\Delta t \rightarrow \infty$ and $\Delta t \rightarrow 0$. Based on numerical experiments we observe that the Newton tolerance of Eq. (37) seems good enough to maintain temporal convergence accuracy, as will be shown in Sect. 3.1. However, the assumptions made in the derivation of the Newton tolerance may be invalid in some simulations, and if accurate estimate of Newton tolerance are needed, $\mathcal{L}'_\delta (\mathbf{I}/\Delta t - a_{ii} \mathcal{L}'_\delta)^{-1}$ can be estimated using the Jacobian matrix or simply by a calibration for a specific problem.

Although based on some assumptions, Eq. (37) is still much less problem dependent compared with Newton tolerances commonly used such as

$$\begin{cases} \tau_1 = \theta_1 \|\mathbf{v}^0\|, \\ \tau_2 = \theta_2 \|\mathbf{N}(\mathbf{v}^0)\|, \end{cases} \quad (46)$$

as is shown in references [5, 21, 22] and will be shown in Sect. 3.1. It will also change in a more consistent manner with the temporal error when Δt changes. For simulations not using the adaptive Newton tolerance in the following, $\theta_2 = 10^{-3}$ in Eq. (46) is adopted unless otherwise specified.

2.4 Summary of the Adaptive Strategy and Discussion

The basic idea behind the time step adaptation strategy proposed in Sect. 2.2 is the relation between temporal and spatial errors that is sufficient for maintaining temporal accuracy in unsteady simulations. Here, we give a brief discussion to illustrate that it is a rational choice because it shares a very similar logic behind the convergence tests in CFD, which is also a key point in maintaining the high efficiency of implicit time integration methods.

The implicit time integration methods could be more efficient than the explicit methods in some stiff problems such as problems with low Mach number, high Reynolds number and highly stretched grids [2, 21, 22, 27, 29]. The assumption behind this conclusion is that we can safely increase the time steps and temporal errors when using implicit time integration methods without obviously degrading the simulation results. In another word, the comparison of implicit and explicit time integration in efficiency is not based on the computation time of the same temporal error level as is done in most studies of time integrations methods [3, 11] but is based on the computation time as long as the results are sufficiently accurate¹. This implicitly defined sufficiently accurate condition is the maximum error level without obviously degrading the target quantity of interest, which is the one we are seeking when performing convergence tests in CFD simulations. However, this condition usually depends on the unsteady properties of the problem (the magnitude and frequency of the unsteady waves), the scale of interest, the acceptable error level, etc. and is not known before an accurate simulation result is obtained.

Similarly, the error condition behind the time adaptation strategy in Sect. 2.2, is the maximum temporal error level without obviously degrading the results of the discrete partial differential equation (PDE) system. Although, convergence tests are still needed to maintain the accuracy physically, the strategy avoids improper error relations, which may seriously lower the efficiency without obviously improving the results. In a case presented in reference [21], the improper relations of the spatial and temporal errors can increase the CPU time by about 20 times without obviously improving the results.

In some adaptive time-stepping strategies, the adapted time step is limited to be no larger than a time step Δt_{limit} , which is user defined based on their understanding

¹ These studies are still of great value because as long as the time step can be freely chosen for the two methods in comparison, the results can always get the same (desired) error level. However, there is a large difference between these two conditions in comparison between explicit and implicit methods, in which the time step of explicit methods is restricted by the stability condition.

of the physical time scale to be resolved as in reference [22]. However, Eqs. (28) and (29) indicate that limiting the time step will not improve the results but possibly will increase the computational costs. Instead, refining the spatial grid will be much more efficient in improving the results. Therefore, it is advised to refine the mesh instead of limiting the time step to improve the computation accuracy in our time adapting strategy when the adaptive time step is obviously too large to capture the unsteady flow.

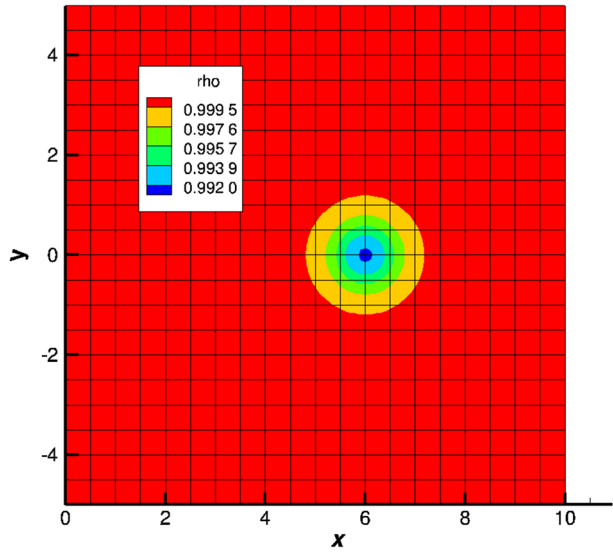
As a by-product of this observation, the adapted time step can also serve as a good indicator of mesh refinement if the temporal scale of interest is well defined and the properties of the ODE solver (ESDIRK in this paper) are well understood. For a specific ODE solver, such as DIRK, we can assess the ODE solver in predicting a single wave at a specific time with a given time step [11]. If the unsteady flow scale (unsteady frequency) of interest is clearly defined in our simulations, this allows us to determine if the unsteady flow scale is satisfactorily captured with the ODE solver and the adapted Δt . If the temporal error is too large to be satisfactory, the relation (29) tells us that decreasing the time step cannot improve the results, and that a more efficient way to increase accuracy is to refine the spatial mesh instead. Therefore, we can use the adaptive time steps as a criteria for spatial mesh refinement.

In the adaptative strategy, extra computational costs are required to estimate spatial and temporal errors. The embedded scheme of Eq. (14) requires an extra implicit RK stage, which will approximately increase the computational cost by $1/(S - 1)$, where S is the number of stages of the main ESDIRK scheme with values of 3, 4 and 6 for the second, third and fourth order ESDIRK schemes adopted, respectively. Similarly, the spatial error estimation in Eq. (30) requires the calculation of a higher order spatial discretization operator $L_\delta^{P+1}(\mathbf{u}_i^{n+1})$. The extra cost from spatial error estimation is relatively lower than those of temporal error estimation for stiff problems since up to an order of magnitude or more L_δ operations are needed in one implicit RK stage. For example, in simulations of $P = 2$ and $P = 3$, the extra cost of $L_\delta^{P+1}(\mathbf{u}_i^{n+1})$ is only about 1.5 times the cost of $L_\delta^P(\mathbf{u}_i^{n+1})$, while the extra cost in temporal error estimating is more 10 times of that. Therefore, we can roughly estimate the extra cost to be $100/(S - 1)\%$. For simulations with little or slow time scale changes, a freezing of the adapted time step can be adopted to lower the additional costs.

Compared with adaptive time-stepping strategies in references [4, 5, 21, 22], the current strategy is different in the following ways.

- The adaptive time step is determined by the relation between the temporal and spatial errors instead of a user defined temporal error level. Therefore, highly problem dependent user defined tolerances are avoided.
- An embedded ESDIRK scheme with an order of $N + 1$ is employed instead of schemes of $(N - 1)$ th order, which predicts the temporal error more accurately. This also helps avoid problem dependent calibration parameters of the estimated errors.
- The adaptation is determined element by element first instead of using the L_2 norm of the temporal error of the whole flow field. Therefore, it has the potential to avoid time step being determined primarily by the large scale unsteady structures and avoid small-scale unsteady structures being under-resolved.

Fig. 1 Isentropic vortex problem: mesh and density distribution



3 Numerical Results

Numerical experiments are conducted in this section to demonstrate the properties of the methods proposed in Sect. 2. The isentropic vortex is simulated in Sect. 3.1 to validate the Newton tolerance choice and illustrate some basic properties of the adaptive time-stepping. The steady-state flow past a flat plate is adopted in Sect. 3.2 to demonstrate the performance of the adaptive method in steady-state simulations. In Sects. 3.3 and 3.4, the Taylor-Green vortex problem and the turbulent flow over a circular cylinder at $Re = 3900$ are studied to illustrate the performance of the methods in freely decaying and wall-bounded turbulent flow simulations, respectively.

3.1 Isentropic Vortex Convection

This test case is adopted because its analytical solution is available, which makes it easier for spatial and temporal error estimations. In the computational domain $[0, 10] \times [-5, 5]$, an inviscid isentropic vortex is convected down stream, the exact solution of which at time t is

$$\begin{cases} \rho = \left(1 - \frac{\varphi^2(\gamma-1)}{16\gamma\pi^2} e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}}, \\ u = u_0 - \frac{\varphi(y-y_0-v_0t)}{2\pi} e^{(1-r^2)}, \\ v = v_0 + \frac{\varphi(x-x_0-u_0t)}{2\pi} e^{(1-r^2)}, \\ p = \rho^\gamma, \end{cases} \tag{47}$$

where $(u_0, v_0) = (1.0, 0.0)$ is the velocity vector of the mean flow, the vortex center at time t is located at $(x_0 + u_0t, y_0 + v_0t)$, r is the distance from the vortex center, and $\varphi = 0.5$ is a parameter that controls the strength of the vortex. Periodic boundary conditions are applied

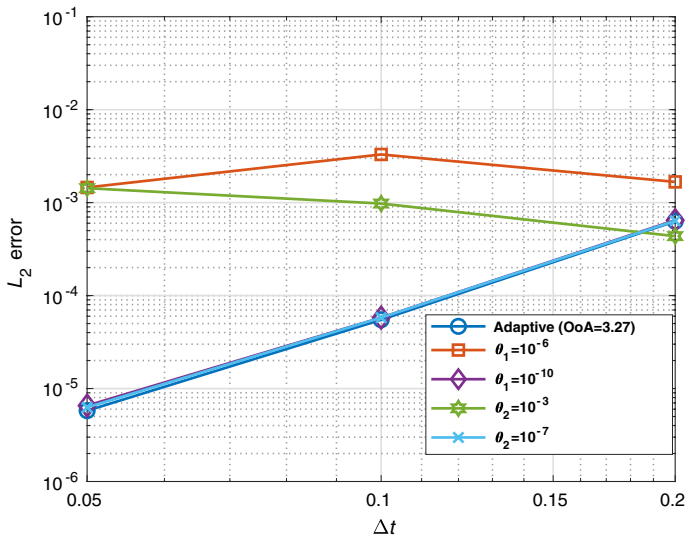


Fig. 2 Isentropic vortex problem: effect of different Newton tolerances on temporal accuracy

Table 1 Isentropic vortex problem: comparison of the adaptive Newton tolerance and the maximum Newton tolerance that maintains temporal accuracy

Δt	τ_{\max}	$\eta \ e_t\ $	$\eta \ e_t\ / \tau_{\max}$
0.05	1.0×10^{-5}	5.7×10^{-6}	0.57
0.1	4.0×10^{-5}	5.5×10^{-5}	0.14
0.2	2.0×10^{-4}	6.3×10^{-4}	0.31

to the boundaries in both directions. The uniform mesh and the density distribution are illustrated in Fig. 1.

To estimate different errors, different solutions are adopted as the reference in the error estimation. The total discretization error of a solution is obtained by subtracting it with the analytical solution of Eq. (47). As in reference [3], the difference between the solution and a solution with a very small time step is regarded as the sum of temporal and iterative errors as the two solutions share the same spatial discretization. The difference between the solution with a very small time step and the analytical solution is regarded as the spatial error, since the error is dominated by the spatial error with a very small time step.

3.1.1 Analysis of the Adaptive Newton Tolerance

This section compares the performance of the proposed Newton tolerance, given by Eq. (37) and $\eta = 0.1$, with commonly adopted tolerances given by Eq. (46) using values $\theta_1 = (10^{-6}, 10^{-10})$ and $\theta_2 = (10^{-3}, 10^{-7})$. The main purpose is to show that Eq. (37) can

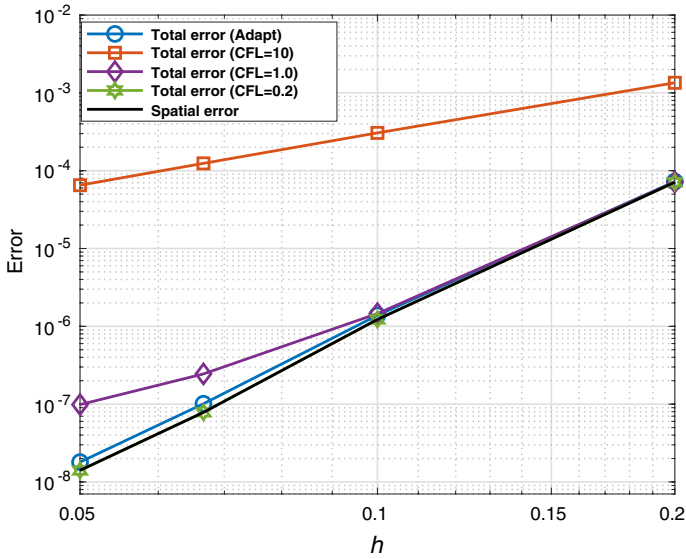


Fig. 3 Isentropic vortex problem: effect of time-stepping method on total errors

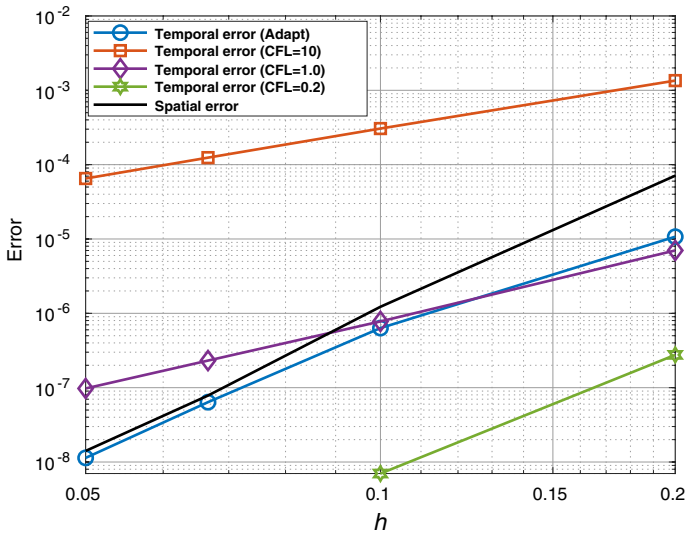


Fig. 4 Isentropic vortex problem: effect of time-stepping method on temporal errors and spatial errors

give a reasonable tolerance and still maintain temporal accuracy. In this case, we employ $P = 2$ and $N = 3$ with $h = 1/3$ and three different time steps: $\Delta t = 0.05, 0.1$, and 0.2 .

Figure 2 shows that large values of the parameters θ_1 and θ_2 in the Newton tolerances from Eq. (46) lead to a degradation of the order of temporal accuracy. Small values of these parameters achieve desired temporal order of accuracy but are not necessarily the best choice or the most efficient choice. Furthermore, they need to be calibrated for each

test case and for each time step to obtain the best performance because in general the term $\Delta t \bar{\mathbf{e}}_{it}$ based on the tolerances of Eq. (46) does not scale simultaneously with the temporal error. On the other hand as shown in Fig. 2, the adaptive Newton tolerance of Eq. (37) achieves the optimal order of accuracy.

To verify that the Newton tolerance is not unnecessarily small, we numerically seek the maximum Newton tolerance τ_{\max} which maintains a difference between $\|\mathbf{e}_t^{n+1} + \Delta t \bar{\mathbf{e}}_{it}\|$ and $\|\mathbf{e}_t^{n+1}\|$ is within 1%. Compared with τ_{\max} , the designed adaptive Newton tolerance $\eta \|\mathbf{e}_t\|$ should be no larger to maintain temporal accuracy and should not be too small to maintain efficiency. Table 1 shows that the ratios between the adaptive Newton tolerance and the τ_{\max} is of the same order of 0.1 for different time steps. This indicates that the iterative error scales simultaneously with temporal error and $\eta = 0.1$ is close to the maximum choice that can maintain temporal accuracy in this test.

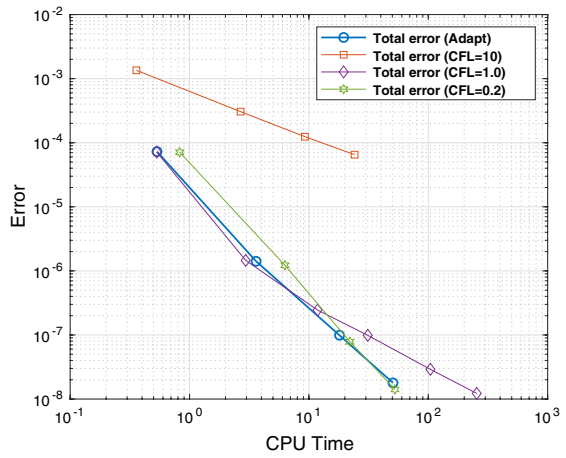
3.1.2 Analysis of Errors in the Adaptive Time Stepping

We described in Sect. 3.1.1 how to choose the tolerances in Newton iteration to ensure that the iterative error is smaller than the temporal error. This section aims at studying spatial, temporal, and total discretization errors characterized in terms of parameters such as the CFL number and the mesh size, h . The adaptive time-stepping strategy is evaluated at a polynomial order $P = 6$. The total discretization errors and temporal errors are calculated with respect to the analytical solution and a reference solution computed at a CFL number of 0.01.

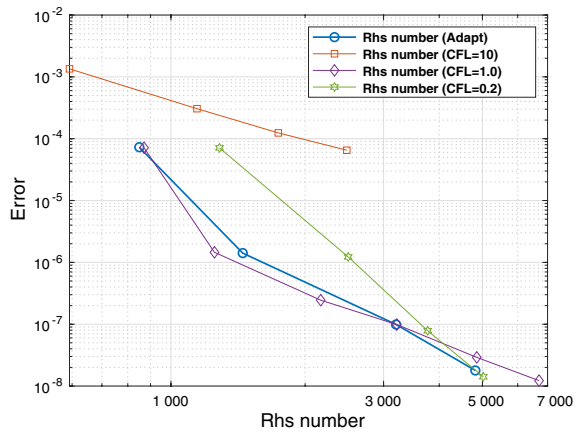
The total errors of the third-order ESDIRK (ESDIRK3) with the adaptive time-stepping, CFL = 10.0, CFL = 1.0, and CFL = 0.2 using different mesh sizes, $h = 0.2$, $h = 0.1$, $h = 0.067$ and $h = 0.05$, are presented in Fig. 3. The corresponding temporal errors are shown in Fig. 4. The spatial errors are also presented for comparison. The temporal errors with the adaptive time steps are always smaller than the spatial errors, as per design, and the total error converges at the same rate as the spatial error. However, for simulations with CFL = 10.0, the total errors are dominated by the temporal errors and converge at a lower order of 3 instead. For simulations with CFL = 0.2, the total errors are dominated by spatial errors. However, the temporal errors are much smaller than the spatial errors, which may lead to extra costs without obviously improving the results. For simulations with CFL = 1.0, temporal error is smaller than the spatial error for $h = 0.2$ but larger than that for $h = 0.05$, which emphasizes the need for calibration for each time step size when using a time step based on a CFL number. Referring to the adaptive time-stepping methods in references [5, 22], the temporal error curve should be horizontal lines because the temporal error tolerance in these methods is user defined and independent of h .

The CPU time of the above simulations is compared in Fig. 5. The simulation with the adaptive strategy is close to the most efficient ones at all the error levels. It is slightly more expensive than the most efficient one at some error levels partly due to the extra costs in estimating the errors. Figure 5 also shows the corresponding curves of GMRES iteration number and residual evaluation number. These numbers are the summation of all the time steps during the time interval. The GMRES iteration number and the residual evaluation number of the adaptive method are also close to the smallest among different time-stepping methods. For the parameters adopted in this test, the efficiency of the adaptive stepping can be up to ten times more efficient at the same error level than the simulations with a naive choice of CFL, which highlights the importance of balancing the spatial and temporal errors from the point view of efficiency.

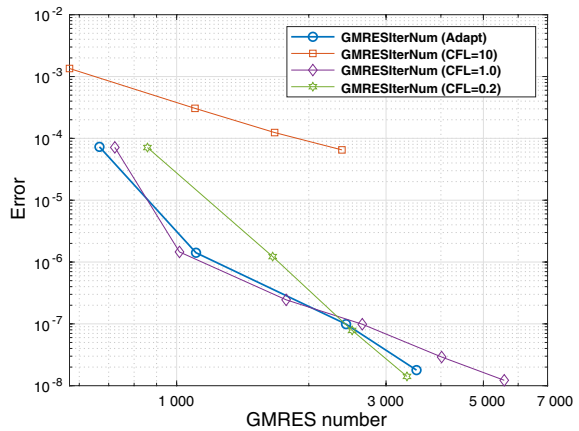
Fig. 5 Isentropic vortex problem: effect of time-stepping method on efficiency



(a) CPU time



(b) Residual evaluation numbers



(c) GMRES iteration numbers

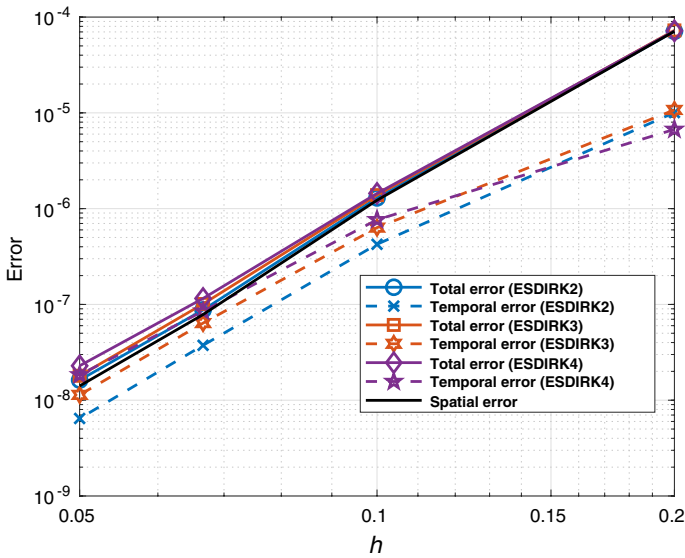


Fig. 6 Isentropic vortex problem: performance of adaptive time-stepping for different ESDIRK schemes

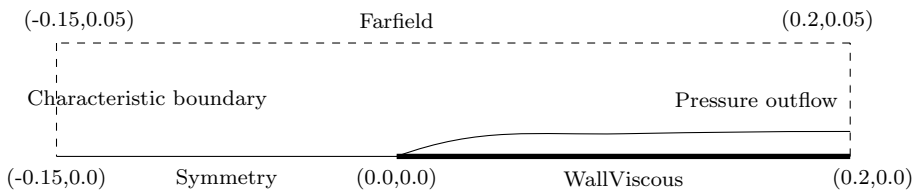
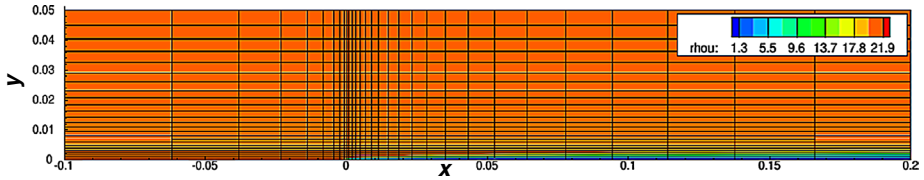


Fig. 7 Boundary-layer flow past a flat plate: geometry and boundary conditions

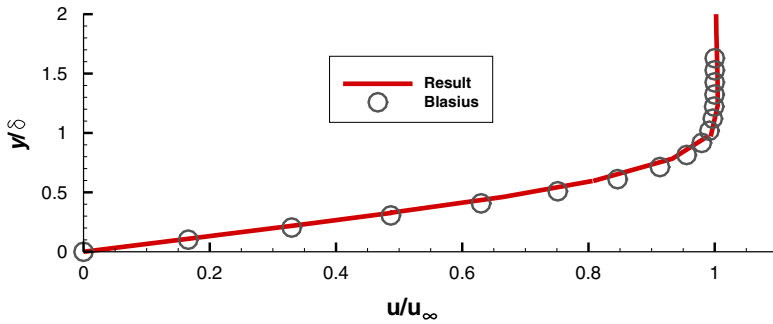
The errors of ESDIRK schemes with $N = 2, 3, 4$ are compared in Fig. 6. All temporal errors are relatively smaller than the spatial errors and the total errors almost coincide with each other for different values of N . Therefore, the simulation results are almost independent of the ESDIRK scheme adopted in the adaptive time-stepping. This property should also be achieved by the adaptive methods proposed in [5, 22], but it requires an additional calibrating process which is problem dependent.

3.2 Steady-State Flat Plate Boundary-Layer Flow

The objective of this example is to illustrate how the adaptive time-stepping strategy could be used to accelerate convergence towards steady-state in a practical case. The case considered here is the boundary-layer flow past a flat plate at a Mach number $Ma = 0.1$ and a Reynolds number $Re = 1.6 \times 10^6$. The geometry and boundary conditions are depicted in Fig. 7. The details of boundary conditions can be found in references [20, 29]. The simulations are performed using $P = 2$ and $N = 3$ with the highly stretched mesh shown in Fig. 8a. The computed profile of the horizontal velocity is shown and compared with the



(a) Mesh and x -momentum



(b) Velocity profile, where δ is the boundary layer thickness

Fig. 8 Boundary-layer flow past a flat plate: mesh and velocity profile

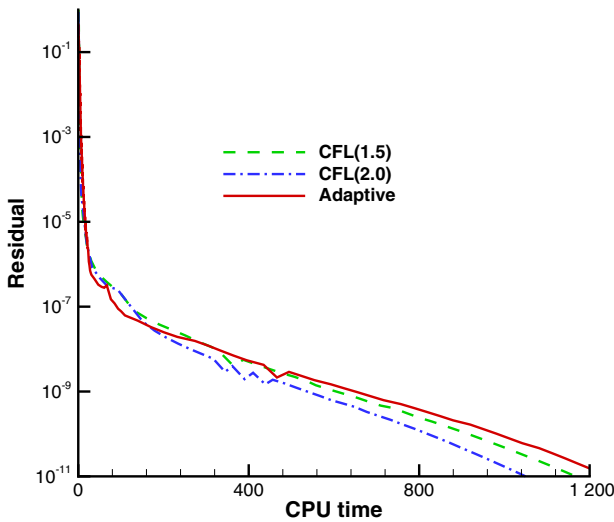
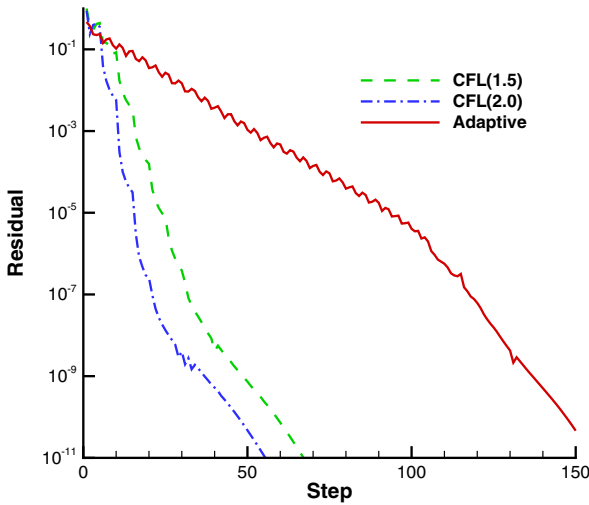
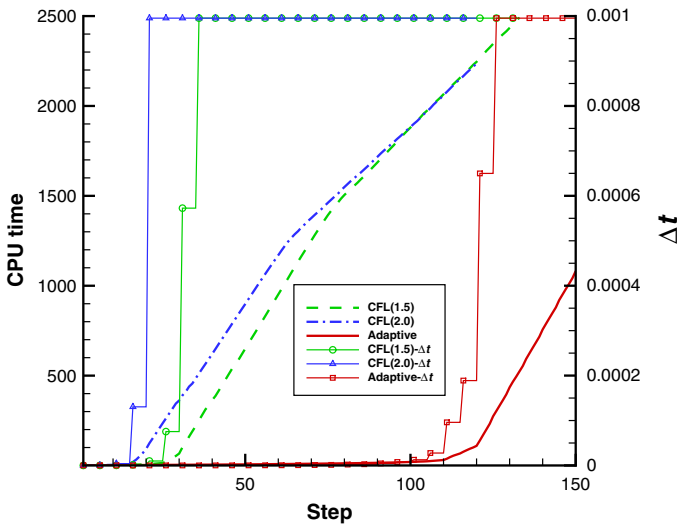


Fig. 9 Boundary-layer flow past a flat plate: residual convergence history. The number in the parenthesis indicates the growth rate of the CFL number

analytical Blasius profile in Fig. 8b. In all the simulations below, the time step is set to be no larger than that corresponding to a CFL number of 5×10^4 to maintain the stability of the simulations.



(a) Residual



(b) CPU time and Δt

Fig. 10 Boundary-layer flow past a flat plate: residual, CPU time and time step evolution curves as a function of steps. The number in the parenthesis indicates the growth rate of the CFL number

For the steady-state problem we adopt the same adaptive time-stepping strategy and parameters used for the unsteady simulations to show that the proposed adaptive strategy can adjust itself based on the properties of the problem. For comparison, a time-stepping strategy based on a growing CFL is also presented, which starts from $CFL = 0.1$ and grows by a rate of 1.5 and 2.0 up to $CFL = 5 \times 10^4$. A relative Newton tolerance of $\theta_2 = 10^{-6}$ in

Fig. 11 Taylor-Green vortex: evolution of the enstrophy

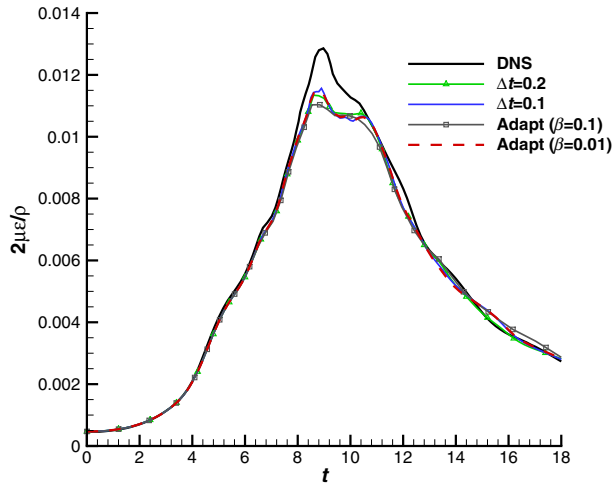
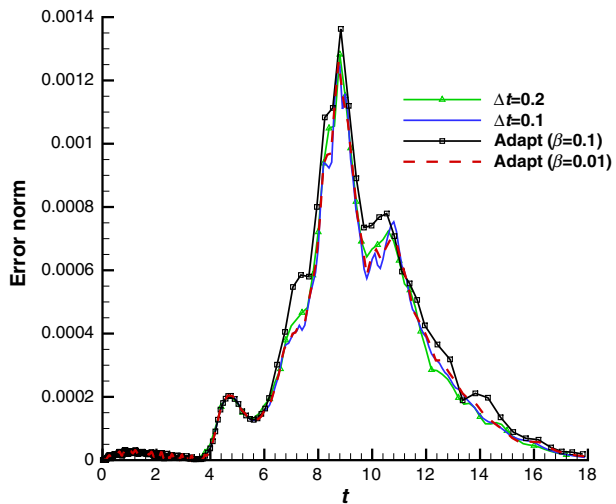


Fig. 12 Taylor-Green vortex: error norms of the kinetic energy dissipation rate



Eq. (46) is adopted since the default choice of $\theta_2 = 10^{-3}$ easily leads to blowing up at such a large CFL number. The growing CFL number method is a commonly adopted strategy for steady-state simulations because a small time step is needed to maintain stability in the initial transient phase, while a large time step is needed to accelerate the convergence when the flow field is close to the steady-state solution [6, 28].

The convergence history of the normalized residual norm $\|L_\delta(\mathbf{u})\|$ is presented in Fig. 9. The simulation with the adaptive time-stepping proposed in Sect. 2.2 converges almost as fast as the simulations with a CFL growth rate of 2.0. The residual, Δt and CPU time at different steps are illustrated in Fig. 10. It shows that the Δt of the adaptive method grows much slower at the first 100 time steps. However, the extra CPU cost because of the extra steps is low since it is cheap to converge at a small Δt . Figure 10 also demonstrates that the adaptive Δt is small at the beginning of the simulation. This is because the transient flow is highly unsteady leading to a large temporal derivative term $D_t(\mathbf{u})$ in Eq. (15) and

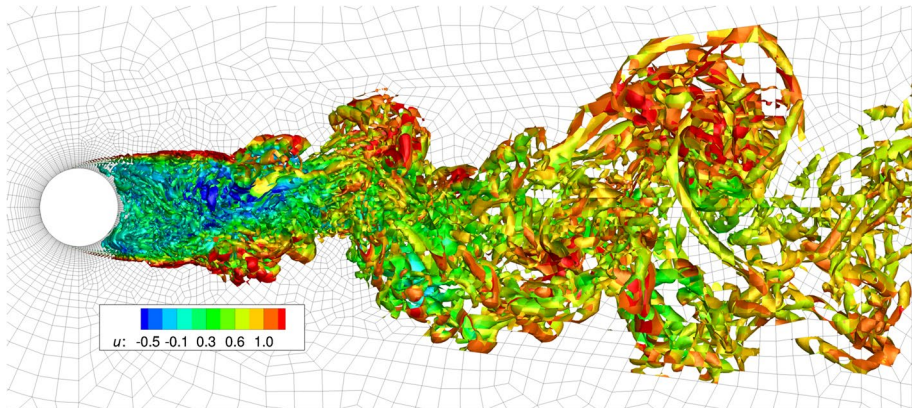


Fig. 13 Turbulent flow over a circular cylinder: Q criteria iso-surface ($Q = 5$) and mesh from reference [29]

consequently to a smaller Δt . As the flow field becomes closer to the steady state, the Δt increases because the magnitude of the unsteady waves in the flow field becomes smaller, which leads to a smaller term $D_t(\mathbf{u})$ and larger Δt based on the relation of Eq. (29). The adaptive time-stepping adjusts the Δt based on unsteady properties of the flow field and achieves fast convergence while maintaining stability.

Although the adaptive time-stepping may not be as efficient as methods specially designed for steady-state simulations, such as local time-stepping and multi-grid, it is general and, with a fixed set of parameters, is capable to efficiently simulate a wide range of steady and unsteady flows.

3.3 Taylor-Green Vortex

Here, we apply the adaptive time-stepping strategy to the simulation of the Taylor-Green vortex, which represents an unsteady flow of decaying vortices. The initial large vortices break down into smaller scale vortices and eventually into turbulent flow. The simulations are run with $Ma = 0.1$, $Re = 1600$ and periodic boundary conditions on a 32^3 uniformly distributed mesh with $P = 4$ and $N = 4$. Note that, to simulate nearly incompressible conditions, the flow is taken to be compressible but at a low Mach number of $Ma = 0.1$.

Figure 11 presents the evolution of $2\mu\epsilon/\rho$, where ϵ denotes the enstrophy. According to reference [7], this variable is a good estimate of the kinetic energy dissipation rate in the incompressible limit. Figure 12 presents the errors of the dissipation rate as defined by Eq. (18) of reference [7]. The simulations are run with the set of values $\beta = 0.1$ and $\beta = 0.01$ for the adaptive time-stepping strategy and they are also run with $\Delta t = 0.1$ and $\Delta t = 0.2$ for comparison. The decay curves of $2\mu\epsilon/\rho$ are in relatively good agreement with each other for all the simulations. Their differences can be analyzed from the distributions of the errors in Fig. 12. The error with adaptive time-stepping and $\beta = 0.1$ is slightly larger than other errors. The error with $\beta = 0.01$ almost coincides with the error of $\Delta t = 0.1$. Both simulations with $\beta = 0.1$ and $\beta = 0.01$ are accurate in temporal accuracy especially compared with the spatial error, which can be roughly estimated by the deviation from the DNS result [7].

We enforce the local temporal error to be smaller than the spatial error by design, but there is no reason to introduce a bias towards the spatial error if the main consideration

Fig. 14 Turbulent flow over a circular cylinder: adaptive time steps (Δt) of ESDIRK2, ESDIRK3 and ESDIRK4

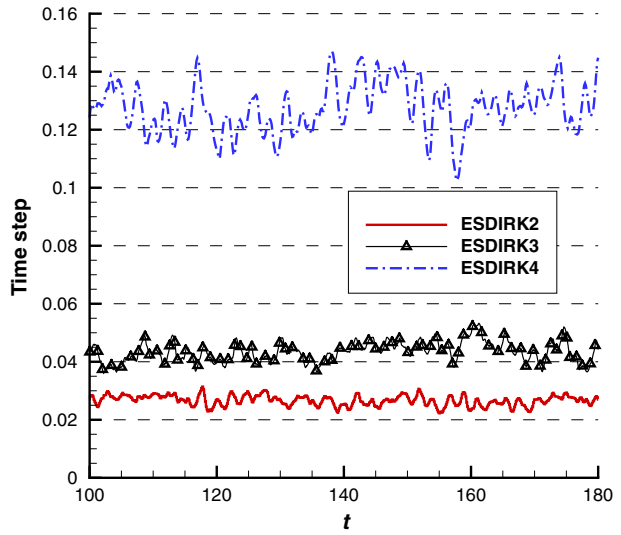
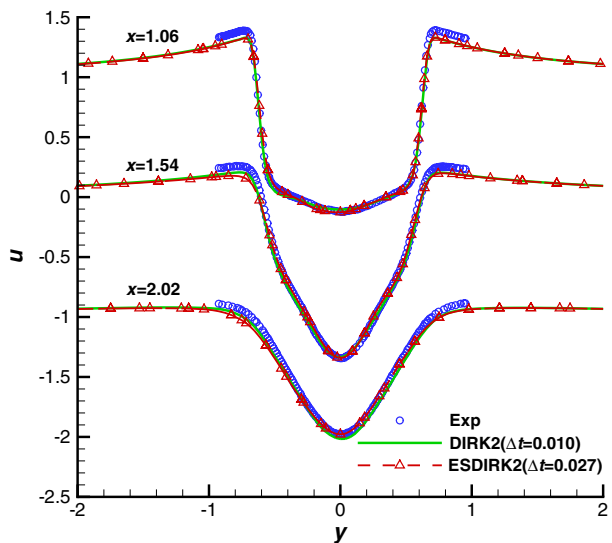


Table 2 Turbulent flow over a circular cylinder: comparisons of averaged time steps and CPU time for different DIRK schemes with the CPU time ratio defined as the ratio of CPU time between the current method and the DIRK2 method

Methods	$\Delta \bar{t}$	CPU time	CPU time ratio
DIRK2	0.010	7.2×10^2	1.00
ESDIRK2	0.027	7.9×10^2	1.09
ESDIRK3	0.043	9.8×10^2	1.36
ESDIRK4	0.127	1.1×10^3	1.53

Fig. 15 Turbulent flow over a circular cylinder: comparison of time-averaged velocity distribution. The velocity profiles at $x = 1.54$ and $x = 2.02$ are shifted down by increments of $u = -1$ and $u = -2$, respectively



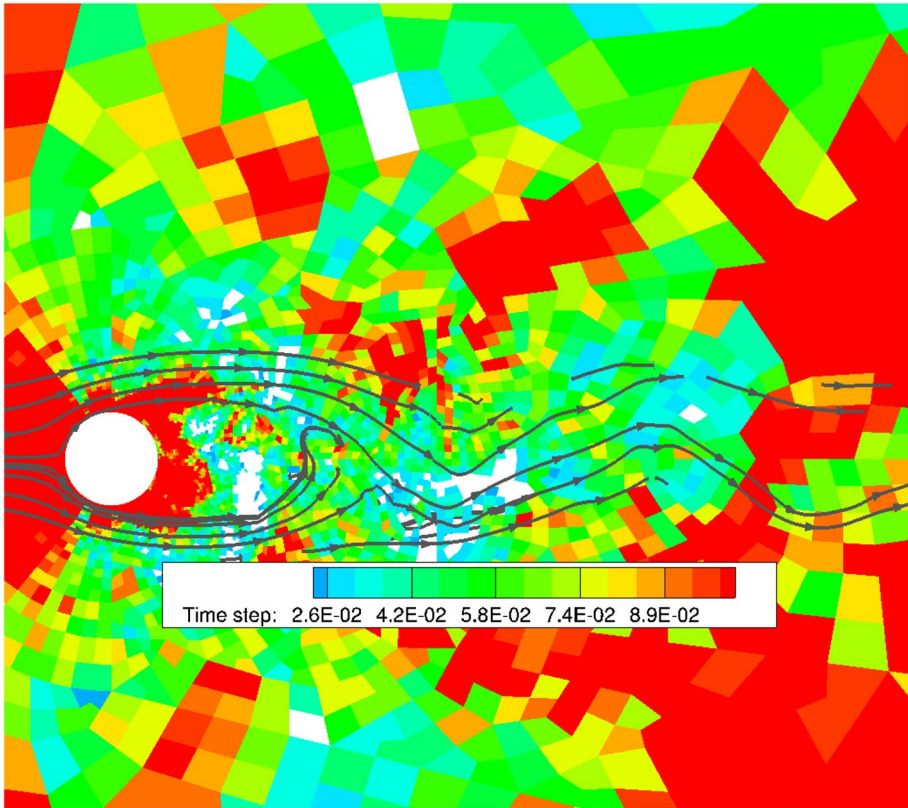


Fig. 16 Turbulent flow over a circular cylinder: values of the elemental time steps

is simply to minimize the error magnitude. This consideration may be of relevance for under-resolved turbulent simulations where the spatial truncation error term sometimes serve as an implicit modeling of the under-resolved scales, which is referred to as the implicit large eddy simulation model [9].

To avoid the temporal error from influencing the modeling of the under-resolved scales, a smaller temporal error compared with the spatial error, $\beta = 0.01$, is adopted in implicit LES simulations of turbulent flows in our study. This approach gives reasonable results as shown in Fig. 12 and in Sect. 3.4. The computational CPU time of the simulations with $\beta = 0.01$ is slightly larger than that of simulation with $\Delta t = 0.1$ (1.27 times) partly because of the additional costs required for the error estimation.

3.4 Turbulent Flow Over a Circular Cylinder at $Re = 3\,900$

The widely studied test case of the turbulent flow over a circular cylinder at $Re = 3\,900$ is adopted here to further illustrate the performance of the adaptive strategy in wall-bounded turbulence simulations. Details of the settings of the test case can be found in reference [29], which employ 123 360 curved unstructured hexahedral elements and a polynomial order of $P = 2$. The unstructured mesh distributions and the vortex structures behind the

Table 3 Summary of tunable parameters in the adaptive strategy

Parameter	Position	Range	Current choice
β	Equation (29)	$0 < \beta < 1$	$\beta = 0.1$ or $\beta = 0.01$
η	Equation (37)	$0 < \eta < 1$	$\eta = 0.1$
r	Equation (35)	$r \in [0, \infty)$	$r = 1$
ϵ_m	Equation (33)	Small relative to \mathbf{U}	Equation (34)

Table 4 Summary of parameters in different test cases and comparison of efficiency. The CPU time ratio is defined as the ratio of CPU time using the adaptive strategy and that using the parameters listed above

Test case	CFL	Newton tolerance	CPU time ratio
Isentropic vortex ($h = 0.2$)	1.0	$\theta_2 = 10^{-3}$	0.96
Isentropic vortex ($h = 0.05$)	0.2	$\theta_2 = 10^{-7}$	1.01
Boundary layer	$0.1 - 5 \times 10^4$	$\theta_2 = 10^{-6}$	1.21
Taylor-Green vortex	≈ 29	$\theta_2 = 10^{-3}$	1.27
Circular cylinder	≈ 40	$\theta_2 = 10^{-3}$	1.09

cylinder are illustrated in Fig. 13. Reference [29] adopted a second-order DIRK scheme (DIRK2) with a time step of 0.01, which corresponds to a CFL number of about 40. The time step was determined based on some numerical tests on its efficiency and accuracy in reference [29]. Here, we simulate the same case using the ESDIRK2, ESDIRK3 and ESDIRK4, together with the adaptive time-stepping strategy, to demonstrate their performance.

The adaptive time steps for a duration corresponding to approximately 19 vortex shedding periods are shown in Fig. 14. Table 2 presents the corresponding averages, $\Delta \bar{t}$. The averaged time step of ESDIRK2 is approximately 0.027, which is 2.7 times larger than the $\Delta t = 0.01$ of the DIRK2. The adapted time steps are larger for the ESDIRK3 and ESDIRK4 with averaged values of 0.043 and 0.127, respectively, because of the improved resolution of the high-order ESDIRK schemes.

Table 2 summarizes the averaged time steps and CPU time of the simulations. The results show that the simulations with adaptive time-stepping strategy achieve good efficiency with only a small overhead in CPU time partly due to the extra costs of error estimations. The time-averaged velocity distributions are presented in Fig. 15. The result of ESDIRK2 with $\Delta t = 0.027$ almost coincides with the result of DIRK2 with $\Delta t = 0.01$, both of which are in a good agreement with experimental results [23]. The simulations with adaptive time step give accurate predictions of the time-averaged quantities.

Figure 16 depicts the distribution of elemental Δt for the ESDIRK2 scheme corresponding to the variable ρu on a slice of the flow field. To highlight the elements with a smaller elemental time step than the global time step, the contour coloring is cut below the global time step and these elements are presented in white. One important observation is that the relation of Eq. (29) is maintained in the majority of elements because the elemental time step is smaller than the global time step in fewer than 1% of the elements. The influence of ϵ_m in Eq. (33) is negligible in the global time step in this test case, which is only effective in elements with very small error magnitudes.

3.5 Summary of Parameters in the Simulations

The tunable parameters in the adaptive strategy are listed in Table 3. The parameter β controls the relation between the temporal error and spatial error, which is set to 0.1 for general simulations, but to 0.01 for implicit large-eddy simulations as discussed in Sect. 3.3. Similarly, η controls the relation between the temporal error and iterative error, which is set to 0.1. The value r in Eq. (35) determines the contribution of elemental time steps to the global time step. For $r = 0$, a simple algebraic average is used. A larger r leads to a larger contribution from elements with larger temporal errors. The term ϵ_m in Eq. (33) avoids division by zero and also serves as a threshold values for the error level of interest. The current choice of Eq. (34) is motivated by the truncation error of the time integration process. In determining β and η , we assume the spatial error is dominating. However, no optimization on the values of β and η is performed, since the optimal values are usually problem dependent.

On the contrary, in the simulations without the adaptive strategy, the choices of Δt and Newton tolerance have to be changed case by case to achieve a good performance as summarizes in Table 4. These case by case choices are obtained at costs of repeated numerical experiments, which could be expensive especially when starting the simulations of a new test case. The comparison of computational efficiency in Table 4 shows that the adaptive strategy is close to the most efficient choices in all the cases without case by case optimized parameters, which demonstrates the remarkable generality of the adaptive strategy.

4 Discussion and Conclusions

We have proposed a balanced adaptive time-stepping strategy based on the idea of balancing the errors generated within one time step which has been implemented in an implicit DG high-order spectral/*hp* element solver.

This adaptive time-stepping strategy maintains temporal accuracy in the sense that the total error is dominated by the spatial error and further decreasing the temporal error will not obviously improve the results of the discrete PDE system, which is verified by numerical experiments in the isentropic vortex, Taylor-Green vortex and turbulent flow over a circular cylinder problem. Moreover, this adaptive time step is a relatively efficient choice because it is around the maximum value with temporal accuracy, the efficiency of which is tested in a variety of steady-state and unsteady problems, including the isentropic vortex, Taylor-Green vortex, flat plate boundary layer and turbulent flow over a circular cylinder. In all these tests, the adaptive methods are close to the most efficient one possibly with small overhead partly due to extra costs in error estimating.

This adaptive time-stepping strategy performs well in a variety of problems without the need of tuning the parameters or requiring a priori knowledge of the flow properties, thus reducing the necessity for user intervention. This feature will facilitate the development of automatic CFD simulation pipelines in a variety of application areas such as, for instance, shape optimization.

The main idea of the adaptive time stepping is to construct a proper relation between different errors, which is not limited to the spatial discretization methods, temporal discretization methods and error estimators adopted in this paper. Provided properly defined spatial and temporal error estimators, the idea should also be applicable to other unsteady PDE solvers. Currently, the adaptive method is based on the local errors generated within one time step, which in theory can not guarantee the global error is still spatial error dominated. This could be improved by adopting some global error estimators.

The adaptive strategy provides an upper bound of the time step based on the requirement of temporal accuracy. However, this upper bound does not necessarily maintain simulation stability in challenging problems. Methods for improving robustness could be considered in the future.

Appendix A Butcher Tableau of ESDIRK Schemes

The coefficients of ESDIRK schemes adopted in this paper can be expressed in the form of the general Butcher tableau shown in Table A1 [17], where S and $R = S + 1$ are the numbers of stages for the main scheme and the embedded scheme, respectively. The coefficients of the matrix A are sufficient to define the whole Butcher tableau since $c_i = \sum_{j=1}^{S+1} a_{ij}$, $b_i = a_{Si}$, and $\hat{b}_i = a_{Ri}$. The last two relations indicate that the main and

Table A1 General form of the Butcher tableau of the embedded ESDIRK. Here R is shorthand for $R = S + 1$

$$\begin{array}{c|cccccc}
 & 0 & 0 & \dots & \dots & \dots & 0 \\
 & c_2 & a_{21} & \alpha & 0 & \dots & \dots & 0 \\
 & c_3 & a_{31} & a_{32} & \alpha & 0 & \dots & 0 \\
 \begin{array}{l} c \\ \hline \hat{b}^T \end{array} & A & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
 & & c_S & a_{S1} & a_{S2} & \dots & \dots & \alpha & 0 \\
 & & c_R & a_{R1} & a_{R2} & \dots & \dots & \dots & \alpha \\
 \hline
 & & & b_1 & \dots & \dots & \dots & b_S & 0 \\
 & & & \hat{b}_1 & \dots & \dots & \dots & \dots & \hat{b}_R
 \end{array}$$

Table A2 Butcher array of embedded ESDIRK2 with $S + 1 = 4$

	0	0	0	0
	0.292 893 218 800 000	0.292 893 218 800 000 0	0	0
$A =$	0.353 553 390 567 523	0.353 553 390 632 477	0.292 893 218 800 000	0
	0.215 482 203 122 508	0.686 886 723 913 539	- 0.195 262 145 836 047	0.292 893 218 800 000

Table A3 Butcher array of embedded ESDIRK3 with $S + 1 = 5$

	0	0	0	0	0
	0.435 866 521 508 46	0.435 866 521 508 46 0	0	0	0
$A =$	0.140 737 774 724 71	- 0.108 365 551 381 32	0.435 866 521 508 46	0	0
	0.102 399 400 619 91	- 0.376 878 452 255 56	0.838 612 530 127 19	0.435 866 521 508 46	0
	0.157 024 897 860 32	0.117 330 441 370 44	0.616 678 030 392 12	- 0.326 899 891 131 34	0.435 866 521 508 46

Table A4 Butcher array of embedded ESDIRK4 with $S + 1 = 7$

	0	0	0	0	0	0	0
	0.270 000 000 000 000 00	0.270 000 000 000 000 00	0	0	0	0	0
	0.135 000 000 000 000 00	0.872 653 718 043 596 86	0.270 000 000 000 000 00	0	0	0	0
$A =$	0.248 142 112 344 473 22	0.132 820 885 228 593 22	− 0.038 866 866 589 177 71	0.270 000 000 000 000 00	0	0	0
	0.254 944 798 221 504 71	0.131 061 964 223 472 00	− 0.045 220 939 302 357 08	0.033 891 216 820 516 42	0.270 000 000 000 000 00	0	0
	0.175 499 755 231 829 41	0	− 0.016 417 259 314 923 83	3.593 571 752 900 106 25	− 3.022 654 248 817 011 82	0.270 000 000 000 000 0	0
	0.158 476 126 436 704 10	0	− 0.073 847 037 320 949 83	5.260 567 763 976 348 93	− 4.839 469 477 584 075 00	0.224 272 624 491 971 80	0.270

the embedded schemes are both stiffly accurate. The diagonal coefficients are given by $a_{11} = 0$ and $a_{ii} = \alpha; i = 2, \dots, S + 1$. The coefficients of the matrix A for the ESDIRK2, ESDIRK3 and ESDIRK4 schemes are presented in Tables A2, A3 and A4, respectively.

Acknowledgements Zhen-Guo Yan acknowledges supports from the National Natural Science Foundation of China (Grant no. 11902344) and National Numerical Windtunnel Project. The development of the implicit solver in Nektar++ has been supported by EPSRC grant (EP/R029423/1) and UK Turbulence Consortium grant (EP/R029326/1). We would like to gratefully acknowledge access to the computing facilities provided by the Imperial College Research Computing Service (<https://doi.org/10.14469/hpc/2232>).

Compliance with Ethical Standards

Conflict of Interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Arévalo, C., Söderlind, G., Hadjimichael, Y., Fekete, I.: Local error estimation and step size control in adaptive linear multistep methods. *Numer. Algor.* **86**, 537–556 (2021)
2. Bassi, F., Botti, L., Colombo, A., Crivellini, A., Ghidoni, A., Massa, F.: On the development of an implicit high-order discontinuous Galerkin method for DNS and implicit LES of turbulent flows. *Eur. J. Mech. B/Fluids* **55**, 367–379 (2016)

3. Bijl, H., Carpenter, M.H., Vatsa, V.N., Kennedy, C.A.: Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: laminar flow. *J. Comput. Phys.* **179**(1), 313–329 (2002)
4. Birken, P., Gassner, G., Haas, M., Munz, C.-D.: Preconditioning for modal discontinuous Galerkin methods for unsteady 3D Navier-Stokes equations. *J. Comput. Phys.* **240**, 20–35 (2013)
5. Blom, D.S., Birken, P., Bijl, H., Kessels, F., Meister, A., van Zuijlen, A.H.: A comparison of Rosenbrock and ESDIRK methods combined with iterative solvers for unsteady compressible flows. *Adv. Comput. Math.* **42**(6), 1401–1426 (2016)
6. Bücker, H.M., Pollul, B., Rasch, A.: On CFL evolution strategies for implicit upwind methods in linearized Euler equations. *Int. J. Numer. Methods Fluids* **59**(1), 1–18 (2009)
7. De Wiart, C.C., Hillewaert, K., Duponcheel, M., Winkelmann, G.: Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. *Int. J. Numer. Methods Fluids* **74**(7), 469–493 (2014)
8. Eisenstat, S.C., Walker, H.E.: Choosing the forcing terms in an inexact newton method. *SIAM J. Sci. Comput.* **17**(1), 16–32 (1996)
9. Grinstein, F.F., Margolin, L.G., Rider, W.J.: *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge University Press, Cambridge (2007)
10. Hartmann, R., Houston, P.: An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations. *J. Comput. Phys.* **227**(22), 9670–9685 (2008)
11. Holst, K.R., Glasby, R.S., Bond, R.B.: On the effect of temporal error in high-order simulations of unsteady flows. *J. Comput. Phys.* **402**, 108989 (2020)
12. Kalkote, N., Assam, A., Eswaran, V.: Acceleration of later convergence in a density-based solver using adaptive time stepping. *AIAA J.* **57**(1), 352–364 (2019)
13. Karniadakis, G., Sherwin, S.: *Spectral/hp Element Methods for Computational Fluid Dynamics*, 2nd edn. Oxford Science Publications, Oxford (2013)
14. Kelley, C.T.: Numerical methods for nonlinear equations. *Acta Numerica* **27**, 207–287 (2018)
15. Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review. NASA report TM-2016-219173 (2016)
16. Knoll, D.A., Keyes, D.E.: Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.* **193**(2), 357–397 (2004)
17. Kværnø, A.: Singly diagonally implicit Runge-Kutta methods with an explicit first stage. *BIT Numer. Math.* **44**(3), 489–502 (2004)
18. Lian, C., Xia, G., Merkle, C.L.: Solution-limited time stepping to enhance reliability in CFD applications. *J. Comput. Phys.* **228**, 4836–4857 (2009)
19. Meisrimel, P., Birken, P.: Goal oriented time adaptivity using local error estimates. *Algorithms* **13**(5), 113 (2020)
20. Mengaldo, G.: *Discontinuous Spectral/hp Element Methods: Development, Analysis and Applications to Compressible Flows*. PhD thesis, Imperial College London, UK (2015)
21. Noventa, G., Massa, F., Bassi, F., Colombo, A., Franchina, N., Ghidoni, A.: A high-order discontinuous Galerkin solver for unsteady incompressible turbulent flows. *Comput. Fluids* **139**, 248–260 (2016)
22. Noventa, G., Massa, F., Rebay, S., Bassi, F., Ghidoni, A.: Robustness and efficiency of an implicit time-adaptive discontinuous Galerkin solver for unsteady flows. *Comput. Fluids* **204**, 104529 (2020)
23. Parnaudeau, P., Carlier, J., Heitz, D., Lamballais, E.: Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. *Phys. Fluids* **20**(8), 085101 (2008)
24. Söderlind, G.: Automatic control and adaptive time-stepping. *Numer. Algorithms* **31**(1), 281–310 (2002)
25. Söderlind, G., Wang, L.: Adaptive time-stepping and computational stability. *J. Comput. Appl. Math.* **185**(2), 225–243 (2006)
26. Toro, E.F.: *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd edn. Springer, Berlin (2009)
27. Vandenhoeck, R., Lani, A.: Implicit high-order flux reconstruction solver for high-speed compressible flows. *Comput. Phys. Commun.* **242**, 1–24 (2019)
28. Vanderstraeten, D.: An expert system to control the CFL number of implicit upwind methods. In: Toro, E.F. (ed) *Godunov Methods: Theory and Applications*, pp. 977–984. Springer, Boston (2001)
29. Yan, Z.-G., Pan, Y., Castiglioni, G., Hillewaert, K., Peiró, J., Moxey, D., Sherwin, S.J.: Nektar++: design and implementation of an implicit, spectral/hp element, compressible flow solver using a Jacobian-free Newton Krylov approach. *Comput. Math. Appl.* **81**, 351–372 (2020)
30. Yildirim, A., Kenway, G.K.W., Mader, C.A., Martins, J.R.R.A.: A Jacobian-free approximate Newton-Krylov startup strategy for RANS simulations. *J. Comput. Phys.* **397**, 108741 (2019)