Journal of Computational and Applied Mathematics 224 (2009) 152–167

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A generalization of the Runge–Kutta iteration

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

[a] Royal Military Academy, Department of Mathematics (MWMW), Renaissancelaan 30, B-1000 Brussels, Belgium
[b] Ghent University, Department of Flow, Heat and Combustion Mechanics, St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium

ARTICLE INFO

Article history:
Received 16 March 2007

MSC:
65F10
65N55

Keywords:
Iterative solution
Multi-grid
Multi-stage

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. We show that a generally used formulation [A. Jameson, Numerical solution of the Euler equations for compressible inviscid fluids, in: F. Angrand, A. Dervieux, J.A. Désidéri, R. Glowinski (Eds.), Numerical Methods for the Euler Equations of Fluid Dynamics, SIAM, Philadelphia, 1985, pp. 199–245] does not allow to form all possible polynomial transmittance functions and we propose a new formulation to remedy this, without using an excessive number of coefficients.

After having converted the optimal parameters found in previous studies (e.g. [B. Van Leer, C.H. Tai, K.G. Powell, Design of optimally smoothing multi-stage schemes for the Euler equations, AIAA Paper 89–1923, 1989]) we compare them with those that we obtain when we optimize for an integrated 2-grid V-cycle and show that this results in superior performance using a low number of stages. We also propose a variant of our new formulation that roughly follows the idea of the Martinelli–Jameson scheme [A. Jameson, Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiter and their effect on multigrid convergence, Int. J. Comput. Fluid Dyn. 4 (1995) 171–218; J.V. Lassaline, Optimal multistage relaxation coefficients for multigrid flow solvers. http://www.ryerson.ca/~jvl/papers/cfd2005.pdf] used on the advection–diffusion equation which that can be extended to other types. Gains in the order of 30%–50% have been shown with respect to classical iterative schemes on the advection equation. Better results were also obtained on the advection–diffusion equation than with the Martinelli–Jameson coefficients, but with less than half the number of matrix-vector multiplications.

1. Introduction

Euler/Navier–Stokes solvers have long been using explicit multi-stage time-marching schemes because of their simplicity. Acceleration techniques like multi-grid have been successfully added, which depend heavily on the smoothing properties of the multi-stage scheme. Despite its 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 the observation 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 is used [4,5]. For that reason we focus on this simple equation.

* Corresponding author. Tel.: +32 2 742 6527; fax: +32 2 742 6512.
E-mail addresses: Robby.Haelterman@rma.ac.be (R. Haelterman), Jan.Vierendeels@UGent.be (J. Vierendeels), Dirk.Van.Heule@rma.ac.be (D. Van Heule).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.04.021

It has already been discovered [5] 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 extend the Fourier analysis to other components of an idealized two-grid V-cycle (restriction, defect correction and prolongation) and look at the resulting transmittance for which we try and find the optimal set of coefficients.

This paper is organized as follows. In Sections 2 and 3 we give a brief introduction on multi-stage solvers of the Runge–Kutta type; in Section 4 we propose our alternative formulation; in Section 5 we detail the different objective functions that we use in the optimization procedure and discuss the results which show that this integrated approach gives improved convergence speeds, but also show that a high number of stages is not necessarily beneficial. Finally, in Section 6, we introduce a variant that splits the discretization matrix in its Hermitian and anti-Hermitian part and show that further improvements can be obtained.

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

We want to solve a linear algebraic system of the type

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

where $A \in \mathbb{R}^{p\times p}$; $\mathbf{u}, \mathbf{b} \in \mathbb{R}^{p\times 1}$, which typically results from the discretization of a linear ordinary (or partial) differential equation; we will go into more detail in Section 5. If the dimension of the above system becomes too large, the solution is often found in an iterative way. The subclass of iterative solvers that we consider here starts from a regular splitting of $A : A = M - N$ where M is regular. This results in

$$\mathbf{u}_{n+1} = M^{-1}N\mathbf{u}_n + M^{-1}\mathbf{b} \tag{2}$$

starting from an initial guess $\mathbf{u}_o$, or, after rearranging the terms,

$$\mathbf{u}_{n+1} = \mathbf{u}_n - M^{-1}\mathbf{r}_n \tag{3}$$

where we call $\mathbf{r}_n = A\mathbf{u}_n - \mathbf{b}$ the residual at iteration n. The splitting that will concern us most in this paper is that with

$$M = \frac{1}{\tau} I_p \tag{4}$$

where $I_p \in \mathbb{R}^{p\times p}$ is the identity matrix and τ is an iteration parameter. With a fixed value of τ this scheme is called the stationary Richardson iteration. It is obvious that it corresponds to

$$\mathbf{u}_{n+1} = \mathbf{u}_n - \tau\,(A\mathbf{u}_n - \mathbf{b}) \tag{5}$$

which can be interpreted as stemming from a time discretization, with a (pseudo-) time-step τ):

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\tau} + A\mathbf{u}_n = \mathbf{b} \tag{6}$$

where we assume that we are only interested in the steady state solution. In the following paragraphs we build upon this single-stage iterative scheme to construct the various multi-stage schemes. In Section 6 we will consider yet other ways to split A.

3. The Runge–Kutta time-stepping scheme

The Runga–Kutta (R–K) scheme can be implemented for a scalar time-stepping scheme. In a commonly used matrix form it is given by

$$\begin{aligned}
&\mathbf{U}_{(o)} = \mathbf{u}_n \\
&\mathbf{U}_{(1)} = \mathbf{U}_{(o)} - \alpha_1 M^{-1}\mathbf{r}_{(o)} \\
&\vdots \\
&\mathbf{U}_{(l+1)} = \mathbf{U}_{(o)} - \alpha_{l+1} M^{-1}\mathbf{r}_{(l)} \\
&\vdots \\
&\mathbf{U}_{(m)} = \mathbf{U}_{(o)} - \alpha_m M^{-1}\mathbf{r}_{(m-1)} \\
&\mathbf{u}_{n+1} = \mathbf{U}_{(m)}
\end{aligned} \tag{7}$$

where $\alpha_1, \ldots, \alpha_m$ are the iteration parameters. We can write $\mathbf{u}_n = \mathbf{u}_{\text{exact}} + \mathbf{e}_n$, where $\mathbf{e}_n$ is the error of $\mathbf{u}_n$ with respect to the exact solution $\mathbf{u}_{\text{exact}} = A^{-1}\mathbf{b}$.

After some algebra we find

$$\mathbf{e}_{n+1} = P_m(-M^{-1}A)\mathbf{e}_n \tag{8}$$

where the transmittance function P_m is given by the polynomial

$$P_m(z) = 1 + \sum_{l=1}^{m} \left(\prod_{i=m-l+1}^{m} \alpha_i \right) z^l \tag{9}$$

$$= 1 + \sum_{l=1}^{m} \beta_l z^l \tag{10}$$

with $\beta_l = \prod_{i=m-l+1}^{m} \alpha_i$.

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

$$(\mathbf{e}_{n+1})_i = P_m(-\lambda_i)(\mathbf{e}_n)_i. \tag{11}$$

In order to have a stable scheme we require that

$$|P_m(-\lambda_i)| \leq 1 \quad (\forall i \in \{1, \ldots, p\}). \tag{12}$$

The main limitation of the Runge–Kutta solver in this formulation is that the following statement holds

$$\exists l < m : \beta_l = 0 \Rightarrow \quad \forall k \in \{l+1, l+2, \ldots, m\} : \beta_k = 0. \tag{13}$$

Remark. Other formulations exist, e.g. in [2], but these use $m + (m-1)!$ coefficients to form a polynomial of degree m and require more storage.

4. An alternative Runge–Kutta formulation

To avoid the limitation (13) of the Runge–Kutta scheme we propose the following formulation, which can be interpreted as a periodic non-stationary method [12].

$$\begin{aligned}
&\mathbf{U}_{(o)} = \mathbf{u}_n \\
&\mathbf{V}_{(o)} = M^{-1}\mathbf{r}_{(o)} \\
&\mathbf{U}_{(1)} = \mathbf{U}_{(o)} - \gamma_1 \mathbf{V}_{(o)} \\
&\mathbf{V}_{(1)} = -M^{-1}A\mathbf{V}_{(o)} \\
&\mathbf{U}_{(2)} = \mathbf{U}_{(1)} - \gamma_2 \mathbf{V}_{(1)} \\
&\mathbf{V}_{(2)} = -M^{-1}A\mathbf{V}_{(1)} \\
&\vdots \\
&\mathbf{V}_{(l)} = -M^{-1}A\mathbf{V}_{(l-1)} \\
&\mathbf{U}_{(l+1)} = \mathbf{U}_{(l)} - \gamma_{l+1} \mathbf{V}_{(l)} \\
&\vdots \\
&\mathbf{V}_{(m-1)} = -M^{-1}A\mathbf{V}_{(m-2)} \\
&\mathbf{U}_{(m)} = \mathbf{U}_{(m-1)} - \gamma_m \mathbf{V}_{(m-1)} \\
&\mathbf{u}_{n+1} = \mathbf{U}_{(m)}
\end{aligned} \tag{14}$$

where $\gamma_1, \ldots, \gamma_m$ are the iteration parameters. This results in

$$\mathbf{e}_{n+1} = P'_m(-M^{-1}A)\mathbf{e}_n \tag{15}$$

$$(\mathbf{e}_{n+1})_i = P'_m(-\lambda_i)(\mathbf{e}_n)_i \quad (i = 1, \ldots, p) \tag{16}$$

where

$$P'_m(z) = 1 + \sum_{l=1}^{m} \gamma_l z^l. \tag{17}$$

It is easy to see that any real m-th degree polynomial respecting $P'_m(0) = 1$ can be constructed in this way, thus avoiding the limitation in (13).

If $\forall \lambda_i \in \sigma(M^{-1}A) : P_m(\lambda_i) = P'_m(\lambda_i)$ Eqs. (9) and (17) then the coefficients α_l and $\gamma_l (l = 1, \ldots, m)$ are related by the following expression

$$\prod_{m-l+1}^{m} \alpha_i = \gamma_l \quad (l = 1, \ldots, m). \tag{18}$$

We see that in this new formulation intermediate results need not be stored which reduces storage cost. Another added benefit of the scheme is that it uses the coefficients of the transmittance polynomial directly which will result in fewer rounding errors when finite precision coefficients are used, whereas for (9) it is obtained by multiplication. From Eq. (17) we also see that

$$P'_m(-M^{-1}A) = I_p + \left(\sum_{l=1}^{m} \gamma_l (-M^{-1}A)^{l-1} \right) (-M^{-1}A). \tag{19}$$

We can thus interpret the above equation as a preconditioned form of (3), where the polynomial preconditioner is given by $\sum_{l=1}^{m} \gamma_l (-M^{-1}A)^{l-1}$.

5. Optimization of parameters

5.1. The equations under consideration

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

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \tag{20}$$

$(a \in \mathbb{R}_o^+)$ with suitable boundary conditions. We are only interested in the steady-state solution of Eq. (20). The spatial derivative is discretized with a first or second order upwind scheme (resp. U1 and U2) or a K3 upwind biased scheme.

With the mesh size given by Δx, the discretization of the space operator $\frac{\partial u}{\partial x}$ (on a regular mesh) gives

- U1

$$\frac{u_j - u_{j-1}}{\Delta x} \tag{21}$$

- U2

$$\frac{3u_j - 4u_{j-1} + u_{j-2}}{2\Delta x} \tag{22}$$

- K3

$$\frac{2u_{j+1} + 3u_j - 6u_{j-1} + u_{j-2}}{6\Delta x}. \tag{23}$$

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 then shows that it is given by

- U1

$$\lambda(\theta) = \frac{1 - \mathrm{e}^{-\mathrm{i}\theta}}{\Delta x} \tag{24}$$

- U2

$$\lambda(\theta) = \frac{3 - 4\mathrm{e}^{-\mathrm{i}\theta} + \mathrm{e}^{-2\mathrm{i}\theta}}{2\Delta x} \tag{25}$$

- K3

$$\lambda(\theta) = \frac{3 + 2\mathrm{e}^{\mathrm{i}\theta} - 6\mathrm{e}^{-\mathrm{i}\theta} + \mathrm{e}^{-2\mathrm{i}\theta}}{6\Delta x}. \tag{26}$$

5.2. Objective functions

With every $\theta \in [-\pi, \pi]$ we associate an eigenvalue $\lambda(\theta)$ and an eigenvector $\mathbf{E}_\theta$. We use $(\mathbf{e}_n)_\theta$ for the corresponding component of $\mathbf{e}_n$. This allows us to write

$$(\mathbf{e}_{n+1})_\theta = P'_m(-\lambda(\theta))(\mathbf{e}_n)_\theta. \tag{27}$$

The eigenvector corresponding to the eigenvalue 0 is the "constant vector" ($= [1 \ 1 \ \dots \ 1]^{\mathrm{T}}$), which we assume to be eliminated due to the presence of boundary conditions.

While it is possible to find coefficients that allow this equation to be solved with (14) 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 we turn our attention to multi-grid schemes, and use (14) 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 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 considerations both approaches are similar.

When looking for good coefficients in this context 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

$$\rho_{HF} = \sup_{\theta \in \Phi_{HF}} |P'_m(-\lambda(\theta))|. \tag{28}$$

A first objective function to minimize would simply be

$$\mathcal{I}_1 = \rho_{HF} \tag{29}$$

which depends on the coefficients $\gamma_1, \dots, \gamma_m$. We add the stability constraint

$$\sup_{\theta \in [-\pi, \pi]} |P'_m(-\lambda(\theta))| \leq 1. \tag{30}$$

This objective function is the one used in most previous studies (e.g. [1,11]) and means that we ignore the effect of the remainder of the multi-grid cycle. Additionally, [11] required that the polynomial had a number of zeroes in the high frequency domain.

We could refine this objective function by adding the effect of the defect correction. We consider a two-grid V-cycle, where we assume that the solution on the coarser grid (defect correction) is exact (ideal two-grid cycle). This 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. Their effect can be quantified by the following transmittance functions (resp. for the restriction and prolongation) that act on the residual [3]

$$\mu_R = \frac{1}{4}\left(\mathrm{e}^{\mathbf{i}\theta} + 2 + \mathrm{e}^{-\mathbf{i}\theta}\right) \tag{31}$$

$$= \left(\cos\frac{\theta}{2}\right)^2 \tag{32}$$

$$\mu_P = \mu_R. \tag{33}$$

This means that even the best defect correction will only be able to reduce a fraction of the low frequency part of the residual, which is given by

$$\mathbf{r}_n = A\mathbf{e}_n \tag{34}$$

$$(\mathbf{r}_n)_\theta = A(\mathbf{e}_n)_\theta = \lambda(\theta)(\mathbf{e}_n)_\theta. \tag{35}$$

The resulting objective function then becomes

$$\mathcal{I}_2 = \sup_{\theta \in [-\pi, \pi]} \left|\mu_{V,1}(\theta)\right| \tag{36}$$

where

$$\mu_{V,1}(\theta) = \left(1 - \left(\cos\frac{\theta}{2}\right)^4\right)\left(P'_m(-\lambda(\theta))\right)^2 \quad \text{for } \theta \in \Phi_{LF} \tag{37}$$

$$\left(P'_m(-\lambda(\theta))\right)^2 \quad \text{for } \theta \in \Phi_{HF}. \tag{38}$$

(Minimizing $\mathcal{I}_2$ will automatically take into account the stability constraint $\mathcal{I}_2(\theta) \leq 1, \forall\theta \in [-\pi, \pi]$.)

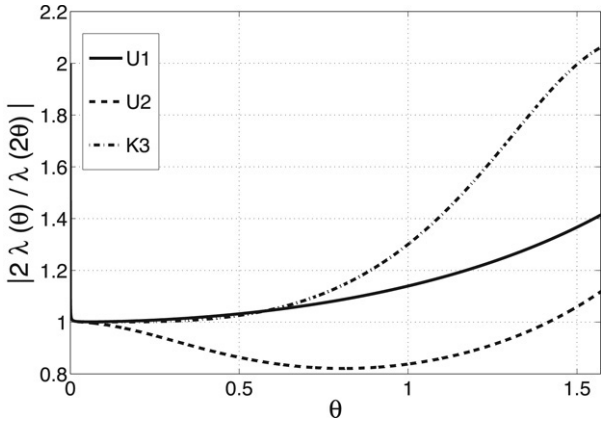

Fig. 1. $\left|\frac{2\lambda(\theta)}{\lambda(2\theta)}\right|$ for the advection equation; different discretization schemes.

Using this objective function means that we take a pre-smoothing step followed by an idealized defect correction, that is only limited by the filtering done by restriction and prolongation, and finally a post-smoothing step. Using the coefficients that minimize this function will lead to a smoother that will sacrifice damping of the high frequencies for a better resolution of the somewhat lower frequencies. Due to the fact that the stability constraint is now taken over the whole V-cycle, it is also quite possible that the smoother itself is not stable.

A last refinement with a respect to the previous objective function takes into account the limitations of the defect correction itself. Even if we consider a direct solver on the coarsest grid, which is the ideal situation, it is limited by its ability to eliminate the low frequency errors if we use a similar discretization scheme on the coarsest grid. If we use a coarse grid spacing of $2\Delta x$, this can be quantified by the Fourier symbol on the coarsest grid which is given by $\frac{\lambda(2\theta)}{2}$ ($\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$) [3]. The solution process on the coarsest grid then computes

$$\frac{\lambda(2\theta)}{2}\,(\mathbf{v}_\theta)_{\text{coarse}} = (\mathbf{r}_\theta)_{\text{coarse}} = \lambda(\theta)\,(\mathbf{e}_\theta)_{\text{coarse}} \left(\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right) \tag{39}$$

where $(\mathbf{v}_\theta)_{\text{coarse}}$ stands for the defect correction on the coarse grid, and $(\mathbf{r}_\theta)_{\text{coarse}}$ ($(\mathbf{e}_\theta)_{\text{coarse}}$) for the value of the residual (error) on the coarse grid, after restriction. $\frac{2\lambda(\theta)}{\lambda(2\theta)}$ is then a measure of how well the coarse grid operator can resolve the low frequencies. We finally propose our third objective function

$$\mathcal{I}_3 = \sup_{\theta\in[-\pi,\pi]} \left|\mu_{V,2}(\theta)\right| \tag{40}$$

where

$$\mu_{V,2}(\theta) = \left(1 - \left(\cos\frac{\theta}{2}\right)^4 \frac{2\lambda(\theta)}{\lambda(2\theta)}\right) \left(P'_m(-\lambda(\theta))\right)^2 \quad \text{for } \theta \in \Phi_{LF} \tag{41}$$

$$\left(P'_m(-\lambda(\theta))\right)^2 \quad \text{for } \theta \in \Phi_{HF}. \tag{42}$$

Again the minimization procedure will result in $\mathcal{I}_3(\theta) \leq 1, \forall\theta \in [-\pi, \pi]$ if an optimum exists.

The last two objective functions ($\mathcal{I}_2$ and $\mathcal{I}_3$) 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 note that we are in effect minimizing a spectral radius, which means that we are optimizing the asymptotic rate of convergence [12].

To illustrate the effect of $\frac{2\lambda(\theta)}{\lambda(2\theta)}$ for the advection equation, we plot this ratio for U1,U2 and K3 in Fig. 1. We see that this results in a slight over-correction for U1 and a more pronounced over-correction for K3. For U2 the ratio approximates unity quite closely over a wide range of values of θ, but with a slight under-correction.

In itself, over-correction is not necessarily a bad thing, as it compensates for the filtering done by the restriction and the prolongation. For that reason we also look at $\left(1 - \left(\cos\frac{\theta}{2}\right)^4 \frac{2\lambda(\theta)}{\lambda(2\theta)}\right)$ for the three schemes and see (Fig. 2) that all three have difficulties reducing errors around $\theta = \frac{\pi}{2}$, which is most pronounced for K3. However, K3 manages to reduce errors close to $\theta = 0$ best.

5.3. Results

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

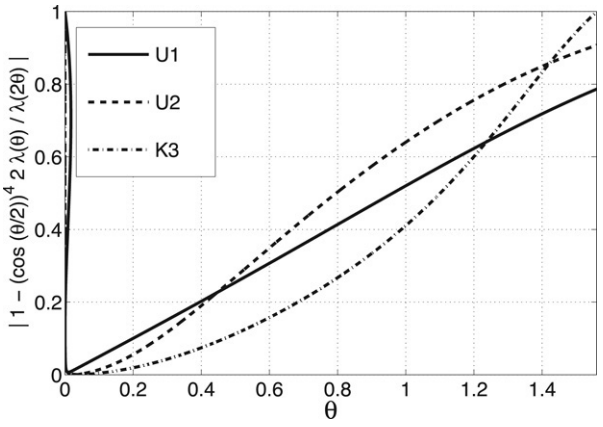

Fig. 2. $\left|1-\left(\cos\frac{\theta}{2}\right)^4\frac{2\lambda(\theta)}{\lambda(2\theta)}\right|$ for the advection equation; different discretization schemes.

Table 1
Optimal m-stage coefficients for $\mathcal{I}_1$

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.7078	0.5786	0.5226	0.4941	0.4780	0.4785
$\sqrt[2m]{\mathcal{I}_2}$	0.7056	0.6651	0.7179	0.7566	0.7849	0.8066
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using U1 after conversion from [11].

Table 2
Optimal m-stage coefficients for $\mathcal{I}_1$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.7872	0.7260	0.6958	0.6690	0.6590
$\sqrt[2m]{\mathcal{I}_2}$	0.7909	0.8208	0.8470	0.8667	0.8814
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using U2 after conversion from [11].

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. The stability constraint was implemented as a penalty function on the objective function.

The optimal coefficients for the U1, U2 and K3 scheme, when considered as a smoother for multi-grid and using $\mathcal{I}_1$ are taken from [11] and converted to corresponding γ-values. These can be found in Tables 1–3.

All schemes under consideration tacitly assume $\Delta x = 1$, and use no preconditioner ($M = I_p$). If different values of Δx are needed the following re-scaling can be used

$$\gamma_k \to (\Delta x)^k \gamma_k. \tag{43}$$

This is a similar relationship that is obtained when the CFL-number is explicitly retained as an iteration parameter (e.g. [1,11]).

For each optimum set of coefficients we also give the values of the other objective functions in Tables 4–9. To be able to compare the merit of schemes with different number of stages we give the values of $\sqrt[m]{\mathcal{I}_1}$, $\sqrt[2m]{\mathcal{I}_2}$ and $\sqrt[2m]{\mathcal{I}_3}$.

Table 3
Optimal m-stage coefficients for $\mathcal{I}_1$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.8383	0.7769	0.7385	0.7150	0.7018
$\sqrt[2m]{\mathcal{I}_2}$	0.8383	0.8290	0.8492	0.8664	0.8799
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using K3 after conversion from [11].

Table 4
Optimal m-stage coefficients for $\mathcal{I}_2$

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.5000	1.0000	1.5000	2.0015	2.4049	2.8864
γ_2		0.3573	1.0422	2.0510	3.1960	4.6695
γ_3			0.2711	1.0496	2.4078	4.3522
γ_4				0.2166	0.9788	2.3736
γ_5					0.1681	0.6978
γ_6						0.0851
$\sqrt[m]{\mathcal{I}_1}$	0.7078	0.6551	0.6913	0.7208	0.7450	0.7661
$\sqrt[2m]{\mathcal{I}_2}$	0.7056	0.6551	0.6912	0.7207	0.7441	0.7661
$\sqrt[2m]{\mathcal{I}_3}$	0.7056	0.7091	0.7536	0.7855	0.8092	0.8291
$\sqrt[2m]{\mathcal{I}_E}$	0.6267	0.7117	0.7519	0.7864	0.8094	0.8293

Advection equation using U1.

Table 5
Optimal m-stage coefficients for $\mathcal{I}_3$

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.7078	0.7046	0.7475	0.7791	0.8032	0.8202
$\sqrt[2m]{\mathcal{I}_2}$	0.7056	0.7046	0.7475	0.7791	0.8011	0.8202
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using U1.

Table 6
Optimal m-stage coefficients for $\mathcal{I}_2$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.4067	0.6095	0.8113	1.0163	1.2324
γ_2	0.0780	0.2202	0.4217	0.6834	1.0135
γ_3		0.0303	0.1117	0.2639	0.5145
γ_4			0.0121	0.0557	0.1613
γ_5				0.0050	0.00286
γ_6					0.0022
$\sqrt[m]{\mathcal{I}_1}$	0.7878	0.8062	0.8220	0.8349	0.8481
$\sqrt[2m]{\mathcal{I}_2}$	0.7879	0.8062	0.8219	0.8349	0.8459
$\sqrt[2m]{\mathcal{I}_3}$	0.8623	0.8714	0.8799	0.8872	0.8935
$\sqrt[2m]{\mathcal{I}_E}$	0.8611	0.8741	0.8746	0.8858	0.8900

Advection equation using U2.

Table 7
Optimal m-stage coefficients for $\mathcal{I}_3$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.8557	0.8636	0.8721	0.8801	0.8863
$\sqrt[2m]{\mathcal{I}_2}$	0.8558	0.8636	0.8721	0.8801	0.8863
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using U2.

Table 8
Optimal m-stage coefficients for $\mathcal{I}_2$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
γ_1	0.6683	1.0707	1.3979	1.7340	2.1202
γ_2	0.3118	0.7810	1.3395	2.0442	3.0050
γ_3		0.2454	0.7033	1.4717	2.7662
γ_4			0.1719	0.6468	1.7201
γ_5				0.1431	0.6843
γ_6					0.1398
$\sqrt[m]{\mathcal{I}_1}$	0.8167	0.7829	0.7921	0.8054	0.8165
$\sqrt[2m]{\mathcal{I}_2}$	0.8144	0.7829	0.7921	0.8054	0.8165
$\sqrt[2m]{\mathcal{I}_3}$	0.8309	0.7888	0.7921	0.8054	0.8165
$\sqrt[2m]{\mathcal{I}_E}$	0.8880	0.7903	0.7986	0.8040	0.8163

Advection equation using K3.

Table 9
Optimal m-stage coefficients for $\mathcal{I}_3$

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
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
$\sqrt[m]{\mathcal{I}_1}$	0.8241	0.7861	0.7894	0.8018	0.8137
$\sqrt[2m]{\mathcal{I}_2}$	0.8241	0.7861	0.7933	0.8068	0.8183
$\sqrt[2m]{\mathcal{I}_3}$	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

Advection equation using K3.

It was found that the frequencies that converged most slowly were those around $\theta = \frac{\pi}{2}$, which is not entirely unexpected. To validate the results obtained by the model an actual solver using 2000 nodes was used and its asymptotic convergence rate measured in the L2-norm. This is given by $\mathcal{I}_E$. The agreement between this value and $\mathcal{I}_3$ was in general very good, thus proving that the latter serves as an adequate model, except for $m = 1$ when aliasing was to prominent.

We can see that an increase in m will lead to a net computational gain in the value of $\mathcal{I}_1$. For that reason a high number of stages has been advocated in the past. However, looking at the complete picture ($\sqrt[2m]{\mathcal{I}_2}$ and $\sqrt[2m]{\mathcal{I}_3}$), including the defect correction, the interpretation is markedly different, and the improvement is absent. (We use $\sqrt[2m]{}$ because we apply the same smoother twice in the cycle.) We recall that in our study the resolution of the low frequency has been idealized. This means that we assume to have obtained the best defect correction theoretically possible within the framework of the study and have ignored aliasing. Real multi-grid solvers that use iterative solvers on (a) coarser grid(s) will always give somewhat less efficient defect corrections.

When we now optimize our solver for $\mathcal{I}_2$ and $\mathcal{I}_3$ we get better asymptotic convergence rates (over the whole cycle) than when optimizing for $\mathcal{I}_1$. However, when we look at the reduction per stage ($\sqrt[2m]{}$) we see that using a higher number of stages – while resulting in a better smoother – does not necessarily result in a better V-cycle. The reason is the limited ability to resolve modes slightly below $\frac{\pi}{2}$ (Fig. 2), even though we are using an idealized coarse grid solution. Judging from the model, the optimal number of stages to use would be 1 or 2 for U1, 2 for U2, 3 for K3 and 1 for the Poisson equation. Experiments on real solvers confirmed these results, and showed that the optimal value of m was 1 for the U1 scheme. In

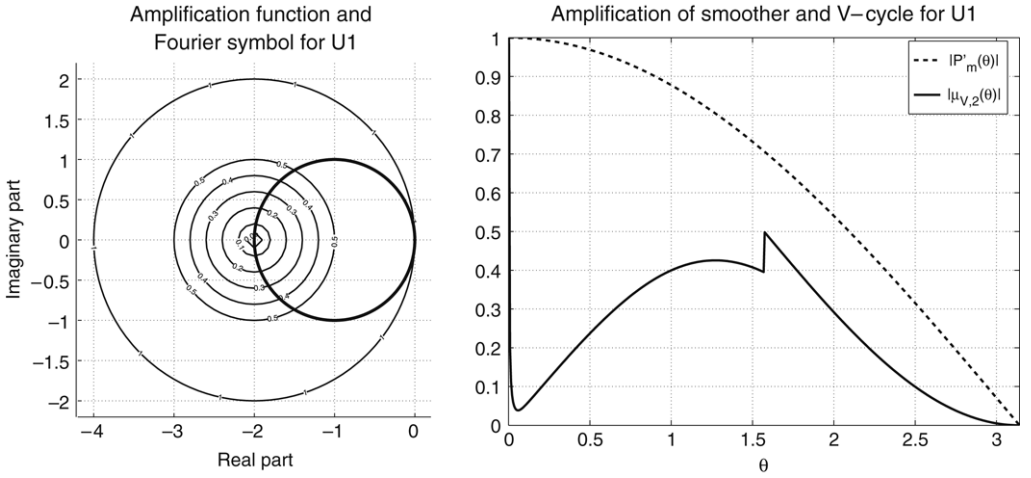

Fig. 3. Optimal U1 scheme ($m = 1$; $\gamma_1 = 0.5$). Figure left: Stability region and Fourier symbol. Figure right: Amplification curves $|P'_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line).

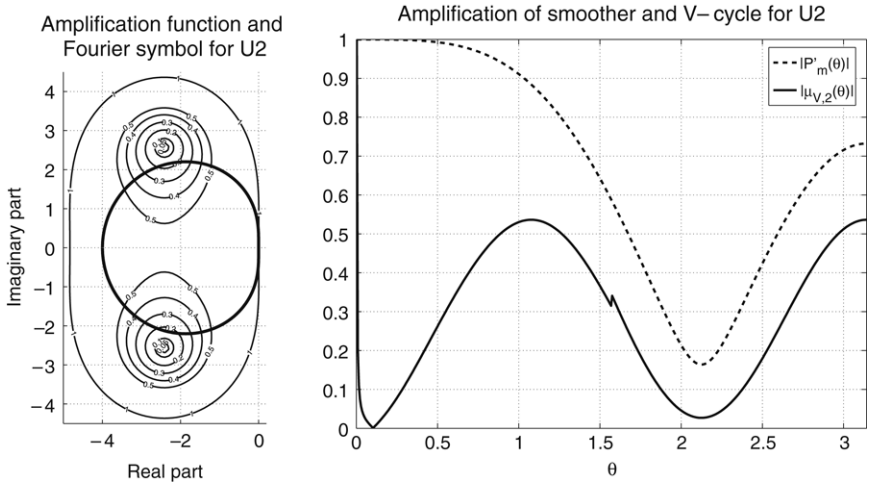

Fig. 4. Optimal U2 scheme ($m = 2$; $\gamma_1 = 0.3881$, $\gamma_2 = 0.0803$). Figure left: Stability region and Fourier symbol. Figure right: Amplification curves $|P'_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line).

general the experimental convergence rates again differed only within 0.5%–5% of the values of the theoretical model (based on $\mathcal{I}_3$), except for $m = 1$ in the U1 scheme, and $m = 2$ for the K3 scheme, which is believed to be due to the effect of aliasing (neglected in the model). The integrated optimization approach ($\mathcal{I}_3$) yielded an improvement over the old coefficients, which for the optimal number of stages amounted to 33% for the K3 scheme ($m = 3$), 5% for the U2 ($m = 2$) scheme and equal performance for U1 ($m = 1$). For higher values of m the optimization of $\mathcal{I}_3$ steadily outperformed that of $\mathcal{I}_1$.

We illustrate the results for the different optimal schemes by drawing the stability region $S = \{z \in \mathbb{C}; |P'_m(z)| \leq 1\}$ on which we superpose $(-\lambda(\theta))$ and also give $|P'_m(\lambda(\theta))|$ (the amplification of the smoother) and $|\mu_{V,2}(\lambda(\theta))|$ (the amplification of the V-cycle). We point out that, due to rounding errors in the actual computation of Eq. (39), $|\mu_{V,2}(\lambda(\theta))|$ sometimes tended to 1 in the neighborhood of $\theta = 0$ (see Figs. 3–5).

It is clear that the new optimization procedure results in an amplification curve that respects the equal excursion principle, but now taken over the complete V-cycle and that smoothing has been sacrificed to some extent.

6. Iterative scheme using a splitting of the discretization matrix

6.1. Formulation

In analogy to the Modified-Runge Kutta scheme [5] we propose a scheme that splits A in two parts

$$A = B + C. \tag{44}$$

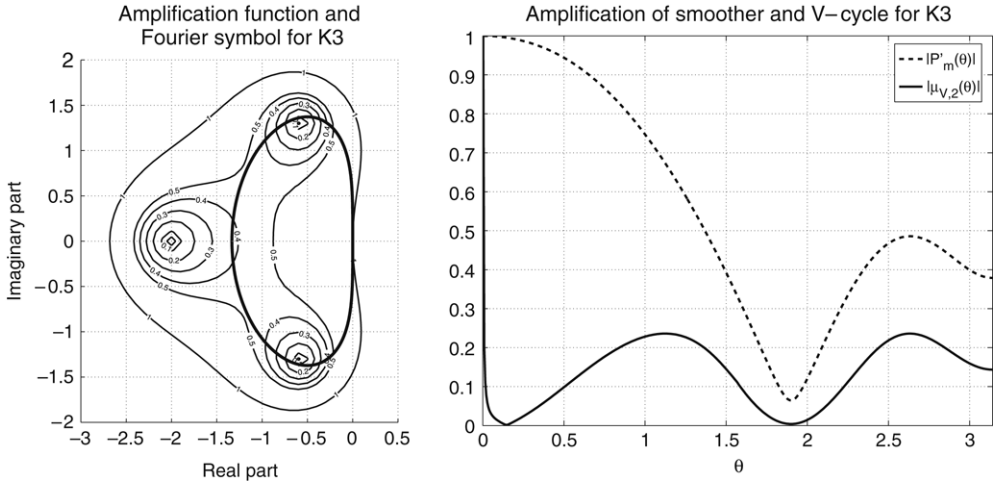

Fig. 5. Optimal K3 scheme ($m = 3$; $\gamma_1 = 1.0537$, $\gamma_2 = 0.7632$, $\gamma_3 = 0.2417$). Figure left: Stability region and Fourier symbol. Figure right: Amplification curves $|P'_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line).

The scheme is given by

$$
\begin{aligned}
&\mathbf{U}_{(o)} = \mathbf{u}_n \\
&\mathbf{V}_{(o)} = M^{-1}\mathbf{r}_{(o)} \\
&\mathbf{U}_{(1)} = \mathbf{U}_{(o)} - \gamma_1 \mathbf{V}_{(o)} \\
&\mathbf{V}_{B(1)} = -M^{-1}B\mathbf{V}_{(o)} \\
&\mathbf{V}_{C(1)} = -M^{-1}C\mathbf{V}_{(o)} \\
&\mathbf{U}_{(2)} = \mathbf{U}_{(1)} - \gamma_{B,2}\mathbf{V}_{B(1)} - \gamma_{C,2}\mathbf{V}_{C(1)} \\
&\mathbf{V}_{B(2)} = -M^{-1}B\mathbf{V}_{B(1)} \\
&\mathbf{V}_{C(2)} = -M^{-1}C\mathbf{V}_{C(1)} \\
&\vdots \\
&\mathbf{V}_{B(l)} = -M^{-1}B\mathbf{V}_{B(l-1)} \\
&\mathbf{V}_{C(l)} = -M^{-1}C\mathbf{V}_{C(l-1)} \\
&\mathbf{U}_{(l+1)} = \mathbf{U}_{(l)} - \gamma_{B,l+1}\mathbf{V}_{B(l)} - \gamma_{C,l+1}\mathbf{V}_{C(l)} \\
&\vdots \\
&\mathbf{V}_{B(m-1)} = -M^{-1}B\mathbf{V}_{B(m-2)} \\
&\mathbf{V}_{C(m-1)} = -M^{-1}C\mathbf{V}_{C(m-2)} \\
&\mathbf{U}_{(m)} = \mathbf{U}_{(m-1)} - \gamma_{B,m}\mathbf{V}_{B(m-1)} - \gamma_{C,m}\mathbf{V}_{C(m-1)} \\
&\mathbf{u}_{n+1} = \mathbf{U}_{(m)}
\end{aligned}
\tag{45}
$$

which results in

$$\mathbf{e}_{n+1} = P''_m(-M^{-1}B, -M^{-1}C)\mathbf{e}_n \tag{46}$$

where

$$P''_m(z_1, z_2) = 1 + \left(\gamma_1 + \sum_{l=2}^{m}\left(\gamma_{B,l}z_1^{l-1} + \gamma_{C,l}z_2^{l-1}\right)\right)(z_1 + z_2). \tag{47}$$

Again we note that it is not necessary to store intermediate results and that $\mathbf{V}_B$ and $\mathbf{V}_C$ can be computed in parallel.

We limit ourselves to splittings where $M^{-1}B$ and $M^{-1}C$ have the same eigenvectors as $M^{-1}A$, which is straightforward if A results from a differencing scheme, as illustrated in Section 5. We number the eigenvalues $\lambda_{B,i}$ and $\lambda_{C,i}$ $(i = 1, \ldots, p)$ of $M^{-1}B$ and $M^{-1}C$ respectively in such a way that

$$\lambda_{B,i} + \lambda_{C,i} = \lambda_i \quad (i = 1, \ldots, p). \tag{48}$$

Table 10
Optimal coefficients for $\mathcal{I}_3$ when using the split scheme

	U1	U2	K3
γ_1	0.5000	0.2071	0.4580
$\gamma_{B,2}$	0.0000	0.0000	0.0000
$\gamma_{C,2}$	0.6043	0.1703	0.5471
$\sqrt[4]{\mathcal{I}_3}$	0.6462	0.7779	0.6240
$\sqrt[4]{\mathcal{I}_E}$	0.6413	0.7762	0.6391

Advection equation using different discretizations.

In this case, according to a well-known theorem in [10], $(M^{-1}B)$ and $(M^{-1}C)$ commute, which avoids problems with the commutative property of the scalars in expression (47) and we can write

$$\mathbf{e}_i^{n+1} = P_m''(-\lambda_{B,i}, -\lambda_{C,i})\mathbf{e}_i^n. \tag{49}$$

We will now propose splits such that λ_B is the real part and λ_C the imaginary part of λ_i. In other words where $B = \frac{1}{2}(A + A^{\mathrm{T}})$ and $C = \frac{1}{2}(A - A^{\mathrm{T}})$.

6.1.1. Advection equation with U1

We propose to split the differencing scheme (and hence A) as

$$\frac{1}{\Delta x}\left(\left(\frac{-u_{j+1} + 2u_j - u_{j-1}}{2}\right) + \left(\frac{u_{j+1} - u_{j-1}}{2}\right)\right) \tag{50}$$

which will result in

$$\lambda(\theta) = \underbrace{\frac{1}{\Delta x}(1 - \cos\theta)}_{\lambda_B(\theta)} + \underbrace{\frac{1}{\Delta x}(\mathbf{i}\sin\theta)}_{\lambda_C(\theta)}. \tag{51}$$

This splitting is a very logical one as it writes the upwind discretization as the sum of a central discretization (imaginary part) and a first order diffusion term (real part).

6.1.2. Advection equation with U2

$$\frac{\left(\frac{1}{2}u_{j+2} - 2u_{j+1} + 3u_j - 2u_{j-1} + \frac{1}{2}u_{j-2}\right) + \left(-\frac{1}{2}u_{j+2} + 2u_{j+1} - 2u_{j-1} + \frac{1}{2}u_{j-2}\right)}{2\Delta x} \tag{52}$$

$$\lambda(\theta) = \underbrace{\frac{1}{2\Delta x}(3 - 4\cos\theta + \cos(2\theta))}_{\lambda_B(\theta)} + \underbrace{\frac{\mathbf{i}}{2\Delta x}(4\sin\theta - \sin(2\theta))}_{\lambda_C(\theta)}. \tag{53}$$

Again the real part corresponds to a diffusion term, this time of third order.

6.1.3. Advection equation with K3

$$\frac{\left(\frac{1}{2}u_{j+2} - 2u_{j+1} + 3u_j - 2u_{j-1} + \frac{1}{2}u_{j-2}\right) + \left(-\frac{1}{2}u_{j+2} + 4u_{j+1} - 4u_{j-1} + \frac{1}{2}u_{j-2}\right)}{6\Delta x} \tag{54}$$

$$\lambda(\theta) = \underbrace{\frac{1}{6\Delta x}(3 - 4\cos\theta + \cos(2\theta))}_{\lambda_B(\theta)} + \underbrace{\frac{\mathbf{i}}{6\Delta x}(8\sin\theta - \sin(2\theta))}_{\lambda_C(\theta)}. \tag{55}$$

6.2. Results

The optimal coefficients obtained with this approach can be found in Table 10. It was found that $m = 2$ gave the optimal efficiency for the three discretizations. Using more coefficients only improved the performance slightly but not enough to compensate for the extra numerical work. It was also found that setting $\gamma_{B,2}$ (corresponding to the diffusive part) to zero did not degrade performance too much. As this allowed us to reduce the number of coefficients (and hence matrix-vector

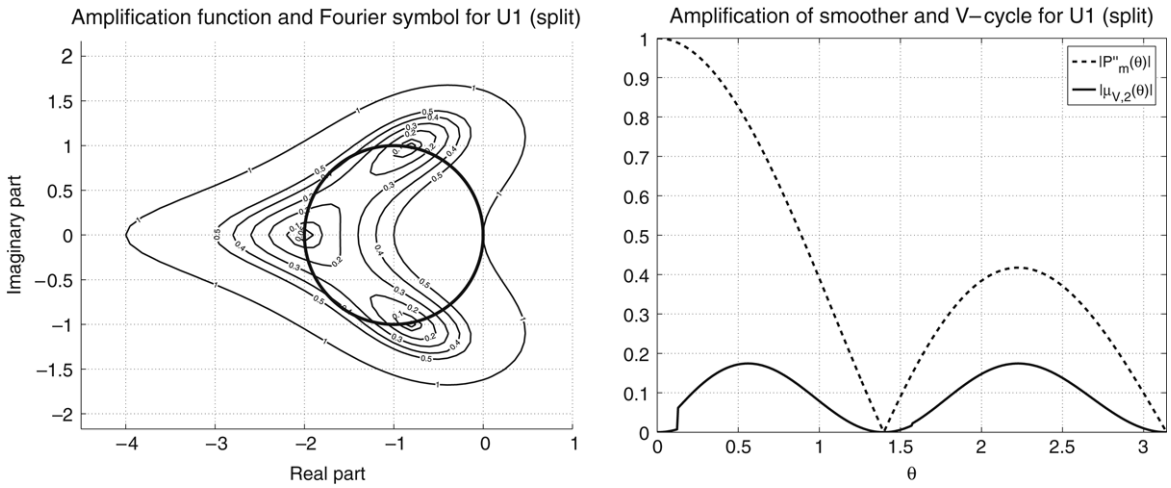

Fig. 6. Figure left: stability region and Fourier symbol. Figure right: amplification curves $|P''_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line) for the optimal split U1 scheme ($m = 2$; $\gamma_1 = 0.5000$, $\gamma_{B,2} = 0.0000$, $\gamma_{C,2} = 0.6043$).

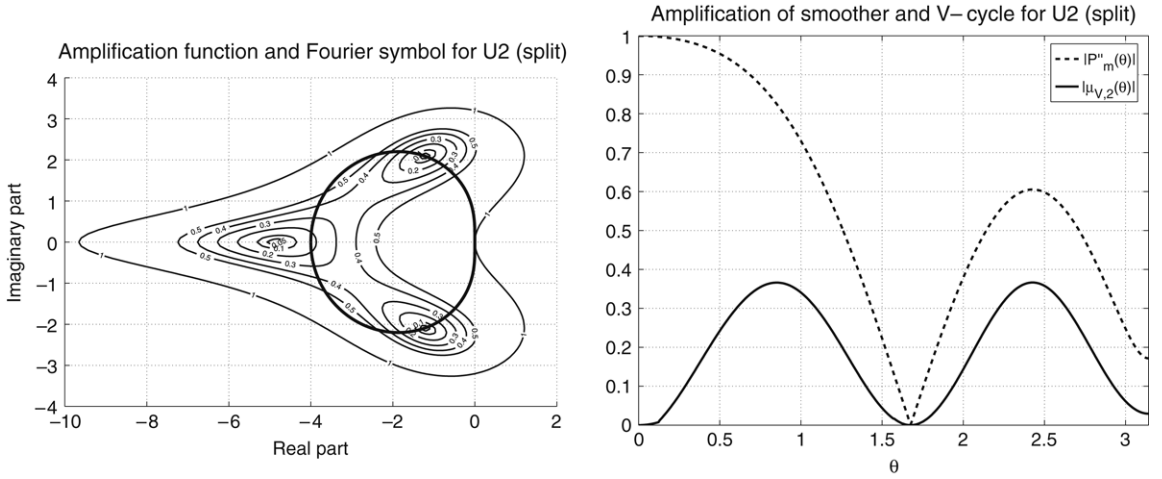

Fig. 7. Figure left: stability region and Fourier symbol. Figure right: amplification curves $|P''_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line) for the optimal split U2 scheme ($m = 2$; $\gamma_1 = 0.2071$, $\gamma_{B,2} = 0.0000$, $\gamma_{C,2} = 0.1703$).

multiplications) from 3 to 2 it was found that a higher efficiency was attained. The schemes were optimized for $\mathcal{I}_3$ and compared with $\mathcal{I}_E$.

Again, agreement between the theoretical model and the experimental results was very good. From the data it is clear that this split approach with $m = 2$ and $\gamma_{B,2} = 0$ can lead to improvements in the order of 10% for U1 and U2 and 20% for K3 with respect to our optimal results in Section 5. We illustrate the results for the different optimal split schemes by drawing the stability region $S = \{z \in \mathbb{C};\ |P''_m(z)| \leq 1\}$ on which we superpose $(-\lambda(\theta))$ and also give $|P''_m(\lambda(\theta))|$ and $|\mu_{V,2}(\lambda(\theta))|$ (see Figs. 6–8).

7. Advection–diffusion equation

Often a certain amount of artificial diffusion is added to the advection Eq. (20) to stabilize it when the advection term is discretized with a central discretization. If we choose a third order dissipative term this becomes

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + \mu \Delta x^3 \frac{\partial^4 u}{\partial x^4} = 0 \tag{56}$$

where $a, \mu \in \mathbb{R}_o^+$. Again we are only interested in the steady-state solution and take $a = 1$ for the sake of simplicity. Discretization of the space operators gives

$$\frac{1}{2\Delta x}\left(u_{j+1} - u_{j-1}\right) + \frac{\mu}{\Delta x}\left(u_{j+2} - 4u_{j+1} + 6u_j - 4u_{j-1} + u_{j-2}\right). \tag{57}$$

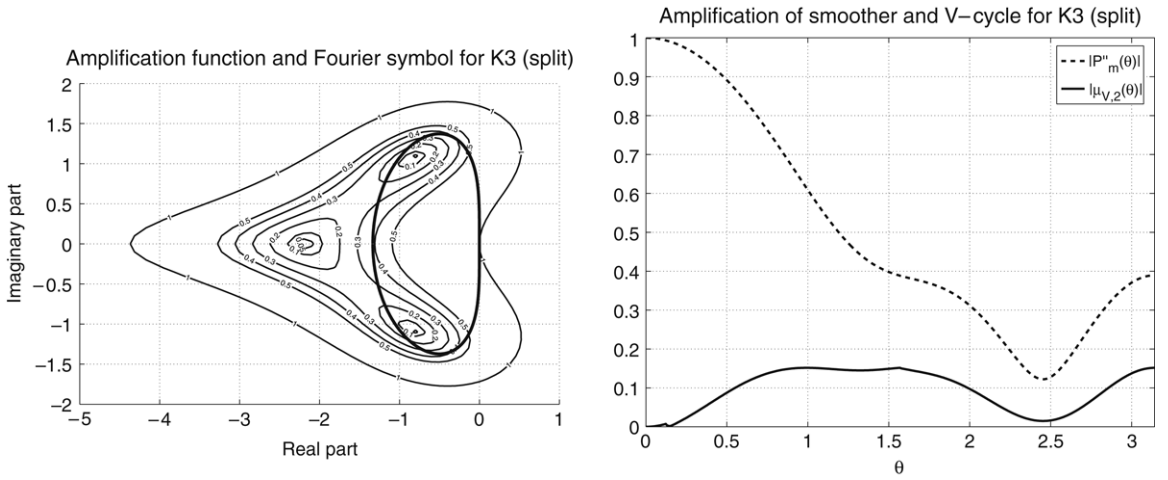

Fig. 8. Figure left: stability region and Fourier symbol. Figure right: amplification curves $|P_m''(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line) for the optimal split K3 scheme ($m = 2$; $\gamma_1 = 0.4580$, $\gamma_{B,2} = 0.0000$, $\gamma_{C,2} = 0.5471$).

This is quite similar to the discretizations of the previous section with a natural splitting between the Hermitian (diffusive) and anti-Hermitian (convective) part.

The resulting Fourier symbol is

$$\lambda(\theta) = \underbrace{\frac{2\mu}{\Delta x}\left(3 - 4\cos\theta + \cos(2\theta)\right)}_{\lambda_B(\theta)} + \underbrace{\frac{\mathbf{i}}{\Delta x}\left(\sin\theta\right)}_{\lambda_C(\theta)}. \tag{58}$$

The iterative solution of this equation has been studied in [6,7,9], which led to the so called modified Runge–Kutta scheme and the Martinelli–Jameson coefficients, which were obtained by trial and error. Later Hosseini [5] analytically confirmed the results of Martinelli and Jameson. The Modified Runge–Kutta scheme treats the convective and diffusive parts of the residual differently. We use the formulation given in [8], adapting it to the conventions and notations used in this paper.

$$\begin{aligned}
&\mathbf{U}_{(o)} = \mathbf{u}_n \\
&\mathbf{U}_{(1)} = \mathbf{U}_{(o)} - \alpha_1 h\left(\mathbf{r}_C^{(o)} + \mathbf{r}_B^{(o)}\right) \\
&\vdots \\
&\mathbf{U}_{(l)} = \mathbf{U}_{(o)} - \alpha_l h\left(\mathbf{r}_C^{(l-1)} + \sum_{k=0}^{l-1} \Gamma_{l,k}\mathbf{r}_B^{(k)}\right) \\
&\mathbf{u}_{n+1} = \mathbf{U}_{(m)}.
\end{aligned} \tag{59}$$

Here $\mathbf{r}_C^{(l)}$ stands for the convective, and $\mathbf{r}_B^{(l)}$ for the dissipative part of the residual at stage l and

$$\Gamma_{l,k} = \begin{cases} \beta_l & \text{if } k = l-1 \\ (1-\beta_l)\Gamma_{l-1,k} & \text{if } k \neq l-1. \end{cases} \tag{60}$$

The Martinelli–Jameson coefficients are given by:

$$\begin{aligned}
\alpha_1 &= \frac{1}{4} \qquad \beta_1 = 1 \\
\alpha_2 &= \frac{1}{6} \qquad \beta_2 = 0 \\
\alpha_3 &= \frac{3}{8} \qquad \beta_3 = \frac{14}{25} \\
\alpha_4 &= \frac{1}{2} \qquad \beta_4 = 0 \\
\alpha_5 &= 1 \qquad \beta_5 = \frac{11}{25}
\end{aligned}$$

and $h = 3.93$, $\mu = 1/32$. The resulting stability region, Fourier symbol and amplification curves are given in Fig. 9.

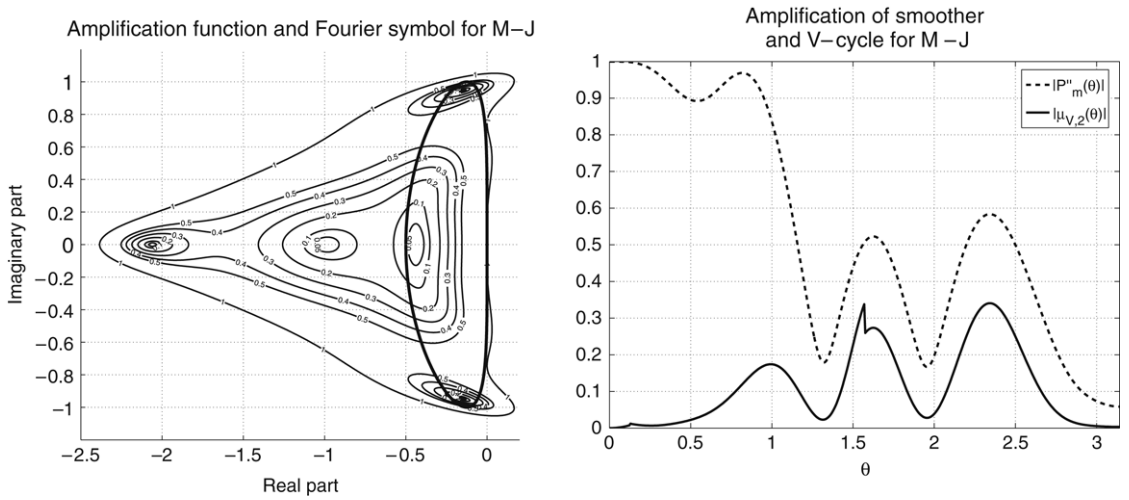

Fig. 9. Advection–diffusion equation with 5-stage Martinelli–Jameson coefficients and $h = 3.8058$. Figure left: stability region and Fourier symbol. Figure right: amplification curves $|P''_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line).

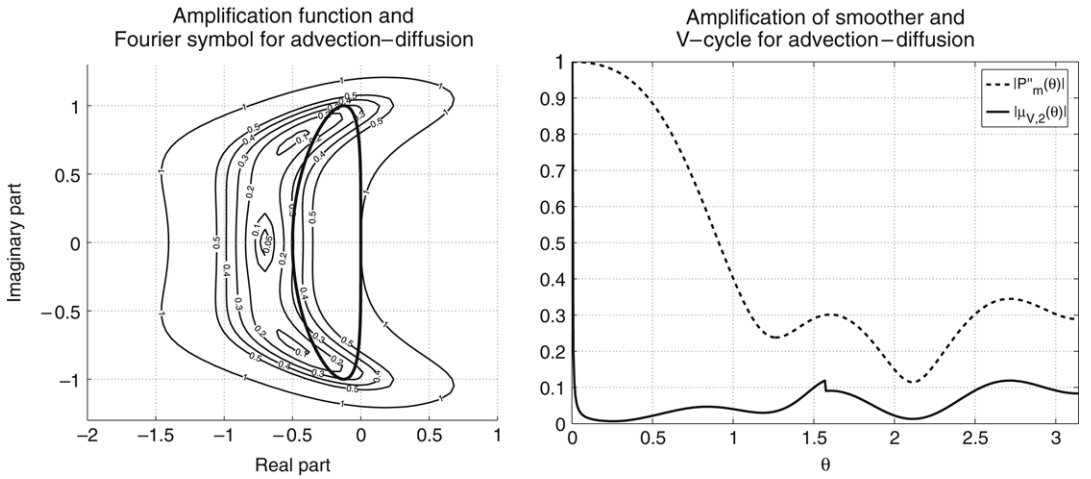

Fig. 10. Advection–diffusion equation with split scheme ($m = 3$; $\gamma_1 = 1.4226$, $\gamma_{B,2} = 0$, $\gamma_{C,2} = 1.2826$, $\gamma_{B,3} = 0$, $\gamma_{C,3} = 1.3000$). Figure left: Stability region and Fourier symbol. Figure right: Amplification curves $|P''_m(\lambda(\theta))|$ (dotted line) and $|\mu_{V,2}(\lambda(\theta))|$ (solid line).

Testing our objective functions on this scheme, keeping the α and β coefficients and μ but optimizing h for ℓ_1 and ℓ_3 we found

- $\ell_1 = 0.5831$ for $h = 3.8231$
- $\ell_3 = 0.3407$ for $h = 3.8058$

which is not that different from the value found in [5]. We illustrate these results by drawing the stability region $S = \{z \in \mathbb{C};\ |P''_m(z)| \leq 1\}$ on which we superpose $(-\lambda(\theta))$ and also give $|P''_m(\lambda(\theta))|$ and $|\mu_{V,2}(\lambda(\theta))|$. We only do this for $h = 3.8058$ as both results are qualitatively similar.

This scheme uses 5 stages, with 5 evaluations of the advective and 3 evaluations of the diffusive residual, which is quite costly. We like to point out that the spike in the amplification curve around $\theta = \frac{\pi}{2}$ is due to the way we model the low and high frequencies separately. It seems reasonable that a better model would smoothen this curve and that therefore the optimal value of ℓ_3 will be slightly lower.

We now try to find a split scheme, according to our own formulation in Section 6 for this equation, minimizing ℓ_3 and using a low number of stages. It was found that a scheme with $m = 3$ gave the best result, keeping $\mu = 1/32$. Setting $\gamma_{B,2}$ and $\gamma_{B,3}$ to zero again did not degrade performance too much and offered a computational saving of 40%. With this in mind the optimal scheme was given by $(\gamma_1, \gamma_{B,2}, \gamma_{C,2}, \gamma_{B,3}, \gamma_{C,3}) = (1.4226, 0, 1.2826, 0, 1.3000)$, which gave an optimal value $\ell_3 = 0.1190$ ($\sqrt[3]{\ell_3} = 0.4919$), which only differed by less than 1% from values measured on the real solver.

We note that these values are substantially lower than the optimal values found for the Martinelli–Jameson scheme and effectively use only 3 coefficients. We illustrate this schemes in Fig. 10.

8. Conclusions

Previous studies [1,4,5,11], that optimized the Runge–Kutta parameters for the high frequency domain, showed that a better smoother can be obtained when using a high number of stages. While this is true, we discovered that for the complete 2-grid cycle the gain from a higher number of stages does not outweigh the extra computational cost, due to the limitations of the defect correction. It actually appears that using a low number of stages gives the best results. We optimized the coefficients for a complete V-cycle and obtained improvements up to 20% for higher order discretizations (measured per stage). When we choose to treat the Hermitian part of the discretization separately from the anti-Hermitian part, further gains could be obtained; these were of the order of 30%–50% using a low number of stages (measured per stage).

We compared the latter with the Martinelli–Jameson scheme for the advection–diffusion equation and found an asymptotic convergence rate that was 66% better (over a whole V-cycle) but using 3, instead of 8, matrix-vector multiplications in the smoother.

References

[1] L.A. Catalano, 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.
[2] L.V. Fauset, Applied Numerical Analysis Using MATLAB, Prentice Hall, Upper Saddle River, NJ, 1999.
[3] R. Haelterman, An analytical quantification of the optimization of multi-stage solvers, in: ICCAM 2006, Twelfth International Congress on Computational and Applied Mathematics, Leuven, Belgium, 10–14 July, 2006.
[4] K. Hosseini, 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.
[5] K. Hosseini, Practical implementation of robust preconditioners for optimized multistage flow solvers, Ph.D. Thesis, Stanford University, 2005.
[6] A. Jameson, Numerical solution of the Euler equations for compressible inviscid fluids, in: F. Angrand, A. Dervieux, J.A. Dé sidé ri, R. Glowinski (Eds.), Numerical Methods for the Euler Equations of Fluid Dynamics, SIAM, Philadelphia, 1985, pp. 199–245.
[7] A. Jameson, Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiter and their effect on multigrid convergence, Int. J. Comp. Fluid Dyn. 4 (1995) 171–218.
[8] J.V. Lassaline, Optimal multistage relaxation coefficients for multigrid flow solvers, http://www.ryerson.ca/~jvl/papers/cfd2005.pdf.
[9] L. Martinelli, Calculations of viscous flows with a multigrid method, Ph.D. Thesis, Princeton University, 1987.
[10] R.S. Varga, Matrix Iterative Analysis, in: Springer Series in Computational Mathematics, vol. 27, Springer-Verlag, Berlin, 2000.
[11] B. Van Leer, C.H. Tai, K.G. Powell, Design of optimally smoothing multi-stage schemes for the Euler equations, AIAA Paper 89–1923, 1989.
[12] D.M. Young, Iterative Solution of Large Linear Systems, Dover Publications, New York, 1971.