Tenth International Conference on
Computational Fluid Dynamics (ICCFD10),
Barcelona,Spain, July 9-13, 2018

ICCFD10-003

Adaptive time steps for compressible flows based on dual-time stepping and a RK/Implicit smoother scheme

O. Peles*,** and E. Turkel*
Corresponding author: oren_peles@walla.co.il

* School of Mathematics/Tel-Aviv University, Israel.
** IMI systems, Ramat Ha-Sharon, 4711001, Israel.

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 large variations of time scales during the physical time and hence a wide a range of required physical time steps. We discuss the method and present several examples of compressible flow problems.

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]). There are also optimized versions of BDF [28]. 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 accuracy decreases when using large time steps.

Acceleration of the convergence in pseudo-time is one way to reduce the CPU time. Many techniques are used in the literature such as multigrid methods, low Mach preconditioning and residual smoothing. [17] and [20] introduced the pseudo-time RK/imlicit smoother as a convergence acceleration method for steady viscous flows. The method was extended by Peles et al. [14] for turbulent flows and for reactive flows in [15] and general multiphysics problems in [16]. For low Mach problem preconditioning methods were developed in [21, 22, 24, 26, 27]. Instead, Rossow [17] used an intrinsic treatment for the low Mach accuracy. This was combined with the RK/Implicit smoother in [15] and [12] using the corresponding flux Jacobian in a Roe-type scheme. Langer [13] 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 in time. Thus, if we wish a higher order accurate scheme, in time, implicit RK methods for the physical time steps can be a good choice. 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 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 of constant time steps. 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 using constant time steps, the optimum time step of BDF method can be more efficient then the optimal DIRK time step. However, in the BDF scheme, without prior knowledge about the required time steps, a time-step sensitivity analysis 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]. A method for adaptive time steps with IRK scheme was presented in [11] for incompressible Navier-Stokes flows.

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 DIRK scheme for the physical time steps with adaptive time stepping. The governing equations will presented in section 2. In section 3.1 the dual time stepping with BDF method will be discussed and in section 3.2 the dual-time stepping with DIRK will be presented. In section 4 the adaptive time step technique will be described and several examples of compressible flows will be presented in section 5.

2 Governing equations

We consider the three dimensional, compressible Navier-Stokes questions, given in conservation form as

$$\frac{\partial Q}{\partial t} + \nabla \vec{F}(Q) = \nabla \vec{F}_v(Q) + S \tag{1}$$

where Q is the vector of variables - $Q = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, 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

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

F is the inviscid flux, given by

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

S is any source term and F_v is the viscous flux, given by

$$F_v = (0, \vec{\tau}_x, \vec{\tau}_y, \vec{\tau}_z, u\vec{\tau}_x + v\vec{\tau}_y + w\vec{\tau}_z + k\nabla T)^T$$

where $\vec{\tau}_x$, $\vec{\tau}_y$ and $\vec{\tau}_z$ 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 Differencing

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

$$\frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} + \nabla \vec{F}(W) = \nabla \vec{F}_v(W) \tag{2}$$

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

$$\frac{\partial W}{\partial \tau} = -\frac{\partial W}{\partial t} - R \tag{3}$$

where the residual R is given by

$$R = \frac{1}{V} \oint_S (\vec{F} - \vec{F}_v) \cdot d\vec{S} = \sum_{cellfaces} (\vec{F} - \vec{F}_v) \cdot d\vec{S}. \tag{4}$$

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

$$\frac{\partial W}{\partial t} \approx \frac{3W^{n+1} - 4W^n + W^{n-1}}{2\Delta t}. \tag{5}$$

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

$$\begin{aligned} W^{(0)} &= W^n \\ W^{(k+1)} &= W^{(0)} - \alpha_{k+1} R(W^{(k)}) \quad ; \quad k = 0...p-1 \\ W^{n+1} &= W^{(p)} \end{aligned} \tag{6}$$

where p is the number of stages. Following [17] and [20] linearization of (3) with respect to the pseudo time yields

$$\left(I + \frac{\Delta\tau}{V} \sum_{cellfaces} A \cdot d\vec{S} + \Delta\tau \frac{\partial S}{\partial W} \right) \Delta W = R \tag{7}$$

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

Following [17] and [20] for the residual smoothing operator we write 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^+ in the left side multiply ΔW of the local cell and move A^- to the right side multiply ΔW of the neighbor cells. This yields a set of linear equations for the smoothed ΔW which replaces the unsmoothed residuals -

$$\left(I + \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^+ \cdot d\vec{S} + \Delta\tau \frac{\partial S}{\partial W} \right) \Delta W_{local} = R - \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^- \cdot \Delta W_{NB} \cdot d\vec{S}. \tag{8}$$

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). Since the solver is explicit and we use the resulting ΔW as smoothed residuals we perform only few number of implicit iterations, typically 2-3, in each RK step. ϵ is used as a relaxation parameter. Usually $\epsilon \leq 1$ is used. $\epsilon > 1$ is an over-relaxation and can be used for very stiff problems.

In [25] (2) was preconditioned and was solved in semi-discrete form using the scheme

$$P^{-1} \frac{\partial W}{\partial \tau} + \frac{\partial W}{\partial t} + \nabla \vec{F}(W) = \nabla \vec{F}_v(W) - h_x \frac{\partial}{\partial x} \left(P^{-1} |PA| \frac{\partial W}{\partial x} \right) \tag{9}$$

where the last term is the artificial viscosity and the matrix P is the low-Mach number preconditioner which aims to control the consistency of the scheme for low Mach number flow as $M \to 0$ (see [23] and [2]). Since in the approximation of the derivative with respect to physical time ((5)), W^{n+1} is unknown, it was approximated by $W^{(k+1)}$. They derived the point implicit scheme -

$$\left(I + \alpha_k c_t \frac{\Delta\tau}{\Delta t} P \right) w^{(k+1)} = w^0 - \alpha_k \Delta\tau P \left\{ R^k + \frac{c_t w^k - F(w^n, w^{n-1}, ...)}{\Delta t} \right\} + \alpha_k c_t \frac{\Delta\tau}{\Delta t} P w^k. \tag{10}$$

Later in [20], the matrix P^{-1} multiplying the $\frac{\partial W}{\partial \tau}$ term 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 where its Jacobian is $\frac{c_t}{\Delta t}$, the smoother operator takes the form

$$\left(I + \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^+ \cdot d\vec{S} + c_t \frac{\Delta\tau}{\Delta t} + \Delta\tau \frac{\partial S}{\partial W} \right) \Delta W_{local} = R^k - \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^- \cdot \Delta W_{NB} \cdot d\vec{S} \tag{11a}$$

where

$$R^k = \sum_{cellfaces} (\vec{F} - \vec{F_v}) \cdot d\vec{S} - \frac{3W^k - 4W^n + W^{n-1}}{2\Delta t}. \tag{11b}$$

Equations (11a) and (11b) define the standard scheme for dual time stepping with RK/implicit smoother.

We adopt the strategy of [17] and [20] for low Mach treatment by using a modified speed of sound c' in the flux Jacobian matrix in the implicit smoother operator. We also 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 [12]). Hence, we set the preconditioning matrix $P = I$ and (10) can be written (for scalar equation) as

$$W^{(k+1)} = W^0 - \alpha_k \Delta\tau \frac{\widetilde{R}^k}{1 + \alpha_k c_t \frac{\Delta\tau}{\Delta t}} \tag{12}$$

where

$$\widetilde{R}^k = R^k + \frac{c_t W^0 - F(W^n, W^{n-1}, ...)}{\Delta t}. \tag{13}$$

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

$$\left(I + \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^+ \cdot d\vec{S} + \alpha_k c_t \frac{\Delta\tau}{\Delta t} + \Delta\tau \frac{\partial S}{\partial W}\right) \Delta W_{local} = R^k - \epsilon \frac{\Delta\tau}{V} \sum_{cellfaces} A^- \cdot \Delta W_{NB} \cdot d\vec{S}. \tag{14a}$$

where

$$R^k = \sum_{cellfaces} (\vec{F} - \vec{F}_v) \cdot d\vec{S} - \frac{3W^0 - 4W^n + W^{n-1}}{2\Delta t}. \tag{14b}$$

To summarize the differences between the standard (equations (11a) and (11b)) and the new scheme (equations (14a) and (14b)) -

1. In the right hand side of the standard scheme (11b) the approximation of the derivative with respect to physical time contains W^k and in the new scheme W^k is replaced by W^0 (14b).
2. In the smoother operator of the new scheme (14a) 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 [19].

For this problem the initial conditions are

$$\begin{cases} p = 1 \text{ PSI and } T = 231.11\text{K on the right side of the tube} \\ p = 10 \text{ PSI and } T = 288.89\text{K on the left side} \end{cases} \tag{15}$$

with Mach number zero in both sides.

The convergence for a series of CFL numbers is presented in figure 3.1 for three physical time steps. Improved convergence is obtained as the CFL number increases until reaching an asymptotic convergence. Comparison of the results to the analytic solution is presented in figure 3.2.

The computations demonstrate that the new approach is more robust especially at very low Mach numbers. Figure 3.3 shows a comparison between the conversion histories of the standard and the new methods for the Shu-Osher problem (see details in [18]). 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.

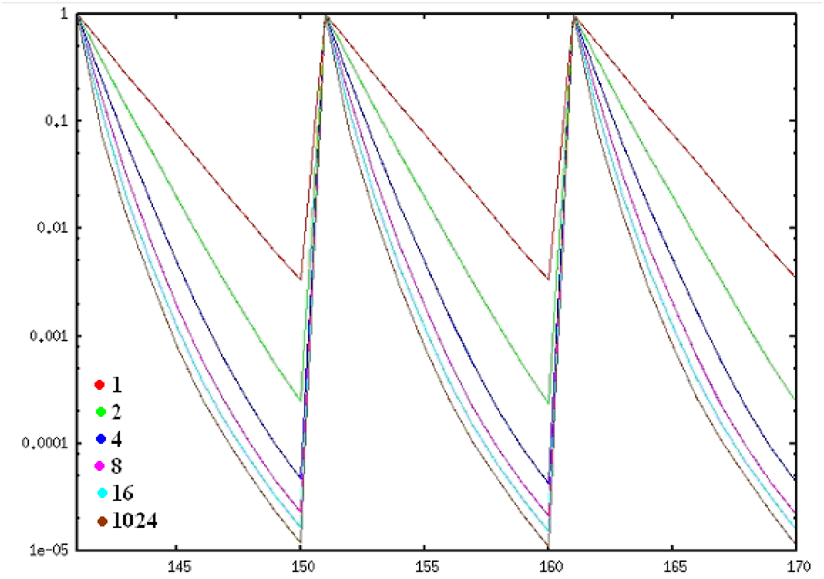

Figure 3.1: Convergence history for series of CFL numbers for three physical time steps

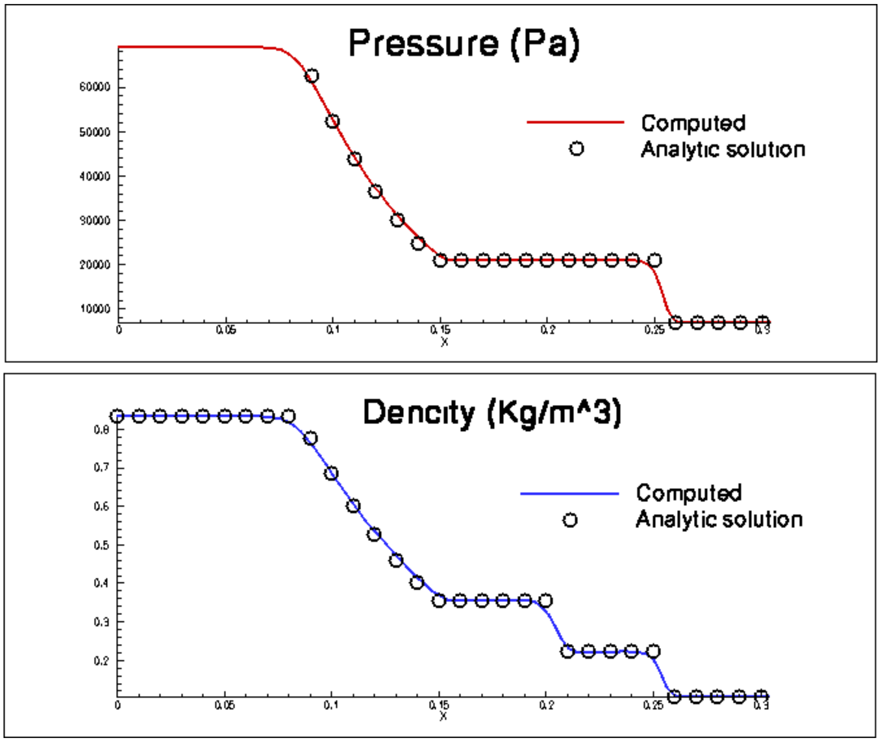

Figure 3.2: Sod's tube results - Analytical solution vs. numerical simulation. Pressure (top) and density (bottom)

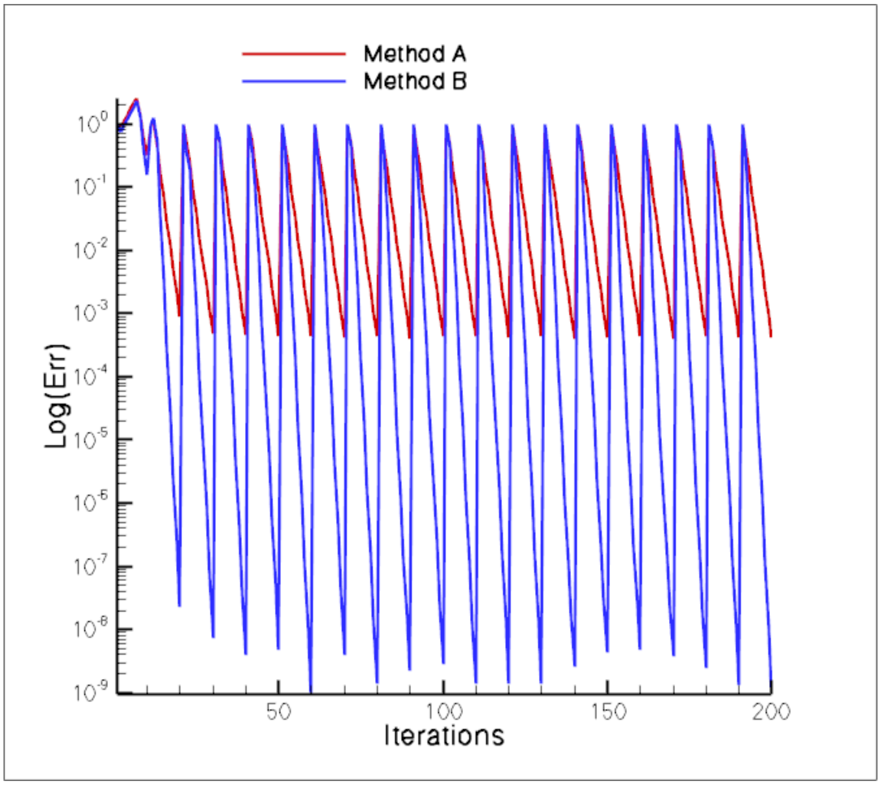

Figure 3.3: Convergence history comparison between the standard dual time-stepping method (A) and the new dual-time stepping method (B) for the one-dimension Shu-Osher problem. The CFL used was 64

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

$$\frac{du}{dt} = -R(u). \tag{16}$$

The DIRK scheme is given by

$$u^{(j)} = u^n - \Delta t \sum_{i=1}^{j} a_{ij} R(u^{(i)}) \quad j = 1..s \tag{17a}$$

$$u^{n+1} = u^n - \Delta t \sum_{j=1}^{s} b_j R(u^{(i)}) \tag{17b}$$

where $u^n = u(t_n)$, $u^{(j)}$ is the solution at the j-th internal RK stage and s is the number of RK stages. Also, $a_{11} = 0$ and so $u^{(1)} = u^n$. Furthermore, $a_{jj} \neq 0 \quad \forall j > 1$ and $b_j = a_{js}$ hence $u^{n+1} = u^{(s)}$. We then write (17a) as

$$R^*(u^{(j)}) = R(u^{(j)}) - \frac{u^n - u^{(j)}}{a_{jj}\Delta t} - \frac{1}{a_{jj}} \sum_{i=1}^{j-1} a_{ij} R(u^{(i)}) = 0. \tag{18}$$

Equation (18) is solved for $u^{(j)}$ with pseudo-time stepping

$$\frac{du^{(j)}}{d\tau} = R^*(u^{(j)}). \tag{19}$$

We solve (19) with a low-storage RK with an implicit smoother, where the residual smoother operator is defined similar to the new scheme (14a) by

$$\left(I+\epsilon\frac{\Delta\tau}{V}\sum_{cellfaces}A^{+}\cdot d\vec{S}+\frac{1}{a_{jj}}\frac{\Delta\tau}{\Delta t}+\Delta\tau\frac{\partial S}{\partial W}\right)\Delta W_{local}=R^{k}-\epsilon\frac{\Delta\tau}{V}\sum_{cellfaces}A^{-}\cdot\Delta W_{NB}\cdot d\vec{S}. \tag{20}$$

3.3 DIRK coefficients

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

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

We recall that $b_j = a_{js}$ while the coefficients $\hat{b}_j$ for the lower order embedded scheme are given by

$$\begin{aligned} \hat{b}_1&=\frac{5}{12}+\frac{\sqrt{3}}{12}\\ \hat{b}_2&=\frac{9}{12}+\frac{\sqrt{3}}{12}\\ \hat{b}_3&=-\frac{2}{12}-\frac{2\sqrt{3}}{12} \end{aligned} \tag{22}$$

4 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 u^{n+1} be the solution of the given DIRK scheme after one time step of size Δt and let $\hat{u}^{n+1}$ be the solution of the embedded scheme, both with initial condition u^n. We therefore have:

$$u^{n+1}=u+c\Delta t^{p} \tag{23}$$

and

$$\hat{u}^{n+1}=u+\hat{c}\Delta t^{p-1}. \tag{24}$$

Subtracting equation (23) from (24) yields

$$\hat{u}^{n+1}-u^{n+1}=\hat{c}\Delta t^{p-1}. \tag{25}$$

Hence,

$$\frac{\|\hat{u}^{n+1}-u^{n+1}\|}{\|\hat{u}^{n}-u^{n}\|}=\left(\frac{\Delta t_{n+1}}{\Delta t_{n}}\right)^{p-1}. \tag{26}$$

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

$$\|\hat{u}^{n+1}-u^{n+1}\|=\|\hat{u}^{n}-u^{n}\|\cdot\left(\frac{\Delta t_{n+1}}{\Delta t_{n}}\right)^{p-1}\leq\epsilon\|u^{n}\|. \tag{27}$$

From (27) we get

$$\Delta t_{n+1} \leq \Delta t_n \cdot \left(\frac{\epsilon \|u^n\|}{\|\hat{u}^n - u^n\|} \right)^{\frac{1}{p-1}} \tag{28}$$

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

$$\Delta t_{n+1} = max\Bigg(min\Bigg(\Delta t_n \cdot \left(\frac{TOL \|u^n\|}{\|\hat{u}^n - u^n\|} \right)^{\frac{1}{p-1}}, C_2 \Delta t_n \Bigg), C_1 \Delta t_n \Bigg) \tag{29}$$

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).

5 Examples

We consider three test cases. Two of them are one-dimensional and one is a two dimensional problem.

5.1 Riemann problem (Sod's tube)

Sod's tube problem was defined in (15). Figure 5.1 presents the solution for two values of the tolerance parameter (TOL) and compares this result also to the analytical solution. We see that the solution is getting better as TOL is getting smaller especially in the vicinity to the shock-wave. For $TOL = 10^{-3}$, figure 5.2 presents 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.

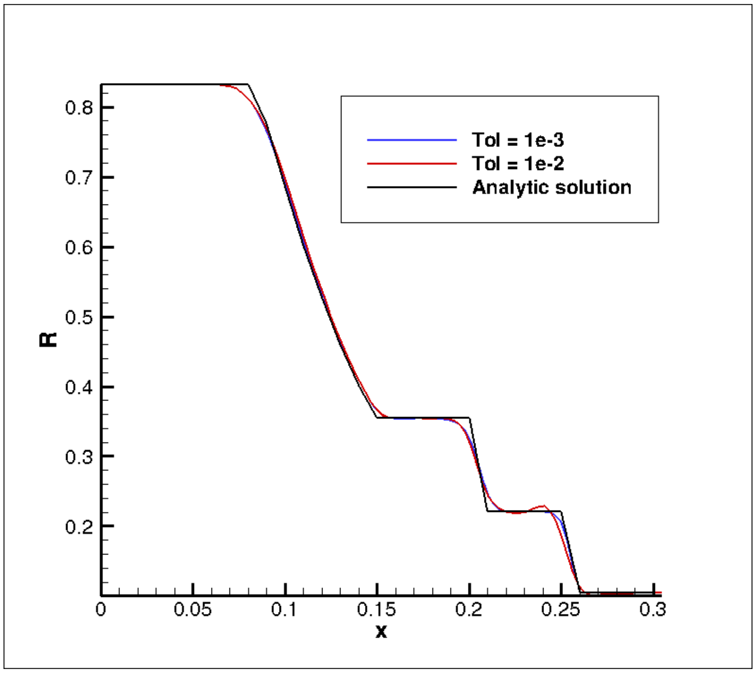

Figure 5.1: Comparison solutions for two values of convergence tolerances and the analytical solution at t=200 ms.

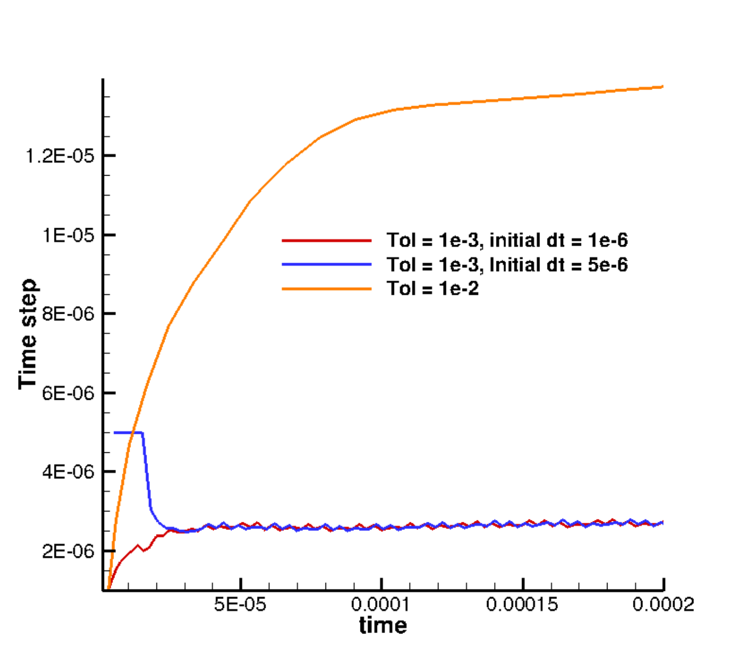

Figure 5.2: Evolution of the selected time steps for different tolerances and initial time steps (Sods tube problem)

	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

Table 5.1: initial conditions for the shockwave bubble interaction

5.2 Shock-Sine wave interaction (Shu-Osher problem)

The Shu-Osher problem is described the interaction between a one-dimensional shock-wave with a sine wave [18]. The initial conditions are given by

$$\begin{cases} \rho = 3.857143; & u = 2.629369; \quad P = 10.33333 \quad \text{when} \quad x \leq -4 \\ \rho = 1 + 0.2\sin(5x); & u = 0 \qquad\qquad\quad P = 1 \qquad\qquad\quad \text{when} \quad x > -4 \end{cases} \tag{30}$$

Figure 5.3 compares the solutions for two values of the tolerances to the solution of the BDF method (there is no analytical solution). It can be seen that for a smaller tolerance, the solution is closer to the BDF solution. Figure 5.4 shows the evolution of the selected time step. The oscillatory behavior of the problem is manifested in the time steps evolution. In addition, a constant trend of increasing of the time steps can be seen. The reason for this phenomena is unclear.

5.3 Shock-wave-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 figure 5.5. The exact initial condition are presented in table 5.1.

The properties of air are $\gamma = 1.4$ and molecular weight 29 gr/cm^3 and the properties of helium are $\gamma = 1.67$ and molecular weight 4 gr/cm^3.

This problem, is characterized by strongly varying time scales in physical time. We display in figure 5.6 the time step evolution. Before the bubble, the grid itself is getting denser and so the time step must be decrease in order to

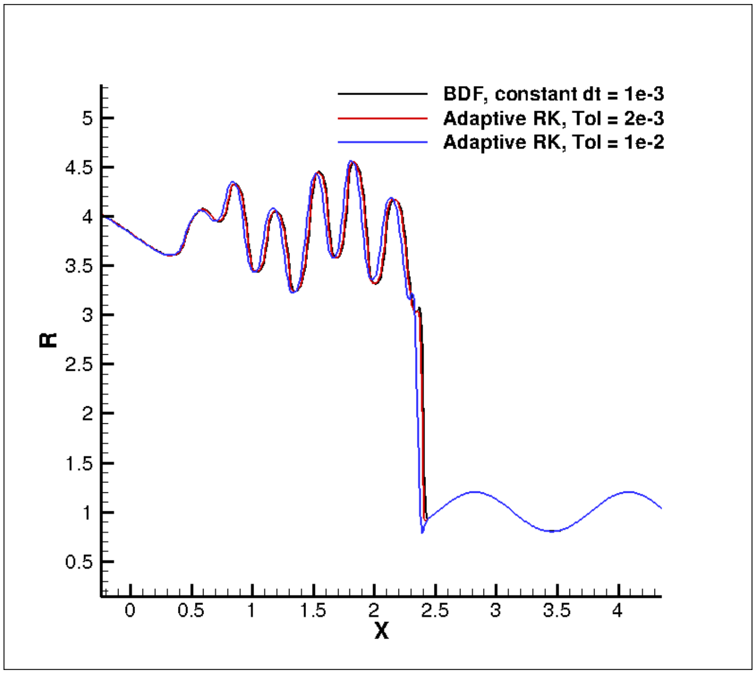

Figure 5.3: Comparison of two adaptive time steps DIRK simulations with two values of Tol and simulation with BDF with $\Delta t = 10^{-3}$. The smaller Tol = 2e-3 is almost identical to BDF but with time steps more about twice larger. The larger Tol = 1e-2 with times steps about 3 to 5 larger.

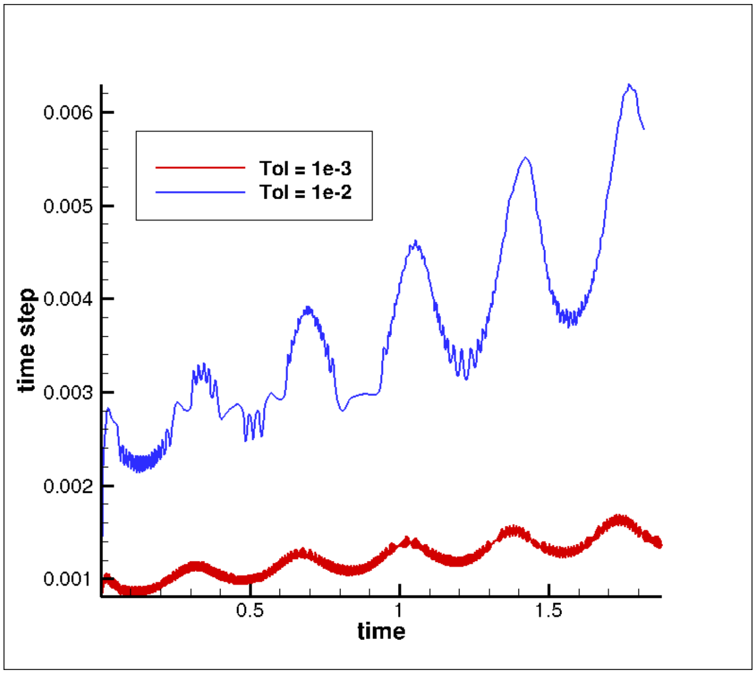

Figure 5.4: Time steps evolution for the Shu-Osher problem for two values of Tol. The oscillatory nature of the problem is reflected in the time steps as well as unexplained yet of time step increasing trend

keep a constant error. After exiting from the bubble, the grid becomes sparser and the time step again increases. The standard time steps for this problem with the BDF method is about $10^{-4} sec.$ and the CPU time gain, using DIRK, was approximately a factor of four. In figure 5.7 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.

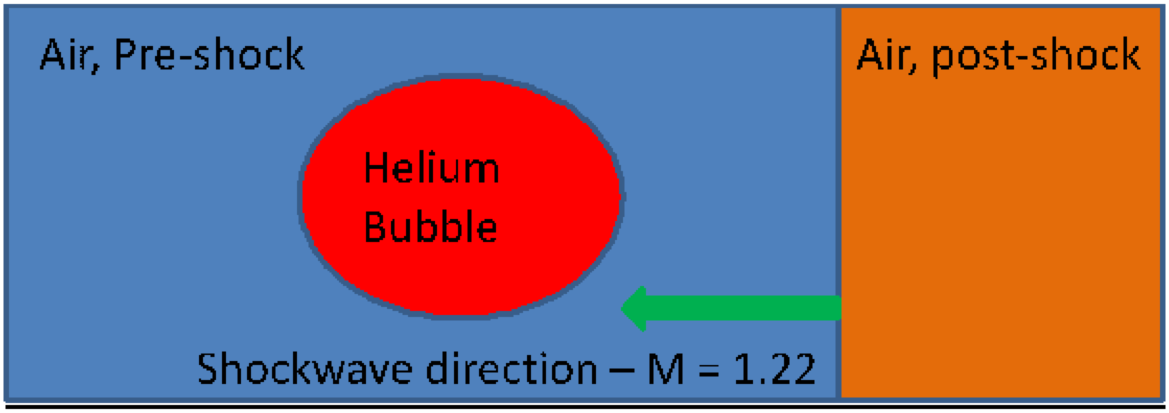

Figure 5.5: Schematic description of the initial conditions for the shockwave-bubble interaction problem

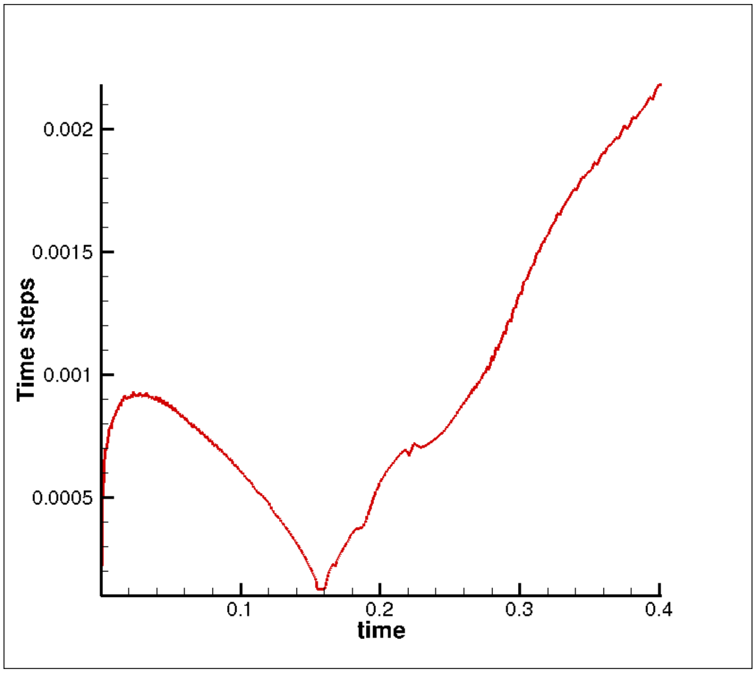

Figure 5.6: Time steps evolution for the Shock-wave/bubble interaction problem. the deceasing of the time step before the shock-wave/bubble impingement due to the grid clustering can be seem as well as the time step increasing after the shock-wave is leaves the bubble.

6 Conclusions

We have presented an adaptive time stepping technique based on dual time steps method combined with a high-order implicit Runge-Kutta method in physical time and a RK/implicit smoother for the pseudo time integration. The method gives a high gain of CPU time for problems with a large variation of time scales. It also replaces the process of time-

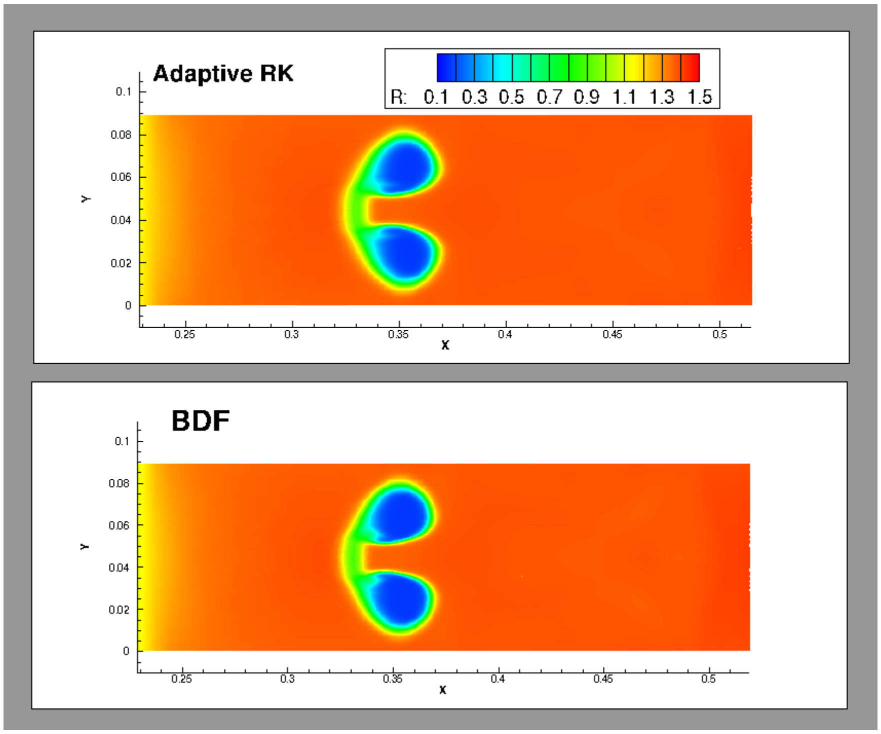

Figure 5.7: Comparison of the results between adaptive time steps DIRK and BDF (density) at time $t = 0.03$ sec.

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 and the CPU time is a direct outcome of the desired accuracy.

We also found that the present method requires more inner iterations in pseudo-time and the improvement of the internal convergence will improve the worthiness of the method. Another yet, unresolved issue is the stability analysis of the method. [1] gives the linear stability of the implicit time integration using a dual-time stepping. They analyzed several temporal schemes and show that the inner iterations in pseudo-time introduces a secondary convergence criterion to the numerical method.

References

[1] J. J. Chiew and T. H. Pulliam. Stability Analysis of Dual-Time Stepping. *46th AIAA Fluid Dynamics Conference*, 2016.

[2] D. Darmofal and B. Van Leer. Local preconditioning: Manipulating mother nature to fool father time. *Computing the Future II: Advances and Prospects in Computational Aerodynamics*, 1998.

[3] L. S. de Brito Alves. Dual time stepping with multi-stage schemes in physical time for usteady low Mach number compressible flows. *EPTT-VII Escola de Primavera de TransiÃğÃčo e TurbulÃłncia*, 2010.

[4] P. Eliasson and P. Weinerfelt. High-order implicit time integration for unsteady turbulent flow simulations. *Computers & Fluids*, 112:35–49, 2015.

[5] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28(2):270–287, 1988.

[6] J.F. Haas and B. Sturtevant. Interaction of weak shock waves with cylindrical and spherical gas inhomogenities. *Journal of Fluid Mechanics*. 181:41–76, 1987.

[7] A. Hay, S. Etienne, D. Pelletier and A. Garon. hp-Adaptive time integration based on the BDF for viscous flows. *Journal of Computational Physics*, 291:151–176, 2015.

[8] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper*, p. 1596, 1991.

[9] A. Jameson. Evaluation of Fully Implicit Runge Kutta Schemes for Unsteady Flow Calculations. *Journal of Scientific Computing*, 1–34, 2017.

[10] A. Jameson, W. Schmidt and E. Turkel Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA paper*, 1259, 1981.

[11] V. John and J. Rang. Adaptive time step control for the incompressible NavierâĂŞStokes equations. *Comput. Methods Appl. Mech. Engrg*, 199:514-524, 2010.

[12] S. Langer. Accuracy investigations of a compressible second order finite volume code towards the compressible limit. *Computers & Fluids*, 149:110–137, 2017.

[13] S. Langer. Agglomeration multigrid methods with implicit Runge-Kutta smoothers applied to aerodynamic simulations on unstructured grids. *Journal of Computational Physics*, 277:72–100, 2014.

[14] O. Peles, S. Yaniv and E. Turkel. Convergence Acceleration of Runge-Kutta Schemes using RK/Implicit Smoother for Navier-Stokes Equations with SST Turbulence. *Proceedings of 52nd Israel Annual Conference on Aerospace Sciences*, 2012.

[15] O. Peles, E. Turkel and S. Yaniv. Fast Iterative Methods for Navier-Stokes Equations with SST Turbulence Model and Chemistry (ICCFD7). *Seventh International Conference on Computational Fluid Dynamics*, 2012.

[16] O. Peles and E. Turkel Acceleration Methods for Multi-physics Compressible Flow. *Journal of Computational Physics* 358:201–234, 2018.

[17] C. C. Rossow. Efficient computation of compressible and incompressible flows. *Journal of Computational Physics*, 220(2):879–899, 2007.

[18] C. W. Shu and S. Osher. Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes. *Journal of Computational Physics*, 83(1):32–78, 1989.

[19] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, 1978.

[20] R. C. Swanson, E. Turkel and C. C. Rossow. Convergence Acceleration of Runge-Kutta Schemes for Solving the Navier-Stokes Equations. *Journal of Computational Physics*, 224(1):365–388, 2007.

[21] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics*, 72(2):277-298, 1987.

[22] E. Turkel. Review of Preconditioning Methods for Fluid Dynamics. *Applied Numerical Mathematics*, 12:257-284, 1993.

[23] E. Turkel, A. Fiterman and B. Van Leer. Preconditioning and the limit to the incompressible flow equations. *Frontiers of Computational Fluid Dynamics 1994*, 215-234. D.A. Caughey and M.M. Hafez editors, John Wiley and Sons, 1994.

[24] E. Turkel, R. Radespiel and N. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics,. *Computers & Fluids*, 26(2):613-634, 1997

[25] E. Turkel, V.N. Vatsa. Choice of variables and preconditioning for time dependent problems. *AIAA paper*, 3692, 2003.

[26] E. Turkel, V.N. Vatsa. Local Preconditioners for Steady State and Dual Time Stepping.. *ESAIM: M2AN*, 39(3):515-536, 2005.

[27] B. Van Leer, W. Lee. and P. Roe. Characteristic time-stepping or local preconditioning of the Euler equations.. *10th Computational Fluid Dynamics Conference*, 1991.

[28] V.N. Vatsa, M.H. Carpenter and D.P. Lockard Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications *48th AIAA Aerospace Sciences Meeting 2010-0122*, Orlando, FL, Jan 4-7, 2010,