




Adaptive Time Steps for Compressible Flows Based on Dual-Time Stepping and a RK/Implicit Smoother

O. Peles¹ · E. Turkel² 

Received: 24 January 2019 / Revised: 16 July 2019 / Accepted: 29 July 2019 / Published online: 7 August 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

We present an adaptive time stepping technique, based on a dual-time stepping scheme together with a RK/implicit smoother for physical time marching. This method is found to be very efficient for transient problem characterized by a large variations of time scales during the physical time and hence a wide range of required physical time steps. We discuss the method and present several examples of compressible flow problems.

Keywords Navier–Stokes · DIRK · Adaptive time steps · Acceleration · Dual time stepping

Mathematics Subject Classification 65L06 · 65M06

1 Introduction

Dual time stepping with Backward Differentiation (BDF) of order two in physical time is a common algorithm for time dependent problems, both for inviscid, viscous flows and URANS (Unsteady RANS) simulations (See [8]). Jameson [8] combined BDF with a low-storage Runge–Kutta for the pseudo-time marching technique [10]. This scheme is point implicit in the physical time and unconditionally stable with respect to the physical time steps. However, the scheme requires constant time steps and the accuracy decreases when using large time steps.

Acceleration of the convergence in pseudo-time is one way to reduce the CPU time and many techniques are used in the literature such as multigrid methods, low Mach preconditioning and residual smoothing. The RK/Implicit scheme was extended to solve transport

We dedicate this paper to the memory of Saul (Shalom) Abarbanel who was a mentor to whole generation of students and colleagues.

✉ E. Turkel
eliturkel@gmail.com

O. Peles
oren_peles@walla.co.il

¹ IMI Systems, Ramat Ha-Sharon, Israel

² School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel

equations for modeling the effects of turbulence in Swanson and Rossow [22–24]. Peles et al. [15] applied this scheme to the two-equation turbulence model (SST model) of Menter. Later, Peles et al. [16] used this idea to develop the capability of solving chemically reacting flows as well as general multi-physics problems [17]. For low Mach number problems preconditioning methods were developed in [26,27,29,31,32]. Alternatively, Rossow [18] used a modified form of the numerical dissipation for low Mach number accuracy. Such an approach was further extended by Swanson et al. [25]. It was also combined with the RK/Implicit smoother in [16] and [13] using the flux Jacobian in a Roe-type scheme. Langer [14] extended the method for unstructured grids.

Large physical time steps, using implicit schemes, is another way to reduce the CPU time. A standard implicit scheme used is BDF (backward difference formula). However, for the BDF method to be unconditionally stable it can only be at most second-order accurate in time. Thus, if one wishes to use a higher order accurate scheme, in time, a good choice is implicit RK methods for the physical time steps. Several techniques have been developed for the implementation of dual time stepping with implicit RK for the physical time. In [3] a method using dual time stepping combined with multi-stage schemes in physical time for unsteady low Mach number compressible flows was developed. [4] and [9] presented a formulations of a dual time stepping procedures for implicit Runge–Kutta schemes. In [9] Jameson used a pseudo-time RK/Implicit smoother for time dependent problems. The method was tested on both subsonic and transonic tests cases. The CPU time cost for DIRK methods with n stages is n evaluations of the residuals and $n - 1$ residual smoothing so the gain in terms of CPU time is small. However, an intrinsic weakness of the BDF method is the need for constant time steps while DIRK methods allow varying and adaptive time steps. This is important for problems with time scales that vary sharply during the physical time range. For problems requiring only constant time steps, the optimum time step of BDF method can be more efficient than the optimal DIRK time step. However, in the BDF scheme, without a prior knowledge about the required time steps, a time-step sensitivity needs to be done. This process also costs CPU time so in total, an adaptive time step with DIRK methods can be more efficient. Adaptive time stepping for incompressible flows based on a BDF formulation was developed by Hay et al. [7] and for an IRK scheme was presented in [11].

In this paper, we present a dual time stepping technique for compressible flow based on a low-storage RK with an implicit residual smoother (RK/implicit smoother) scheme for the pseudo time marching and a DIRK scheme for the physical time steps with adaptive time stepping. The governing equations will be presented in Sect. 2. In Sect. 3.1 the dual time stepping with BDF method will be discussed, and in Sect. 3.2 the dual-time stepping with DIRK will be presented. Linear stability and accuracy analysis will be discussed in Sect. 4. In Sect. 5 the adaptive time step technique will be described and several examples of compressible flows will be presented in Sect. 6.

2 Governing Equations

We consider the three dimensional, compressible Navier–Stokes questions, given in conservative form as

$$\frac{\partial W}{\partial t} + \nabla \mathbf{F}(W) = \nabla \mathbf{F}_v(W) + S(W) \quad (1)$$

where W is the vector of variables - $W = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, ρ is the gas density, u , v and w are the components of the velocity vector, p is the pressure, related to the total internal energy E by

$$\rho E = \frac{p}{\gamma - 1} + \frac{1}{2} \cdot \rho \cdot (u^2 + v^2 + w^2).$$

\mathbf{F} is the inviscid flux, given by

$$\mathbf{F} = (\rho \mathbf{u}, \rho \mathbf{u} \mathbf{u} + p \hat{\mathbf{x}}, \rho \mathbf{u} \mathbf{v} + p \hat{\mathbf{y}}, \rho \mathbf{u} \mathbf{w} + p \hat{\mathbf{z}}, \rho \mathbf{u} (E + p))^T.$$

$S(W)$ is any source term and \mathbf{F}_v is the viscous flux, given by

$$\mathbf{F}_v = (0, \boldsymbol{\tau}_x, \boldsymbol{\tau}_y, \boldsymbol{\tau}_z, u \boldsymbol{\tau}_x + v \boldsymbol{\tau}_y + w \boldsymbol{\tau}_z + k \nabla T)^T$$

where $\boldsymbol{\tau}_x$, $\boldsymbol{\tau}_y$ and $\boldsymbol{\tau}_z$ are the stress vectors in the directions x , y and z and k is the thermal conductivity.

3 Dual Time Stepping Approach

3.1 RK/Implicit Smoother Scheme for Dual Time Stepping with Backward Differentiating

When solving (1) with a dual-time stepping method we solve the equation set

$$\frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} + \nabla \mathbf{F}(W) = \nabla \mathbf{F}_v(W) + S(W) \quad (2)$$

where τ is the pseudo-time. We integrate in space and apply the Gauss theorem for a control volume (grid cell) to get a semi-discrete scheme

$$\frac{\partial W}{\partial \tau} = -R^*(W) \quad (3)$$

where

$$R^*(W) = \frac{\partial W}{\partial t} + R \quad (4)$$

and the residual R is given by

$$R = \frac{1}{V} \oint_S (\mathbf{F} - \mathbf{F}_v) \cdot d\mathbf{S} \approx \frac{1}{V} \sum_{\text{cell faces}} (\mathbf{F} - \mathbf{F}_v) \cdot d\mathbf{S} - S(W). \quad (5)$$

We approximate the physical time derivative with respect to the physical time in (3) by a second order backward difference formula (BDF2) -

$$\frac{\partial W}{\partial t} \approx \frac{3W^{n+1} - 4W^n + W^{n-1}}{2\Delta t}. \quad (6)$$

For each physical time step we solve (3) using a low-storage Runge–Kutta scheme for the pseudo-time marching -

$$\begin{aligned} W^{(0)} &= W^n \\ W^{(k+1)} &= W^{(0)} - \alpha_{k+1} \Delta \tau R^*(W^{(k)}) \quad ; \quad k = 0 \dots p-1 \\ W^{n+1} &= W^{(p)} \end{aligned} \quad (7)$$

where p is the number of stages. For a second order upwind scheme we use optimized coefficients from [33]—either $\alpha_k = [0.1918, 0.4929, 1]$ for a three stage RK or $\alpha_k = [0.0695, 0.1602, 0.2898, 0.506, 1]$ for a five stage RK. Following [18] and [25] linearization of (3) with respect to the pseudo time yields

$$\left(I + \frac{\Delta\tau}{V} \sum_{\text{cell faces}} A \cdot d\mathbf{S} + \Delta\tau \frac{\partial S}{\partial W}\right) \Delta W = \Delta\tau R^* \quad (8)$$

where A is the flux Jacobian matrix and $\frac{\partial S}{\partial W}$ is the Jacobian of the source term.

We introduce a relaxation parameter ϵ multiplying $\frac{\Delta\tau}{V} \sum A \cdot d\mathbf{S}$. Usually, $\epsilon \leq 1$. $\epsilon > 1$ is an over-relaxation and can be used for very stiff problems. We then express the flux Jacobian as the sum of two matrices, A^+ with non-negative eigenvalues and A^- with non-positive eigenvalues, defined as $A^\pm = \frac{1}{2}(A \pm |A|)$. We leave A^+ on the left side multiplying ΔW of the local cell and move A^- to the right side to multiply ΔW of the neighbor cells. This yields a set of linear equations for the smoothed ΔW which replaces the unsmoothed residuals R^* in Eq. (7) -

$$\left(I + \epsilon \frac{\Delta\tau}{V} \sum_{\text{cell faces}} A^+ \cdot d\mathbf{S} + \Delta\tau \frac{\partial S}{\partial W}\right) \Delta W_{\text{local}} = \Delta\tau R^* - \epsilon \frac{\Delta\tau}{V} \sum_{\text{cell faces}} A^- \cdot \Delta W_{NB} \cdot d\mathbf{S}. \quad (9)$$

This linear system for the smoothed residual can be solved using an iterative solver such as Gauss-Seidel or red-black iterations (RB can be used for parallel computing). The solver is explicit and we use the resulting ΔW as smoothed residuals. We perform only a few iterations, typically 2–3, in each RK step.

In [30] the conservative equation (2) was preconditioned and was solved in semi-discrete form using the scheme

$$P^{-1} \frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} + \nabla \mathbf{F}(W) = \nabla \mathbf{F}_v(W) - h_x \frac{\partial}{\partial x} \left(P^{-1} |P A| \frac{\partial W}{\partial x} \right). \quad (10)$$

Multiplying $\frac{\partial W}{\partial \tau}$ by P^{-1} changes the eigenvalues of the system and hence the convergence rate but not the steady state solution. The last term is the artificial viscosity and the matrix P is a low-Mach number preconditioner which controls the consistency of the scheme for low Mach number flow as $M \rightarrow 0$ (see [2,28]). Since in the approximation of the derivative with respect to physical time (6), W^{n+1} is unknown, it was approximated by $W^{(k+1)}$ so

$$R^*(W^{(k)}) \approx R(W^{(k)}) + \frac{3W^{(k+1)} - 4W^n + W^{n-1}}{2\Delta t}. \quad (11)$$

This yields the point implicit scheme:

$$\left(I + \alpha_k c_t \frac{\Delta\tau}{\Delta t} P\right) W^{(k+1)} = W^{(0)} - \alpha_k \Delta\tau P \left\{ R(W^{(k)}) + \frac{3W^{(k)} - 4W^n + W^{n-1}}{2\Delta t} \right\} + \alpha_k c_t \frac{\Delta\tau}{\Delta t} P W^{(k)}. \quad (12)$$

where $c_t = \frac{3}{2}$. Later, in [25], the matrix P^{-1} multiplying $\frac{\partial W}{\partial \tau}$ was replaced by any L^{-1} convergence acceleration operator (e.g. multigrid or residual smoothing). Since $\frac{\partial W}{\partial t}$

is considered to be a source term whose Jacobian is $\frac{c_t}{\Delta t}$, the smoother operator takes the form

$$\begin{aligned} & \left(I + \epsilon \frac{\Delta \tau}{V} \sum_{\text{cell faces}} A^+ \cdot d\mathbf{S} + c_t \frac{\Delta \tau}{\Delta t} + \Delta \tau \frac{\partial S}{\partial W} \right) \Delta W_{\text{local}} \\ &= \Delta \tau R^*(W^{(k)}) - \epsilon \frac{\Delta \tau}{V} \sum_{\text{cell faces}} A^- \cdot \Delta W_{NB} \cdot d\mathbf{S} \end{aligned} \quad (13a)$$

where

$$R^*(W^{(k)}) = \frac{1}{V} \sum_{\text{cell faces}} (\mathbf{F} - \mathbf{F}_v) \cdot d\mathbf{S} - \frac{3W^{(k)} - 4W^n + W^{n-1}}{2\Delta t} - S(W^{(k)}). \quad (13b)$$

Equations (13a) and (13b) defined the standard scheme for dual time stepping with RK/implicit smoother.

Instead, we adopt the strategy of [18] and [25] for low Mach treatment by using a modified speed of sound c^* in the flux Jacobian matrix in the implicit smoother operator. We apply a Roe-like scheme with artificial viscosity with the same modified flux Jacobian $A(c^*)$ where the artificial viscosity flux is given by $f_{ad} = -|A(c^*)| \frac{\partial W}{\partial x}$ (see also [13]). Hence, we set the preconditioning matrix $P = I$ and (12) can be written as

$$W^{(k+1)} = W^{(0)} - \alpha_k \Delta \tau \frac{\tilde{R}^*(W^{(k)})}{1 + \alpha_k c_t \frac{\Delta \tau}{\Delta t}} \quad (14)$$

where

$$\tilde{R}^*(W^{(k)}) = R(W^{(k)}) + \frac{3W^{(0)} - 4W^n + W^{n-1}}{2\Delta t}. \quad (15)$$

We identify $(1 + \alpha_k c_t \frac{\Delta \tau}{\Delta t})^{-1}$ as the smoother operator for the scalar equation and keep the same structure for the residual smoother. We define the new residual smoother scheme as

$$\begin{aligned} & \left(I + \epsilon \frac{\Delta \tau}{V} \sum_{\text{cell faces}} A^+ \cdot d\mathbf{S} + \alpha_k c_t \frac{\Delta \tau}{\Delta t} + \Delta \tau \frac{\partial S}{\partial W} \right) \Delta W_{\text{local}} \\ &= \Delta \tau \tilde{R}^*(W^{(k)}) - \epsilon \frac{\Delta \tau}{V} \sum_{\text{cell faces}} A^- \cdot \Delta W_{NB} \cdot d\mathbf{S}. \end{aligned} \quad (16a)$$

where

$$\tilde{R}^*(W^{(k)}) = \frac{1}{V} \sum_{\text{cell faces}} (\mathbf{F} - \mathbf{F}_v) \cdot d\mathbf{S} - \frac{3W^{(0)} - 4W^n + W^{n-1}}{2\Delta t} - S(W^{(k)}). \quad (16b)$$

To summarize the differences between the standard [Eqs. (13a), (13b)] and the non-standard scheme [Eqs. (16a), (16b)] -

1. In the right hand side of the standard scheme (13b) the approximation of the derivative with respect to physical time contains $W^{(k)}$ and in the new scheme (16b) $W^{(k)}$ is replaced by $W^{(0)}$.
2. In the smoother operator of the new scheme (16a) we have α_k multiplying $c_t \frac{\Delta \tau}{\Delta t}$.

To validate the new dual time stepping scheme, included within the RK/implicit smoother, we consider the solution of the Riemann problem for Sod's shock tube problem [21].

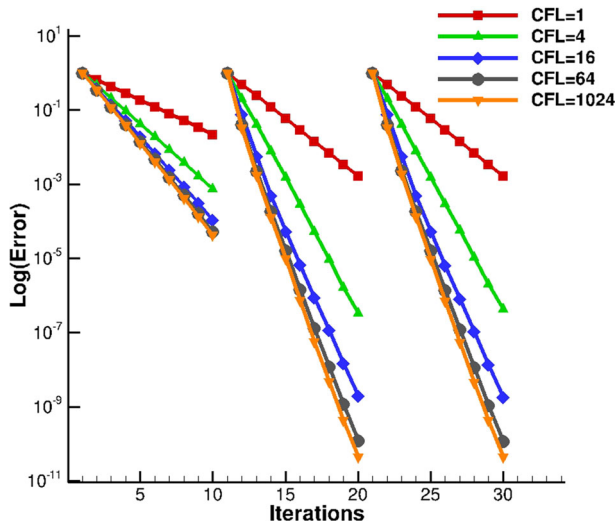


Fig. 1 Convergence history for series of CFL numbers for three physical time steps

For this problem the initial conditions are

$$\begin{cases} p = 1 \text{ PSI and } T = 231.11\text{K on the right side of the tube} \\ p = 10 \text{ PSI and } T = 288.89\text{K on the left side} \end{cases} \quad (17)$$

with Mach number zero on both sides.

For the a discretization of the one dimensional hyperbolic equation $u_t + au_x = 0$ the CFL number is defined as $a \frac{\Delta t}{\Delta x}$. For Euler equations we take the maximum eigenvalue $-a = |u| + c$ where u is velocity and c is the speed of sound. The convergence for a series of CFL numbers is presented in Fig. 1 for three physical time steps. Improved convergence is obtained as the CFL number increases until reaching an asymptotic convergence.

The computations demonstrate that the new approach converge faster and more stable especially at very low Mach numbers. Figure 2 shows a comparison between the convergence histories of the standard and the new methods for the Shu–Osher problem (see details in [20]). For ten pseudo-time iterations with CFL=64, the results are practically the same but the new method converges much quicker. It reaches the same error level after about five iterations so it saves almost half of the CPU time.

3.2 Dual Time Stepping with DIRK

The algorithm for dual-time stepping with a DIRK scheme for the physical time marching is based on [4]. We want to solve the semi-discrete equation of the form

$$\frac{dW}{dt} = -R(W). \quad (18)$$

The DIRK scheme is given by

$$W^{(i)} = W^n - \Delta t \sum_{j=1}^i a_{ij} R(W^{(j)}) \quad i = 1..s \quad (19a)$$

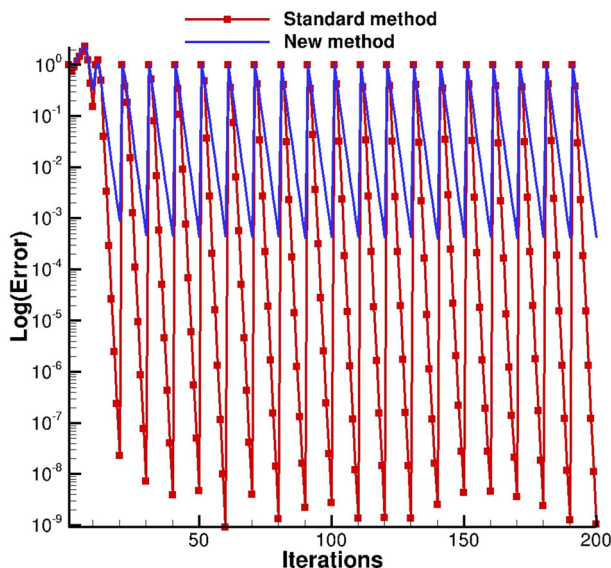


Fig. 2 Convergence history comparison between the standard dual time-stepping method and the new dual time stepping method for the one-dimension Shu–Osher problem with $CFL = 64$

$$W^{n+1} = W^n - \Delta t \sum_{i=1}^s b_i R(W^{(i)}) \quad (19b)$$

where $W^n = W(t_n)$, $W^{(i)}$ is the solution at the i -th internal RK stage and s is the number of RK stages. $a_{11} = 0$ and so $W^{(1)} = W^n$. Furthermore, $a_{ii} \neq 0 \quad \forall i > 1$ and $b_i = a_{is}$ hence $W^{n+1} = W^{(s)}$. We then write (19a) as

$$R^*(W^{(i)}) = R(W^{(i)}) - \frac{W^n - W^{(i)}}{a_{ii} \Delta t} + \frac{1}{a_{ii}} \sum_{j=1}^{i-1} a_{ij} R(W^{(j)}) = 0. \quad (20)$$

Equation (20) is solved for $W^{(i)}$ with pseudo-time stepping

$$\frac{dW^{(i)}}{d\tau} = -R^*(W^{(i)}). \quad (21)$$

We solve (21) with a low-storage RK with an implicit smoother, where the residual smoother operator is defined similar to the new scheme (16a) by

$$\begin{aligned} & \left(I + \epsilon \frac{\Delta \tau}{V} \sum_{\text{cellfaces}} A^+ \cdot dS + \frac{1}{a_{ii}} \frac{\Delta \tau}{\Delta t} + \Delta \tau \frac{\partial S}{\partial W} \right) \Delta W_{\text{local}} \\ &= \Delta \tau R^*(W^{(k)}) - \epsilon \frac{\Delta \tau}{V} \sum_{\text{cellfaces}} A^- \cdot \Delta W_{NB} \cdot dS. \end{aligned} \quad (22)$$

3.3 DIRK Coefficients

We choose the DIRK3 scheme from [11]. The non-zero coefficients a_{ij} are given by

$$\begin{aligned} a_{21} &= a_{22} = a_{33} = \frac{1}{2} + \frac{\sqrt{3}}{6} \\ a_{32} &= -\frac{1}{12a_{22}(2a_{22} - 1)} \\ a_{31} &= 1 - a_{32} - a_{33} \end{aligned} \quad (23)$$

We recall that $b_i = a_{is}$ while the coefficients \hat{b}_i for the lower order embedded scheme are given by

$$\begin{aligned} \hat{b}_1 &= \frac{5}{12} + \frac{\sqrt{3}}{12} \\ \hat{b}_2 &= \frac{9}{12} + \frac{\sqrt{3}}{12} \\ \hat{b}_3 &= -\frac{2}{12} - \frac{2\sqrt{3}}{12} \end{aligned} \quad (24)$$

4 Accuracy and Stability Analysis

4.1 Dual Time Steps

In [1] a linear stability of an implicit time integration using a dual-time stepping is given. They analyze several temporal schemes and show that the inner iterations in pseudo-time introduces a secondary convergence criterion to the numerical method. Thus, we consider the dual time step equation

$$\frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} = R(W) \quad (25)$$

where t is the physical time and τ is the pseudo time. For the physical time step $n+1$, We solve for the solution W^{n+1} using Runge–Kutta iterations in the pseudo time. The combined iterative process is given by

$$W^{k+1,n+1} = F_{k \mapsto k+1}(W^{k,n+1}; W^n, W^{n-1}, W^{n-2}, \dots) \quad (26)$$

where $k = 1..K$ is the number of pseudo-time iterations and n is the physical time step. After K pseudo-time steps $W^{n+1} = W^{K,n+1}$ so (26) should converge to

$$W^{n+1} = F_{n \mapsto n+1}(W^n, W^{n-1}, W^{n-2}, \dots, W^{n-N+1}). \quad (27)$$

where N is the number of backward time steps needed. The amplification factor σ for schemes with $N = 1$ (e.g. Runge–Kutta and first order Euler) is defined as $W^{n+1} = \sigma W^n$. For schemes with $N > 1$ the amplification factor is defined as $W^{n+1} = \sigma^m W^{n-m+1} \forall m \leq N$.

In order to evaluate the amplification factor for the dual time step algorithm defined in (26), we consider the linear equation (25) with $R(W) = \lambda W$. We then need to find the solution for σ of the non-linear equation

$$\sigma = F_{n \mapsto n+1}(1, 1/\sigma, 1/\sigma^2, \dots) \quad (28)$$

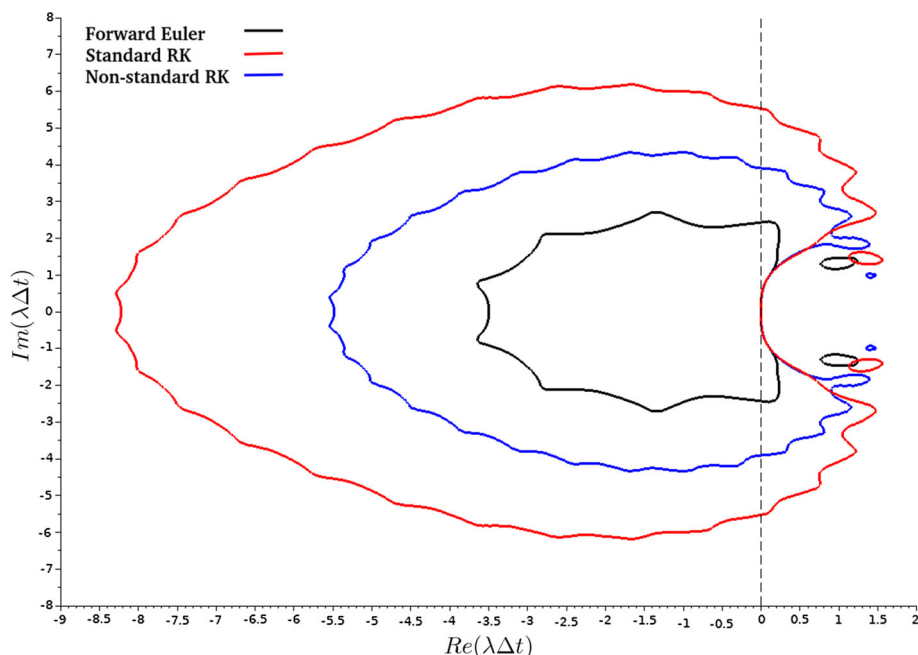


Fig. 3 Comparison stability region for 3 BDF2 schemes with different pseudo-time schemes. $|\sigma| \leq 1$ is region inside the closed curves

From a dimensional analysis, the amplification factor obeying (28) is a function of the ratio between the physical time step size and the pseudo time step size $\frac{\Delta t}{\Delta \tau}$, the number of pseudo time iterations K and the phase $\lambda \Delta t$. Following [1] we explore the stability region where $|\sigma| \leq 1$ in the complex plane of $z = \lambda \Delta t$ for $\frac{\Delta t}{\Delta \tau} = 1$ and a typical value of inner pseudo-time iterations $K = 10$.

4.2 Analysis of BDF2 Scheme for Physical Time

We compare the stability region for three schemes in pseudo-time - 1. Forward Euler scheme as in [1], 2. Standard Runge–Kutta scheme defined in Eq. (13b) and 3. Non-standard Runge–Kutta defined in Eq. (16b). Note, that when using pseudo time Runge–Kutta with only one stage with $\alpha = 1$, the three methods are the same. The stability region, $|\sigma| \leq 1$, can be seen in Fig. 3. We can see that the stability region of the standard method is larger compared to the stability region of the non-standard method so larger time steps can be used. However, this analysis can not explain the better convergence observed for the non standard method. For both RK schemes the stability region is wider than the stability region of forward Euler.

4.3 Analysis of Runge–Kutta Methods for Physical Time

In the case of full convergence in pseudo-time, the intermediate solutions $W^{(i)}$ obey Eq. (20). In this case Eq. (20) can be multiplied by any constant without changing the solution. Otherwise multiplication by constants changes the solution and the convergence properties of the scheme. In [1], a slightly different algorithm was adopted in which the scheme for $W^{(i)}$ is multiplied by a_{ii} so that the modified scheme is given by

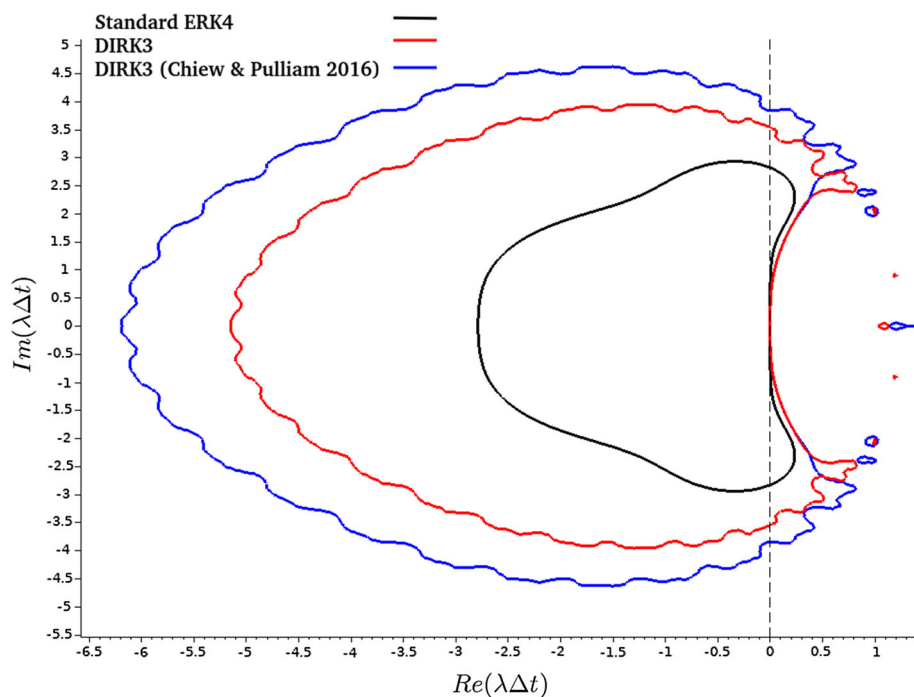


Fig. 4 Comparison stability region for three Runge–Kutta schemes. $|\sigma| \leq 1$ region is inside the closed curves. The wiggly nature of the DIRK stability region curves is due to the inner pseudo-time iterations

$$R^*(W^{(i)}) = a_{ii} R(W^{(i)}) - \frac{W^n - W^{(i)}}{\Delta t} + \sum_{j=1}^{i-1} a_{ij} R(W^{(j)}) = 0. \quad (29)$$

In Fig. 4 we present the stability region for three methods—1. Standard explicit Runge–Kutta (ERK) of order four, 2. DIRK3 and 3. [1] modified DIRK3 [Eq. (29)]. We see that both versions of DIRK3 have a wider stability region with respect to the stability region of ERK. The algorithm defined by Eq. (29) has not yet been tested for the Navier–Stokes equation.

4.4 Rate of Convergence

We stress that the level curves $|\sigma| = 1$ in Figs. 3 and 4 only give the domain of stability. A larger domain implies stability for larger time steps. However, this does not imply faster convergence to a steady state. For example, Crank–Nicolson is stable for the entire left half plane and so is unconditionally stable. However, $|\sigma| = 1$ everywhere and so the solution does not decay to a steady state. For faster convergence rates we need to investigate the level curves $|\sigma| < 1$. For example, if the domain $|\sigma| = 0.8$ is larger, this implies a convergence rate of 0.8 for larger time steps.

In Figs. 5 and 6 we plot the level curves $|\sigma| = \{1, 0.9, 0.7, 0.5, 0.3\}$ for BDF2 and DIRK3 respectively. We see that $|\sigma| = 1$ give a larger domain for DIRK3. However for DIRK3 $|\sigma| = 0.7$ no longer contains the imaginary axis for small real λ . In general, from numerical tests it seems that BDF2 converge faster.

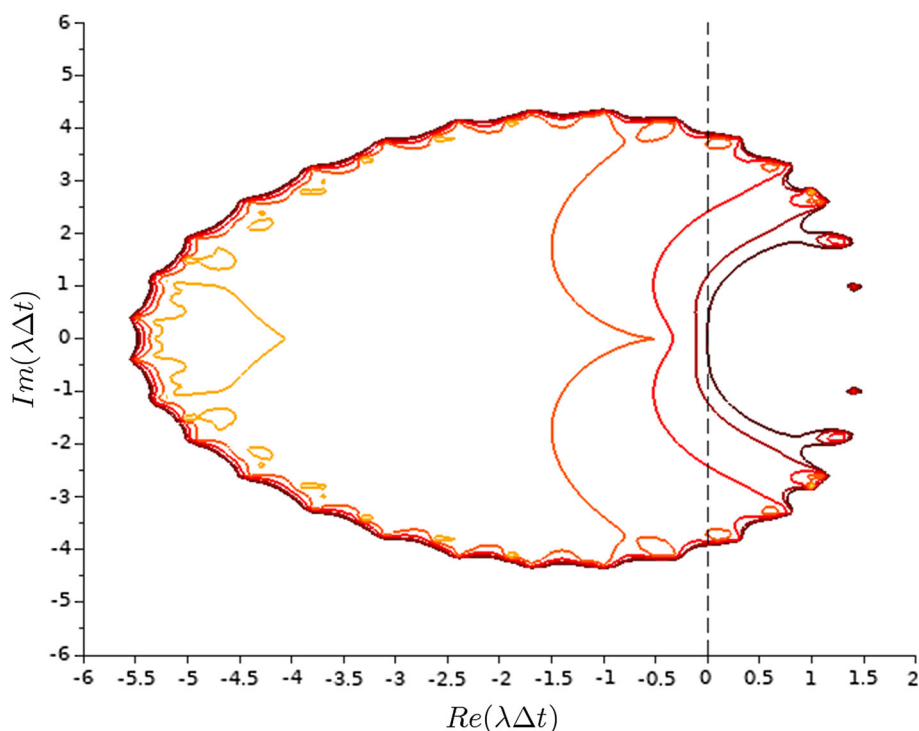


Fig. 5 Amplification factor curves lines for BDF2 scheme. Level curves $|\sigma| = \{1, 0.9, 0.7, 0.5, 0.3\}$ are plotted

4.5 Influence of the Number of Pseudo Time Iterations

We now investigate the influence of the number of pseudo-time iterations K . Without proof, the solution after K pseudo-time iteration is a numerical approximation for u^{n+1} . The convergence plots for ESDIRK of order four [12] scheme in physical time for linear equation with $\lambda = -1$ for $K = 1.4$ can be seen in Fig. 7. The resulting order are approximately $p(K) = \max(K + 1, 4)$. Based on numerical tests for ESDIRK and other DIRK schemes, we conjecture that for DIRK of order $p = P$, $p(K) = \max(K + 1, P)$. Since $b_i = a_{i,s}$, for full convergence in pseudo time, $W^{n+1} = W^{(s)}$ where $W^{(s)}$ is the solution of the last intermediate RK stage s . For insufficient number of pseudo-time iterations K , this is not true. If we still assume that $W^{n+1} = W^{(s)}$, the result is an order reduction for small K so $p(K) = \max(K, P)$. Convergence plots for $K = 1..4$ under this assumption can be seen in Fig. 8.

5 Adaptive Time Steps

For a given DIRK scheme of order p , the embedded scheme is a scheme of lower order, $p - 1$ with the same a_{ij} but \hat{b}_i instead of b_i . Hence, we don't need another evaluation of the residual to find the lower order approximation. We note that this notation is slightly different than that used for explicit RK schemes where an embedded scheme means one using fewer stages than the full scheme. Let W^{n+1} be the solution of the given DIRK scheme after one time step of size Δt and let \hat{W}^{n+1} be the solution of the embedded scheme, both with initial condition W^n . We therefore have:

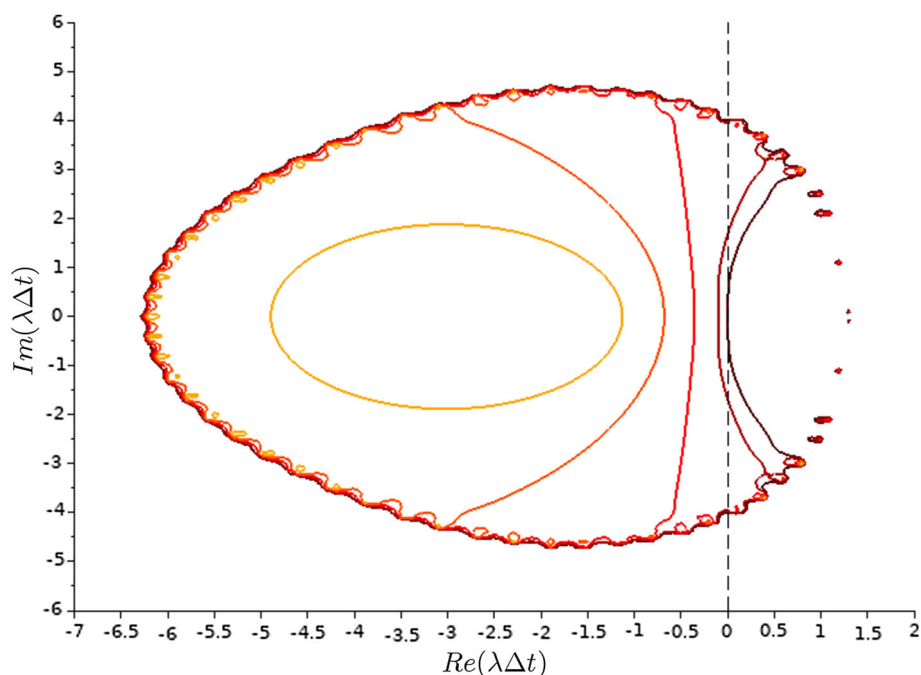


Fig. 6 Amplification factor curves lines for DIRK3 scheme. Level curves $|\sigma| = \{1, 0.9, 0.7, 0.5, 0.3\}$ are plotted

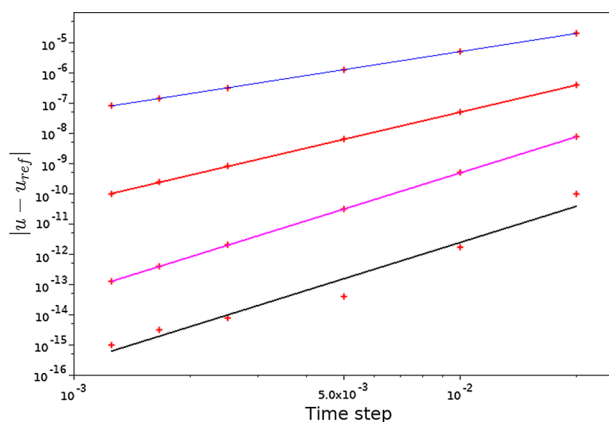


Fig. 7 Convergence plots for ESDIRK4 scheme in physical time for a series of number of pseudo-time iterations K . The resulting orders are 1.992 for $K=1$, 2.984 for $K=2$, 3.978 for $K=3$ and 3.986 for $K=4$

$$W^{n+1} = W + c\Delta t^p \quad (30)$$

and

$$\hat{W}^{n+1} = W + \hat{c}\Delta t^{p-1}. \quad (31)$$

For Δt sufficiently small, $c\Delta t^p$ is negligible compared with $\hat{c}\Delta t^{p-1}$. Therefore, subtracting Eq. (30) from (31) yields

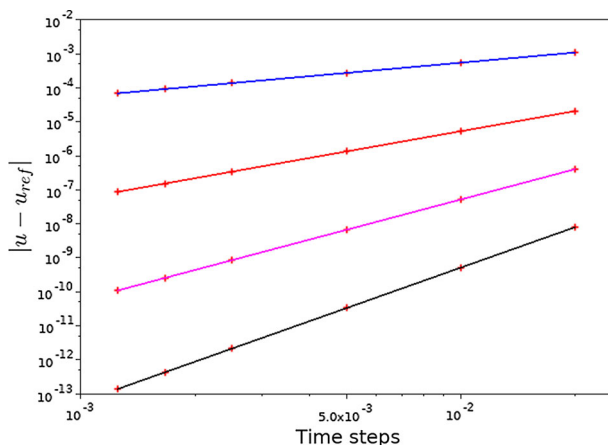


Fig. 8 Convergence plots for ESDIRK4 scheme in physical time, assuming $W^{n+1} = W^{(s)}$, for a series of number of pseudo-time iterations K . The resulting orders are 0.998 for $K = 1$, 1.981 for $K = 2$, 2.973 for $K = 3$ and 3.969 for $K = 4$

$$\hat{W}^{n+1} - W^{n+1} = \hat{c} \Delta t^{p-1}. \quad (32)$$

Hence,

$$\frac{\|\hat{W}^{n+1} - W^{n+1}\|}{\|\hat{W}^n - W^n\|} = \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^{p-1}. \quad (33)$$

We want $\|\hat{W}^{n+1} - W^{n+1}\|$ bounded relative to the norm of the solution $\|W\|$ such that

$$\|\hat{W}^{n+1} - W^{n+1}\| = \|\hat{W}^n - W^n\| \cdot \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^{p-1} \leq \epsilon \|W^n\|. \quad (34)$$

From (34) we get

$$\Delta t_{n+1} \leq \Delta t_n \cdot \left(\frac{\epsilon \|W^n\|}{\|\hat{W}^n - W^n\|} \right)^{\frac{1}{p-1}}. \quad (35)$$

We use the PI-like controller [5]. The time step size for the next time step is thus given by

$$\Delta t_{n+1} = \max \left(\min \left(\Delta t_n \cdot \left(\frac{TOL \|W^n\|}{\|\hat{W}^n - W^n\|} \right)^{\frac{1}{p-1}}, C_2 \Delta t_n \right), C_1 \Delta t_n \right) \quad (36)$$

The TOL parameter determines the required accuracy. A large TOL enables large time steps with low accuracy while smaller TOL yields a more accurate solution with smaller time steps (so an increased CPU time).

6 Examples

We consider three test cases, a one-dimensional case and two 2-dimensional problems—the first one is a pure subsonic and the second one with moving shock.

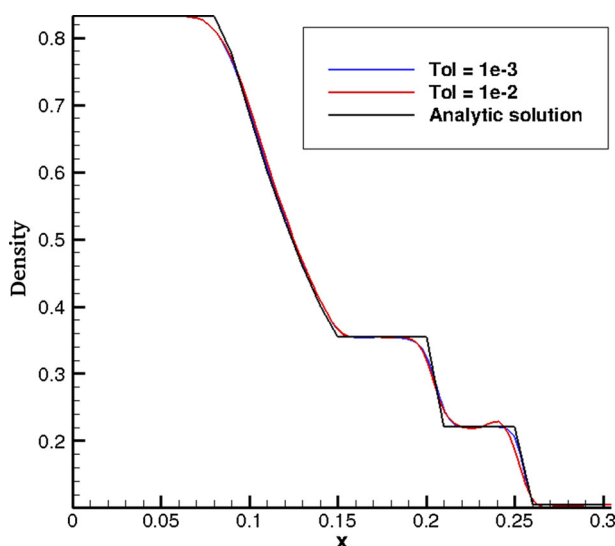


Fig. 9 Comparison solutions for two values of convergence tolerances and the analytic solution at $t=200$ ms

6.1 Riemann Problem (Sod's Shock Tube Problem)

Sod's shock tube problem was defined in (17). Figure 9 presents the solution for two values of the tolerance parameter (TOL) and compares this result also to the analytic solution. We see that the solution gets better as TOL becomes smaller especially in the vicinity to the shock-wave. For $TOL = 10^{-3}$, Fig. 10 present the evolution of the selected time steps for two different initial time steps. The shock-wave is moving with a constant speed along a uniform mesh. We indeed see that we reach constant time steps without a dependency on the initial time step.

For the Riemann problem, the CPU time with the adaptive time steps was about equal to the optimal time step with the BDF method. However, unlike in the BDF method, the algorithm reaches the final time step automatically.

6.2 Subsonic Example—Rayleigh–Taylor Instability

The Rayleigh–Taylor instability case was proposed in [35] and is a pure subsonic example. The problem describes a two dimensional inviscid flow with $x \in [0, 1/4]$ and $y \in [0, 1]$ in which a heavy fluid accelerates into a light one under the influence of gravitation. The acceleration effect is achieved by adding a linear source terms ρg to the y -direction momentum equation and $\rho v g$ to the energy equation with $g = 1$ (m/s^2 for dimensional codes). The initial conditions divide the domain into two layers of fluids. A heavy fluid is in the bottom half, and a light fluid is in the top part. A small velocity perturbation is also added:

$$\begin{aligned} \rho &= 2; & u &= 0; & v &= -0.025c \cos(2\pi x); & P &= 2y + 1 & \text{when } y \leq 1/2 \\ \rho &= 1; & u &= 0; & v &= -0.025c \cos(2\pi x); & P &= y + 3/2 & \text{when } y > 1/2 \end{aligned} \quad (37)$$

where $c = \sqrt{\gamma p / \rho}$ and $\gamma = 5/3$. The boundary conditions at $x=0$ and $x=1/4$ along the y axis was set to $\frac{\partial \rho}{\partial n} = \frac{\partial P}{\partial n} = 0$ and $\mathbf{v} \cdot \mathbf{n} = 0$. At $y=0$ along x axis the boundary conditions

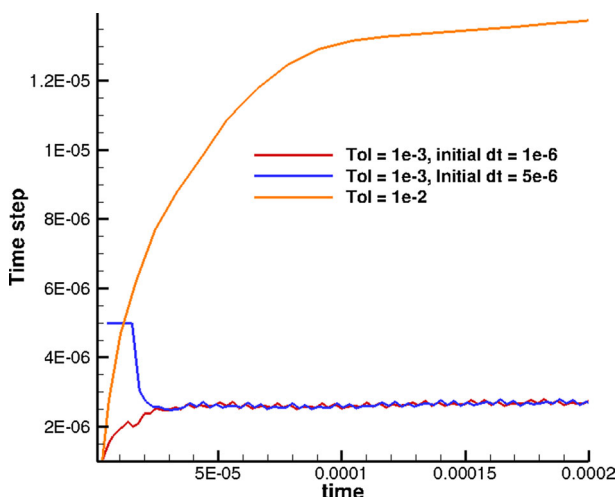
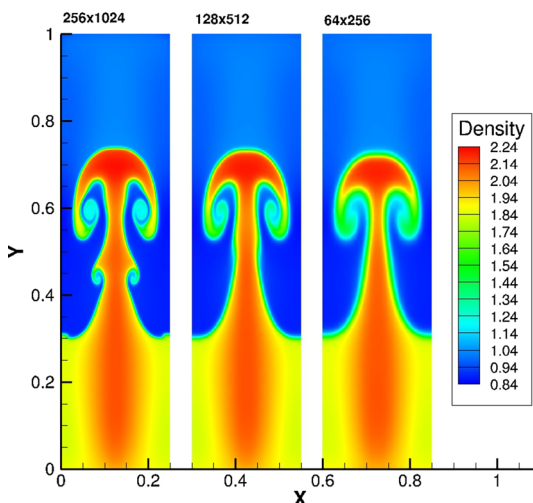


Fig. 10 Evolution of the selected time steps for different tolerances and initial time steps (Sod's shock tube problem)

Fig. 11 Solutions at time $t = 1.95$ using the DIRK method with accuracy tolerance of $TOL = 2 \times 10^{-4}$ for three mesh sizes



imposed are $\rho = 2$, $P = 1$ and $\mathbf{v} = 0$ and At $y = 1$ along x axis the boundary conditions imposed are $\rho = 1$, $P = 2.5$ and $\mathbf{v} = 0$. The Rayleigh–Taylor instability is characterized by a fingering nature. Since the problem is inviscid, the exact details of the complex pattern is controlled by the numerical diffusion of the scheme [36]. In Fig. 11 a comparison between three grids with different mesh sizes using the DIRK scheme with $TOL = 2 \times 10^{-4}$ is presented. It can be seen that as the grid become denser, more patterns appears. Most of the publications which consider and present results for the Rayleigh–Taylor instability test case use high-order spatial schemes, for example the Weighted Compact Nonlinear Scheme (WCNS) [36], WENO [19] and WENO-Z [34]. In this paper we use a second order scheme. The spatial accuracy is not in the main purpose of this paper, instead, we are exploring the temporal scheme.

Fig. 12 Solutions at time $t = 1.95$ using the BDF method for four time steps: from the left: 1×10^{-2} , 5×10^{-3} , 2.5×10^{-3} and 1×10^{-3}

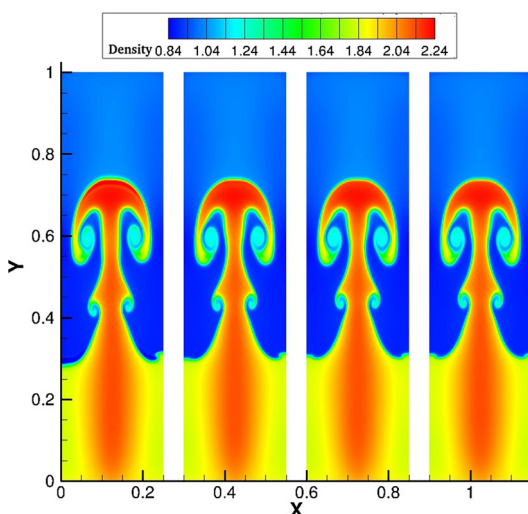


Table 1 Summary of numerical tests for RT instability case

Method	Accuracy tolerance (TOL)	Average time step	Mesh size	Number of time steps	Work
BDF	–	0.01	256×1024	195	195
BDF	–	0.005	256×1024	390	390
BDF	–	0.0025	256×1024	780	780
BDF	–	0.001	256×1024	1950	1950
DIRK	2×10^{-3}	0.0156	256×1024	125	250 (375)
DIRK	1×10^{-3}	0.0105	256×1024	185	370 (555)
DIRK	5×10^{-4}	0.0071	256×1024	275	550 (825)
DIRK	2×10^{-4}	0.0043	256×1024	457	914 (1371)
DIRK	2×10^{-4}	0.00615	128×512	317	634 (951)
DIRK	2×10^{-4}	0.009	64×256	216	512 (768)

We present, in Fig. 12, the solution at time $t = 1.95$ for a series of four times steps using the BDF method. For the largest time step, 1×10^{-2} , the solution is seem asymmetric and wiggles can be seen close to the top part of the front between the two fluids. For a smaller time step with 5×10^{-3} the wiggles disappear but the solution is still slightly asymmetric. The last two time steps are 2.5×10^{-3} and 1×10^{-3} . For these time steps the solution is sufficiently similar and symmetric to the eyes. Therefore, we refer the larger between the two, $dt = 2.5 \times 10^{-3}$, as the reference time step for the BDF method. A similar test was done for the DIRK method but for a series of four values of the accuracy parameter TOL . For the largest value of 2×10^{-3} the solution is very asymmetric. For the two last values of TOL , 5×10^{-4} and 2×10^{-4} , the solution is sufficiently similar and symmetric, so we refer $TOL = 5 \times 10^{-4}$ as the reference tolerance solution for DIRK.

The summary of these 10 tests is presented in Table 1. The table lists the number of time steps required to reach the final time for each numerical test (constant for BDF method and average for DIRK), the total number of time steps and the number of work units. For the BDF method the work units is the number of time steps. For the n-stage DIRK method there are n

Fig. 13 Solutions at time $t = 1.95$ using adaptive time steps DIRK method for four accuracy tolerances: from the left: 2×10^{-3} , 1×10^{-3} , 5×10^{-4} and 2×10^{-4}

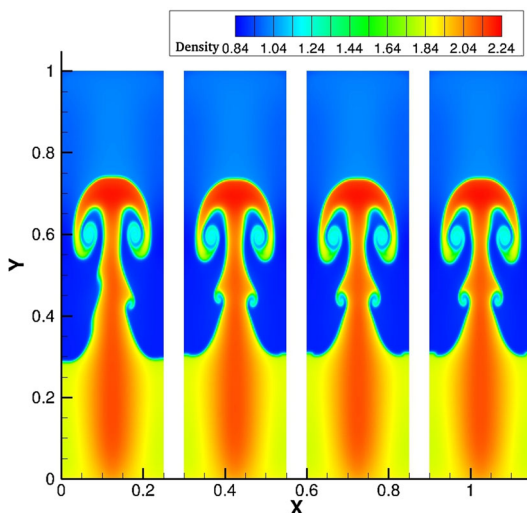


Fig. 14 Schematic description of the initial conditions for the shockwave–bubble interaction problem

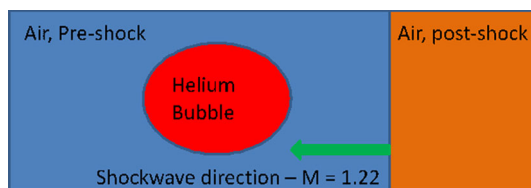


Table 2 Initial conditions for the shockwave bubble interaction

	Velocity	Pressure	Density	Gas
Post-shock	0.3118	1.57	1.3764	Air
Pre-shock	0	1	1	Air
Bubble	0	1	0.1358	Helium

evaluations of the residuals. However, there are only $n - 1$ stages with a full solve (residual smoothing and solution update) of the intermediate solutions $u^{(j)}$. These $n - 1$ stages occupied most of the CPU time so for DIRK the presented work units are $n - 1$ times the number of time steps (n times the number of time steps is also presented in brackets). In terms of work units the reference solution of the adaptive DIRK (Fig. 13) costs about 30% less than the reference solution with the BDF method (Fig. 12).

6.3 Shockwave–Bubble Interaction

The shockwave–bubble interaction is a two-dimensional problem describing the interaction of a shock-wave with a bubble. Details and experimental results can be found in [6]. The schematic description of the problem is presented in Fig. 14. The initial condition are presented in Table 2.

The properties of air are $\gamma = 1.4$ and the molecular weight is 29 gram/mole. The properties of helium are $\gamma = 1.67$ and the molecular weight is 4 gram/mole.

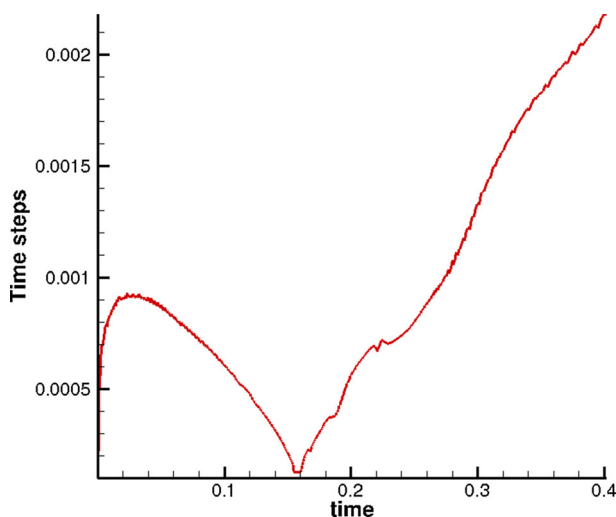


Fig. 15 Time steps evolution for the shockwave–bubble interaction problem. The decreasing of the time step before the shock-wave/bubble impingement due to the grid clustering can be seen as well as the time step increasing after the shock-wave leaves the bubble

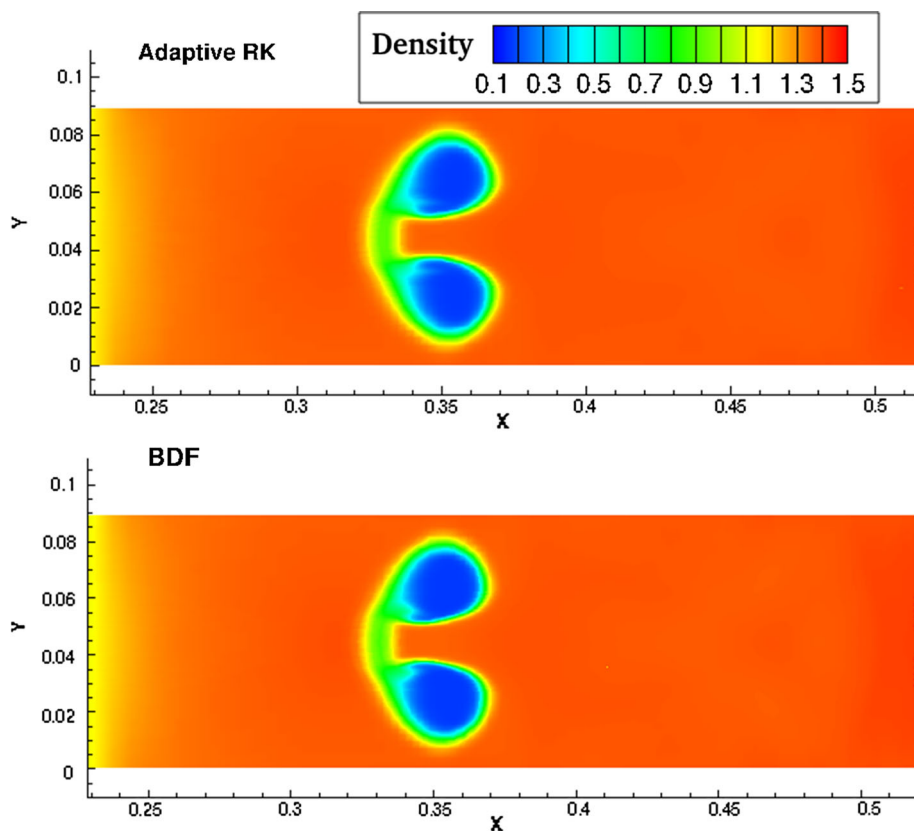


Fig. 16 Comparison of the results (density) between adaptive time steps DIRK and BDF at time $t = 0.03$ s

This problem is characterized by strongly varying time scales in physical time. In Fig. 15 we present the time step evolution. Before the bubble, the grid itself is getting denser and so the time step decreases in order to keep a constant error. After exiting from the bubble, the grid becomes sparser and the time step increases. The standard time steps for this problem with the BDF method is about 10^{-4} s. and the CPU time gain for RK was approximately a factor of four. In Fig. 16 we display a comparison between the solutions (density) of BDF and RK with adaptive time steps. We see a good agreement between the two solutions while the adaptive RK scheme is much more efficient.

7 Conclusions

Linear analysis and numerical tests shows that the DIRK method requires more inner iterations in pseudo-time than the BDF2 for given convergence tolerance. In the future, improvement of the internal convergence will improve the efficiency of the method. Despite this disadvantage we have presented an adaptive time stepping technique based on the dual time steps method combined with a high-order implicit Runge–Kutta method in physical time and RK/implicit smoother for the pseudo time integration. This method gives a high gain of CPU time for problems with a large variation of time scales. It also replaces the process of time-steps sensitivity check needed to find the optimal time step for BDF. The most important parameter is the tolerance parameter, TOL , which determines the required accuracy. The CPU time is a direct outcome of the desired accuracy.

References

- Chiew, J. J., Pulliam, T. H.: Stability analysis of dual-time stepping. In: 46th AIAA Fluid Dynamics Conference (2016)
- Darmofal, D., Van Leer, B.: Local preconditioning: manipulating mother nature to fool father time. In: Caughey, D.A., Hafez, M. (eds.) *Advances and Prospects in Computational Aerodynamics*, Computing the Future II. John Wiley and Sons (1998)
- de Brito Alves, L. S.: Dual time stepping with multi-stage schemes in physical time for unsteady low Mach number compressible flows. EPTT-VII Escola de Primavera de Transição e Turbulência (2010)
- Eliasson, P., Weinerfelt, P.: High-order implicit time integration for unsteady turbulent flow simulations. *Comput. Fluids* **112**, 35–49 (2015)
- Gustafsson, K., Lundh, M., Süderlind, G.: A pi stepsize control for the numerical solution of ordinary differential equations. *BIT Numer. Math.* **28**(2), 270–287 (1988)
- Haas, J.F., Sturtevant, B.: Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mech.* **181**, 41–76 (1987)
- Hay, A., Etienne, S., Pelletier, D., Garon, A.: hp-Adaptive time integration based on the BDF for viscous flows. *J. Comput. Phys.* **291**, 151–176 (2015)
- Jameson, A.: Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. AIAA paper, p. 1596 (1991)
- Jameson, A.: Evaluation of fully implicit Runge Kutta schemes for unsteady flow calculations. *J. Sci. Comput.* **73**, 1–34 (2017)
- Jameson, A., Schmidt, W., Turkel, E.: Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. AIAA paper, 1259 (1981)
- John, V., Rang, J.: Adaptive time step control for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.* **199**, 514–524 (2010)
- Kennedy, C.A., Carpenter, M.H.: Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Appl. Numer. Math.* **44**(1), 139–181 (2003)
- Langer, S.: Accuracy investigations of a compressible second order finite volume code towards the compressible limit. *Comput. Fluids* **149**, 110–137 (2017)

14. Langer, S.: Agglomeration multigrid methods with implicit Runge–Kutta smoothers applied to aerodynamic simulations on unstructured grids. *J. Comput. Phys.* **277**, 72–100 (2014)
15. Peles, O., Yaniv, S., Turkel, E.: Convergence Acceleration of Runge–Kutta Schemes using RK/implicit smoother for Navier–Stokes equations with SST turbulence. In: *Proceedings of 52nd Israel Annual Conference on Aerospace Sciences* (2012)
16. Peles, O., Turkel, E., Yaniv, S.: Fast Iterative methods for Navier–Stokes equations with SST turbulence model and chemistry (ICCFD7). In: *Seventh International Conference on Computational Fluid Dynamics* (2012)
17. Peles, O., Turkel, E.: Acceleration methods for multi-physics compressible flow. *J. Comput. Phys.* **358**, 201–234 (2018)
18. Rossow, C.C.: Efficient computation of compressible and incompressible flows. *J. Comput. Phys.* **220**(2), 879–899 (2007)
19. Shi, J., Zhang, Y.T., Shu, C.W.: Resolution of high order WENO schemes for complicated flow structures. *J. Comput. Phys.* **186**, 690–696 (2003)
20. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **83**(1), 32–78 (1989)
21. Sod, G.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27**(1), 1–31 (1978)
22. Swanson, R.C., Rossow, C.C.: An efficient solver for the RANS Equations and a one-equation turbulence model. *Comput. Fluids* **42**, 13–25 (2011)
23. Swanson, R.C., Rossow, C.C., Yaniv, S.: Analysis of a RK/implicit smoother for multigrid, computational fluid dynamics. In: Kuzmin, A. (Ed.) *The Sixth International Conference on Computational Fluid Dynamics*. Springer, St. Petersburg, Russia, July 12–16 (2010)
24. Swanson, R.C., Turkel, E.: Analysis of a fast iterative method in a dual time algorithm for the Navier–Stokes equations. In: *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, Vienna, Austria, September 10–14, 2012
25. Swanson, R.C., Turkel, E., Rossow, C.C.: Convergence acceleration of Runge–Kutta schemes for solving the Navier–Stokes equations. *J. Comput. Phys.* **224**(1), 365–388 (2007)
26. Turkel, E.: Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comput. Phys.* **72**(2), 277–298 (1987)
27. Turkel, E.: Review of preconditioning methods for fluid dynamics. *Appl. Numer. Math.* **12**, 257–284 (1993)
28. Turkel, E., Fitterman, A., Van Leer, B.: Preconditioning and the limit to the incompressible flow equations. In: Caughey, D.A., Hafez, M.M. (eds.) *Frontiers of Computational Fluid Dynamics*, pp. 215–234. Wiley, New York (1994)
29. Turkel, E., Radespiel, R., Kroll, N.: Assessment of preconditioning methods for multidimensional aerodynamics. *Comput. Fluids* **26**(2), 613–634 (1997)
30. Turkel, E., Vatsa, V.N.: Choice of variables and preconditioning for time dependent problems. *AIAA paper*, 3692 (2003)
31. Turkel, E., Vatsa, V.N.: Local preconditioners for steady state and dual time stepping. *ESAIM: M2AN* **39**(3), 515–536 (2005)
32. Van Leer, B., Lee, W., Roe, P.: Characteristic time-stepping or local preconditioning of the Euler equations. In: *10th Computational Fluid Dynamics Conference* (1991)
33. Van Leer, B., Lee, E.T., Roe, P.L., Powell, K.G., Tai, C.H.: Design of optimally smoothing multi-stage schemes for the Euler equations. *Commun. Appl. Numer. Methods* **8**(10), 761–769 (1992)
34. Wang, B.S., Li, P., Gao, Z., Don, W.S.: An improved fifth order alternative WENO-Z finite difference scheme for hyperbolic conservation laws. *J. Comput. Phys.* **374**, 469–477 (2018)
35. Xu, Z., Shu, C.W.: Anti-diffusive flux corrections for high order finite difference WENO schemes. *J. Comput. Phys.* **205**, 458–485 (2005)
36. Zhao, G., Sun, M., Xie, S., Wang, H.: Numerical dissipation control in an adaptive WCNS with a new smoothness indicator. *Appl. Math. Comput.* **330**, 239–253 (2018)

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com