Implicit Explicit Residual Smoothing for Upwind Schemes

Rickard Enander, Björn Sjögreen
Department of Scientific Computing,
Uppsala University,
Box 120, 751 04 Uppsala, Sweden,

October 7, 1996

Abstract

In this paper we will investigate if implicit residual smoothing can be used together with upwind discretizations when steady state solutions are computed. We will numerically examine the second order accurate Roe's method in different flow situations i two and three space dimensions. These experiments indicate that a significant amount of CPU-time can be saved by residual smoothing and especially the explicit version.

Classification 65N05,76G15.

Key words Finite differences, Euler equations, Upwind schemes, Convergence acceleration

1 Introduction

The computation of steady state solutions to hyperbolic systems, such as the Euler equations, are today of interest in various parts of industrial development. The inviscid flow equations are often written in the following form:

$$\frac{\partial}{\partial t} w = \sum_{i=1}^{m} \frac{\partial}{\partial x_i} F_i(w). \tag{1}$$

In (1) m is the number of space dimensions, w some state vector and F_i flux functions.

In this paper we are interested in solving for the steady state by an upwind spatial discretization. We will use the time dependent equation to iterate to a steady state solution. We will present a new Runge-Kutta method, suited for both TVD and centered approximation of spatial derivates.

An iteration process of this kind is easily implemented but has the drawback of slow convergence due to the explicit time stepping. Several techniques can be used to accelerate the convergence to the steady state. Local time stepping means that the time step is not uniform over the whole computational domain. In areas where the grid cells are larger, the time step is allowed to be larger as well. Multigrid techniques are also in frequent use. In this paper we will focus on the concept of *implicit residual smoothing, IRS*, which was introduced for schemes based on centered discretization with artificial viscosity in space and Runge-Kutta time stepping. IRS can also be used for upwind discretization. In IRS there is one free parameter, which in this paper will be called α, describing the increase in effective time step. This increase is balanced by an implicit operator to ensure the stability of the scheme. It has been observed that this

parameter cannot be larger than 2-3 in real applications due to a deterioration of the damping properties. In the original form, [5], the implicit part is taken in a symmetric centered way without regard to wave propagation directions.

$$\begin{aligned} R_{i,j}^{IRS} &= IRS(\alpha)\, R_{i,j} \\ IRS(\alpha) &= \alpha \cdot (1 - \beta\Delta_{\pm,x})^{-1} (1 - \beta\Delta_{\pm,y})^{-1} \end{aligned} \quad . \tag{2}$$

In (2) $R_{i,j}$ is the 2D spatial discretization computed by, for instance, an upwind scheme. In the basic iteration, $R_{i,j}$ is used to iterate in time. $R_{i,j}^{IRS}$ is the spatial discretization, modified by multiplying with α and smoothed implicitly, which is used in the time iteration instead of $R_{i,j}$ to accelerate the scheme.

In connection with upwind schemes there has been proposals of improvements by performing the smoothing in an upwind way. One way is described in [1] as:

$$\begin{array}{lcccl} -\epsilon\tilde{R}_{i-i,j}^{+} & + & (1+\epsilon)\,\tilde{R}_{i,j}^{+} & & = R_{i,j}^{+} \\ & & (1+\epsilon)\,\tilde{R}_{i,j}^{-} & -\epsilon\tilde{R}_{i+i,j}^{-} & = R_{i,j}^{-} \end{array} \quad . \tag{3}$$

In (3) R^{+} is the part of the spatial discretization corresponding to positive eigenvalues and correspondingly for R^{-}. Numerical experiments indicate that the number of iterations to obtain a steady state can be reduced. However, the CPU-time is reduced only marginally due to the cost of eigenvalue decomposition.

We will here investigate a different concept: *implicit explicit residual smoothing, IERS* to accelerate convergence properties of upwind schemes. IERS was introduced for Jameson type of schemes, see [2, 3].

2 The model equation in 1D

Let us consider a simple scalar equation in 1D to explore the general ideas and basic features of residual smoothing:

$$\begin{array}{lll} v_t & = -v_x & 0 \le x \le 2\pi \quad , \quad 0 \le t \\ v(x,0) & = f(x) & \\ v(0,t) & = g(t) & \end{array} \quad . \tag{4}$$

Define a grid

$$x_j = j \cdot h, \qquad j = [0, 1, \ldots, N], \qquad h = 2\pi/N \tag{5}$$

and introduce a grid function $U = \{u_j\}_{j=0}^{N}$ approximating the solution v in (4). The term v_x is approximated by the grid function U using an upwind difference:

$$\begin{array}{ll} \frac{\partial}{\partial x} u_j & \rightarrow \frac{1}{h} \Delta_+ u_j \\ \Delta_+ u_j & = (u_{j+1} - u_j) \end{array} \tag{6}$$

When the discretization in space is defined, we have a system of ordinary differential equations,

$$\frac{\partial}{\partial t} U = R(U) \tag{7}$$

which is integrated by an explicit Runge-Kutta method.

The numerical scheme described so far is easily implemented but converges poorly in general. The scheme is explicit and the time step Δt must be of the same order as the spatial step size h. To analyze schemes of this kind the Fourier transform is often used. We then assume periodic boundary conditions and express the solution in Fourier components $\hat{u}$:

$$\begin{array}{l} \{\hat{u}_\omega\}_{\omega=-r}^{r} \\ u_j = \sum_{\omega=-r}^{r} \hat{u}_\omega e^{i\omega x_j} \end{array} \tag{8}$$

with $r = N/2$. Here ω represents the *wave numbers* of the Fourier components with $|\omega|$ small meaning smooth components and $|\omega|$ large, oscillatory ones. By inserting (8) in (6) we get the Fourier symbol of the spatial discretization:

$$\widetilde{R}(\xi) = \frac{1}{h} \cdot e^{i\xi/2} \cdot 2i \cdot \sin(\xi/2) = \frac{1}{h} \cdot \Big(\cos(\xi) + i \cdot \sin(\xi) - 1 \Big) \tag{9}$$

where $\xi = \omega h$. In the top of Fig. 1 the Fourier symbol of the upwind discretization is plotted together with the stability region of a three-stage Runge-Kutta method. The ratio, λ, between the time step, Δt, and h is in this case 0.7.

By applying the implicit residual smoothing operator to the spatial discretization the Fourier symbol can be modified. The aim is to use as large effective time step as possible. The IRS is applied to the spatial discretization, R, and the general form in 1D is:

$$\begin{aligned} R_j^{IRS} &= IRS(\alpha)\, R_j \\ IRS(\alpha) &= \alpha \cdot (1 - \beta \Delta_{\pm})^{-1} \end{aligned} \tag{10}$$

with Fourier symbol

$$\widetilde{IRS}(\xi) = \frac{\alpha}{\Big(1 + 4\beta \sin^2\Big(\frac{\xi}{2}\Big)\Big)}. \tag{11}$$

The parameter α represents the increase in effective time step. Thus we would like to make it as large as possible. The parameter β is determined by the choice of α and must be of order α^2 for the scheme to remain stable. Our choice is

$$\beta = \frac{1}{4}\Big(\alpha^2 - 1\Big). \tag{12}$$

The drawback of the IRS is that the Fourier symbol of the total spatial discretization is contracted towards the imaginary axis, as indicated by the plots

in Fig. 1 for $\alpha = 15$. This results in poor damping of oscillatory modes for large choices of α. By examining (11) the contraction of the symbol is of order α in the 1D case. In 2D and 3D the situation is even worse, the contraction is of order α in 2D and α in 3D. This is the reason why larger α:s than 2-3 cannot be used in real applications since effective damping of the oscillatory modes is important for convergence and stability.

To improve the situation, we now introduce explicit residual smoothing and obtain the IERS, previously described in [2, 3]. The general form is:

$$\begin{aligned} R_j^{IERS} &= IERS(\alpha)\, R_j \\ IERS(\alpha) &= IRS(\alpha) \cdot (1 - \gamma \Delta_{\pm}) \end{aligned} \tag{13}$$

with Fourier symbol

$$\widetilde{IERS}(\xi) = \widetilde{IRS}(\xi) \cdot \left(1 + 4\gamma \sin^2\left(\frac{\xi}{2}\right)\right) \tag{14}$$

The parameter γ is used to recover the good damping properties of the unsmoothed scheme. This is illustrated in Fig. 1 where the Fourier symbol for IERS is well spread out in the Runge-Kutta stability region. The parameter is in this case determined by the choice of α and β in the following way [5]:

$$\gamma = \left(\frac{(1 + 4\beta)}{\alpha} - 1\right)/4 \tag{15}$$

The choice is made to make the damping of the most oscillatory modes as effective in the IERS case as in the original scheme.

To summarize, IRS makes the effective time step larger, which can be seen in Fig. 1. The symbol is pushed up along the imaginary axis for small wave numbers. The drawback is the contraction of the symbol toward the imaginary

axis for the oscillatory modes resulting in less effective damping of these modes. The explicit part of the IERS is designed to counteract the bad damping of oscillatory modes.

3 The time stepping method

The simplest form of time stepping is the Euler forward method which sometimes is used together with upwind methods. In general this method can not be used together with implicit smoothing, since the effect of the IRS for small wave numbers is to push the Fourier symbol upwards along the imaginary axis. One basic property needed for the time stepping scheme is that a part of the imaginary axis around the origin is part of the stability region.

It is discussed in [7] how to design Runge-Kutta methods for TVD schemes. Assume that we are given the forward Euler method

$$u^{n+1} = u^n - \Delta t L(u^n), \tag{16}$$

with $L(u)$ representing a TVD difference method. The idea is then to write a Runge-Kutta method as a convex combination of such forward Euler steps. For a m stage method we then have the general form

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(i)} &= \textstyle\sum_{k=0}^{i-1} \alpha_k^{(i)} u^{(k)} - \beta_k^{(i)} \Delta t L(u^{(k)}) \quad i = 1, 2, \ldots, m \\ u^{n+1} &= u^{(m)} \end{aligned} \tag{17}$$

For each stage, the weights $\alpha_k^{(i)}$, $k = 0, \ldots, i-1$ satisfy

$$\begin{aligned} \alpha_k^{(i)} &\geq 0 \\ \textstyle\sum_{k=0}^{i-1} \alpha_k^{(i)} &= 1 \end{aligned} \tag{18}$$

and the coefficients $\beta_k^{(i)}$ are non-negative. If the forward Euler method is TVD under the CFL condition $\lambda \leq \lambda_0$, ($\lambda = \Delta t/\Delta x$), then the method (17) is TVD under the CFL condition

$$\lambda \leq \lambda_0 \min_{i,k} \frac{\alpha_k^{(i)}}{\beta_k^{(i)}} \tag{19}$$

which is proven in [7].

As mentioned above, the stability region should include a part of the imaginary axis. This has been discussed in generality in [4]. This is not guaranteed to hold for a TVD/RK method. For example, the TVD/RK method of order two does not have the imaginary axis in the stability domain. However, TVD and ENO methods are sometimes modified to have a bias toward centered differencing, [6], which then could make the spectrum of the method come closer to the imaginary axis. Furthermore, the same effect could be caused by one-sided differences at boundaries.

We will therefore construct a TVD/RK method which includes a part of the imaginary axis in its stability domain. We consider the general two stage method

$$\begin{aligned} u^1 &= u^n - a\Delta t L(u^n) \\ u^{n+1} &= b(u^n - c\Delta t L(u^n)) + d(u^1 - e\Delta t L(u^1)) \end{aligned} \tag{20}$$

We require $b + d = 1$, $0 \leq b \leq 1$, and $a > 0, c > 0, e > 0$. (20) is then TVD/RK under the CFL condition

$$cfl = \min(\frac{1}{a}, \frac{1}{c}, \frac{1}{e}) cfl_0 \tag{21}$$

where cfl_0 is the CFL number of the original forward Euler scheme.

The characteristic polynomial of (20) is

$$P(z) = b + d + (bc + ad + de)z + adez^2. \tag{22}$$

We require that

$$\begin{array}{ll} b + d & = 1 \\ bc + ad + de & = 1 \\ ade & = \varphi \end{array} \tag{23}$$

such that the characteristic polynomial becomes $1 + z + \varphi z^2$. With d, e as parameters we obtain from (23)

$$\begin{array}{ll} a & = \varphi/de \\ b & = 1 - d \\ c & = (1 - de - \varphi/e)/(1 - d) \end{array} \tag{24}$$

We next minimize a, c, and e The equation $ae = \varphi/d$ indicates that we should choose $a = e = \sqrt{(}\varphi/d)$. Inserting this into the c equation gives

$$c = (1 - 2\sqrt{\varphi d})/(1 - d) \tag{25}$$

which is decreasing in d for $\varphi > 1/4$. We must have $\varphi > 1/2$ to include a part of the imaginary axis in the stability region. a and e are decreasing in d too. Thus we chose d maximal. The condition $c \geq 0$ gives $\varphi d \leq 1/4$. Thus we take $d = 1/4\varphi$.

All coefficients are now determined as functions of φ. The method is

$$\begin{array}{l} u_1 = u^n - 2\varphi \Delta t L(u^n) \\ u^{n+1} = u^n - \frac{1}{2}\Delta t L(u^n) - \frac{1}{2}\Delta t L(u_1) \end{array} \tag{26}$$

An analysis of the polynomial $R(z) = 1 + z + \varphi z^2$ shows that $z = iy$ is included in its stability region (defined by $|R(z)| \leq 1$) if $\varphi > 1/2$, and that the included part of the imaginary axis is $0 \leq |y| \leq \sqrt{2\varphi - 1}/\varphi$.

The method above is TVD for CFL $\leq \frac{1}{2\varphi}$ CFL$_0$. On the imaginary axis stability holds up to $y = \sqrt{2\varphi - 1}/\varphi$. Of these two stability limits one increases

and one decreases with φ. The optimal point is then to take them equal, which occurs for $\varphi = 5/8 = 0.625$. Both stability limits are then 0.8.

4 IRS and IERS in 2D

In two space dimensions the implicit residual smoother is applied as:

$$(1 - \beta\Delta_{\pm,x})(1 - \beta\Delta_{\pm,y})\, R_{IRS} = \alpha R \tag{27}$$

and the explicit version as:

$$\begin{aligned}(1 - \beta\Delta_{\pm,x})(1 - \beta\Delta_{\pm,y})\, R_{IERS} = \\ \alpha\left(1 + \gamma\Delta_{\pm,x}\Delta_{\pm,y} - \delta\Delta_{\pm,x} - \delta\Delta_{\pm,y}\right) R\end{aligned} \quad . \tag{28}$$

The parameters on the explicit side are chosen to restore good damping and have the following form:

$$\begin{aligned}\delta &= \left(\tfrac{(1+4\beta)}{\alpha} - 1\right)/4 && = O(\alpha) \\ \gamma &= \left(\tfrac{(1+4\beta)}{\alpha} - 8\delta - 1\right)/16 && = O(\alpha^3)\end{aligned} \tag{29}$$

The additional computational effort in the IERS case compared with the IRS case is a 9 point explicit stencil which has to be computed for each grid point and component in the solution vector.

5 Numerical examples in 2D

In this section two numerical examples are presented. The principle in all examples is that we have taken a working code, [8], without any residual smoothing and "plugged in" a subroutine of about 70 statements. Then the code has been run for several values of the parameter α and the best value is then compared for IRS and IERS. In all cases the Runge Kutta scheme discussed previously has

been used as time integrator. In some cases we have used local time stepping in combination with residual smoothing to accelerate the convergence.

5.1 Test case 1

In this case we consider a flow past a wing profile. The free stream Mach number is 0.8. The solution is presented in Fig. 2. Local time stepping was used in the time iteration.

The spatial discretization was made by a second order MUSCL-type approximation with the van Albada limiter applied in primitive variables. The first order Roe's method together with an entropy fix was used as basic scheme.

By using IERS with $\alpha = 3.6$ we earned $\approx$ 15 % CPU-time compared with IRS with $\alpha = 2.6$. In Fig. 3 the convergence history is plotted for IRS, IERS and the basic scheme.

The IERS also turned out to be much more robust with respect to the choice of α. In Fig. 4 the CPU-time is plotted as a function of α. It can be seen that in the IRS case the optimum is much more sharper than in the IERS case.

5.2 Test case 2

In this case we have a flow past a half cylinder. The free stream Mach number is 5.0 and the solution is presented in Fig. 5. The same basic numerical scheme was used as in the previous case. The convergence history is plotted in Fig. 6. About 30% CPU time was gained by using IERS compared with IRS. The same observation about robustness with respect to parameter choice was made in this case.

6 IRS and IERS in 3D

In three space dimensions the implicit residual smoother is applied as:

$$(1-\beta\Delta_{\pm,x})(1-\beta\Delta_{\pm,y})(1-\beta\Delta_{\pm,z})\,R_{IRS} = \alpha R. \tag{30}$$

The explicit version is defined as:

$$\begin{aligned}(1-\beta\Delta_{\pm,x})(1-\beta\Delta_{\pm,y})(1-\beta\Delta_{\pm,z})\,R_{IERS} = \\ \alpha\,(1-\gamma\Delta_{\pm,x}\Delta_{\pm,y}\Delta_{\pm,z} \\ +\delta\Delta_{\pm,x}\Delta_{\pm,y}+\delta\Delta_{\pm,x}\Delta_{\pm,z}+\delta\Delta_{\pm,y}\Delta_{\pm,z} \\ -\epsilon\Delta_{\pm,x}-\epsilon\Delta_{\pm,y}-\epsilon\Delta_{\pm,z})\,R\end{aligned}. \tag{31}$$

The parameters on the explicit side are chosen to restore good damping and have the following form:

$$\begin{aligned}\epsilon &= \left(\tfrac{1+4\beta}{\alpha}-1\right)/4 && = O(\alpha)\\ \delta &= \left(\tfrac{(1+4\beta)^2}{\alpha}-8\epsilon-1\right)/16 && = O(\alpha^3)\\ \gamma &= \left(\tfrac{(1+4\beta)^3}{\alpha}-48\delta-12\epsilon-1\right)/64 && = O(\alpha^5)\end{aligned} \tag{32}$$

The implicit side is computed independently in each direction and corresponds to three tridiagonal systems to solve. The explicit side is a 27 point stencil which for $u_{i,j,k}$ can be computed as:

$$
\begin{aligned}
&\text{Explicit side of IERS} \;= \\
&\quad c_1 \cdot \quad u_{i,j,k} \quad + \\
&\quad c_2 \cdot (\quad u_{i,j,k+1} + u_{i,j,k-1} + \\
&\qquad\quad u_{i,j+1,k} + u_{i,j-1,k} + \\
&\qquad\quad u_{i+1,j,k} + u_{i-1,j,k} \quad) \quad + \\
&\quad c_3 \cdot (\quad u_{i,j+1,k+1} + u_{i,j+1,k-1} + u_{i,j-1,k+1} + u_{i,j-1,k-1} + \\
&\qquad\quad u_{i+1,j,k+1} + u_{i+1,j,k-1} + u_{i-1,j,k+1} + u_{i-1,j,k-1} + \\
&\qquad\quad u_{i+1,j+1,k} + u_{i+1,j-1,k} + u_{i-1,j+1,k} + u_{i-1,j-1,k} \quad) \quad + \\
&\quad c_4 \cdot (\quad u_{i+1,j+1,k+1} + u_{i+1,j+1,k-1} + \\
&\qquad\quad u_{i+1,j-1,k+1} + u_{i+1,j-1,k-1} + \\
&\qquad\quad u_{i-1,j+1,k+1} + u_{i-1,j+1,k-1} + \\
&\qquad\quad u_{i-1,j-1,k+1} + u_{i-1,j-1,k-1} \quad)
\end{aligned}
\tag{33}
$$

with

$$
\begin{aligned}
c_1 &= 1 + 8\gamma + 12\epsilon + 6\delta \\
c_2 &= -4\gamma - 4\delta - \epsilon \\
c_3 &= 2\gamma + \delta \\
c_4 &= -\gamma
\end{aligned}
\tag{34}
$$

7 Numerical examples in 3D

In three space dimensions we have calculated a flow past an ONERA M6 wing, plotted in Fig. 7. The computations were made for one coarse grid with $32 \times 16 \times 16$ grid points and one finer with $64 \times 32 \times 32$ grid points. The computations were made with the first and second order accurate Roe's method of the same type as used in two space dimensions. As seen in Fig. 8 and in Table.I the CPU time needed decreases with up to 30% depending on the case. In this experiment the gain is larger in the ROE 2 case and also larger when more grid points are used. We again found that the IERS is more robust with respect to α in these 3D cases.

8 Implementation

There are several ways of implementing IRS and IERS and in this section we will describe the way that we have found to be most robust. The description is made in terms of a finite difference scheme in 2D but the discussion is valid as well for a finite volume scheme and in 3D. The steps described below are performed independently for all components in the solution vector.

The computations are made in a *region of computation*. In Fig. 9 we have a 2D region of computation. Suppose that the boundaries of this region are described by the indices $[I_0..I_N] \times [J_0..J_N]$. Depending of the nature of the boundaries a *region of smoothing* is defined. Typically the grid line nearest to each boundary are excluded as indicated in Fig. 9. If the boundary is periodical, as for an O-mesh, or if many ghost cells are used in the scheme, then more grid lines should be excluded. Let this region be defined by the following indices $[I_0^s..I_N^s] \times [J_0^s..J_N^s]$. The region of smoothing defines the *region of explicit smoothing*. In the IRS case the residual in this region is only multiplied with α and in the IERS case the explicit part of (28) is applied. The region of explicit smoothing is always one grid line smaller than the region of smoothing, see Fig. 9. Define this region by the indices $[I_0^e..I_N^e] \times [J_0^e..J_N^e]$ with $I_0^e = I_0^s + 1, I_N^e = I_N^s - 1$ and likewise for the J indices. The last step is to perform the implicit part which is done grid line by grid line. There are more than one way to treat the boundaries of the tridiagonal system but we have found that Dirichlet conditions is the most robust approach and the matrix to be solved for looks like:

$$\begin{pmatrix} 1 & 0 & \dots & & \\ -\beta & 1+2\beta & -\beta & 0 & \\ 0 & \ddots & \ddots & \ddots & 0 \\ & 0 & -\beta & 1+2\beta & -\beta \\ & & \dots & 0 & 1 \end{pmatrix} \quad . \tag{35}$$

In the I-direction the system is solved for $I_0^s..I_N^s$ for the grid lines $J_0^s+1..J_N^s-1$ and correspondingly in the J-direction. Hence there are a total of $J_N^s - J_0^s - 1$ systems of length $I_N^s - I_0^s + 1$ and $I_N^s - I_0^s - 1$ systems of length $J_N^s - J_0^s + 1$ to be solved for each component in the solution vector. Depending on the complexity of the underlying scheme, the number of operations per time step increase with about 20 % in the IRS case and 40% in the IERS case.

The residual on the boundary of the region of smoothing is unchanged but do influence all interior points.

The only *free implementation parameter* is the how much smaller the region of smoothing should be compared to the region of computation. Out strategy is to start with the guess that one grid line should be excluded from each boundary. If the scheme is unstable, then exclude more grid lines. The most likely candidate is grid lines nearest to periodic boundaries. If one excludes to many grid lines, the only thing happening is that the scheme will not converge as well as for an optimal choice. The reason for the scheme to become unstable with a too large smoothing region is not clear to us. One plausible explanation is that near boundaries there are points where the residual is computed with less accuracy. These point are normally taken care of by boundary conditions and do not cause any problems in form of instabilities. In the IRS and IERS cases, the influence of these points are global due to the implicit nature of the methods.

9 Conclusions

In this paper we have numerically investigated whether the concept of IERS can be used together with upwind schemes. IERS has been compared with the pure implicit version, IRS. We have found that the CPU-time needed to compute a steady state solution can be reduced with up to 30%. The IERS is much more robust with respect to the one free parameter α present in both versions. Typically the optimal choice of α is near the stability limit in the IRS case whereas in the IERS we have a very flat optimal region. Furthermore, we have not used any upwind information in the smoothers, making them very efficient and easy to implement.

References

[1] J. BLAZEK, N. KROLL, C.C. ROSSOW *A comparison of several implicit smoothing methods* Proceedings of ICFD Conference on Numerical Methods for Fluid Dynamics, Reading, 1992, pp.451-460.

[2] R. ENANDER *Improved Implicit Residual Smoothing for Steady State Computations of First-Order Hyperbolic Systems* , J. Comp. Phys., vol 107 No. 2, pp.291-296 1993

[3] R. ENANDER *Grid Patching and Residual Smoothing for Computations of Steady State Solutions of First Order Hyperbolic Systems* , Doctorial Thesis, Uppsala University, Department of Scientific Computing, 1991.

[4] H.-O. KREISS, G. SCHERER *Method of Lines for Hyperbolic Differential Equations* , SIAM J. Num. Anal., 29 (1992), pp. 640–646.

[5] A. JAMESON *Computational Transonics,* Comm. on Pure and Appl. Math., Vol. XLI, pp. 507-549, 1988.

[6] C-W. SHU *Numerical Experiments on the Accuracy of ENO and Modified ENO Schemes*, J. Sci. Comput., 5 (1990), pp. 151–167.

[7] C-W. SHU, S. OSHER *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes II*, J. Comput. Phys., 83 (1989), pp. 32–78.

[8] B. SJÖGREEN *A FORTRAN Subroutine Library for the Computation of Two Dimensional Compressible Flows with Shocks,* Internal report 92-06, Department of Scientific Computing, Uppsala University, 1992.

Scheme	Grid	IRS best α	IERS best α	Gain CPU IERS / IRS
ROE1	16*16*32	2.0	3.0	≈ 0 %
ROE1	32*32*64	2.5	6.0	≈ 20 %
ROE2	16*16*32	1.8	3.0	≈ 15 %
ROE2	32*32*64	2.0	4.0	≈ 30 %

Table I: Comparison of 3D cases

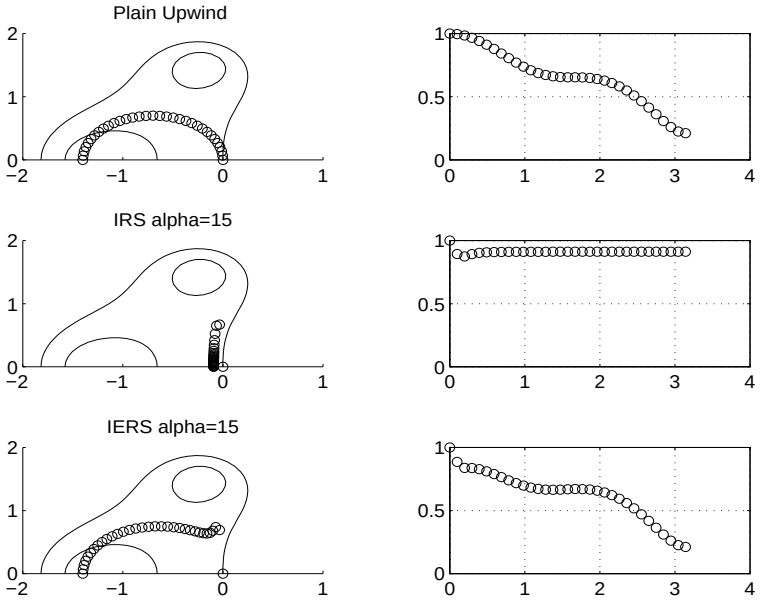

Figure 1: Fourier symbol and amplification factors for RK-iteration.

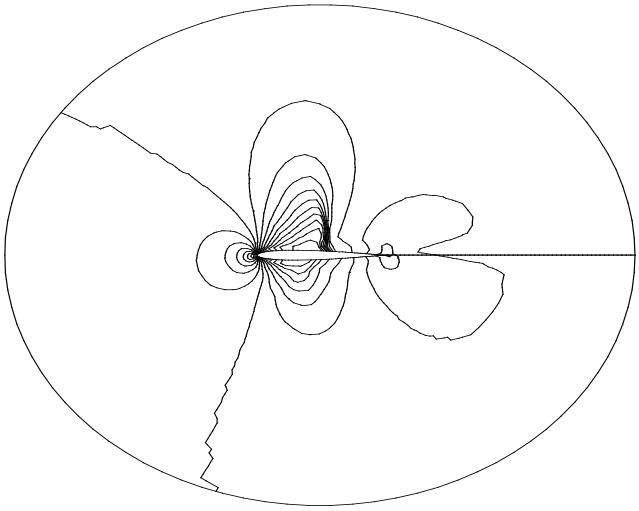

Figure 2: Test case 1. Flow past a wing profile.

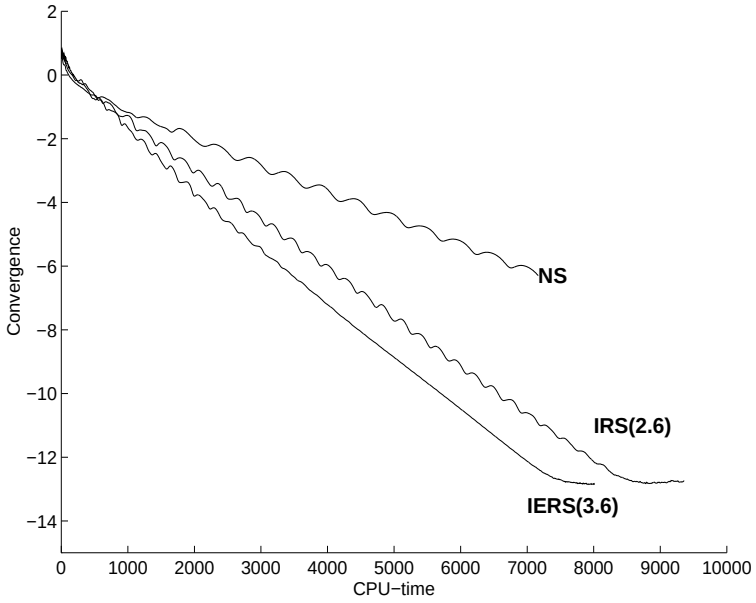

Figure 3: Convergence history for test case 1.

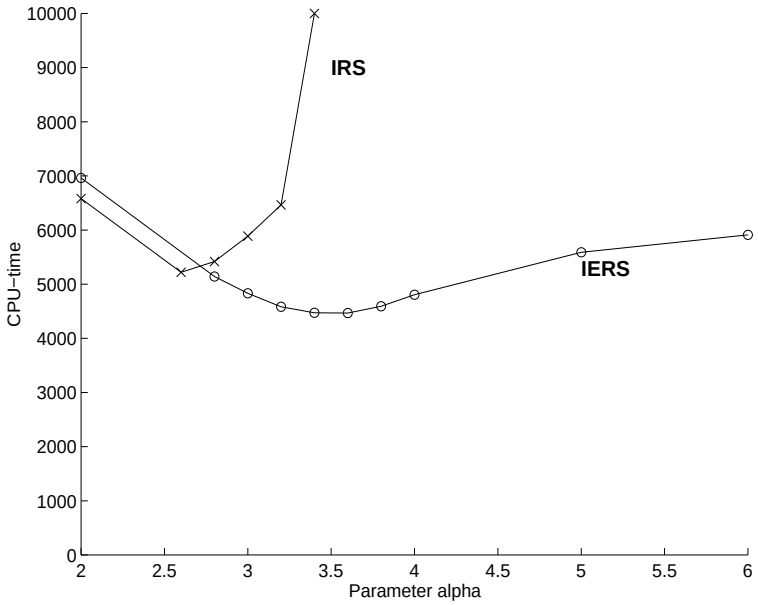

Figure 4: Robustness with respect to the parameter α.

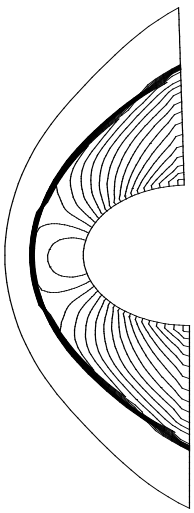

Figure 5: Test case 2. Flow past past a 2D sphere at Mach 5.

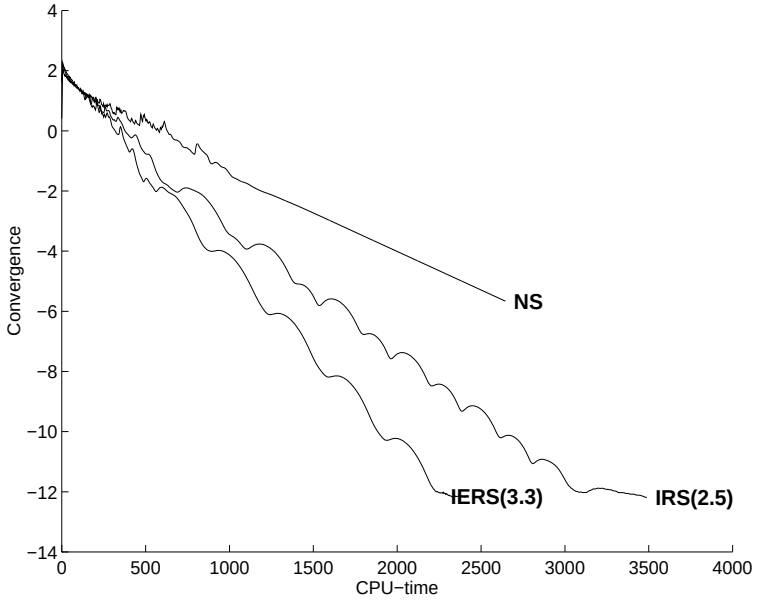

Figure 6: Convergence history for test case 2.

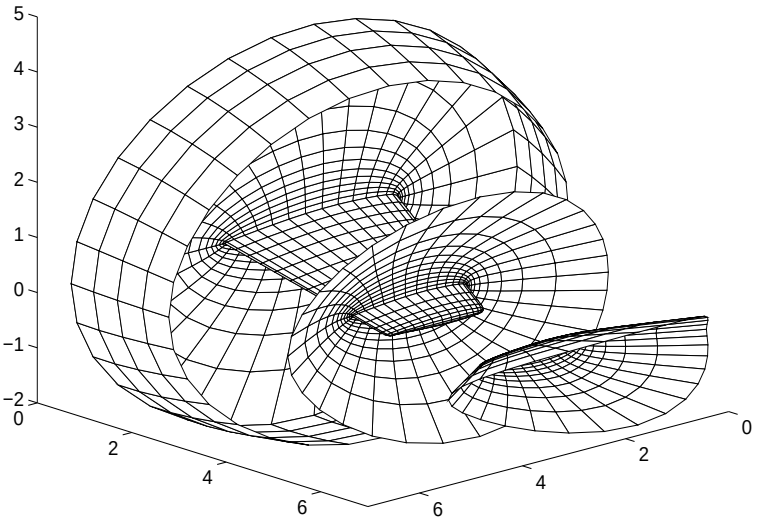

Figure 7: Geometry of 3D case.

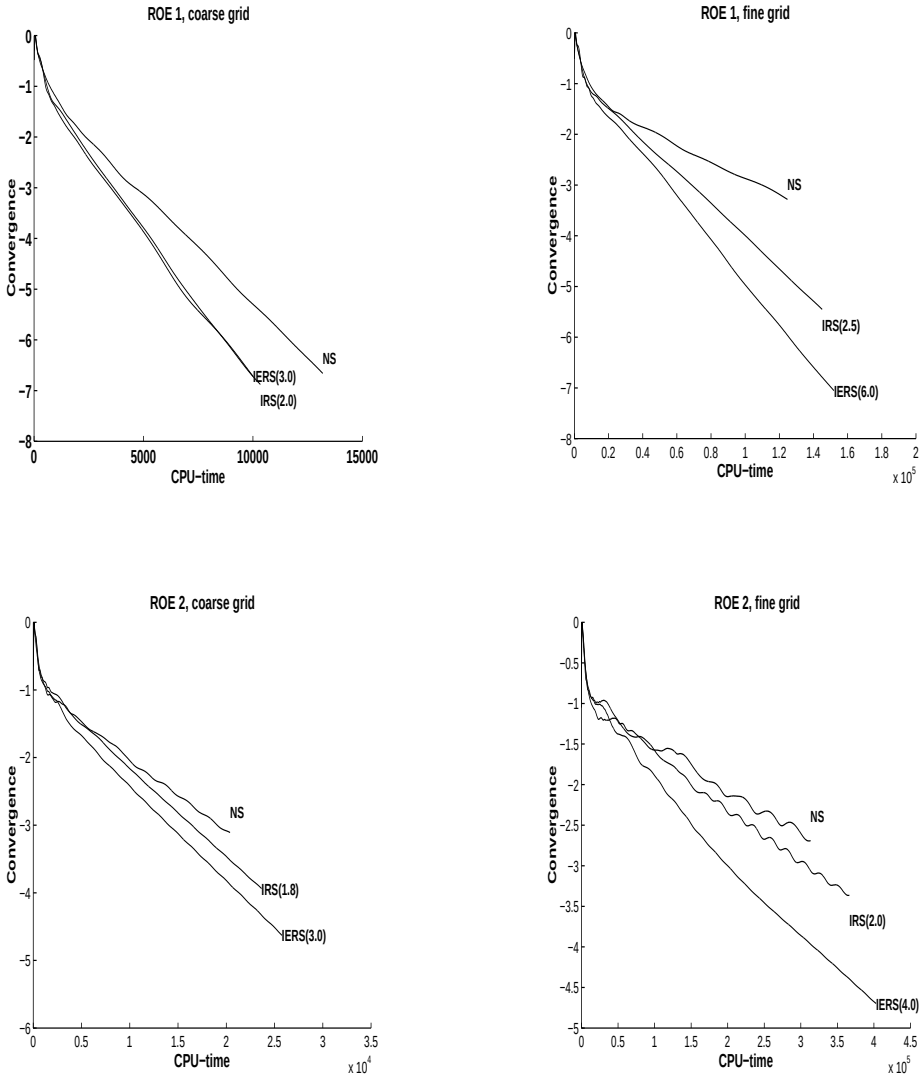

Figure 8: Convergence history for 3D cases

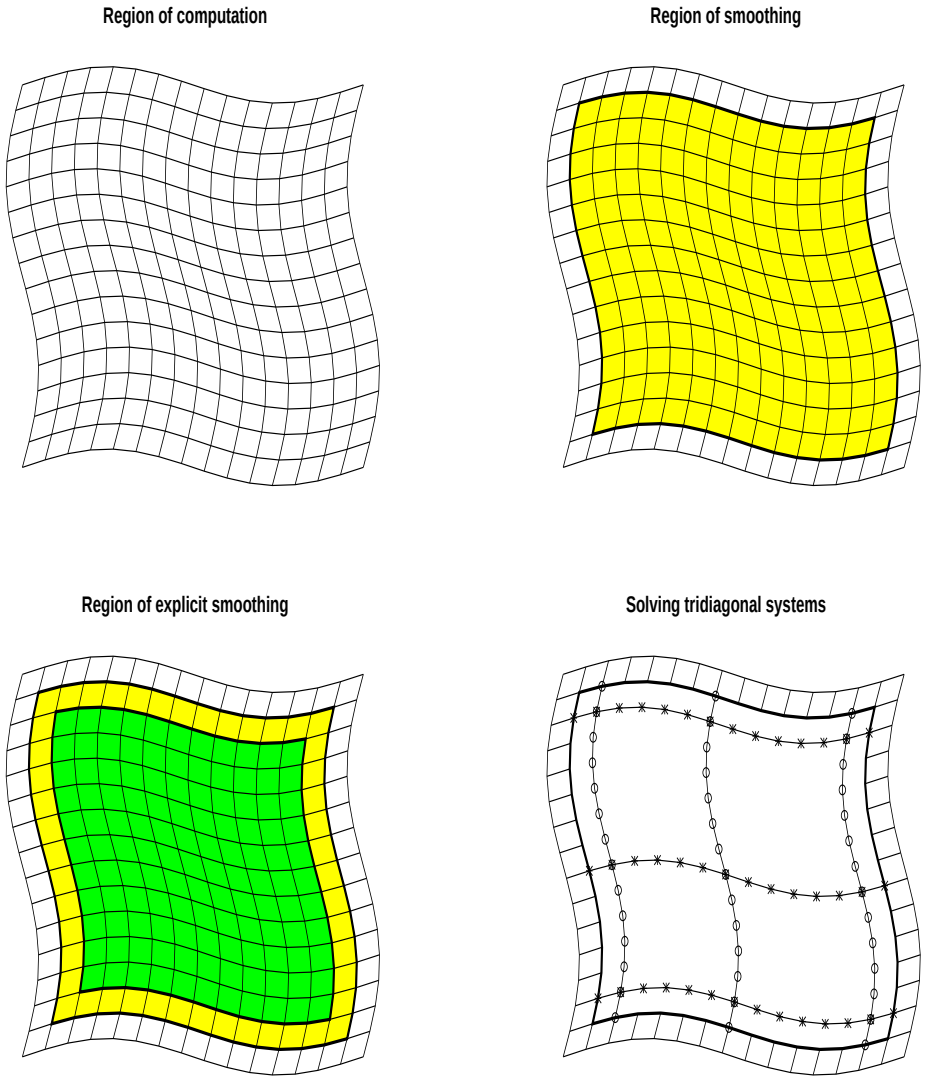

Figure 9: Implementation