Optimization of the Runge-Kutta iteration with residual smoothing

R. Haelterman [a,*], J. Vierendeels [b], D. Van Heule [a],

[a] *Royal Military Academy, Dept. of Mathematics (MWMW), Renaissancelaan 30, B-1000 Brussels, Belgium*

[b] *Ghent University, Dept. of Flow, Heat and Combustion Mechanics, St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium*

Abstract

Iterative solvers in combination with multi-grid have been used extensively to solve large algebraic systems. One of the best known is the Runge-Kutta iteration. Previously [4] we reformulated the Runge-Kutta scheme and established a model of a complete V-cycle which was used to optimize the coefficients of the multi-stage scheme and resulted in a better overall performance. We now look into aspects of central and upwind residual smoothing within the same optimization framework. We consider explicit and implicit residual smoothing and either apply it within the Runge-Kutta time-steps, as a filter for restriction or as a preconditioner for the discretized equations. We also shed a different light on the very high CFL numbers obtained by upwind residual smoothing and point out that damping the high frequencies by residual smoothing is not necessarily a good idea.

Key words: iterative solution, multi-grid, multi-stage
1991 MSC: 65F10, 65N55

1 Introduction

Euler/Navier-Stokes solvers have used explicit multi-stage time-marching schemes for a long time because of their simplicity. Acceleration techniques like multi-grid have been successfully added, which depend heavily on the smooth-

* Corresponding author.
Email addresses: Robby.Haelterman@rma.ac.be (R. Haelterman), Jan.Vierendeels@UGent.be (J. Vierendeels), Dirk.Van.Heule@rma.ac.be (D. Van Heule).

ing properties of the multi-stage scheme. Despite its extreme simplicity, the one-dimensional wave equation has been used as a model with remarkably good results to find optimal coefficients for real flow solvers for the Euler or Navier-Stokes equations. One of the main reasons is that the locus of the one-dimensional scalar Fourier symbol of the advection equation forms the envelope of the loci of the eigenvalues of the discretization matrix of the two-dimensional Euler equations if block-Jacobi preconditioning and a regular grid is used [6,7]. For that reason we focus on this simple equation.
It has already been discovered [7] that coefficients that optimize the smoothing properties of a multi-stage scheme used in conjunction with multi-grid does not automatically result in the fastest overall performance. To find out why this is, we previously extended the Fourier analysis to other components of an idealized two-grid V-cycle (restriction, defect correction and prolongation) and looked at the resulting transmittance [4]. Within this framework optimal coefficients were obtained and it was verified that the transmittance of the V-cycle closely followed the results obtained during numerical experiments. Whereas most studies looked at the smoothing properties of a multi-stage scheme, the model was able to capture the convergence of actual multi-grid solvers quite closely and resulted in faster convergence when measured over a complete V-cycle. We now extend the same formulation to allow for central and upwind residual smoothing. This residual smoothing can be found in various stages of the multi-grid cycle and can be written in explicit or implicit form.
This paper is organized as follows: in section 2 we give a brief introduction about iterative solvers and the currently used formulation for Runge-Kutta type solvers. In section 3 we discuss central and upwind residual smoothing. In section 4 we give the model equation and the different discretizations, together with the impact of residual smoothing on their Fourier symbols. Section 5 treats the optimization procedure, the model of the V-cycle we use and, finally, the resulting optimal coefficients for the model equation.

Remark :
We use $\mathbf{i}$ as a symbol for the complex unit ($= \sqrt{-1}$) and i as an index.

2 Iterative solution of an algebraic system of linear equations with Runge-Kutta iteration

We want to solve a linear algebraic system of the type

$$A\mathbf{u} = \mathbf{b} \tag{1}$$

where $A \in I\!R^{p\times p}; \mathbf{u}, \mathbf{b} \in I\!R^{p\times 1}$, which typically results from the discretization of a linear differential equation; we will go into more detail in section 4. If the dimension of the above system becomes too large, the solution is often found in an iterative way. We use the formulation of the the Runga-Kutta (R-K) scheme that was established in matrix form in [4]:

$$\begin{aligned}
\mathbf{U}_{(o)} &= \mathbf{u}_n \\
\mathbf{V}_{(o)} &= M^{-1}\mathbf{r}_{(o)} \\
\mathbf{U}_{(1)} &= \mathbf{U}_{(o)} - \gamma_1 \mathbf{V}_{(o)} \\
\mathbf{V}_{(1)} &= -M^{-1}A\mathbf{V}_{(o)} \\
\mathbf{U}_{(2)} &= \mathbf{U}_{(1)} - \gamma_2 \mathbf{V}_{(1)} \\
&\vdots \\
\mathbf{V}_{(l)} &= -M^{-1}A\mathbf{V}_{(l-1)} \\
\mathbf{U}_{(l+1)} &= \mathbf{U}_{(l)} - \gamma_{l+1} \mathbf{V}_{(l)} \\
&\vdots \\
\mathbf{V}_{(m-1)} &= -M^{-1}A\mathbf{V}_{(m-2)} \\
\mathbf{U}_{(m)} &= \mathbf{U}_{(m-1)} - \gamma_m \mathbf{V}_{(m-1)} \\
\mathbf{u}_{n+1} &= \mathbf{U}_{(m)}
\end{aligned} \tag{2}$$

starting from an initial guess $\mathbf{u}_o$, where we call $\mathbf{r}_n = \mathbf{r}_{(o)} = A\mathbf{u}_n - \mathbf{b}$ the residual at iteration n. $\gamma_1, \ldots, \gamma_m$ are the iteration parameters. M serves as a preconditioning matrix, which is optional. We can write $\mathbf{u}_n = \mathbf{u}_{exact} + \mathbf{e}_n$, where $\mathbf{e}_n$ is the error of $\mathbf{u}_n$ with respect to the exact solution $\mathbf{u}_{exact} = A^{-1}\mathbf{b}$. After some algebra we find that

$$\mathbf{e}_{n+1} = P_m(-M^{-1}A)\mathbf{e}_n \tag{3}$$

where the transmittance function P_m is given by the polynomial

$$P_m(z) = 1 + \sum_{l=1}^{m} \gamma_l z^l \tag{4}$$

Let $\sigma(M^{-1}A) = \{\lambda_1, \ldots, \lambda_p\}$ denote the spectrum of $M^{-1}A$. We assume that $M^{-1}A$ has p distinct orthonormal eigenvectors $\mathbf{E}_i$, corresponding to eigenvalues λ_i $(i = 1, \ldots, p)$, so that every error $\mathbf{e}_n$ can be written as a linear combination of these eigenvectors $(\mathbf{e}_n = \sum_{i=1}^{p} (\mathbf{e}_n)_i = \sum_{i=1}^{p} (a_n)_i \mathbf{E}_i)$. Due to the linear nature of the equations we only need to consider one component at a time and write

$$(\mathbf{e}_{n+1})_i = P_m(-\lambda_i)(\mathbf{e}_n)_i \tag{5}$$

3 Residual Smoothing

3.1 Residual Smoothing within the Runge-Kutta iteration

3.1.1 Central residual smoothing

To improve the stability and/or convergence speed of the iterative scheme, central explicit residual smoothing (CERS) can be used [3]. In this case the residual in equation (2) is replaced with a smoothed residual $\tilde{\mathbf{r}}_n$. This can be done with the nodal equation

$$(1 + 2\varepsilon)\tilde{r}_j = r_j + \varepsilon(r_{j+1} + r_{j-1}) \tag{6}$$

(where we have dropped the index denoting the iteration count, and the remaining index denotes the node). It can also be implemented in an implicit way (CIRS)

$$(1 + 2\varepsilon)\tilde{r}_j - \varepsilon(\tilde{r}_{j+1} + \tilde{r}_{j-1}) = r_j \tag{7}$$

where ε is the smoothing parameter.

We note that residual smoothing is a special case of preconditioning and that the smoothing operator can be included in the matrix M in equation (2). Apart from smoothing the residual, it also widens the support of the discretization scheme. For instance CERS with $\varepsilon = 0.5$ is the same as full weighting that is, for instance, used in the restriction operator of some multi-grid methods.

3.1.2 Upwind residual smoothing

For advection equations discretized with upwind(-biased) schemes a different residual smoothing has been proposed to include upwinding [10].

In an explicit way (UERS) this gives

$$(1 + \varepsilon)\tilde{r}_j = r_j + \varepsilon r_{j-1} \tag{8}$$

and in its implicit (UIRS) form

$$(1+\varepsilon)\tilde{r}_j - \varepsilon\tilde{r}_{j-1} = r_j \tag{9}$$

3.1.3 Discussion about the implicit forms

While equations (7) and (9) are valid formulations, we must treat them with caution. Contrary to equations (6) and (8) the parameter is on one side of the equation only, meaning that as ε becomes bigger, the magnitudes of the smoothed residuals must become smaller to balance the equation. This means that these formulations of residual smoothing not only change the shape of the Fourier symbols, but also the size (see section 4 for a graphical illustration). It has to be noted that multiplying the residual in equation (2) with a scalar $\kappa \in I\!R$ (i.e. $M = \kappa^{-1} I_p$) will not lead to a better optimal convergence. The corresponding coefficients will only be re-scaled to compensate:

Residual	$\mathbf{r}$	$\kappa\ \mathbf{r}$
Coefficient	γ_k	$\kappa^{-k}\ \gamma_k$

$(k = 1, \ldots, m)$. This effect, which can lead to very high coefficients (especially for upwind residual smoothing), has been interpreted as allowing for extremely high CFL numbers and thus very fast convergence by convection [1]. While this can improve convergence speed of single-grid solvers, this effect is generally absent in a multi-grid setting, where the main improvements results from a change in the shape of the Fourier symbol, not the size.

This leads us to propose a different formulation for implicit residual smoothing, resp. central and upwind:

$$\tilde{r}_j - \frac{\varepsilon}{1+2\varepsilon}(\tilde{r}_{j+1} + \tilde{r}_{j-1}) = r_j \tag{10}$$

$$\tilde{r}_j - \frac{\varepsilon}{1+\varepsilon}\tilde{r}_{j-1} = r_j \tag{11}$$

We emphasis that apart from the re-scaling, equations (10) and (11) will not change the behavior of the scheme but are only used to separate the effect of scaling and smoothing and to avoid round-off errors in the optimization procedure later on.

3.2 Residual Smoothing as a filter

In the previous paragraph we have looked at residual smoothing as a preconditioner within the R-K iteration. In the context of multi-grid we can also apply it at other stages of the multi-grid cycle. For instance, if we fear that the residual has been insufficiently smoothed before passing it to the coarser grid, we can use it as a filter just before the restriction phase to limit aliasing.

3.3 Residual Smoothing as a preconditioner of the original equations

We can also precondition the original equation (12) directly, replacing it with

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{b} \tag{12}$$

where in this case M contains the effect of residual smoothing. In this way we effectively alter the discretization scheme in every step of the multi-stage cycle.

4 The equations under consideration

4.1 Discretization and Fourier symbol

We will try to find the optimal parameters γ_l $(l = 1, \ldots, m)$ for the one-dimensional advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{13}$$

$(a > 0)$, to which we add suitable initial and boundary conditions. We are only interested in the steady-state solution of equation (13). The spatial derivative is discretized with a first or second order upwind scheme (resp. U1 and U2) or a K3 upwind biased scheme.

On a regular mesh the discretization of the space operator gives

- U1

$$\frac{u_j - u_{j-1}}{\Delta x} \tag{14}$$

- U2

$$\frac{3u_j - 4u_{j-1} + u_{j-2}}{2\Delta x} \tag{15}$$

- K3

$$\frac{2u_{j+1} + 3u_j - 6u_{j-1} + u_{j-2}}{6\Delta x} \tag{16}$$

To study the behavior of the iterative scheme when using the resulting discretization matrix, we pass from the discrete representation of the p eigenvalues to the continuous representation given by the Fourier symbol $\lambda(\theta)$, $(\theta \in [-\pi, \pi])$. Von Neumann analysis shows that it is given by

- U1

$$\lambda(\theta) = \frac{1 - e^{-\mathbf{i}\theta}}{\Delta x} \tag{17}$$

- U2

$$\lambda(\theta) = \frac{3 - 4e^{-\mathbf{i}\theta} + e^{-2\mathbf{i}\theta}}{2\Delta x} \tag{18}$$

- K3

$$\lambda(\theta) = \frac{3 + 2e^{\mathbf{i}\theta} - 6e^{-\mathbf{i}\theta} + e^{-2\mathbf{i}\theta}}{6\Delta x} \tag{19}$$

4.2 Effect of residual smoothing on the discretization and Fourier symbol

4.2.1 Central residual smoothing

Central residual smoothing alters $\lambda(\theta)$ in the following way (for equations (6), (7) and (10) respectively)

$$\tilde{\lambda}_{CE}(\theta) = \frac{1 + 2\varepsilon\cos\theta}{1 + 2\varepsilon}\lambda(\theta) = \mathcal{R}_{CE}(\theta)\lambda(\theta) \tag{20}$$

$$\tilde{\lambda}_{CI^*}(\theta) = \frac{\lambda(\theta)}{1 + 2\varepsilon(1 - \cos\theta)} = \mathcal{R}_{CI^*}(\theta)\lambda(\theta) \tag{21}$$

$$\tilde{\lambda}_{CI}(\theta) = \frac{1+2\varepsilon}{1+2\varepsilon(1-\cos\theta)}\lambda(\theta) = \mathcal{R}_{CI}(\theta)\lambda(\theta) \tag{22}$$

Comparing $\tilde{\lambda}_{CI^*}$ and $\tilde{\lambda}_{CI}$ we clearly distinguish the effect of the multiplication of the Fourier symbol by a scalar.

We illustrate the effect of the smoothing by drawing curves of $\mathcal{R}_{CE}(\theta)$, $\mathcal{R}_{CI^*}(\theta)$ and $\mathcal{R}_{CI}(\theta)$ for various values of ε in figures 1, 2 and 3.

Fig. 1. $\mathcal{R}_{CE}(\theta) = \frac{1+2\varepsilon\cos\theta}{1+2\varepsilon}$ for various values of ε.

Fig. 2. $\mathcal{R}_{CI^*}(\theta) = \frac{1}{1+2\varepsilon(1-\cos\theta)}$ for various values of ε.

We have already hinted that the amplification curves of figures 1, 2 and 3 can be misleading, as they would give the impression that a low value of the amplification factor for high frequencies would mean a good reduction of the corresponding error component. This is not necessarily the case. We will see below that a low value of the amplification will draw the Fourier footprint of a mode towards the origin, actually making it harder to smooth with a

Fig. 3. $\mathcal{R}_{CI}(\theta) = \frac{1+2\varepsilon}{1+2\varepsilon(1-\cos\theta)}$ for various values of ε.

multi-stage scheme; when used solely within the Runge-Kutta iteration it is as if the smoother does not "see" the higher frequencies, which impairs its ability to resolve them. In this context the beneficial effects will result from either a change in form of $\lambda(\theta)$ and/or a clustering of the high-frequency modes, preferably away from the origin. We note that for CERS with $\varepsilon \geq 0.5$ (figure 1), $\tilde{\lambda}_{CE}(\theta)$ will become zero for a certain value $\theta^* \in [-\pi, \pi]$, even though $\lambda(\theta^*) \neq 0$. As $P_m(0) = 1$ it thus becomes impossible to eliminate this component in the Runge-Kutta iteration.

To illustrate what happens to the Fourier symbol under the effect of smoothing, we use the U1 discretization with $\Delta x = 1$ as an example (figures 4, 5 and 6).

Fig. 4. $\tilde{\lambda}_{CE}(\theta)$ for U1. Various values of ε; bold line represents high frequency modes.

We see that in all cases the shape of the Fourier symbol changes. What is not readily apparent is that for the implicit cases we get a slight clustering of high frequency Fourier modes (the boldest line in the figures), which is

Fig. 5. $\tilde{\lambda}_{CI^*}(\theta)$ for U1. Various values of ε; bold line represents high frequency modes.

Fig. 6. $\tilde{\lambda}_{CI}(\theta)$ for U1. Various values of ε ; bold line represents high frequency modes.

not present in the explicit upwind formulation. Also, for the explicit residual smoothing the high frequency modes are drawn towards the origin, which is bad for convergence.

4.2.2 Upwind residual smoothing

Upwind residual smoothing alters $\lambda(\theta)$ in the following way (for equations (8), (9) and (11) respectively)

$$\tilde{\lambda}_{UE}(\theta) = \frac{1+\varepsilon e^{-\mathbf{i}\theta}}{1+\varepsilon}\lambda(\theta) = \mathcal{R}_{UE}(\theta)\lambda(\theta) \tag{23}$$

$$\tilde{\lambda}_{UI^*}(\theta) = \frac{\lambda(\theta)}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}} = \mathcal{R}_{UI^*}(\theta)\lambda(\theta) \tag{24}$$

$$\tilde{\lambda}_{UI}(\theta) = \frac{1+\varepsilon}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}}\lambda(\theta) = \mathcal{R}_{UI}(\theta)\lambda(\theta) \tag{25}$$

We draw the curves of $\mathcal{R}_{UE}(\theta)$, $\mathcal{R}_{UI^*}(\theta)$ and $\mathcal{R}_{UI}(\theta)$ for various values of ε in figures 7, 8 and 9.

Fig. 7. $\mathcal{R}_{UE}(\theta) = \frac{1+\varepsilon e^{-\mathbf{i}\theta}}{1+\varepsilon}$ for various values of ε.

Fig. 8. $\mathcal{R}_{UI^*}(\theta) = \frac{1}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}}$ for various values of ε.

We note that in the explicit case, the limit for $\varepsilon \to \infty$ reduces the operator to the unity circle in the complex plane, meaning that the modulus of the Fourier symbol will not be altered. We draw the modulus of the operators $\mathcal{R}_{UE}(\theta)$, $\mathcal{R}_{UI^*}(\theta)$ and $\mathcal{R}_{UI}(\theta)$ in figures 10, 11 and 12 and note that for the upwind explicit residual smoothing there will only exist a value $\theta^* \in [-\pi, \pi]$

Fig. 9. $\mathcal{R}_{UI}(\theta) = \frac{1+\varepsilon}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}}$ for various values of ε.

Fig. 10. Modulus of $\mathcal{R}_{UE}(\theta) = \frac{1+\varepsilon e^{-\mathbf{i}\theta}}{1+\varepsilon}$ for various values of ε.

Fig. 11. Modulus of $\mathcal{R}_{UI^*}(\theta) = \frac{1}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}}$ for various values of ε.

such that $\tilde{\lambda}_{UE}(\theta^*) = 0$ with $\lambda(\theta^*) \neq 0$ when $\varepsilon = \pm 1$. UIRS does not contain

Fig. 12. Modulus of $\mathcal{R}_{UI}(\theta) = \frac{1+\varepsilon}{1+\varepsilon-\varepsilon e^{-\mathbf{i}\theta}}$ for various values of ε.

this singularity except at $\theta = 0$ in the limit $\varepsilon \to \infty$ and for $\varepsilon = -1$.

We show what happens to the Fourier symbol under the effect of smoothing, for which we again use the U1 discretization with $\Delta x = 1$ as an example (figures 13, 14,and 15).

Fig. 13. $\tilde{\lambda}_{UE}(\theta)$ for U1. Various values of ε; bold line represents high frequency modes.

Both the implicit formulations give similar results, apart from the re-scaling, which makes their effect more apparent. At first sight it appears that the shape of the Fourier symbol does not change, but this is deceptive, as the main alteration is the clustering of high frequency Fourier modes (the boldest line in the figures). This clustering is not present in the explicit upwind formulation. It is also apparent that the clustering in UIRS is more pronounced than in CIRS. In the limit for $\varepsilon \to \infty$, $\tilde{\lambda}_{UI}(\theta)$ will even be reduced to a single point $1 + 0\mathbf{i}$, apart from the singularity at the point for which $\theta = 0$ ($\lambda(0) = 0$). We will discuss this further below.

Fig. 14. $\tilde{\lambda}_{UI^*}(\theta)$ for U1. Various values of ε; bold line represents high frequency modes.

Fig. 15. $\tilde{\lambda}_{UI}(\theta)$ for U1. Various values of ε; bold line represents high frequency modes.

4.3 Further discussion about upwind implicit residual smoothing

We saw in the previous paragraph that UIRS would allow us to obtain extreme clustering of the high frequency modes of the U1 scheme for $\varepsilon \to \infty$. This is hardly surprising, as the smoothing operator in that case reduces to

$$\tilde{\mathbf{r}}_j - \tilde{\mathbf{r}}_{j-1} = \mathbf{r}_j \tag{26}$$

Solving this implicit equation would imply computing the inverse of a matrix which is exactly the discretization operator of the U1 scheme. In other words, we would be preconditioning the scheme with A^{-1}. This would incur the same cost as solving the model problem. For that reason we have dropped the use of UIRS on the U1 discretization in section 5, after it had been verified that the

optimization procedure indeed led to the optimal value $\varepsilon \approx \infty$. Nevertheless this discussion clearly shows whence the performance of UIRS originates and how it can be successfully employed on more complex problems like the Euler equations, if the latter have been preconditioned such that its Fourier footprint closely matches that of (17). As in most cases implicit smoothing is done by using a sequence of Jacobi iterations (effectively creating an approximate inverse of the smoothing operator), it would be necessary to investigate which would be the optimal way to do so.
We get markedly different effects on the U2 and K3 discretizations (figures 16 and 17). The clustering of the high frequency modes is no longer discernable and, strangely, from $\varepsilon = 1$ onwards the smoothed Fourier symbol for K3 starts to "fold back" on itself, meaning that some points in the complex plane correspond to two different frequencies (figure 17). This lack of clustering leads us to believe that the current formulation of UIRS is only well suited for first order upwind discretizations and that the use of a higher order residual smoother might improve the results for higher order discretizations.

Fig. 16. $\tilde{\lambda}_{UI}(\theta)$ for U2. Various values of ε; bold line represents high frequency modes.

Remarks :
It might seem counter-intuitive to use negative smoothing factors (ε), but it can be easily seen from the mathematical expression in (25) that apart from $\varepsilon \in [-1, 0[$ these are perfectly acceptable, although we have to note that for large values of ε we would get about the same result for ε as for $-\varepsilon$.

Something that is less obvious in figures 16 and 17 is how fast the Fourier symbol "travels" along its curve. By that we mean that, even for small values of θ, $\tilde{\lambda}_{UI}(\theta)$ will already be a respectable distance from the origin, which - as will become clear in section 5 - greatly improves the convergence performance as the modes for which $\lambda(\theta) \approx 0$ are the hardest to smooth.

The above discussion about the effect of UIRS on the U1 discretization would

Fig. 17. $\tilde{\lambda}_{UI}(\theta)$ for K3. Various values of ε; bold line represents high frequency modes.

carry over to CIRS with $\varepsilon \to \infty$ used on the Poisson equation, indicating that it is a very good choice for elliptic problems.

5 Optimization of parameters

5.1 Objective function

5.1.1 Objective function without residual smoothing

When using the Fourier representation of the discretization scheme, we can associate an eigenvalue $\lambda(\theta)$ with every $\theta \in [-\pi, \pi]$. We decompose the error vector with respect to the corresponding eigenvectors. Equation (5) then becomes

$$(\mathbf{e}_{n+1})_\theta = P_m(-\lambda(\theta))(\mathbf{e}_n)_\theta \tag{27}$$

The eigenvector corresponding to the eigenvalue 0 is the "constant vector" ($= [1\ 1\ \ldots\ 1]^T$), which we assume to be eliminated due to the presence of boundary conditions.
While it is possible to find coefficients that allow equation (13) to be solved with (2) it will always suffer from very slow convergence, as for values of θ for which $\lambda(\theta) \approx 0$ we will have $P_m(-\lambda(\theta)) \approx 1$. For that reason multi-grid schemes are used, in which (2) serves as a smoother.
We recall that there are two ways to define "smooth" errors. In geometrical multi-grid, these are the errors that vary slowly over the grid, i.e. those with $\theta \in \Phi_{LF} = [-\frac{\pi}{2}, \frac{\pi}{2}]$. In an algebraic multi-grid, "smooth" denotes the errors that cannot easily be reduced by the smoother, i.e. those with $\lambda(\theta) \approx 0$. In this paper we will limit ourselves to geometrical multi-grid, as for the equations under consideration both approaches are similar.
When looking for good coefficients we want a scheme that adequately reduces errors for which $\theta \in \Phi_{HF} = [-\pi, \pi] \setminus \Phi_{LF}$. We define the smoothing factor ρ_{HF} as

$$\rho_{HF} = \sup_{\theta \in \Phi_{HF}} |P_m(-\lambda(\theta))| \tag{28}$$

Most previous studies (e.g. [2,9]) tried to find the lowest smoothing factor, ignoring the effect of the remainder of the multi-grid cycle. We previously found [4] that an integrated approach, taking the effect of all components of a 2-grid cycle into account gave better results, which corresponded well with the convergence recorded on actual solvers. We assumed that the solution on the coarser grid (defect correction) is exact (ideal two-grid cycle). This could be done by a direct solver or an indirect solver, so that this approach could represent a V,W or F multi-grid cycle depending on how the defect correction is actually computed. Solving the coarse grid equation exactly does not necessarily mean that all errors corresponding to $\theta \in \Phi_{LF}$ will be completely annihilated, as due to the restriction and prolongation process (which act as non-ideal low-pass filters) some high frequencies will be passed to the coarse grid (aliasing), while the low frequencies will be attenuated. We will only quantify the latter effect, ignoring aliasing. We use full weighting for the restriction and linear interpolation for the prolongation. The proposed objective function (depending on the iteration coefficients) is then

$$\mathcal{I} = \sup_{\theta \in [-\pi, \pi]} |\mu(\lambda(\theta))| \tag{29}$$

where

$$\mu(\lambda(\theta)) = D(\lambda(\theta))\Big(P_m(-\lambda(\theta))\Big)^2 \quad \text{for } \theta \in \Phi_{LF} \tag{30}$$

$$(P_m(-\lambda(\theta)))^2 \qquad \text{for } \theta \in \Phi_{HF} \tag{31}$$

where

$$D(\lambda(\theta)) = \left(1 - \left(\cos\frac{\theta}{2}\right)^4 \frac{2\lambda(\theta)}{\lambda(2\theta)}\right) \tag{32}$$

$(P_m(-\lambda(\theta)))^2$ models the effect of pre- and post-smoothing; $D(\lambda(\theta))$ the defect correction on the coarser grid; $\left(\cos\frac{\theta}{2}\right)^4$ is the combined filtering effect of restriction and prolongation; $\frac{2\lambda(\theta)}{\lambda(2\theta)}$ takes into account the discrepancy due to the truncation error between the matrices on the fine and coarse grid. (For more details see [4].)
Note that the minimization procedure will automatically satisfy the stability constraint $\mathcal{I} \leq 1$.
As we have shown in [4], the objective functions (29) will generally result in multi-stage schemes that are not as good as smoothers, but give a better overall performance after one complete V-cycle. Also, the correlation with measured convergence rates of two-grid V-cycles was very good.

5.1.2 Objective function with residual smoothing

For the cases when we use CERS, CIRS, UERS or UIRS we use similar expressions as (29)-(32), but distinguish three subcases for each

(1) Residual smoothing in the Runge-Kutta smoother meaning that we replace $\lambda(\theta)$ by $\tilde{\lambda}(\theta)$ in equation (30),(31) in $P_m(-\lambda(\theta))$ but not in $D(\lambda(\theta))$. We will call these optimizers $\mathcal{I}_{CERS,RK}$ etc. (see §3.1).

(2) Residual smoothing as a filter applied just prior the restriction to the coarser grid, meaning that we replace equation (32) by

$$\tilde{D}(\lambda(\theta)) = \left(1 - \mathcal{R}(\theta)\left(\cos\frac{\theta}{2}\right)^4 \frac{2\lambda(\theta)}{\lambda(2\theta)}\right) \tag{33}$$

where $\mathcal{R}$, can be either $\mathcal{R}_{CE}$,$\mathcal{R}_{CI}$,$\mathcal{R}_{UE}$ or$\mathcal{R}_{CI}$. We will call these optimizers $\mathcal{I}_{CERS,fil}$ etc. (see §3.2).

(3) Residual smoothing as a preconditioner of the original equations: the Fourier symbol is altered by residual smoothing everywhere, meaning that we replace $\lambda(\theta)$ in equation (30), (31) and (32) everywhere by its smoothed counterpart (including the coarser grid). This has the same

effect as using a different discretization scheme from the start. We will call these optimizers $\mathcal{I}_{CERS,all}$ etc. We include these for comparison only. (see §3.3).

For the implicit cases we use the formulations of equations (10) and (11) was used as the original formulation in (7) and (9) lead to rounding errors due to the fact that combinations occurred of very high values of ε (small values of $\lambda(\theta)$) and very high iteration coefficients (linked to CFL).

5.2 Optimization approach

To find the optimal parameters for the different objective functions a routine was written in Matlab 7 that creates a large number of random seed vectors, containing initial guesses for the various parameters. These are then fed to another routine that computes the value of the chosen objective function for these seed vectors and starts an optimization procedure based on the Nelder-Mead simplex method that is implemented in Matlab's *fminsearch* function.

All schemes under consideration tacitly assume $\Delta x = 1$. If different values of Δx are needed, the following re-scaling can be used

$$\gamma_k \rightarrow (\Delta x)^k \gamma_k \tag{34}$$

We will compare our results against the optimal coefficients for the U1,U2 and K3 scheme, when optimized for ρ_{HF} without residual smoothing. These are taken from [9] and can be found in tables A.1, A.2 and A.3 in appendix A. It was found that in these cases the frequencies that converged most slowly were those around $\theta = \frac{\pi}{2}$ as discussed in [4].
To validate the results obtained by the model (equations (30) and (31)) an actual solver using 2000 nodes was used and its asymptotic convergence rate measured in the L2-norm. As in previous studies, the agreement between this value and $\mathcal{I}$ was in general very good, thus proving that the latter serves as an adequate model. To be able to compare the merit of schemes with a different number of stages in the R-K iteration, an average convergence factor per stage was used.

We see that for the values obtained by optimization of ρ_{HF} without residual smoothing a higher number of stages results in a net benefit in the average smoothing factor $\sqrt[m]{\rho_{HF}}$. For that reason a high number of stages has been advocated in the past. However, in [4] we showed that this is not the case when looking at the complete picture ($\sqrt[2m]{\mathcal{I}}$). Our conclusion was that a lower number of stages gave more optimal results when optimizing for $\mathcal{I}$ without

residual smoothing. For the three schemes these can be summarized as

- U1: $m = 2, (\gamma_1, \ldots, \gamma_m) = (1.0000, 0.3741),\ \sqrt[2m]{\mathcal{I}} = 0.7046$
- U2: $m = 2, (\gamma_1, \ldots, \gamma_m) = (0.3881, 0.0803),\ \sqrt[2m]{\mathcal{I}} = 0.8557$
- K3: $m = 3, (\gamma_1, \ldots, \gamma_m) = (1.0537, 0.7632, 0.2417),\ \sqrt[2m]{\mathcal{I}} = 0.7861$

More details can be found in appendix A (tables A.4, A.5 and A.6).
We draw the attention to the fact that the use of a low number of stages is solely based on observation of the scalar advection equation. Transferability of this conclusion to the Euler equations still has to be examined.

5.3 Results and discussion

We will use averaged values of the optimization function ($\sqrt[2m]{\mathcal{I}}$) that takes into account the computational effort of using m stages in both pre- and post-smoothing. When explicit residual smoothing is used within the Runge-Kutta iteration this effectively doubles the cost per stage; for that reason we also give the value $\sqrt[4m]{\mathcal{I}}$ for info. Explicit residual smoothing only adds one matrix-vector operation, in which instance $\sqrt[1+2m]{\mathcal{I}}$ is also given in the tables. When residual smoothing is used as a preconditioner to the discretized equation we assume the preconditioned matrix is stored and invokes no extra cost within the iterations. For implicit residual smoothing, the cost is not quantified as it will depend on the way the implicit equation for the smoothing is actually solved.
For the U1,U2 and K3 discretizations the optimal results when using central residual smoothing can be found in tables 1, 2 and 3, while for the upwind residual smoothing they are summarized in tables 4, 5 and 6. We recall from §4.3 that the optimal implicit upwind residual smoothing for the U1 discretization would happen for $\varepsilon = \infty$.

From these results it immediately becomes clear that upwind residual smoothing is a far better choice for advection equations. This is very logical as the resulting preconditioning resembles the inverse of A better than for central residual smoothing.
It is also clear that UIRS has a higher potential than UERS; their respective true gains will depend on the cost of the actual implementation. For the explicit residual smoothing the improvements are drastically reduced when counting the cost, to the point that only for U2 there will remain a net benefit. It is quite possible that the same happens for implicit residual smoothing, although some basic arithmetic shows that for the U2 discretization with UIRS applied to the Runga-Kutta iteration a real gain is obtained as long as the total cost remains below the equivalent of 31 matrix-vector products.
The reason that most is gained in the U2 scheme is probably due to the fact

that for U2 the high frequencies are the most spread out and hence benefit most from the clustering effect of preconditioning.
Remarkably, the performance of CERS and CIRS are very similar, while the latter bears a much higher computational cost.
Also, when one has to chose from the three possibilities to apply residual smoothing it seems the best approach would be to use it only within the Runge-Kutta iteration, while the least can be gained from using it as a filter.

We add that the values of $\mathcal{I}$ were compared with convergence factors on actual solvers and differed by less than 5%.

We illustrate the results for the optimal solution for "CERS,RK", "CIRS,RK", "UERS,RK" and "UIRS,RK" for the U2 scheme (figures 18, 19, 20 and 21 . We draw the stability region $S = \{z \in \mathbb{C}; |P_m(z)| \leq 1\}$ on which we superpose $(-\tilde{\lambda}(\theta))$ and also give $|P_m(\tilde{\lambda}(\theta))|$ (the amplification of the smoother) and $|\mu(\lambda(\theta))|$ (the amplification of the V-cycle).

From these figures it is easy to see that the main improvement in convergence stems from the clustering of the high frequencies and the altered shape of the Fourier symbol that follows the contours of the amplification function better.

6 Conclusions

Central and upwind residual smoothing have been tested on various discretizations of the simple scalar advection equation: U1, U2 and K3. Both explicit and implicit formulations were looked at. These were applied within a multigrid framework either to the smoother (a multi-stage Runge-Kutta solver), as a filter before the restriction to a coarser grid or as a preconditioner to the discretized equations. A new optimization function that better resembles the real convergence speed was used to find the optimal coefficients for the Runga-Kutta smoother in this context.
The best improvements were obtained with upwind implicit schemes, although one has to take into account their relative high cost. The most was gained in case the U2 discretization was used.
While this simple equation might lead us to a conclusion of excellent convergence, it is not so much the result as it is the reason for the improvement that we would like to emphasis. Often a reduction in magnitude (modulus) of the high frequency modes or the possibility of high CFL numbers has been the main argument. We have pointed out that the former can even be detrimental when used in the smoother as it would bring them closer to the origin, where they are difficult to damp. We have shown, that the dominating acceleration mechanisms of residual smoothing are

Fig. 18. First figure: Stability region and Fourier symbol (locus of high frequencies in bold). Second figure: amplification curves $|P_m(\tilde{\lambda}(\theta))|$ (dotted line) and $|\mu(\lambda(\theta))|$ (solid line) for the optimal U2 scheme with explicit central residual smoothing (CERS,RK). See table 2 for details.

(1) a very rapid progression of $\tilde{\lambda}(\theta)$ away from the origin
(2) a clustering of the high frequency modes
(3) a modification of the shape of the Fourier symbol to better follow the contours of the transmittance function of the multi-stage smoother.

For the scalar equation the cost of implicit residual smoothing is of the same order as the solution of the original problem and thus prohibitive unless an approximate solution to the residual smoothing equation is used. It is believed however that qualitative aspects of this study will carry over to more complex problems like the Euler equations if these are preconditioned to closely follow the contours of the Fourier symbol of the advection equation. In those applications, the improvement obtained by residual smoothing will outweigh its computational cost.

Fig. 19. First figure: Stability region and Fourier symbol (locus of high frequencies in bold). Second figure: amplification curves $|P_m(\tilde{\lambda}(\theta))|$ (dotted line) and $|\mu(\lambda(\theta))|$ (solid line) for the optimal U2 scheme with implicit central residual smoothing (CIRS,RK). See table 2 for details.

References

[1] L.Blazek, N. Kroll, R. Radespiel, C.-C. Rossow, Upwind implicit residual smoothing method for multi-stage scheme, *AIAA Paper 91-1532*, June 1991.

[2] L.A. Catalano and H. Deconinck, Two-dimensional optimization of smoothing properties of multi-stage schemes applied to hyperbolic equations, *Technical Note 173*, Von Karman Institute for Fluid Dynamics, 1990.

[3] E. Dick and K. Riemslagh, Multi-Staging of Jacobi relaxation in multigrid methods for steady Euler equations II. *J. Comp. Appl. Math.* **59 (3)** (1995), 339–348.

[4] R. Haelterman, J. Vierendeels and D. Van Heule, A generalization of the Runge-Kutta iteration, *J. Comp. Appl. Math.*, doi:10.1016/j.cam.2008.04.021

[5] K. Hosseini and J.J. Alonso, Optimization of multistage coefficients for explicit multigrid flow solvers, AIAA 2003-3705., *16th AIAA Computational Fluid Dynamics Conference*, Orlando, FL, June 23-26, 2003.

[6] K. Hosseini and J.J. Alonso, Practical implementation and improvement of preconditioning methods for explicit multistage flow solvers, AIAA 2004-0763, *42nd AIAA Aerospace Meeting and Exhibit*, Reno, NV, January 2004.

Fig. 20. First figure: Stability region and Fourier symbol (locus of high frequencies in bold). Second figure: amplification curves $|P_m(\tilde{\lambda}(\theta))|$ (dotted line) and $|\mu(\lambda(\theta))|$ (solid line) for the optimal U2 scheme with explicit upwind residual smoothing (UERS,RK). See table 5 for details.

[7] K. Hosseini, Practical implementation of robust preconditioners for optimized multistage flow solvers, Ph.D. thesis, Stanford University, 2005.

[8] J.F. Lynn, Multigrid solution of the Euler equations with local preconditioning, Ph.D. Thesis, The University of Michigan, 1995

[9] B. Van Leer, C.H. Tai and K.G. Powell, Design of optimally smoothing multi-stage schemes for the Euler Equations, *AIAA Paper 89-1923* (1989).

[10] Z.W. Zhu, C.Lacor, C. Hirsch, A new residual smoothing method for multigrid multi-stage schemes, in: Proc. of the 11th AIAA CFD Conference, AIAA-93-3356-CP, 1993.

A Optimal coefficients from previous studies

The results below are taken from [9], resp. [4], and give the optimal coefficients for the optimzer ρ_{HF}, resp. (29), without the use of residual smoothing. $\mathcal{I}_E$ is the convergence factor measured on an actual solver over a V-cycle. It can

be seen that this closely matches values of $\mathcal{I}$, validating our model, except for very low values of m when aliasing (which our model ignores) was too important.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{CERS,RK}} = 0.6586$	$\sqrt[4m]{\mathcal{I}_{CERS,RK}} = 0.8115$	2	(1.8255 , 1.4712)	0.2632
$\sqrt[2m]{\mathcal{I}_{CIRS,RK}} = 0.6586$		2	(0.8996 , 0.3573)	0.7155
$\sqrt[2m]{\mathcal{I}_{CERS,all}} = 0.6596$		2	(1.7482 , 1.3477)	0.2444
$\sqrt[2m]{\mathcal{I}_{CIRS,all}} = 0.6735$		2	(0.9928 , 0.4140)	0.2107
$\sqrt[2m]{\mathcal{I}_{CERS,fil}} = 0.6955$	$\sqrt[(1+2m)]{\mathcal{I}_{CERS,fil}} = 0.7479$	2	(1.0000 , 0.3709)	-0.1822
$\sqrt[2m]{\mathcal{I}_{CIRS,fil}} = 0.6955$		2	(1.0000 , 0.3709)	0.1107

Table 1
Advection equation using U1. Optimal coefficients for Runge-Kutta with central residual smoothing.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{CERS,RK}} = 0.7914$	$\sqrt[4m]{\mathcal{I}_{CERS,RK}} = 0.9293$	3	(1.4224 , 1.3753 , 0.5854)	0.3602
$\sqrt[2m]{\mathcal{I}_{CIRS,RK}} = 0.7699$		3	(0.5841 , 0.2187 , 0.0415)	3.1230
$\sqrt[2m]{\mathcal{I}_{CERS,all}} = 0.7630$		3	(1.4586 , 1.3324 , 0.5574)	0.3526
$\sqrt[2m]{\mathcal{I}_{CIRS,all}} = 0.7732$		3	(0.6613 , 0.2848 , 0.0541)	0.9618
$\sqrt[2m]{\mathcal{I}_{CERS,fil}} = 0.8331$	$\sqrt[(1+2m)]{\mathcal{I}_{CERS,fil}} = 0.8641$	2	(0.4000 , 0.0809)	-0.2749
$\sqrt[2m]{\mathcal{I}_{CIRS,fil}} = 0.8463$		2	(0.3960 , 0.0813)	0.2565

Table 2
Advection equation using U2. Optimal coefficients for Runge-Kutta with central residual smoothing.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{CERS,RK}} = 0.7603$	$\sqrt[4m]{\mathcal{I}_{CERS,RK}} = 0.8720$	4	(2.2450 , 3.0512 , 2.5146 , 1.0678)	0.1771
$\sqrt[2m]{\mathcal{I}_{CIRS,RK}} = 0.7561$		4	(1.5830 , 1.5095 , 0.8719 , 0.2663)	0.3263
$\sqrt[2m]{\mathcal{I}_{CERS,all}} = 0.7603$		4	(2.1231 , 2.7389 , 2.1224 , 0.8486)	0.1516
$\sqrt[2m]{\mathcal{I}_{CIRS,all}} = 0.7562$		4	(1.5711 , 1.5088 , 0.8693 , 0.2596)	0.2382
$\sqrt[2m]{\mathcal{I}_{CERS,fil}} = 0.7823$	$\sqrt[(1+2m)]{\mathcal{I}_{CERS,fil}} = 0.8039$	4	(1.4008 , 1.3132 , 0.6865 , 0.1693)	-0.1568
$\sqrt[2m]{\mathcal{I}_{CIRS,fil}} = 0.7847$		4	(1.3989 , 1.3183 , 0.6894 , 0.1692)	0.0710

Table 3
Advection equation using K3. Optimal coefficients for Runge-Kutta with central residual smoothing.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{UERS,RK}} = 0.4965$	$\sqrt[4m]{\mathcal{I}_{UERS,RK}} = 0.7046$	1	(1.0000)	0.5978
$\sqrt[2m]{\mathcal{I}_{UERS,all}} = 0.5836$		1	(0.6653)	0.1888
$\sqrt[2m]{\mathcal{I}_{UERS,fil}} = 0.6549$	$\sqrt[(1+2m)]{\mathcal{I}_{UERS,fil}} = 0.7127$	2	(1.0000 , 0.3572)	0.9903
$\sqrt[2m]{\mathcal{I}_{UIRS,fil}} = 0.6563$		2	(1.0000 , 0.3577)	0.1776

Table 4
Advection equation using U1. Optimal coefficients for Runge-Kutta with explicit upwind residual smoothing.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{UERS,RK}} = 0.7108$	$\sqrt[4m]{\mathcal{I}_{UERS,RK}} = 0.8431$	2	(1.1507 , 0.5169)	0.7458
$\sqrt[2m]{\mathcal{I}_{UIRS,RK}} = 0.2538$		2	(1.3107 , 0.4317)	-9.4865
$\sqrt[2m]{\mathcal{I}_{UERS,all}} = 0.7505$		2	(1.0686 , 0.4887)	0.7605
$\sqrt[2m]{\mathcal{I}_{UIRS,all}} = 0.3296$		2	(1.3299 , 0.4418)	-162.0642
$\sqrt[2m]{\mathcal{I}_{UERS,fil}} = 0.8288$	$\sqrt[(1+2m)]{\mathcal{I}_{UERS,fil}} = 0.8605$	2	(0.3846 , 0.0766)	2.1779
$\sqrt[2m]{\mathcal{I}_{UIRS,fil}} = 0.8165$		2	(0.4032 , 0.0800)	0.4444

Table 5
Advection equation using U2. Optimal coefficients for Runge-Kutta with upwind residual smoothing.

Optimizer	Optimizer	m	$(\gamma_1, \ldots, \gamma_m)$	ε
$\sqrt[2m]{\mathcal{I}_{UERS,RK}} = 0.6812$	$\sqrt[4m]{\mathcal{I}_{UERS,RK}} = 0.8254$	3	(2.2311 , 2.4489 , 1.2241)	0.4230
$\sqrt[2m]{\mathcal{I}_{UIRS,RK}} = 0.3750$		4	(4.2684 , 7.1418 , 5.5676 , 1.7262)	453.6564
$\sqrt[2m]{\mathcal{I}_{UERS,all}} = 0.7094$		3	(1.9347 , 2.1126 , 0.9620)	0.3214
$\sqrt[2m]{\mathcal{I}_{UIRS,all}} = 0.4136$		2	(2.0331 , 1.1846)	$1.167 \cdot 10^{13}$
$\sqrt[2m]{\mathcal{I}_{UERS,fil}} = 0.7620$	$\sqrt[(1+2m)]{\mathcal{I}_{UERS,fil}} = 0.7921$	3	(1.0947 , 0.7717 , 0.2359)	0.6260
$\sqrt[2m]{\mathcal{I}_{UIRS,fil}} = 0.7692$		3	(1.0644 , 0.7387 , 0.2313)	0.2040

Table 6
Advection equation using K3. Optimal coefficients for Runge-Kutta with upwind residual smoothing.

	m =1	m =2	m =3	m =4	m =5	m =6
γ_1	0.5000	1.0000	1.5000	2.0000	2.5000	3.0000
γ_2		0.3333	0.9000	1.7060	2.7588	4.0608
γ_3			0.2000	0.7059	1.6380	3.1211
γ_4				0.1176	0.5172	1.4242
γ_5					0.0689	0.3636
γ_6						0.0404
$\sqrt[m]{\rho_{HF}}$	0.7078	0.5786	0.5226	0.4941	0.4780	0.4785
$\sqrt[2m]{\mathcal{I}}$	0.7056	0.7159	0.7703	0.8071	0.8329	0.8521
$\sqrt[2m]{\mathcal{I}_E}$	0.6267	0.7171	0.7702	0.8075	0.8333	0.8525

Table A.1
Advection equation using U1: optimal m-stage coefficients for ρ_{HF} after conversion from [9] to the formulation in (2).

Fig. 21. First figure: Stability region and Fourier symbol (locus of high frequencies in bold). Second figure: amplification curves $|P_m(\tilde{\lambda}(\theta))|$ (dotted line) and $|\mu(\lambda(\theta))|$ (solid line) for the optimal U2 scheme with implicit upwind residual smoothing (UIRS,RK). See table 5 for details.

	m =2	m =3	m =4	m =5	m =6
γ_1	0.4693	0.6936	0.9214	1.1508	1.3805
γ_2	0.0934	0.2371	0.4289	0.6701	0.9649
γ_3		0.0315	0.1028	0.2235	0.4063
γ_4			0.0103	0.0412	0.1057
γ_5				0.0033	0.0158
γ_6					0.0011
$\sqrt[m]{\rho_{HF}}$	0.7872	0.7260	0.6958	0.6690	0.6590
$\sqrt[2m]{\mathcal{I}}$	0.8655	0.8839	0.9000	0.9127	0.9222
$\sqrt[2m]{\mathcal{I}_E}$	0.8656	0.8840	0.8990	0.9128	0.9222

Table A.2
Advection equation using U2: optimal m-stage coefficients for ρ_{HF} after conversion from [9] to the formulation in (2).

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.8276	1.3254	1.7320	2.1668	2.5975
γ_2	0.4535	0.8801	1.5824	2.4419	3.4956
γ_3		0.3364	0.8296	1.7101	2.9982
γ_4			0.2394	0.7333	1.7118
γ_5				0.1695	0.6194
γ_6					0.1194
$\sqrt[m]{\rho_{HF}}$	0.8383	0.7769	0.7385	0.7150	0.7018
$\sqrt[2m]{\mathcal{I}}$	0.8491	0.8426	0.8531	0.8668	0.8788
$\sqrt[2m]{\mathcal{I}_E}$	0.8585	0.8423	0.8575	0.8646	0.8786

Table A.3
Advection equation using K3: optimal m-stage coefficients for ρ_{HF} after conversion from [9] to the formulation in (2).

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.5000	1.0000	1.5000	1.9951	2.4560	2.8638
γ_2		0.3741	1.1043	2.1789	3.5724	5.1645
γ_3			0.3021	1.1811	2.9506	5.4415
γ_4				0.2557	1.2980	3.3859
γ_5					0.2357	1.1624
γ_6						0.1708
$\sqrt[m]{\rho_{HF}}$	0.7078	0.7046	0.7475	0.7791	0.8032	0.8202
$\sqrt[2m]{\mathcal{I}}$	0.7056	0.7046	0.7475	0.7790	0.8011	0.8201
$\sqrt[2m]{\mathcal{I}_E}$	0.6267	0.7082	0.7452	0.7798	0.7966	0.8173

Table A.4
Advection equation using U1: optimal m-stage coefficients for $\mathcal{I}$. From [4].

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.3881	0.5893	0.7901	0.9863	1.1651
γ_2	0.0803	0.2294	0.4414	0.7090	1.0130
γ_3		0.0331	0.1230	0.2875	0.5320
γ_4			0.0139	0.0630	0.1703
γ_5				0.0058	0.0307
γ_6					0.0024
$\sqrt[m]{\rho_{HF}}$	0.8557	0.8636	0.8721	0.8801	0.8863
$\sqrt[2m]{\mathcal{I}}$	0.8557	0.8636	0.8721	0.8797	0.8863
$\sqrt[2m]{\mathcal{I}_E}$	0.8555	0.8664	0.8688	0.8800	0.8809

Table A.5
Advection equation using U2: optimal m-stage coefficients for $\mathcal{I}$. From [4].

	m =2	m =3	m =4	m =5	m =6
γ_1	0.6499	1.0537	1.3982	1.7255	2.1893
γ_2	0.3070	0.7632	1.3316	2.0143	3.1388
γ_3		0.2417	0.6985	1.4457	2.9109
γ_4			0.1712	0.6372	1.8290
γ_5				0.1412	0.7359
γ_6					0.1512
$\sqrt[m]{\rho_{HF}}$	0.8241	0.7861	0.7894	0.8018	0.8137
$\sqrt[2m]{\mathcal{I}}$	0.8241	0.7861	0.7894	0.8018	0.8137
$\sqrt[2m]{\mathcal{I}_E}$	0.8692	0.7871	0.7960	0.8005	0.8131

Table A.6
Advection equation using K3: optimal m-stage coefficients for $\mathcal{I}$. From [4].