Bericht des Instituts für Aerodynamik und Strömungstechnik

Report of the Institute of Aerodynamics and Flow Technology

IB 124-2007/3

A Fourth Order Semi-Implicit Runge-Kutta Method for the Compressible Euler Equations

Ursula Mayer, Richard P. Dwight

Herausgeber:
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im April 2007

Institutsdirektor:

Prof. Dr.-Ing. habil. C.-C. Rossow

Verfasser:

Ursula Mayer
Richard P. Dwight

Abteilung: Numerical Methods

Abteilungsleiter:

Prof. Dr.-Ing. N. Kroll

Der Bericht enthält:

24 Seiten
10 Bilder
0 Tabellen
15 Literaturstellen

Abstract

Efficient time integration is of the utmost concern for unsteady flow computations. Within this context two semi-implicit Runge-Kutta methods of 3rd- and 4th-order and their application to two-dimensional compressible inviscid flows are presented. The semi-implicit methods developed here are expected to be considerably more stable than fully explicit methods, and more efficient than fully implicit methods. This is achieved by constructing an explicit 3rd-order Runge-Kutta method with the property that an *arbitrary* stabilizing implicit term may be added at each stage without degrading the order of convergence. This feature allows one to apply implicit operators based on approximate Jacobians, the choice of operator being made only with regard to stability, efficiency and storage requirements. A corresponding 4th-order method is obtained by linear Richardson extrapolation on the original 3rd order scheme, and both schemes are supplied with a local error estimation and time-step control. Numerical examples are performed on various cases using the DLR TAU-Code, a finite volume RANS solver, with a Lower-Upper Symmetric Gauss-Seidel implicit operator. It demonstrated that both semi-implicit schemes are significantly more stable than the corresponding explicit schemes, and in addition they are able to outperform standard fully implicit methods by up to a factor of 10 in terms of CPU time for given accuracy.

Contents

1 Introduction

The level of detail and complexity demanded of numerical simulation is increasing continuously in many engineering fields. In computational aerodynamics time-accurate flow computations are becoming increasingly important and relevant to industry, but require disproportionate computing resources in comparison to stationary problems.

Efficiency improvements of flow solvers in aerodynamics have concentrated principally on the rapid solution of steady flows using such techniques as Full Approximation Storage (FAS) multigrid and approximate (and inexact) Newton solvers. As a result the application of these algorithms has become refined and well understood. When unsteady simulations are required, they are typically performed using a *dual time* algorithm [8], where an analogue of the nonlinear steady problem is solved at each time-step using aforementioned steady solvers. The result is an order-of-magnitude higher cost for unsteady simulation, which has precluded its application in many areas. New algorithms for the efficient solution of unsteady flows are therefore of utmost concern.

The commonly used fully explicit and fully implicit time integration schemes each have a major disadvantage: explicit schemes suffer from a restriction on the time-step size for stiff problems; implicit schemes in contrast are often unconditionally stable and the time-step may be chosen based only on resolution of the physical processes of interest, but a nonlinear system of equations must be solved at every integration step.

Any semi-implicit scheme might therefore be expected to combine the advantages of the two approaches. It should be considerably more stable than an explicit scheme, and significantly cheaper per time-step than an implicit scheme.

One candidate method is due to Nikitin [11], who applied his semi-implicit Runge-Kutta method (RK) to the incompressible Navier-Stokes equations for problems including flow in a driven cavity. The scheme is based on a three-stage third order explicit RK scheme, whose structure is carefully chosen such that it is possible to add an *arbitrary* implicit term to each stage without compromising the order of the method. This allows the choice of the implicit part based *only* on the requirements of stability, time-accuracy being automatically satisfied. Therefore the highly tuned approximate Newton methods developed in the pursuit of efficient schemes for steady problems may be used as stabilization terms for unsteady problems.

This is in sharp contrast to such semi-implicit schemes as the second-order Adams-Bashforth-Crank-Nicolson (ABCN) method for the convection-diffusion equation, which treats the convective terms with Adams-Bashforth and the diffusive terms with Crank-Nicolson, and where the formulation and solution of the implicit part must be exact in order for the scheme to be of the desired order. Modern examples for the compressible Navier-Stokes equations are the third-order semi-implicit RK methods of Yoh and Zhong [13], which also require exact treatment of the implicit part. This condition restricts the application of such methods to situations in with the stiffness is confined to a few terms that may be additively separated from non-stiff terms, and whose exact implicit treatment is much cheaper than the exact implicit treatment of

the entire system. For example Yoh and Zhong have applied their scheme to combustion problems with only the strong source terms - which dominate the stability of the scheme - treated implicitly [14]. With the scheme of Nikitin it is possible to treat all terms implicitly at low cost by using approximate implicit methods.

Since the term Semi-Implicit Runge-Kutta (SIRK) is already well-established for denoting schemes of the ABCN type, we will refer to the present class of methods as Unconstrained Implicit Runge-Kutta (UIRK), in reference to the complete absence of constraints on the implicit operator.

This report considers the third-order time-accurate, semi-implicit RK method of Nikitin applied to the compressible Euler equations. A semi-implicit fourth-order method is constructed by linear Richardson extrapolation on the original scheme, and for both schemes an embedded lower-order method is used to provide local error estimation and time-step control.

The most crucial choice involved in the scheme - in terms of storage requirements, efficiency, and accuracy - is the choice of the implicit operator, its approximate flux Jacobian and the solver for the resulting linear system. In this preliminary study we consider only application of a highly optimized version of the Lower-Upper Symmetric Gauss-Seidel (LU-SGS) algorithm [15, 5]. This involves a first-order approximation of the Jacobian, constructed on-the-fly, whereby all terms in the spatial discretization, including boundary conditions, are present. One sweep of symmetric Gauss-Seidel is applied to give a very rough approximation to the solution of this linear system. The result is an implicit method that is unconditionally stable for inviscid problems, while being cheaper per iteration than a three-stage RK scheme, as well as requiring less memory. This cheap unconditional stability makes LU-SGS an ideal scheme for the current application.

The order of both semi-implicit Runge-Kutta methods is verified numerically for several test cases on various structured and unstructured grids. It is demonstrated that in several cases the UIRK scheme outperforms dual time by a factor of 10 in terms of CPU time for a given accuracy. All implementations and tests are done within the finite volume RANS solver the DLR TAU-Code [6].

This report is organized as follows: the governing Euler equations and their unstructured finite volume discretization are described in Sections 2.1 and 2.2 respectively, then the third-order UIRK scheme and its fourth-order extrapolation are discussed in detail in Sections 3.1 and 3.2. Stability of the schemes is examined in Section 3.3, and the time-step control algorithm is developed in Section 3.4. Section 4 describes LU-SGS, and numerical tests are undertaken in Section 5.

2 Spatial Discretization

2.1 The Euler Equations

The governing equations are the compressible Euler equations in two dimensions, which expressed in conservative form are

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x_i} f_i^c(W) = \frac{\partial W}{\partial t} + \mathcal{R}(W) = 0, \tag{2.1}$$

where the summation convention is applied on the index i and the residual $\mathcal{R}(W)$ denotes the sum of the convective flux derivatives.

The conservative state vector W is defined as

$$W = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \tag{2.2}$$

where $U = (u, v)^T$ is the velocity vector and ρ is the density. The convective fluxes f_i^c in the two coordinate directions $i \in \{x, y\}$ are

$$f_x^c = \begin{bmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ \rho H u \end{bmatrix}, \quad f_y^c = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v v + p \\ \rho H v \end{bmatrix}. \tag{2.3}$$

The pressure p for a calorically perfect gas is defined by the state equation

$$p = (\gamma - 1)\rho \{E - \frac{1}{2}U^2\}, \tag{2.4}$$

where E is the specific total energy per unit mass and γ is the gas dependent ratio of specific heats, which is 1.4 for air. The total enthalpy H is given by

$$H = E + \frac{p}{\rho}. \tag{2.5}$$

2.2 Finite Volume Discretization

In the theory of finite volumes the discretization is not applied to the differential form of the Euler equations (2.1), but to the integral form. The advantage in this approach lies in the fact that integral form allows easily ensuring conservation of mass, momentum and energy over an arbitrary control volume Ω. The integral form can be understood such that the rate of change

of each of the conservative quantities, integrated over Ω, equals the flux of each through the walls of Ω:

$$\frac{\partial}{\partial t}\int_{\Omega} W \, d\Omega + \oint_{\partial\Omega} f^c(W) \cdot n \, d(\partial\Omega) = 0. \tag{2.6}$$

The boundary of Ω is denoted $\partial\Omega$ and n its outer normal. The problem domain is decomposed into a grid of non-overlapping, cell-vertex median-dual control volumes that are approximate Voronoï volumes of a primary grid.

This semi-discretization in space leads to

$$\frac{\partial}{\partial t}\int_{\Omega_i} W_i \, d\Omega = -R(W_i), \tag{2.7}$$

where the residual R contains the full discretization of the integral over the convective fluxes.

Assuming that W varies linearly over the control volume Ω, we get

$$|\Omega_i| \frac{d\,W_i}{d\,t} = -R(W_i), \tag{2.8}$$

where $|\Omega_i|$ is the volume of Ω_i.

In the following we use both upwind and central convective fluxes depending on the problem at hand. The central flux is based on the Jameson-Schmitt-Turkel (JST) scheme [9], when upwind is used it is the AUSM-DV scheme of Wada and Liou [12]. Despite the various residual flux discretizations, the Jacobian used in LU-SGS remains the same, based on a first-order Lax-Friedrichs flux. Investigations have demonstrated that the effect of reducing the flux used in the Jacobian to first-order dominates the effect of any particular choice of flux function [5].

3 Unconstrained Implicit Runge-Kutta (UIRK) Methods

3.1 A Third-Order Semi-Implicit Runge-Kutta Scheme

In this section the semi-implicit third-order Runge-Kutta method (RK3_SI) developed by Nikitin [11], is described in relation to a general autonomous ordinary differential equation (2.8). This UIRK scheme is based on a particular third-order explicit Runge-Kutta scheme (RK3_EXP), where in each stage an arbitrary implicit stabilization term is added, whose order is the same as the truncation error of the stage, and which may therefore also be regarded as a perturbation or error term.

The three stages of the explicit part are:

$$|\Omega|\frac{W' - W^n}{\Delta t} = -\frac{2}{3}\,R(W^n), \tag{3.1}$$

$$|\Omega|\frac{W'' - W^n}{\Delta t} = -\frac{1}{3}\,R(W^n) - \frac{1}{3}\,R(W'), \tag{3.2}$$

$$|\Omega|\frac{W^{n+1} - W^n}{\Delta t} = -\frac{1}{4}\,R(W^n) - \frac{3}{4}\,R(W''). \tag{3.3}$$

which is a special case of the one-parameter family of third-order three-stage schemes with RK coefficient tableau

$$\begin{array}{c|ccc} 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & \frac{2}{3} - \frac{1}{4b} & \frac{1}{4b} & \\ \hline & \frac{1}{4} & \frac{3}{4} - b & b \end{array}$$

with $b = 3/4$ [3]. For reasons that will become clear, the construction of the semi-implicit method relies upon two features of this scheme: that the abscissas of the second and third stages are identical, so that W' and W'' are evaluated at the same time increment $t + \frac{2}{3}\Delta t$, and that the term $R(W')$ does not appear in the final quadrature for W^{n+1}.

Observe that the following relations hold for the above scheme:

$$W' = W(t^n + \frac{2}{3}\Delta t) + O(\Delta t^2), \tag{3.4}$$

$$W'' = W(t^n + \frac{2}{3}\Delta t) + O(\Delta t^3), \tag{3.5}$$

$$W^{n+1} = W(t^{n+1}) + O(\Delta t^4), \tag{3.6}$$

which we now perturb by adding an implicit term to each stage as follows:

$$|\Omega|\frac{W' - W^n}{\Delta t} = -\frac{2}{3}R(W^n) - \gamma L\,(W' - W^n), \tag{3.7}$$

$$|\Omega|\frac{W'' - W^n}{\Delta t} = -\frac{1}{3}R(W^n) - \frac{1}{3}R(W') - \gamma L\,(W'' - W'), \tag{3.8}$$

$$|\Omega|\frac{W^{n+1} - W^n}{\Delta t} = -\frac{1}{4}R(W^n) - \frac{3}{4}R(W'') - \gamma L\,(W^{n+1} - \tilde{W}^{n+1}). \tag{3.9}$$

Here the linear operator L is some approximation of the flux Jacobian $\partial R/\partial W$, and γ is a positive coefficient that controls the relative strength of the implicit terms. The need for the - as yet undefined - intermediate solution $\tilde{W}^{n+1}$ is explained later on.

First consider the right-hand side (RHS) of (3.7), which is obviously an $O(\Delta t)$ perturbation of the RHS of (3.1), which results in an $O(\Delta t^2)$ variation in W' and $R(W')$, i.e. the order of accuracy of these terms has not been reduced. Analogously the RHS of (3.8) is an $O(\Delta t^2)$ perturbation of the RHS of (3.2), resulting in an $O(\Delta t^3)$ variation in W'' and $R(W'')$, and again there is no reduction in accuracy. Note that this was only possible because W' and W'' were evaluated at the same time increment, otherwise $W'' - W'$ would have been $O(\Delta t)$.

Finally, assuming that $\tilde{W}^{n+1} = W(t^{n+1}) + O(\Delta t^3)$ the RHS of equation (3.9) is an $O(\Delta t^3)$ perturbation of the RHS of (3.3) causing an $O(\Delta t^4)$ variation in W^{n+1}. Thus at each stage and overall the semi-implicit scheme retains its order of accuracy. In particular the local error $W^{n+1} - W(t^{n+1})$ is still $O(\Delta t^4)$, and the UIRK scheme is third-order.

It remains to define $\tilde{W}^{n+1}$, a second-order accurate approximation to the solution at the full time increment, which may be written:

$$|\Omega|\frac{\tilde{W}^{n+1} - W^n}{\Delta t} = -\frac{1}{4}R(W^n) - \frac{3}{4}R(W') - \gamma L\,(\tilde{W}^{n+1} - \bar{W}^{n+1}), \tag{3.10}$$

where $\bar{W}^{n+1}$ is an $O(\Delta t)$ approximation of $W(t^{n+1})$, which is chosen as

$$\bar{W}^{n+1} = \frac{3}{2}(\alpha W' + (1-\alpha)W'') - \frac{1}{2}W^n, \tag{3.11}$$

where the real parameter α can be chosen arbitrarily.

Of course although the formal order of the method is preserved, the addition of effectively arbitrary terms must negatively influence its *absolute* accuracy. It makes sense therefore to choose as small a value for γ as possible, bearing in mind that the method is more implicit, and therefore more stable, for large values of γ. There is a compromise to be made for each problem, each time-step and each choice of L. For example, if the time-step is in a region for which the original explicit method is stable, then $\gamma = 0$ is very likely to be optimal, as the added "error" has been reduced as far as possible. Since the UIRK scheme is also about a factor of four more expensive than the baseline explicit scheme per time-step, it is therefore only attractive for time-steps greater than four times the maximum stable explicit time-step.

In Section 3.3 it will be shown that for linear R, and L the exact Jacobian, the scheme is A-stable for $\gamma \geq \frac{1}{3}$. But while it has been noted that LU-SGS is unconditionally stable for steady problems, this is only true for γ close to unity.

These issues will end up reducing the absolute accuracy of the scheme, and it therefore becomes desirable to develop UIRK methods of higher order. In the following section a fourth-order scheme is derived.

3.2 Extrapolation to a 4th Order Runge-Kutta Method

The construction of higher order Runge-Kutta-methods is generally very tedious work, and in the case of the semi-implicit scheme additional constraints must be satisfied, making it even more arduous. Firstly, the coefficients have to be chosen such that lower-order parts in the difference of intermediate solutions in perturbation terms cancel out. Secondly, if intermediate solutions are not at the same time-increment, additional intermediate levels at the appropriate time-increment have to be constructed.

On the other hand it is convenient and easy to obtain a higher order method by extrapolation. The disadvantage is that a scheme so constructed is likely to be considerably more expensive than a scheme more directly derived.

Here Richardson extrapolation is applied to the third order scheme. This linear extrapolation is based on two solutions: $W^{(n+\frac{1}{2})+\frac{1}{2}}$, which is computed in two steps of size $\Delta t/2$, and W^{n+1} which is obtained in one step of size Δt. By linear extrapolation we obtain a Runge-Kutta-method of 4th order (RK4_SI)

$$\hat{W}^{n+1} = \frac{1}{2^p - 1}(2^p \, W^{(n+\frac{1}{2})+\frac{1}{2}}(W^{n+\frac{1}{2}}) - W^{n+1}), \tag{3.12}$$

where $p = 3$ represents the order of the underlying scheme. Since four intermediate solutions of RK3_SI have to be used for the determination the solution $\hat{W}^{n+1}$ at one time-step Δt, the computational costs are three times higher than for RK4_SI. The corresponding explicit scheme RK4_EXP, may be written as a twelve-stage Runge-Kutta method, noting that the original scheme has three-stages.

The theoretical justification for extrapolation, which relies on asymptotic expansions of the error, is well-understood for non-stiff problems. For stiff problems, in contrast, the non-polynomial remainder may become unbounded, even if the stability properties of the method are otherwise good. Numerical experience shows that within the present test cases for Euler flow simulations the order of convergence and therefore the time-accuracy of the solution is not affected by the stiffness of the problem. However, for stiffer problems, such as Navier-Stokes flow computations, care should be taken to detect and avoid unboundedness in the remainder. Additional theoretical investigations would be helpful as well to ensure stability. The interested reader is referred to [1, 4, 7, 10] as a starting point.

3.3 Stability Analysis

Linear stability analysis is carried out for the 3rd-order semi-implicit Runge-Kutta method RK3_SI [11]; for the 4th-order method RK4_SI stability is investigated numerically.

The RHS of the system of ordinary differential equations is assumed to be linear and to coincide with the implicit operator $L = -R$. The rational approximation to the discrete system is determined from equations (3.7) to (3.10) as

$$W^{n+1} = Q(\Delta t \cdot L)W^n, \tag{3.13}$$

where the matrix-valued rational function $Q(z)$ with $z = \Delta t \cdot L$ results in:

$$\begin{aligned} Q(z) = (1-\gamma z)^{-4}[1 - (4\gamma - 1)z + (\gamma - \frac{1}{2})(6\gamma - 1)z^2 \\ - (4\gamma^3 - 6\gamma^2 + 2\gamma - \frac{1}{6})z^3 + \gamma(\gamma - \frac{1}{2})(\gamma^2 - 2\gamma + \frac{1}{3})z^4]. \end{aligned} \tag{3.14}$$

The scheme is called A-stable, if the following inequality is fulfilled for all z in the left-half complex plane:

$$|Q(z)| \leq 1, \tag{3.15}$$

which may be readily shown to hold for

$$\gamma \geq \frac{1}{3}. \tag{3.16}$$

The outcomes of the numerical test-cases in Section 5 show that the same stability behavior is observed for RK4_SI.

However, numerical experience also shows that most problems are only stable for γ close to one. This observation can be explained by the stability of the LU-SGS method, which is by no means an accurate approximation of the exact implicit linear system based on R, and is choosen rather becuase it is extremely cheap. In our case then the stability behavior of the complete RK3_SI and RK4_SI iterations is dominated by the stability of LU-SGS.

3.4 Local Error Estimation and Time-Step Control

A local error estimation and a time-step control algorithm based on the idea of embedded formulas [2, 11] can be implemented for the third- and fourth-order Runge-Kutta methods to attempt to achieve a given local error with minimal computational effort. An estimate of the local error can be determined very easily and cheaply, since two approximations of $W(t^{n+1})$ of two different orders are already available in both schemes.

For the third-order Runge-Kutta scheme (RK3_SI), we obtain W^{n+1} from the execution of the last stage and $\tilde{W}^{n+1}$ from execution of the additional stage of the semi-implicit scheme:

$$W^{n+1} = W(t^{n+1}) + O(\Delta t^4) \qquad \tilde{W}^{n+1} = W(t^{n+1}) + O(\Delta t^3). \tag{3.17}$$

The local error ϵ and estimate of the local error $\tilde{\epsilon}$ are defined respectively

$$\epsilon = \|W^{n+1} - W(t^{n+1})\| \approx O(\Delta t^4), \tag{3.18}$$

$$\tilde{\epsilon} = \|W^{n+1} - \tilde{W}^{n+1}\| \approx O(\Delta t^3), \tag{3.19}$$

whereby $\|\cdot\|$ is some appropriate norm.

Similarly for RK4_SI we take the result of the Richardson extrapolation $\hat{W}^{n+1}$, which is of $O(\Delta t^5)$ and the result of the last stage of RK3_SI, W^{n+1}, which is of $O(\Delta t^4)$

$$\hat{W}^{n+1} = W(t^{n+1}) + O(\Delta t^5) \qquad W^{n+1} = W(t^{n+1}) + O(\Delta t^4). \tag{3.20}$$

The local error can again be estimated applying an appropriate norm

$$\tilde{\epsilon} = \|\hat{W}^{n+1} - W^{n+1}\|. \tag{3.21}$$

Since the order of the error estimate is in both cases smaller than the order of the true error, for sufficiently small Δt the inequality $\epsilon < \tilde{\epsilon}$ holds, and $\tilde{\epsilon}$ can be regarded as an upper bound on ϵ.

An improved time-step for the current iteration, Δt_{actual}, can now be found by multiplying the actual time-step with the factor λ, so that

$$\Delta t_{\text{actual}} = \lambda \cdot \Delta t, \qquad \lambda = \left(\frac{\tau}{\tilde{\epsilon}}\right)^{\frac{1}{p+1}}, \tag{3.22}$$

where τ is a given error tolerance and p is the order of the particular Runge-Kutta scheme. If it turns out that $\lambda < \lambda_{\min}$ (which is here taken to be $\lambda_{\min} = 0.5$), then the step that was taken using Δt is considered to have been unsucessful, and is repeated starting with the stored solution W^n and using the smaller time-step $\Delta t_{\text{actual}} = \lambda_{\min} \cdot \Delta t$. Otherwise the stage is considered to have been successful and the time-step for the next stage is computed as

$$\Delta t_{\text{new}} = \lambda \cdot \Delta t \qquad \lambda = \min\{\lambda, \lambda_{\max}\}, \tag{3.23}$$

where the upper limit on λ is set to $\lambda_{\max} = 1.5$, to prevent the time-step exploding if the error estimator happens to be very close to zero.

4 Lower-Upper Symmetric Gauss-Seidel (LU-SGS)

The biggest advantage of a semi-implicit scheme over a fully implicit scheme lies in the fact that the time-accuracy does not depend on the exactness of the flux Jacobian. This particular feature allows the choice of an implicit operator only with regard to efficiency, storage requirements and stability.

As already discussed, there are two choices to be made in the construction of an implicit operator, the linear system solver and the flux Jacobian approximation. These choices are however closely coupled. There is no point in solving the linear system of equations accurately if the Jacobian is only a very rough approximation. Similarly it makes no sense to construct a perfect Jacobian and then apply only one Jacobi iteration for example.

The demand of low memory requirements excludes the possibilty of using an exact second-order Jacobian. Using a Jacobian based on first-order convective fluxes represents a major reduction in accuracy, but is a necessity. Given this approximation, the additional approximation of applying differing flux functions in the Jacobian and residual has little effect on time-accuracy, and this result will be used in Section 4.2 to choose a flux function that results in a particularly simple and well-conditioned Jacobian (the well-conditionedness being important for the solution of the implicit linear system).

Given the simplicity of this Jacobian, a highly accurate linear solver is not required, and symmetric Gauss-Seidel (SGS) is applied. In principle SGS provides communication between every pair of nodes in the grid within one iteration, thereby removing any Courant-Friedrichs-Levy (CFL) restriction. Also, investigations have shown that a single SGS sweep provides the best compromise between stability and iteration cost [5].

The end result of these considerations is the Lower-Upper Symmetric Gauss-Seidel (LU-SGS) iteration [15, 5], which is in practice unconditionally stable for inviscid cases. In the following only the basics of the particular LU-SGS algorithm implemented in the DLR Tau-Code are described; for more detailed information see [5].

4.1 Linear System Solution

The linear system can be rewritten by decomposing the implicit system matrix A into a lower triangular part L, a diagonal matrix D and an upper triangular part U,

$$A \cdot x = (L + D + U) \cdot x = b, \tag{4.1}$$

where the solution x corresponds to the update ΔW and b to some RHS, which in the simplest case is just the residual R.

Two possible Gauss-Seidel iterations with unknown x^n, here denoted as forward sweep and

backward sweep, are:

$$(D + L)x^{n+1} = b - U \cdot x^n \tag{4.2}$$
$$(D + U)x^{n+1} = b - L \cdot x^n. \tag{4.3}$$

If either of the these iterations converges, then $x^{n+1} = x^n = x$ is the solution of the linear system. For the forward sweep x_i^{n+1} is a function of all x^n as well as x_j^{n+1} for $j < i$, while for the backward sweep x_i^{n+1} is a function of all x^n and x_j^{n+1} for $j > i$. A *symmetric* Gauss-Seidel iteration, a composite of the two sweeps, can therefore be written as

$$(D + L)x^* = b - U \cdot x^n \tag{4.4}$$
$$(D + U)x^{n+1} = b - L \cdot x^*, \tag{4.5}$$

where x_i^{n+1} will be a function of all x^n and x^{n+1}, which in turn implies automatic satisfaction of the CFL condition.

A subvariant of SGS can be developed by applying only a single SGS iteration, which was in fact found to be the optimal number of iterations. Restricting the starting value to $x^0 = 0$, the system of equations (4.4) may be written

$$(D + L)x^* = b \tag{4.6}$$
$$(D + U)x^1 = b - L \cdot x^*. \tag{4.7}$$

This so-called LU-SGS iteration may be rewritten

$$(D + L) \cdot D^{-1} \cdot (D + U)x^1 = b, \tag{4.8}$$

which form suggests choosing an approximate flux Jacobian such that the diagonal block only contains elements on its diagonal, as the diagonal blocks are the only blocks that must be inverted, reducing memory requirements and CPU time significantly. Such a flux Jacobian is described in the next section.

4.2 Approximation of the Convective Flux Jacobians

In the finite volume method the integral of a flux over the boundary of a control volume $\partial\Omega$ is approximated by the sum of internal numerical fluxes $\hat{f}$, and boundary fluxes $\hat{f}_b$, over all facets (or faces) of Ω. In a first-order upwind scheme the unknown variables on the left and right sides of a face, W_L and W_R, are approximated by piecewise constant reconstruction of the cell-centered values, so that if W_i is the average value of W on cell i, then $W_L = W_i$ and $W_R = W_j$.

Given this, the residual R in cell i may be written

$$R_i = \sum_{j \in \mathcal{N}(i)} \hat{f}(W_L, W_R; n_{ij}) + \sum_{m \in \mathcal{B}(i)} \hat{f}_b(W_L; n_m), \tag{4.9}$$
$$= \sum_{j \in \mathcal{N}(i)} \hat{f}(W_i, W_j; n_{ij}) + \sum_{m \in \mathcal{B}(i)} \hat{f}_b(W_i; n_m), \tag{4.10}$$

where $\mathcal{N}(i)$ is the set of all intermediate neighbours of grid point i, and $\mathcal{B}(i)$ is the set of all neighbouring boundary faces.

The numerical flux $\hat{f}$ can be expressed in dissipation form as

$$\hat{f}(W_L, W_R; n_{ij}) = \frac{1}{2}(f^c(W_L) + f^c(W_R)) \cdot n_{ij} - \frac{1}{2}D(W_L, W_R; n_{ij}), \tag{4.11}$$

where f^c is the exact convective flux tensor. Inserting equation (4.11) in equation (4.9), the residual may be rewritten as

$$R_i = \frac{1}{2}\, f^c(W_i) \cdot \sum_{j \in \mathcal{N}(i)} n_{ij} + \frac{1}{2} \sum_{j \in \mathcal{N}(i)} f^c(W_j)\, n_{ij}, \tag{4.12}$$

$$- \frac{1}{2} \sum_{j \in \mathcal{N}(i)} D(W_i, W_j; n_{ij}) + \sum_{m \in \mathcal{B}(i)} \hat{f}_b(W_i; n_m). \tag{4.13}$$

For a closed control volume i, not touching any boundaries, the sum of the normal vectors must equal zero

$$\sum_{j \in \mathcal{N}(i)} n_{ij} = 0. \tag{4.14}$$

The diagonal of the Jacobian is determined by differentiating R_i with respect to W_i. For a non-boundary control volume, it becomes obvious that the diagonal elements only depend on the dissipation D

$$\frac{\partial R_i}{\partial W_i} = -\frac{1}{2} \sum_{j \in \mathcal{N}(i)} \frac{\partial D(W_i, W_j; n_{ij})}{\partial W_i}, \tag{4.15}$$

and thus it is a good strategy to select a flux function with a simple dissipation component to obtain a simple Jacobian matrix block diagonal. For example the first-order Lax-Friedrichs flux

$$\hat{f}_{\mathrm{LF}}(W_L, W_R; n_{ij}) = \frac{1}{2}(F(W_L; n) + F(W_R; n)) - \frac{1}{2}|\lambda|(W_R - W_L), \tag{4.16}$$

where λ is treated as a constant in the differentiation. In this case differentiating the dissipative part D_{LF} with respect to W_i leads to positive scalar multiple of the identity matrix and thus to cheap to store and easy to invert in the LU-SGS scheme. In particular

$$\frac{\partial D_{FL}(W_i, W_j; n_{ij})}{\partial W_i} = |\lambda| I. \tag{4.17}$$

In contrast off-diagonal blocks in contrast are constructed explicitly. For the Lax-Friedrichs scheme they are

$$\frac{\partial R_i}{\partial W_k} = \frac{1}{2}\frac{\partial f^c(W_k) \cdot n_{ik}}{\partial W_k} - \frac{1}{2}|\lambda| I, \qquad k \neq i,\ k \in \mathcal{N}(i). \tag{4.18}$$

Applying the Jacobian flux approximation in combination with the LU-SGS solver results in a very efficient scheme in terms of memory requirements and CPU-time. The only parts of the Jacobian that needs to be stored are the inverted diagonal blocks. The elements of the off-diagonal blocks are computed on-the-fly.

5 Numerical Examples

The semi-implicit, time-accurate Runge-Kutta-methods of 3rd (RK3_SI) and 4th order (RK4_SI) as well as their explicit counterparts RK3_EXP and RK4_EXP have been applied to several two-dimensional problems. They are compared to existing time-accurate time integration methods in the TAU-Code, a 2nd order two-stage explicit Runge-Kutta method (RK2_EXP) and a fully implicit 2nd and 3rd order dual-time scheme, denoted DUAL_2 and DUAL_3 respectively (whereby it was ensured that the dual-time inner iterations were fully converged at each time step, to eliminate any error due to non-zero residuals).

The aim of these comparisons is to determine the stability and relative efficiency of the various schemes. Efficiency will be measured in terms of CPU time required to achieve a given accuracy, where accuracy will always be determined with respect to a highly converged reference solution. In this context stability restrictions may be seen as placing a lower limit on the CPU time required to obtain *any* given accuracy. For example, as already discussed, the UIRK schemes will always be less accurate than the corresponding explicit schemes for a given time step, but the explicit schemes have a very small maximum stable time step which may be unsuitable for practical applications.

In the following the stability limits of the methods will be implicitly shown through termination of the error curve for large time steps.

5.1 An Expansion Fan in a Shock tube on a Structured Grid

In the first test-case a Riemann problem of an expansion fan in a shock tube is simulated. The one-dimensional solution is modelled on the two-dimensional structured grid shown in Figure 5.1, which has cells varying in aspect ratio from $1:1$ on the lower wall to $1:100$ near the upper wall. This irregular cell distribution was chosen in an attempt to model a typical computational grid, where cell sizes and aspect ratios vary significantly over the domain. Choosing an isotropic grid would favour explicit methods unrealistically, as local CFL conditions would lead to broadly similar time steps over the entire domain.

The flow is initialized as a discontinuity, and after a short time the normalized pressure and density are a shown in Figure 5.2. The presence of a shock in the solution was deliberately avoided due to spatial discretization effects such as limiter switching. The AUSM-DV scheme was used with Green-Gauss reconstruction.

A convergence study on the time step Δt for the various schemes is depicted in Figure 5.3. The L^2-error of the pressure over all points in the field is plotted logarithmically against the time-step, so that the gradient of each curve is the achieved order of that method.

The explicit scheme RK2_EXP is plotted in the range starting with the largest stable time-step to the point where the error can not be reliably estimated due to machine rounding errors. Both fully implicit dual-time schemes are unconditionally stable, but require a considerable amount

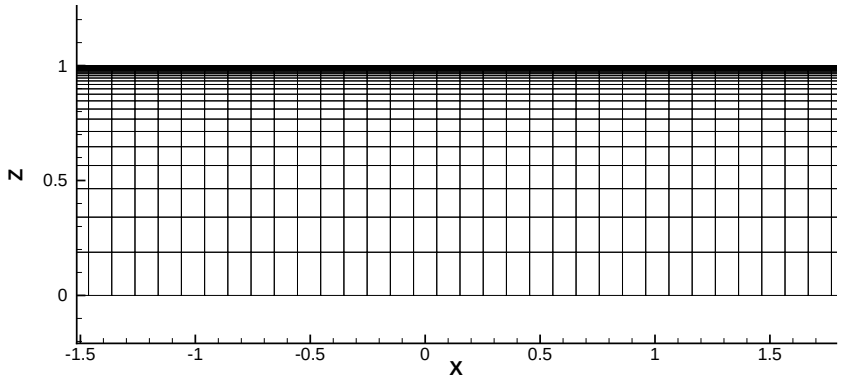

Figure 5.1: Part of structured shock tube grid for the Riemann problem test case. Cell aspect ratio reaches $1:100$.

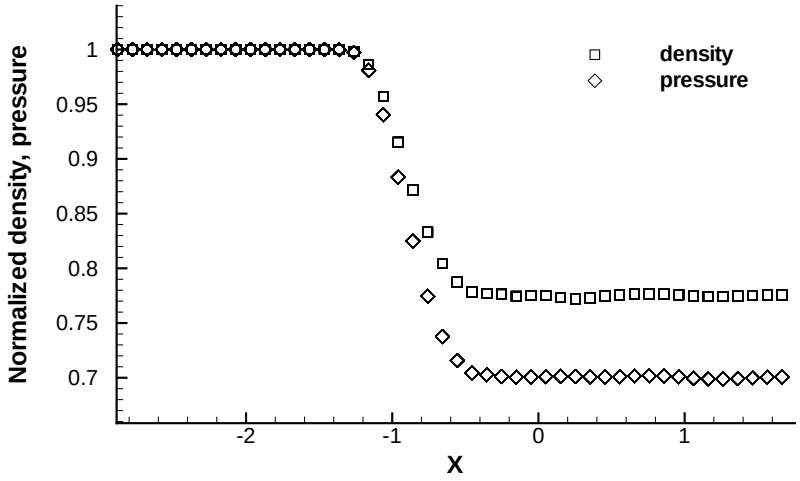

Figure 5.2: Normalized density and pressure distributions of the expansion fan shortly after initialization.

of computational time, and are therefore plotted in a range starting with a time step close to the upper limit for obtaining a reasonable solution, down to the point where the calculation is too expensive to be practical in any engineering situation. In contrast the UIRK results may be readily calculated for the entire time step range, as shown. Over the range for which the explicit 3rd and 4th order Runge-Kutta schemes were stable the resulting error was too small to estimate reliably.

For each scheme the order of convergence is obtained as expected. Both semi-implicit methods RK3_SI and RK4_SI start with a convergence of approximately 2nd order, which increases for smaller time steps to the formal asymptotic order of 3 and 4 respectively. This variation of order is an expected consequence of the addition of the potentially large implicit "error" terms. While they certainly decay at the theoretical rate as $\Delta t \to 0$, for finite and large Δt there is no reason why they may not dominate any other term present. In practical situations this could make it difficult to be certain that the method was behaving time-accurately at all, hence the necessity for step-size control of Section 3.4. It is also notable that for any given time-step the explicit scheme is more accurate than the UIRK schemes, again an expected consequence of

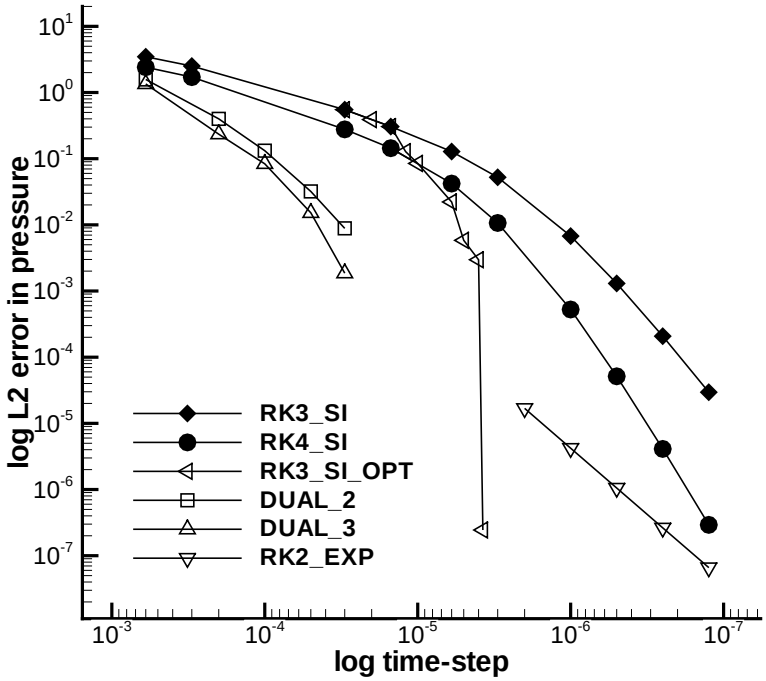

Figure 5.3: Error in solution against time-step for all time integration schemes considered for the Riemann problem.

artificially increasing the level of truncation error.

Curve RK3_SI_OPT shows an attempt to obtain lower errors and potentially 3rd order convergence for larger time-steps by choosing the smallest value of the coefficient γ such that the UIRK method is stable for each time-step. Unfortunately the order could not be improved, as γ could only be reduced to a value significantly less than one as the stability limit of the explicit method was approached. This is likely a consequence of the LU-SGS scheme, which is unconditionally stable only for γ close to one, rather than a feature of the UIRK scheme itself, hence alternative implicit operators are likely to modify this result.

The more practically relevant CPU-time against error results are shown in Figure 5.4. The CPU-times were not measured for each run individually, as variations in load and implementational differences tend to make the results too noisy. Rather the relative costs of individual iterations were estimated on the basis of algorithmic complexity, and the results were verified against representative solver runs. The base unit was taken to be the cost of one LU-SGS iteration. Hence the dual-time schemes are very costly schemes and need about 100 LU-SGS iterations per time-step. RK2_EXP is, with a CPU-time that approximately equals 1.5 LU-SGS iterations, the cheapest scheme. The semi-implicit methods are relatively cheap as well. RK3_SI takes 6 and RK4_SI 18 LU-SGS iterations for the completion of all stages and the explicit part for one time-step. Based on these factors the L^2-errors, multiplied by the number of corresponding LU-SGS iterations, are plotted against the CPU-time in Figure 5.4.

The explicit scheme RK2_EXP provides the highest accuracy for a given time-step, but the serious deficiency of the low stability limit becomes obvious as well. If an L^2-error of order one is required for example, rather than the less realistic 10^{-5}, then the UIRK scheme represent a saving of a factor 10 over RK2_EXP in CPU time. The semi-implicit schemes achieve a similar

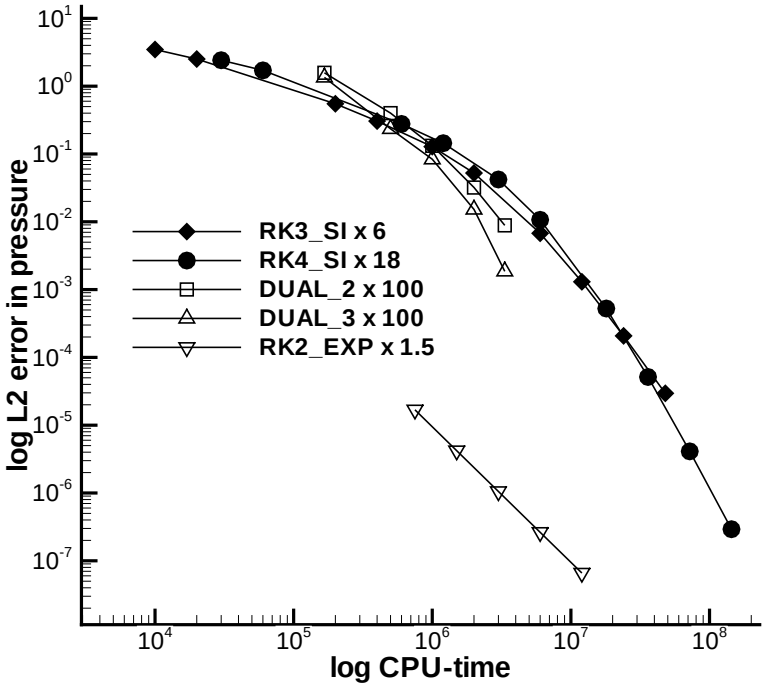

Figure 5.4: Error in solution against normalized CPU-time for the Riemann problem.

performance to the dual-time schemes.

The simulation of the expansion fan demonstrates very well the extension of the stability limit compared to fully explicit methods. Further two test cases show that a significant speed up over implicit dual-time schemes may be achieved in addition.

5.2 Euler Flow over a Forward Facing Step on a Rectangular Grid

Inviscid flow over a forward facing step, impulsively starting from rest to a Mach number $M = 3$ is simulated in two dimensions on a uniform rectangular grid, shown in Figure 5.5. A shock wave emerges in front of the step and eventually grows sufficiently to be reflected at the channel walls. After some time the flow achieves a stationary behavior. The pressure distribution of the partially developed flow after a single shock reflection is also depicted in Figure 5.5.

As before a convergence study of all implemented schemes is shown in Figure 5.6, where the L^2-error in pressure is plotted logarithmically against the time-step. In this case the semi-implicit methods meet their order of convergence immediately. Most interestingly the presence of the convergence curves for the explicit RK schemes RK3_EXP and RK4_EXP allow direct inspection of the magnitude of the error contributed by the implicit stabilization, and it's seen to be about three orders of magnitude in the L^2-error. This large decrease in accuracy would be of more concern if it wasn't for the fact that the fully implicit schemes have a similar disadvantage over the explicit schemes in this case. As it stands the 4th-order UIRK scheme achieves approximately the same accuracy for any given time-step as 2nd and 3rd order dual-time in this case. We might therefore expect a significant gain in efficiency in terms of CPU time through use the

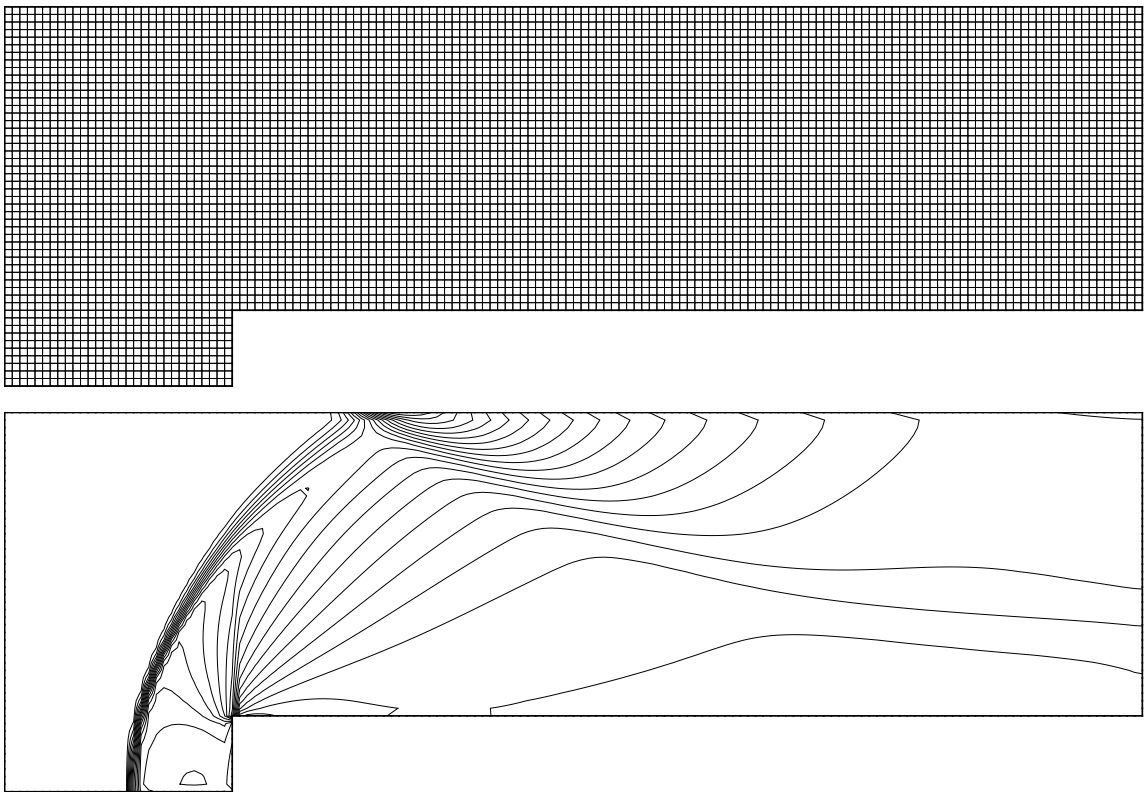

Figure 5.5: Partially developed flow over a forward facing step at Mach 3, with computational grid.

semi-implicit scheme.

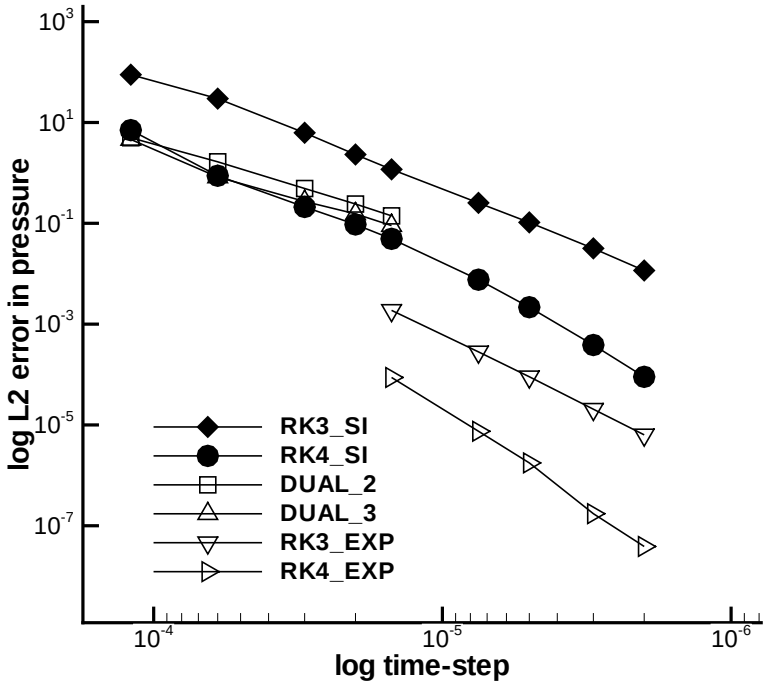

Figure 5.6: Error in solution against time-step for all time integration schemes considered for the forward-facing step.

The improvement in efficiency for both semi-implicit methods is demonstrated in Figure 5.7. The accuracy of the commonly applied dual-time methods for the two largest step sizes is taken as an approximate reference point for typical engineering accuracy. In this range the UIRK schemes provide a reduction in CPU-time of a factor of 3 to 10 over the fully implicit methods. Notable is that the 4th-order UIRK scheme looses much efficiency in comparison to the 3rd-order scheme because of its high per-step expense, indicating that a 4th-order UIRK method with a significantly lower stage count would be very desirable. RK3_EXP and RK4_EXP perform also excellently here, primarily as a consequence of the uniform cell size in the mesh.

The results of the forward facing step problem demonstrate the potential of the presented methods regarding stability and efficiency compared to fully implicit methods.

5.3 Euler Flow over a Ramp on a Triangular Grid

The last test case considers again impulsively started two-dimensional inviscid flow over a ramp at Mach 3. In contrast to the forward facing step problem the grid consists here of triangles which decrease slightly in size as they approach the walls of the domain, and again whose aspect ratios do not vary significantly. The resulting pressure distribution is plotted in Figure 5.8.

Again a convergence study was made, see Figure 5.9, where it can be seen that the 4th-order semi-implicit scheme perform better than both fully implicit schemes in terms of accuracy for a given time step size.

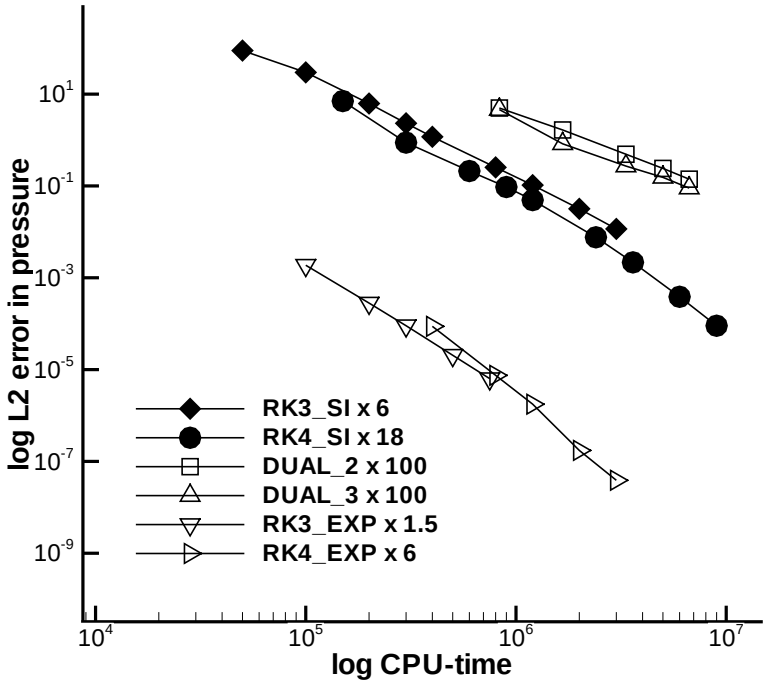

Figure 5.7: Error in solution against CPU-time for the forward-facing step.

The corresponding CPU-time curves are depicted in Figure 5.10, with similar results to the previous case. In particular the UIRK schemes still outperform dual-time by a factor of 3 to 10, and despite the variations in cell size, the explicit schemes are still significantly more efficient than dual-time.

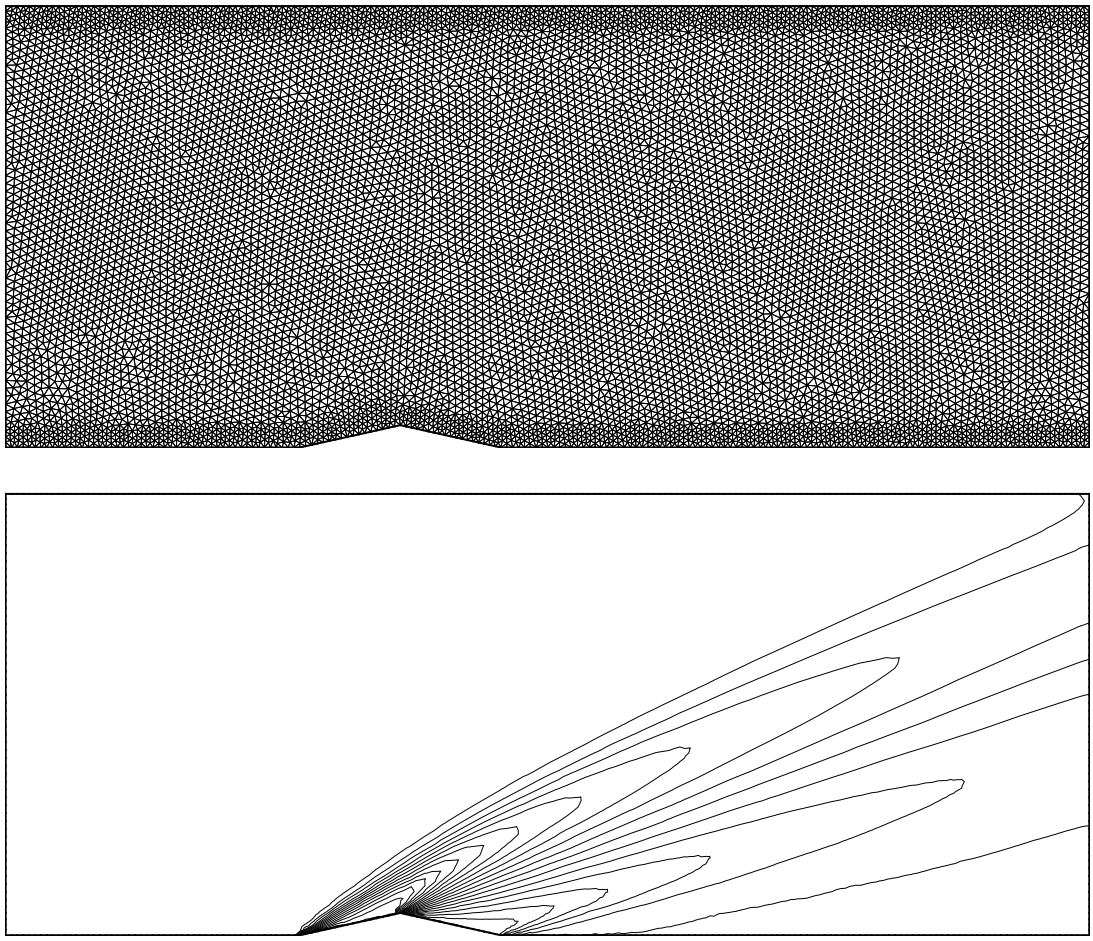

Figure 5.8: Grid and partially developed pressure contours over the wedge.

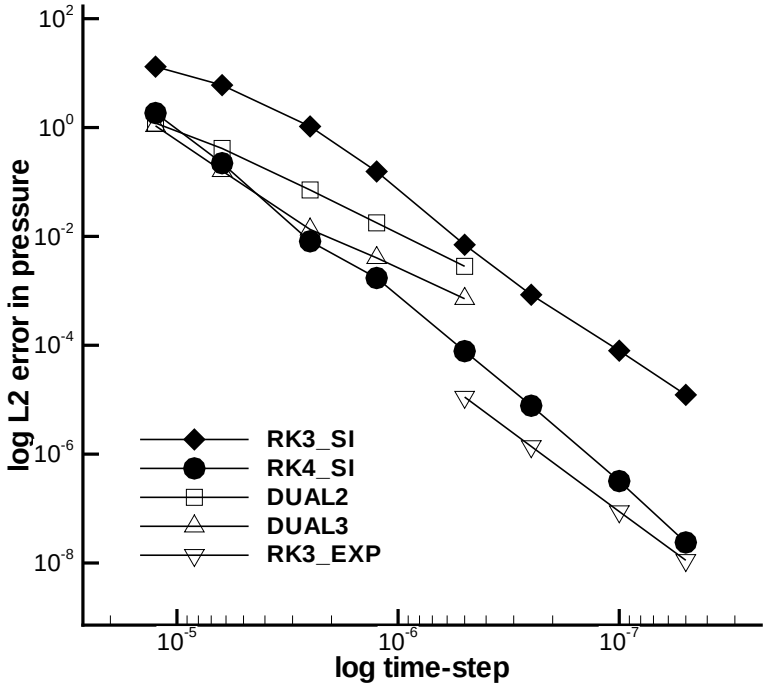

Figure 5.9: Error in solution against time-step for all time integration schemes considered for the supersonic wedge.

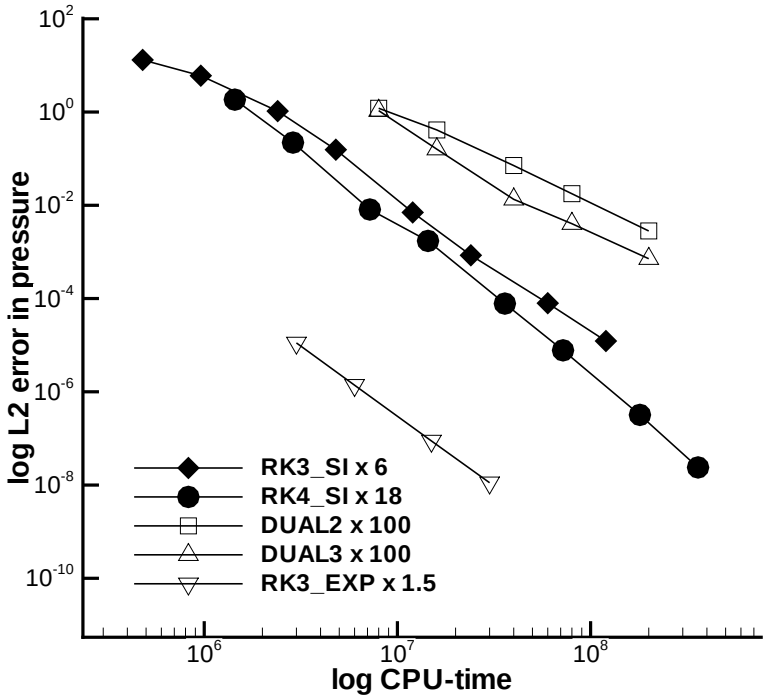

Figure 5.10: Error in solution against CPU-time for the supersonic wedge.

6 Conclusions

Two semi-implicit Runge-Kutta-methods of 3rd (RK3_SI) and 4th order (RK4_SI) and their application to two-dimensional compressible Euler flows have been presented in this paper. The RK3_SI scheme was taken from the work of Nikitin [11], while RK4_SI was developed by applying linear Richardson extrapolation to RK3_SI. Both methods provide local error estimation and time-step control based on embedded lower-order methods.

Semi-implicit methods are generally expected to show a considerably higher stability than fully explicit schemes and they should outperform fully implicit methods regarding CPU-time. This behavior is achieved by careful construction of the methods such that the order of convergence is completely independent of their implicit parts. Therefore we classify RK3_SI and RK4_SI as "Unconstrained-Implicit Runge-Kutta-methods" (UIRK); the term "unconstrained" emphasizes here the complete absence of constraints on the implicit operator. This unique feature allows the choice of the stabilization term to be made with respect to the demands of stability only, independent of accuracy. A highly tuned, low-cost LU-SGS scheme is therefore chosen, which is unconditionally stable for inviscid flows, while being of comparable computational expense to a three-stage RK method.

Several numerical examples are computed on various structured and unstructured grids within the finite volumes RANS solver, the DLR TAU-Code. All test cases demonstrate that the order of convergence is indeed independent of the choice of the linear operator. The gain in CPU time for given accuracy reaches a factor of 10 compared to different dual-time schemes for several test cases.

The presented semi-implicit methods show significant potential to speed up time integration for unsteady flow solvers. However, within this preliminary paper they are only applied to comparatively simple inviscid problems. Further investigations consists of extending the methods to high-Reynolds number Navier-Stokes flows. An implicit operator with accurate treatment of high aspect ratio boundary-layer cells, which dominate the stability of such problems, is likely to be necessary. This could take the form of line-implicit operators. Further, the development of a cheaper 4th-order UIRK scheme has been seen to be of considerable interest.

Bibliography

[1] G. Bader and P. Deuflhard. A semi-implicit mid-point rule for stiff systems of ordinary differential equations. *Numerische Mathematik*, 41:373–398, 1983.

[2] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier Science, 2001. ISBN 0-08-043009-0.

[3] J. C. Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.

[4] P. Deuflhard and F. Bornemann. *Scientific Computing with Ordinary Differential Equations*, volume 42 of *Texts in Applied Mathematics*. Springer, 2002.

[5] Richard Dwight. *Efficiency Improvements of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids*. PhD thesis, School of Mathematics, University of Manchester, 2006.

[6] T. Gerhold, M. Galle, O. Friedrich, and J. Evans. Calculation of complex 3D configurations employing the DLR TAU-Code. In *American Institute of Aeronautics and Astronautics, Paper AIAA-97-0167*, 1997.

[7] E. Hairer and C. Lubich. Extrapolation of stiff differential equations. *Numerische Mathematik*, 52:377–400, 1988.

[8] A. Jameson. Time dependant calculations using multigrid with applications to unsteady flows past airfoils and wings. *AIAA Paper, AIAA-91-1596*, 1991.

[9] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In *AIAA Paper, AIAA-81-1259*, 1981.

[10] P. Kaps, S.W.H. Poon, and T.D. Bui. Rosenbrock methods for stiff ODEs: A comparison of Richardson extrapolation and embedding technique. *Computing*, 34:17–40, 1984.

[11] N. Nikitin. Third-order-accurate semi-implicit runge-kutta scheme for incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 51:221–233, 2006.

[12] Y. Wada and M.-S. Liou. A Flux Splitting Scheme with High-Resolution and Robustness for Discontinuities. *Collection of Papers on Aerospace Science*, pages 10–13, January 1994.

[13] J.J. Yoh and X. Zhong. New hybrid Runge-Kutta methods for unsteady reactive flow simulation. *AIAA Journal*, 42(8):1593–1600, 2004.

[14] J.J. Yoh and X. Zhong. New hybrid Runge-Kutta methods for unsteady reactive flow simulation: Applications. *AIAA Journal*, 42(8):1593–1600, 2004.

[15] S. Yoon and A. Jameson. An LU-SSOR scheme for the Euler and Navier-Stokes equations. *AIAA Journal*, 26:1025–1026, 1988.

IB 124-2007/3

A Fourth Order Semi-Implicit Runge-Kutta Method for the Compressible Euler Equations

Ursula Mayer, Richard P. Dwight

Verteiler:

Institut für Aerodynamik und Strömungstechnik, BS	1	Exemplar
Institut für Aerodynamik und Strömungstechnik, GÖ	1	Exemplar
Verfasser ..... je	3	Exemplare
N. Kroll	1	Exemplar
C.-C. Rossow	1	Exemplar
D. Schwamborn	1	Exemplar
Deutsche Bibliothek Frankfurt am Main	2	Exemplare
Zentralbibliothek	2	Exemplare
Reserve	10	Exemplare
	25	Exemplare