Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 616–636
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Dispersion and dissipation properties of the 1D spectral volume method and application to a *p*-multigrid algorithm

Kris Van den Abeele [*,1], Tim Broeckhoven [1], Chris Lacor

Vrije Universiteit Brussel, Department of Mechanical Engineering, Fluid Dynamics and Thermodynamics Research Group, Pleinlaan 2, 1050 Brussel, Belgium

Received 14 June 2006; received in revised form 22 September 2006; accepted 12 October 2006
Available online 28 November 2006

Abstract

In this article, the wave propagation properties of the 1D spectral volume method are studied through analysis of the Fourier footprint of the schemes. A *p*-multigrid algorithm for the spectral volume method is implemented. Restriction and prolongation operators are discussed and the efficiency of the smoothing operators is analyzed. The results are verified for simple 1D advection problems and for a quasi-1D Euler flow. It is shown that a significant decrease in computational effort is possible with the *p*-multigrid algorithm.

Keywords: Spectral volume method; Wave propagation properties; *p*-Multigrid

1. Introduction

The spectral volume method was developed in a series of papers by Wang et al. [1–4], for the solution of general two-dimensional hyperbolic systems like the Euler equations and on unstructured triangular meshes. Recently, Liu et al. [5] extended the method to general three-dimensional hyperbolic systems on unstructured tetrahedral meshes and Sun et al. [6] implemented the method for the Navier–Stokes equations. The spectral volume method can be interpreted as a finite volume method, where within each spectral volume (SV) cell a miniature structured mesh of control volumes (CVs) is formed. In this way a unique stencil for the flux is defined for each face, eliminating the need for searching operations, which are needed in traditional high-order (>2) finite volume methods on unstructured grids. Alternatively, the method can be interpreted as a Petrov–Galerkin method. The weighting functions then correspond to Heaviside-like functions, which are equal to one within one CV and zero everywhere else in the SV. The trial functions are chosen such that they have an average value equal to one in one CV and average values of zero in the other CVs. The method enjoys many of the advantageous properties of the discontinuous Galerkin method. It is capable of achieving arbitrarily high orders of accuracy on unstructured grids. It also has a compact stencil, since a SV only communicates

* Corresponding author. Tel.: +32 26292399; fax: +32 26292880.
E-mail addresses: kvdabeel@vub.ac.be (K. Van den Abeele), tbroeckh@vub.ac.be (T. Broeckhoven), chris.lacor@vub.ac.be (C. Lacor).
[1] FWO research engineer.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.10.022

with its immediate neighbours, through the use of Riemann fluxes, which makes the method suited for parallel computations. The representation of the solution within a SV, by averages over the CVs, allows for a limiting procedure on a control volume level. This gives the SV method a higher resolution for shocks than the discontinuous Galerkin method.

In previous work [2,3,5], the Lebesgue constant, which is basically a measure for the quality of the polynomial interpolation, was used to develop SV partitions. In this paper, the dispersive and dissipative errors of the schemes, which are very important for applications such as Computational Aeroacoustics (CAA) and Large Eddy Simulation (LES), will be used to analyze the performance of the SV partitions. Such analysis is widely used for finite difference and finite volume methods, and Hu et al. [9] performed it for the discontinuous Galerkin method.

With high-order methods, it is possible to achieve low error levels more efficiently than with traditional first-order and second-order accurate schemes. However, efficient solution methods are necessary to fully fulfill this potential. Multigrid algorithms are by far the best candidate for this purpose. Traditional *h*-multigrid is nowadays a standard tool for CFD-applications. In the last few years, *p*-multigrid algorithms for discontinuous Galerkin methods have been investigated and successfully implemented by Helenbrook et al. [12], Bassi and Rebay [13] and Fidkowski et al. [14,15]. Important speed-ups in convergence were observed. In this paper, a *p*-multigrid implementation for the SV method and adapted Runge–Kutta solvers for optimal smoothing properties are presented.

The paper is organized as follows. Section 2 gives a brief description of the SV method. In Section 3, the dispersive and dissipative properties of the SV method are analyzed through the Fourier footprint. The *p*-multigrid algorithm is then described in Section 4. In Section 5, the Fourier footprints are used to investigate the efficiency of the smoothing operators for the multigrid algorithm. Numerical results are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. The spectral volume method

The spectral volume method can be applied to hyperbolic conservation laws (1)

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

The computational domain V is divided in N_{SV} spectral volumes (SV) V_i with volume $|V_i|$. Each of these SVs V_i is further subdivided into control volumes (CV) $V_{i,j}$. Integrating (1) over such a CV and applying the Gauss theorem gives

$$\frac{\partial \overline{U}_{i,j}}{\partial t}|V_{i,j}| = -\int_{\partial V_{i,j}} \vec{F} \cdot \mathrm{d}\vec{s} + \int_{V_{i,j}} S \,\mathrm{d}V = R_{i,j} \tag{2}$$

where $|V_{i,j}|$ is the volume of $V_{i,j}$, $R_{i,j}$ is the residual corresponding to $V_{i,j}$ and $\overline{U}_{i,j}$ is the CV average defined by

$$\overline{U}_{i,j} \equiv \frac{1}{|V_{i,j}|} \int_{V_{i,j}} U \,\mathrm{d}V \tag{3}$$

On a spectral element V_i, a polynomial approximation of the solution is defined

$$U_{V_i} \approx u_{V_i} \equiv \sum_{j=1}^{N_i(p,d)} \overline{U}_{i,j} L_{i,j} \tag{4}$$

$N_i(p,d)$ is the number of CVs in the SV V_i, depending on the desired degree of the polynomial approximation p and the number of spatial dimensions d. The polynomials $L_{i,j}$ associated to the CVs $V_{i,j}$ are defined by

$$\frac{1}{|V_{i,j}|} \int_{V_{i,j}} L_{i,m} \,\mathrm{d}V = \delta_{jm} \tag{5}$$

where δ_{jm} is the Kronecker delta function. Eq. (5) ensures the following property of the polynomial approximation

$$\frac{1}{|V_{i,j}|}\int_{V_{i,j}} u_{V_i}\,\mathrm{d}V = \overline{U}_{i,j} \tag{6}$$

With the polynomial approximation u_{V_i}, the flux integral and the source term integral in (2) can be approximated to order $p+1$, using Gauss quadrature. On the boundary between two SVs however, there are two available values for the flux $\vec{F}$, one from within each SV. Thus on these boundaries a suitable Riemann flux $\vec{F}^R$, for instance a Lax-Friedrichs flux, must be used. A more elaborate overview of the SV method can be found in [1–6].

3. Wave propagation properties

In this section, the dispersive and dissipative properties of the spatial discretization of the SV method will be analyzed. The approach followed is completely analogous to the one in Hu et al. [9]. It will be applied here only for a 1D SV method, but the methodology can easily be extended to a 2D or a 3D SV method. A related, Fourier type analysis to investigate accuracy and the stability has been carried out by Zhang et al. [8]. Consider the 1D scalar advection equation on a domain with periodic boundary conditions and with a harmonic wave as initial solution.

$$\frac{\partial v}{\partial t} + a\frac{\partial v}{\partial x} = 0 \tag{7}$$

$$v(x,0) = \mathrm{e}^{\mathrm{I}kx} \tag{8}$$

To (7), a $(p+1)$th order accurate SV method is applied, on a uniform mesh with cell size Δx. On each SV, a local coordinate ξ is defined, with $\xi \in [-1,1]$. The polynomial approximation $v_i = \sum_{j=1}^{p+1} \bar{v}_{i,j} L_j$ on the SV with index i can then be written as a function of ξ. The boundaries of the CVs are also defined by the corresponding value of ξ: $\xi_0 = -1, \ldots, \xi_q, \ldots, \xi_{p+1} = 1$. On the boundary between two SVs, a Riemann flux of the following form is used:

$$F^R(v_i(1), v_{i+1}(-1)) = a\left(\frac{1+\alpha}{2}v_i(1) + \frac{1-\alpha}{2}v_{i+1}(-1)\right) \tag{9}$$

In (9), α is an upwinding parameter. $\alpha = 0$ corresponds to a central flux, $\alpha = 1$ to a simple upwind flux. Eq. (7) then becomes:

$$\Delta x \sum_{j=1}^{p+1} Q_{mj}\frac{\mathrm{d}\bar{v}_{i,j}}{\mathrm{d}t} + a\sum_{j=1}^{p+1} N^{-1}_{mj}\bar{v}_{i-1,j} + a\sum_{j=1}^{p+1} N^{0}_{mj}\bar{v}_{i,j} + a\sum_{j=1}^{p+1} N^{1}_{mj}\bar{v}_{i+1,j} = 0, \quad m = 1,\ldots,p+1 \tag{10}$$

The matrix elements Q_{mj} is defined by

$$Q_{mj} = \frac{\Delta x_j}{\Delta x}\delta_{mj}, \quad m = 1,\ldots,p+1 \text{ and } j = 1,\ldots,p+1 \tag{11}$$

where Δx_j is the size of the CV with index j within a SV. The matrix elements N^{-1}_{mj}, N^{0}_{mj} and N^{+1}_{mj} are given by the following expressions:

$$N^{-1}_{mj} = \begin{cases} -\frac{1+\alpha}{2}L_j(1) & m = 1 \\ 0 & m = 2,\ldots,p+1 \end{cases} \tag{12}$$

$$N^{0}_{mj} = \begin{cases} -\frac{1-\alpha}{2}L_j(-1) + L_j(\xi_1) & m = 1 \\ -L_j(\xi_{m-1}) + L_j(\xi_m) & m = 2,\ldots,p \\ -L_j(\xi_{p-1}) + \frac{1+\alpha}{2}L_j(1) & m = p+1 \end{cases} \tag{13}$$

$$N^{+1}_{mj} = \begin{cases} 0 & m = 1,\ldots,p \\ \frac{1-\alpha}{2}L_j(-1) & m = p+1 \end{cases} \tag{14}$$

Substitution of the following expression of a harmonic wave

$$\bar{v}_{i,j}(t) = \tilde{v}_j \mathrm{e}^{\mathrm{I}(ki\Delta x - \omega t)} \tag{15}$$

into (10) yields

$$\sum_{j=1}^{p+1}(-\mathrm{I}\widetilde{\Omega}Q_{mj} + \mathrm{e}^{-\mathrm{I}K}N_{mj}^{-1} + N_{mj}^{0} + \mathrm{e}^{\mathrm{I}K}N_{mj}^{+1})\tilde{v}_j = 0, \quad m = 1,\ldots,p+1 \tag{16}$$

with $K = k\Delta x$ the non-dimensional wave number and $\widetilde{\Omega} = \omega\Delta x/a$ the non-dimensional numerical frequency. (16) is a system of $p+1$ equations in the coefficients $\tilde{v}_j$, which has a solution different from zero only if the determinant of the matrix in the left hand side is equal to zero. This yields the numerical dispersion relation

$$\det(-\mathrm{I}\widetilde{\Omega}\mathbf{Q} + \mathrm{e}^{-\mathrm{I}K}\mathbf{N}^{-1} + \mathbf{N}^{0} + \mathrm{e}^{\mathrm{I}K}\mathbf{N}^{+1}) = 0 \tag{17}$$

from which $\widetilde{\Omega}$ can be found. $\widetilde{\Omega}$ should be compared to the exact dispersion relation $\Omega = K$. Expression (17) has $p+1$ solutions, corresponding to the $p+1$ eigenmodes of the numerical system. The quantity $-\mathrm{I}\widetilde{\Omega}$ is the so-called Fourier footprint $\mathscr{R} = \mathscr{R}^{\mathrm{Re}} + \mathrm{I}\mathscr{R}^{\mathrm{Im}}$ of the spatial discretization. The imaginary part $\mathscr{R}^{\mathrm{Im}}$ is a measure of the dispersive properties of the scheme, whereas the real part $\mathscr{R}^{\mathrm{Re}}$ reflects the diffusive behaviour. In order for the scheme to be stable, $\mathscr{R}^{\mathrm{Re}}$ should be non-positive for all K.

Notice that with a SV scheme using a polynomial approximation of degree p, waves with non-dimensional wave numbers K ranging from $-(p+1)\pi$ to $(p+1)\pi$ are captured, since there are $p+1$ degrees of freedom per cell. For classical finite volume (FV) methods, this range is $-\pi \leqslant K \leqslant \pi$, which corresponds to the one degree of freedom per cell used by such methods. To make a fair comparison between the FV schemes and the SV schemes, the plots for the SV method should be downscaled with a factor $p+1$, to take into account the higher number of degrees of freedom used by this method. It should also be noted that equation (17) is periodic in K, with a period of 2π. As was pointed out in Helenbrook and Atkins [10], the $p+1$ solutions of (17) thus correspond to the wave numbers $K + z2\pi$, with $z = -(p+1), -p, \ldots, p-1, p$ and K ranging from 0 to 2π. Consequently, to get a clear picture of the wave propagation properties, each of the $p+1$ solutions of (17) is reassigned to the corresponding wave number when plotting the dispersive and diffusive properties of the scheme.

Figs. 1 and 2 show $\mathscr{R}^{\mathrm{Im}}$ and $\mathscr{R}^{\mathrm{Re}}$ for the second-order SV scheme ($p = 1$), when an upwind Riemann flux is used ($\alpha = 1$ in (9)). For this scheme, K ranges from -2π to 2π. Naturally, the plots are symmetrical with respect to $K = 0$ ($\mathscr{R}^{\mathrm{Re}}(K) = \mathscr{R}^{\mathrm{Re}}(-K)$ and $\mathscr{R}^{\mathrm{Im}}(K) = -\mathscr{R}^{\mathrm{Im}}(-K)$). It can be seen that the numerical scheme is only accurate for small dimensionless wavenumbers ($|K| < 1$). For $|K| > 1$, both the dispersion and the diffusion error start to increase significantly. Furthermore, since $\mathscr{R}^{\mathrm{Re}}$ is non-positive for all K, one can conclude that the spatial scheme is stable.

In Figs. 3 and 4, the Fourier footprints of some upwind fourth-order SV schemes are plotted, along with the Fourier footprint of the fourth-order discontinuous Galerkin (DG) scheme. Because of the symmetry, the

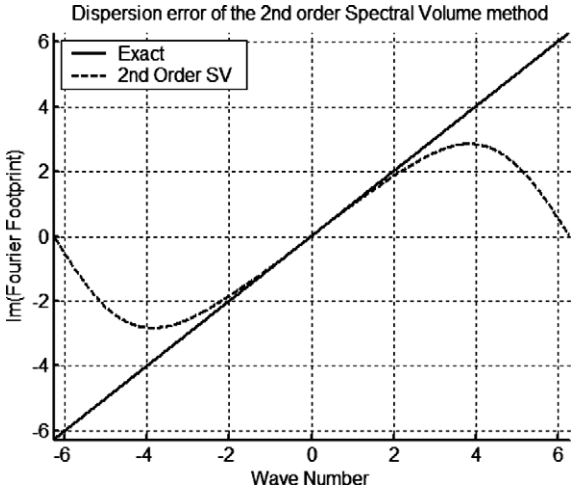

Fig. 1. Dispersion error of the second-order SV method, with upwind Riemann flux ($\alpha = 1$).

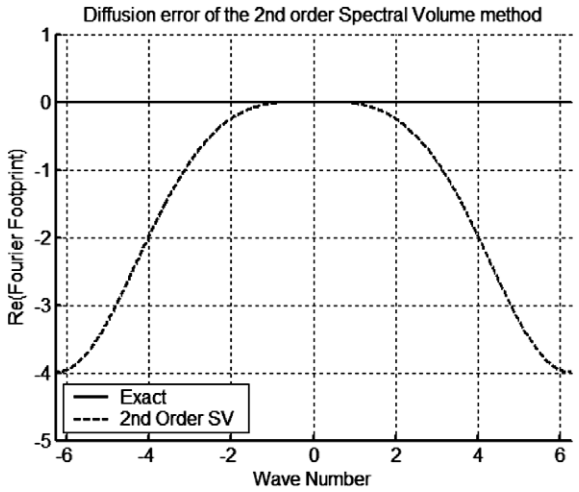

Fig. 2. Diffusion error of the second-order SV method, with upwind Riemann flux ($\alpha = 1$).

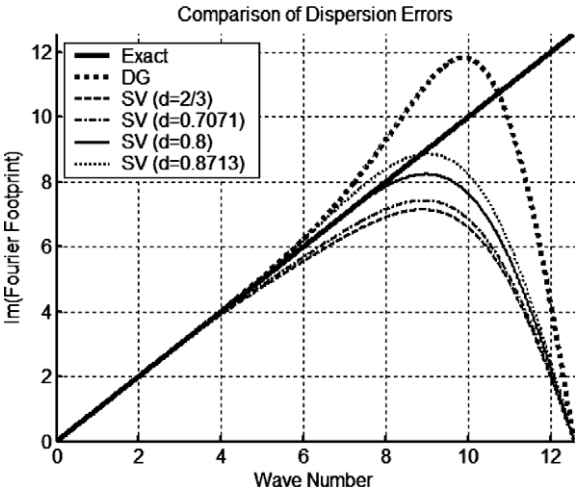

Fig. 3. Comparison of dispersion errors of some fourth-order accurate schemes, with upwind Riemann flux ($\alpha = 1$).

curves are only shown for positive values of the wave number K. For the SV schemes, the positions of the CV boundaries within a SV are given by the following values of the local coordinate $\xi : \{-1, -d, 0, d, 1\}$. Four different values for d are considered. The first is $d = 2/3$, corresponding to the fourth-order SV scheme used in Wang and Liu [2] which they found to be non-convergent for a 2D linear convection problem. They considered a 2D case however, using triangles as SVs. The second value is $d = \cos(\pi/4) \approx 0.7071$, which corresponds to a Gauss–Lobatto distribution of the CV boundaries. For the third scheme, d equals 0.8. This scheme is included because of its nice dispersive and dissipative properties. Finally, $d = \tanh(\mu/2)/\tanh(\mu)$ with $\mu = 2.6$ is considered, resulting in $d \approx 0.8713$. The second and the fourth scheme were discussed in Wang [1]. Table 1 lists the following squarely integrated errors over the wave number range $0 \leqslant K \leqslant \kappa$, for $\kappa = 7.84$.

$$E^{\text{disp}} = \int_0^{\kappa} (\mathscr{R}^{\text{Im}} - K)^2 \, \mathrm{d}K \tag{18}$$

$$E^{\text{diff}} = \int_0^{\kappa} (\mathscr{R}^{\text{Re}})^2 \, \mathrm{d}K \tag{19}$$

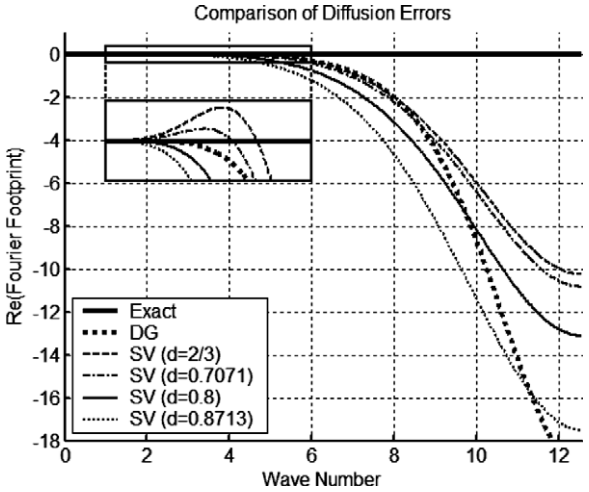

Fig. 4. Comparison of diffusion errors of some fourth-order accurate schemes, with upwind Riemann flux ($\alpha = 1$).

Table 1
Integrated errors for the fourth-order SV and DG schemes with upwind Riemann fluxes, for $\kappa = 7.84$

Scheme	E^{disp}	E^{diff}
DG	0.887	1.698
SV ($d = 2/3$)	1.047	1.973
SV ($d = 0.7071$)	0.566	3.827
SV ($d = 0.8000$)	0.002	6.160
SV ($d = 0.8713$)	0.290	14.135

From Table 1, it can immediately be concluded that the diffusive errors dominate for the SV method with upwind Riemann fluxes, which is also the case for the DG method.

In Fig. 4, it can be seen that for the SV schemes corresponding to $d = 2/3$ and $d = 0.7071$, $\mathscr{R}^{\text{Re}}$ is positive over a certain range of K. Consequently, the amplitude of the waves in this range will grow exponentially in time, meaning that the schemes are unstable, despite the fact that they have smaller squarely integrated diffusion errors than the other two SV schemes. The fact that the scheme with $d = 2/3$ is unstable provides at least an indication to why the equivalent 2D fourth-order scheme in Wang and Liu [2] showed a non-convergent behaviour. With the scheme based on the Gauss–Lobatto distribution however, good results were achieved in Wang [1] and in Wang and Liu [3], which seems to contradict the present result. However, the instability of the scheme is very weak, so it takes a long time to develop. Furthermore, stability analysis shows that the scheme is stable for a small range of CFL-numbers ($0.1625 < \text{CFL} < 0.2275$), when combined with the third-order TVD Runge–Kutta scheme of Shu [17] for time marching, which was used in [1]. This will be shown in Section 5. The third-order accurate SV scheme based on the Gauss–Lobatto distribution was found to suffer from the same weak instability.

The curves in Fig. 4 corresponding to $d = 0.8$ and $d = 0.8713$ show that both these spatial schemes are stable. The dispersive properties of the first are significantly better than those of the latter, as can be seen in Fig. 3 and in Table 1. The first scheme is also less dissipative than the second. It can thus be concluded that the scheme with $d = 0.8$ has the best wave propagation properties of the investigated schemes. The upwind scheme with $d = 0.8$ will be denoted the SV4D08U-scheme in the remainder of the article.

Upon comparison of the dispersive and diffusive behaviour of the SV schemes with that of the discontinuous Galerkin scheme, it can be concluded that the latter is significantly less dissipative. The dispersive error of the DG scheme is slightly larger than that of the SV scheme. However, since the diffusive errors dominate, a smaller error can be expected with the discontinuous Galerkin scheme for a given cell size, as has been reported in the literature [7,8]. However, one should not immediately draw the conclusion that the DG method

is superior to the SV method. As will be shown in Section 5, the latter allows for larger time steps, because of the smaller Fourier footprint. Also, at least in 1D, the evaluation of the residuals requires less CPU-time.

In Fig. 5, the dispersion errors of some fourth-order accurate SV schemes combined with a central Riemann flux are plotted, along with the dispersion error of the fourth-order discontinuous Galerkin scheme with a central flux. The diffusion error of these schemes is uniformely zero. The behaviour of the SV schemes is plotted for values of d equal to 0.6380, 0.7071, 0.8 and 0.8713. The schemes corresponding to the last three values have already been analyzed for an upwind Riemann flux. The scheme with $d = 0.6380$ is included here because it has nice dispersive properties. The square of the dispersive errors integrated over the range $0 \leqslant K \leqslant 7.84$ are included in Table 2.

Upon analysis of Fig. 5 and Table 2, it is immediately clear that, contrary to the case with an upwind Riemann flux, good SV schemes are found for lower values of d, for instance 0.6380. However, below a certain value of d, the resulting schemes are unstable. This is illustrated in Fig. 6, where the solutions of (17) in the wave number range $0 \leqslant K \leqslant 4\pi$ have been plotted for the scheme with $d = 0.6$. Despite the use of a central Riemann flux, a non-zero and positive diffusion error is found for this scheme.

The presence of the discontinuity in the curves should be noted. This means that the numerical group velocity $\partial\mathscr{R}^{\mathrm{Im}}/\partial K$ is not defined for certain wave numbers K, which limits the usefulness of these schemes. However, the evolution of the dispersion error for varying d suggests that there exists one value of d for which the discontinuity disappears and which still yields a stable scheme. Unfortunately, it is not trivial to determine this value.

If one compares the SV schemes with a central Riemann flux to those with an upwind Riemann flux, it can be concluded that the range over which the Fourier footprint $(\mathscr{R})^{\mathrm{Re}} + \mathrm{I}\mathscr{R}^{\mathrm{Im}}$ follows the ideal value of IK closely does not vary significantly. The central schemes have the advantage of not introducing diffusion errors, as opposed to the upwind schemes. However, their usefulness is limited in practice, as discussed above.

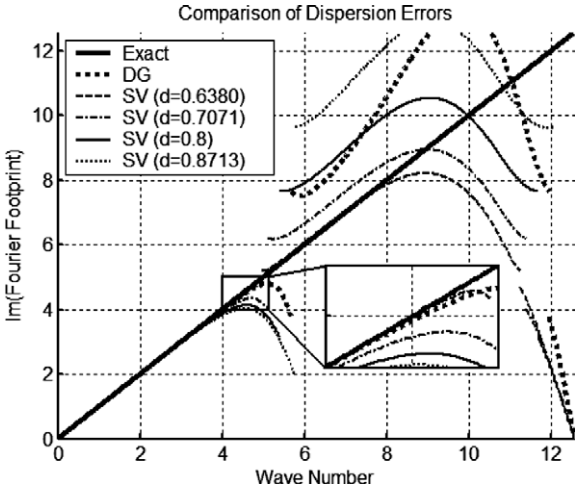

Fig. 5. Comparison of dispersion errors of some fourth-order accurate schemes, with central Riemann flux ($\alpha = 0$).

Table 2
Integrated errors for the fourth-order SV and DG schemes with central Riemann fluxes, for $\kappa = 7.84$

Scheme	E^{disp}
DG	7.504
SV ($d = 0.6380$)	0.427
SV ($d = 0.7071$)	1.542
SV ($d = 0.8000$)	10.894
SV ($d = 0.8713$)	36.069

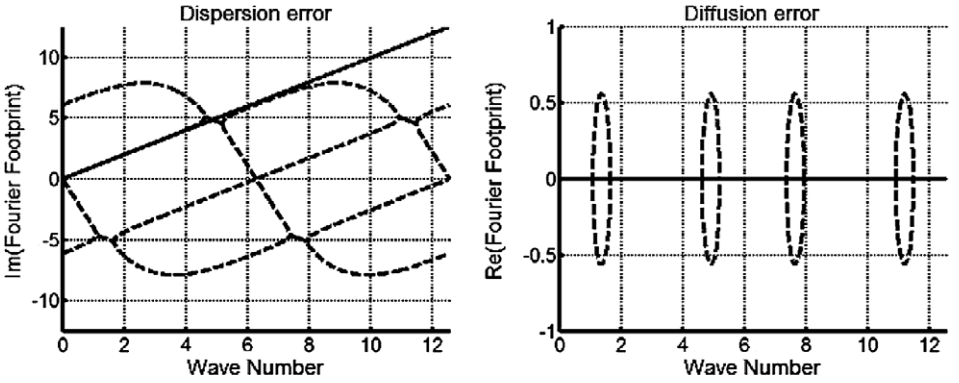

Fig. 6. Dispersion (left) and diffusion errors of the fourth-order accurate SV scheme with $d = 0.6$ and a central Riemann flux ($\alpha = 0$).

When compared with the discontinuous Galerkin scheme, it is seen that the SV scheme with $d = 0.6380$ yields a slightly smaller dispersive error for the low wave number range before the discontinuity. For wave numbers beyond the discontinuity, the SV scheme has a significantly smaller dispersive error.

4. *p*-Multigrid algorithm

The main idea of a multigrid algorithm is based on the observation that error-smoothing operators are generally very efficient in eliminating high-frequency errors, but much less adequate for the low-frequency errors. The multigrid strategy is to switch to a coarser representation of the solution, where the low-frequency errors on the fine representation occur as high-frequency modes, which can thus be efficiently damped out. In the traditional *h*-multigrid approach, this is done by switching to a coarser spatial grid. With a *p*-multigrid algorithm, a high-order solution representation is transferred to a lower-order one. Such *p*-multigrid algorithms have already been studied for high-order discontinuous Galerkin methods, see [12–15]. They have also been applied to spectral element methods, Ronquist and Patera [11]. Interesting properties such as *p*-independent convergence rates and locality of the transfer operators have been reported. The *p*-multigrid algorithm for a SV method described in this section is largely based on the algorithm used by Fidkowski et al. [14,15]. A simple, two-level Full Approximation Scheme algorithm as proposed by Brandt [16] can be summarized in the following way. To solve a fine level problem $\mathbf{R}^f(\overline{\mathbf{U}}^f) = 0$, perform the following operations:

- Perform ν_1 smoothing sweeps on the fine level: $\overline{\mathbf{U}}^f \leftarrow (\mathbf{G}^f)^{\nu_1}\overline{\mathbf{U}}^f$
- Transfer the state and the residual to the coarse level:

$$\overline{\mathbf{U}}_0^c \leftarrow \widetilde{I}_f^c \overline{\mathbf{U}}^f, \quad \mathbf{f}^c \leftarrow \mathbf{R}^c(\overline{\mathbf{U}}^f) - \mathbf{R}^c(\overline{\mathbf{U}}_0^c) = I_f^c \mathbf{R}^f(\overline{\mathbf{U}}^f) - \mathbf{R}^c(\overline{\mathbf{U}}_0^c)$$

- Solve the coarse level problem: $\mathbf{R}^c(\overline{\mathbf{U}}^c) = \mathbf{f}^c$
- Prolongate the coarse level error and correct the fine level state:

$$\overline{\mathbf{U}}^f \leftarrow \overline{\mathbf{U}}^f + I_c^f(\overline{\mathbf{U}}^c - \overline{\mathbf{U}}_0^c)$$

- Perform ν_2 smoothing sweeps on the fine level: $\overline{\mathbf{U}}^f \leftarrow (\mathbf{G}^f)^{\nu_2}\overline{\mathbf{U}}^f$

In this algorithm, $\mathbf{G}^f$ represents an arbitrary smoothing operator on the fine level. $\mathbf{f}^c$ is the so-called forcing function. The coarse level problem could again be solved using a FAS algorithm, and so on. In this way, one arrives at a V-cycle. A further increase in efficiency can be achieved by initializing the solution on coarser levels. In this way, a better initial solution is provided for the fine levels, which will also improve the robustness of the method. This corresponds to a so-called Full Multgrid (FMG) algorithm. In the present work, the decision on when to start solving a finer level problem is made in an analogous way as in Fidkowski et al. [14,15]. The switch to a finer level is made when the L_2 norm of the coarse level residuals is smaller than a factor η_{switch} times the L_2 norm of the fine level residual. Multigrid levels with $p = 0$, $p = 1$ and $p = 3$ are considered,

corresponding to the following relation between p^{fine} and p^{coarse}: $(p^{\text{fine}}+1)/(p^{\text{coarse}}+1)=2$, i.e. doubling the order of accuracy with each higher multigrid level. This choice is made here because it implies that the resolved wavenumber range is doubled with each multigrid level as well, as is the case usually for h-multigrid algorithms. The high wavenumber range, which should be strongly damped, then corresponds to wavenumbers K between $K_{\max}/2$ and $K_{\max}$, with $K_{\max}$ the maximum wavenumber which is resolved on the considered multigrid level.

The prolongation (I_c^f), state restriction $(\widetilde{I}_f^c)$ and residual restriction (I_f^c) operator still have to be defined for the SV method. In this work, these operators are chosen as follows (with omission of the SV index i):

- *Prolongation operator*: this operator can be chosen in the same way as for discontinuous Galerkin methods. On the coarse as well as on the fine level, the solution within a SV is represented by a polynomial $\sum_{j=1}^{N(p,d)}\overline{U}_j L_j$. The coarse level polynomials L_j^c can be written as a function of the fine level polynomials L_m^f:

$$L_j^c=\sum_{m=1}^{N_f}\alpha_{jm}L_m^f,\quad j=1,\ldots,N_c \tag{20}$$

  N_c and N_f are the number of CVs within a SV on the coarse and fine level. By equating the fine level solution to the coarse level solution, the following expression for I_c^f is found: $(I_c^f)_{mj}\equiv\alpha_{jm}$.
- *State restriction operator*: this operator can also be defined completely analogously as for discontinuous Galerkin methods, by projecting the fine level solution onto the coarse level polynomial basis. This results in the following definition for $\widetilde{I}_f^c$: $(\widetilde{I}_f^c)_{mj}=(P^{-1}Q)_{mj}$. The matrices P and Q are defined by

$$P_{jm}=\int_V L_j^c L_m^c\,\mathrm{d}V,\quad j,m=1,\ldots,N_c \tag{21}$$

$$Q_{jm}=\int_V L_j^c L_m^f\,\mathrm{d}V,\quad j=1,\ldots,N_c,\quad m=1,\ldots,N_f \tag{22}$$

- *Residual restriction operator*: for discontinuous Galerkin methods, a very simple formulation for this operator exists, which follows from the property that the residuals are a semilinear form in the basis functions. However, this property is not valid for the SV method, so the residual restriction operator has to be defined in another way. One could evaluate the coarse level residual of the fine level solution by simply using the latter in the points that are used for the calculation of the coarse level residual. However, it is then not immediately clear how to restrict a possible fine level forcing term $\mathbf{f}^f$. Such a term will occur if the considered fine level is not the highest multigrid level. One approach could be to divide the forcing term of a fine level CV between the coarse level CVs which share part of their domain with it. Each coarse level CV would then receive a part proportionate to the fraction of the fine level CV they occupy. In the present work however, only multigrid levels with $p=0$, $p=1$ and $p=3$ are used. The distribution of CVs inside a 1D SV is then such that two fine level CVs, with index $j1$ and $j2$, correspond exactly to one coarse level CV with index j. The restricted residual is then calculated as

$$[I_f^c\mathbf{R}^f(\overline{\mathbf{U}}^f)]_j\equiv R_j^c(\overline{\mathbf{U}}^f)=R_{j1}^f(\overline{\mathbf{U}}^f)+R_{j2}^f(\overline{\mathbf{U}}^f) \tag{23}$$

  The CV distributions for the different p-multigrid levels and the residual restriction operation are illustrated in Fig. 7.

  If another p-multigrid level, with $p=7$, is to be added, then the corresponding SV should be designed in such a way that one of the CV boundaries is $\xi=-d$ and another is $\xi=d$. This is to ensure that the above residual restriction operator can still be used. Notice that for the same reason, a residual restriction operator such as (23) cannot be used for 2D or 3D problems, when simplices are used as SVs. In this case the coarse level CVs don't correspond exactly to a certain number of fine level CVs, except for the restriction from $p=1$ to $p=0$. A possible extension to the operator (23) for this case is to divide the fine level residual proportionally between the coarse level CVs which share part of their domain with the fine level CV. When quadrangles or hexahedrons are used, an operator of the type (23) could still be used. For such SVs, the polynomial basis is not an order-complete basis, but a tensor product basis, which contains more basis functions for a certain order of accuracy. This could be considered a drawback.

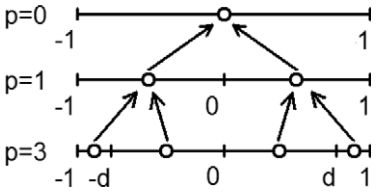

Fig. 7. Distribution of CVs within a 1D SV for a first, second and fourth-order method and illustration of the residual restriction process.

5. Smoother efficiency analysis

The efficiency of a multigrid algorithm largely depends on the quality of the smoothing operators. These should be designed in such a way that the high-frequency modes ($K > K_{\max}/2$) are damped effectively. A variety of smoothing operators are available, varying from explicit Runge–Kutta smoothers to implicit Jacobi, Gauss–Seidel or other smoothers, with or without overrelaxation. In the present work, explicit N_{RK}-stage Runge–Kutta smoothers of the following form are considered:

$$\begin{aligned}
\overline{U}^{0}_{i,j} &= \overline{U}^{n}_{i,j} \\
\overline{U}^{m}_{i,j} &= C^{1}_{m}\overline{U}^{0}_{i,j} + C^{2}_{m}\overline{U}^{m-1}_{i,j} + C^{3}_{m}\frac{\Delta t}{|V_{i,j}|}R^{m-1}_{i,j}, \quad 1 \leqslant m \leqslant N_{\mathrm{RK}} \\
\overline{U}^{n+1}_{i,j} &= \overline{U}^{N_{\mathrm{RK}}}_{i,j}
\end{aligned} \tag{24}$$

To investigate the damping characteristics of such a Runge–Kutta smoother, a Fourier analysis is made, see for example Ramboer et al. [19]. Consider the semi-discretization (10) of the 1D scalar linear advection equation, written in a more compact form

$$\frac{\mathrm{d}\bar{\mathbf{v}}}{\mathrm{d}t} = \frac{1}{\Delta x}\mathbf{Q}^{-1}\mathbf{R}(\bar{\mathbf{v}}) \tag{25}$$

To fix thoughts, consider the simple Euler algorithm, which corresponds to (24) with $N_{\mathrm{RK}} = 1$, $C^{1}_{1} = 1$, $C^{2}_{1} = 0$ and $C^{3}_{1} = 1$, to numerically integrate (25) in time. Then the following expression is obtained:

$$\bar{\mathbf{v}}^{n+1} = \bar{\mathbf{v}}^{n} + \frac{\Delta t}{\Delta x}\mathbf{Q}^{-1}\mathbf{R}(\bar{\mathbf{v}}^{n}) \tag{26}$$

Substituting a Fourier mode solution of the form

$$v(x,t) = F(t)\mathrm{e}^{\mathrm{I}kx}, \quad x = i\Delta x \text{ and } t = n\Delta t \tag{27}$$

gives

$$G(\sigma\mathscr{R}(K)) \equiv \frac{F^{n+1}}{F^{n}} = 1 + \sigma\mathscr{R}(K) \tag{28}$$

where $\mathscr{R}$ is the Fourier footprint of the SV scheme, which was defined in Section 3. The CFL-number σ is given by $a\Delta t/\Delta x$. The complex number $G(\sigma\mathscr{R}(K))$ is the amplification factor of the fully discretized scheme. For a general N_{RK}-stage Runge–Kutta scheme, the amplification factor can be calculated in an analogous way. It will then be a polynomial function of degree N_{RK} in $\sigma\mathscr{R}$. $|G|$ should not be larger than one for stability. In the case of the exact solution in time, $G(\sigma\mathscr{R})$ corresponds to $\mathrm{e}^{\sigma\mathscr{R}}$. Consequently, for a time accurate solution, the amplification factor should be a good approximation of the exponential function, for wavenumbers K ranging from 0 to K_{res}, where K_{res} is the maximum wavenumber which is still accurately resolved by the spatial scheme. However, if one is only interested in the steady state solution, the Runge Kutta scheme is best chosen such that the transient waves are damped out as quickly as possible. Practically, for the multigrid algorithm, this means that $|G(\sigma\mathscr{R})|$ should be small for high wavenumbers, since on each multigrid level only the high-frequency waves should be strongly damped.

Fig. 8 shows the amplitude of the amplification factor for the standard four-stage fourth-order accurate Runge–Kutta scheme as a function of $\sigma\mathscr{R}$. The spatial scheme defines a trajectory for $\mathscr{R}$ in the complex plane,

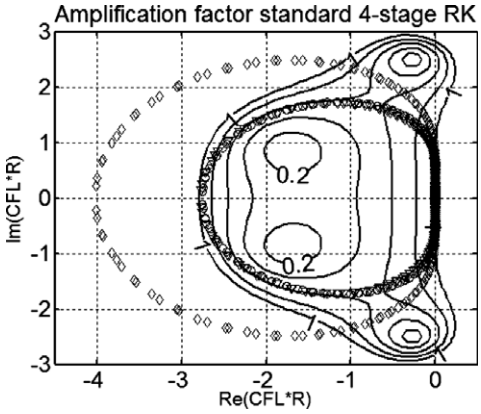

Fig. 8. Amplification factor amplitude $|G|$ of the standard 4-stage fourth-order accurate RK scheme. Fourier footprints $\sigma\mathscr{R}$ of the SV4D08U-scheme with $\sigma = 0.21$ ($\bigcirc$) and the fourth-order upwind DG scheme with $\sigma = 0.21$ ($\diamond$) and with $\sigma = 0.145$ ($\triangledown$).

as a function of the wavenumber. This Fourier footprint, after scaling with σ, should lie entirely inside the domain where $|G| < 1$ for stability. In Fig. 8, the Fourier footprint has been plotted for the SV4D08U-scheme with $\sigma = 0.21$, along with the footprints of the fourth-order upwind discontinuous Galerkin scheme with $\sigma = 0.21$ and $\sigma = 0.145$. It is clearly observed that the SV scheme will be stable for $\sigma = 0.21$. The DG scheme however is unstable for this value and requires a smaller CFL-number. For $\sigma = 0.145$, the Fourier footprint of the latter does lie entirely inside the stability domain. One can thus conclude that because of the generally larger unscaled Fourier footprint, the discontinuous Galerkin method requires smaller CFL-numbers than the SV method, as was mentioned in Section 3. This result is in accordance with the literature [7,8].

We now come back to the stability of the upwind SV scheme with Gauss–Lobatto points. In Section 3 there was an indication that this scheme is unstable because of a positive diffusion error. Fig. 9 shows the Fourier footprint of this scheme for $\sigma = 0.2$ and $\sigma = 0.125$, plotted on the stability domain of the third-order TVD Runge–Kutta scheme [17]. On the right figure it can clearly be seen that this footprint lies just outside the stability domain for $\sigma = 0.125$. Consequently, the scheme is weakly unstable, as was shown in Section 3. For $\sigma = 0.2$ however, the scheme is stable.

The amplitudes of the amplification factors for two five-stage Runge–Kutta schemes (RK5) of the form (24) are shown in Fig. 10. The left figure shows $|G|$ for the standard scheme, which is fifth-order accurate in time for linear problems, see for example Lacor et al., [18]. In the right figure, $|G|$ is shown for a scheme with optimized coefficients, such that better smoothing properties are obtained. For the high-frequency waves to be damped efficiently, the corresponding part of the Fourier footprint should lie in a domain with small amplification factor. The Fourier footprint of the SV4D08U-scheme is shown in both figures, for $\sigma = 0.4$. It is clear that the

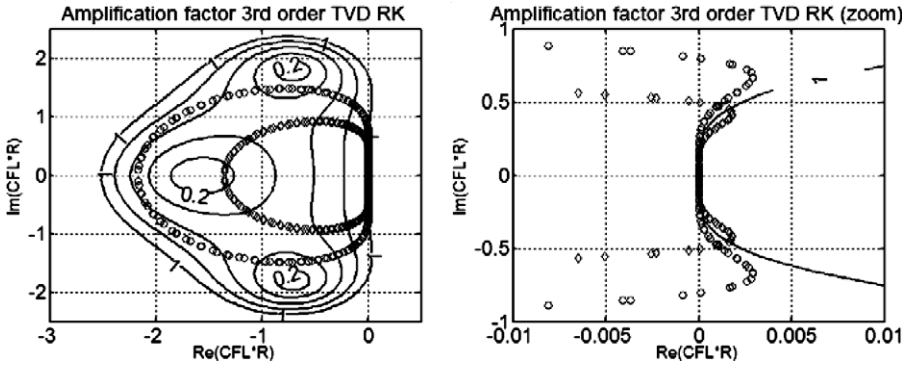

Fig. 9. Amplification factor amplitude $|G|$ of the third-order TVD RK scheme (left) and zoom on the area $[-0.1, 0.1] \times [-1, 1]$ (right). Fourier footprint $\sigma\mathscr{R}$ of the upwind SV scheme based on Gauss–Lobatto points, with $\sigma = 0.2$ ($\bigcirc$) and $\sigma = 0.125$ ($\diamond$).

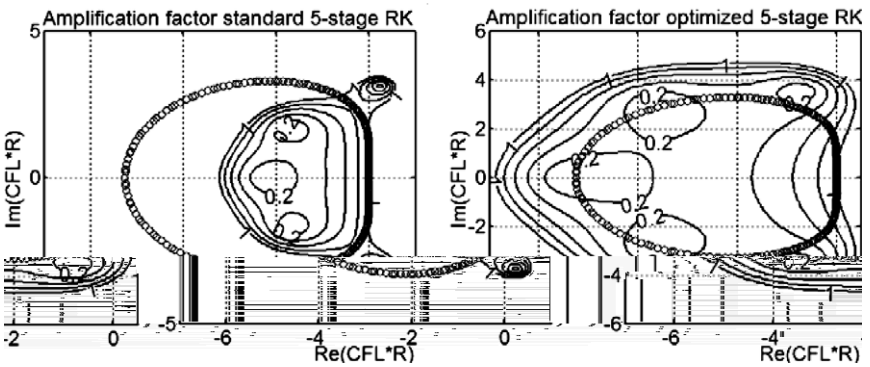

Fig. 10. Amplification factor amplitude $|G|$ of the standard RK5-scheme (left) and the optimized RK5-scheme (right). Fourier footprint $\sigma\mathscr{R}$ of the SV4D08U-scheme with $\sigma = 0.4$ (○).

Table 3
Coefficients of the different Runge–Kutta schemes and CFL-limit in combination with the spatial schemes they are used with. Index m takes the values $1,\ldots,N_{RK}$

Scheme	C_m^1	C_m^2	C_1^3	C_2^3	C_3^3	C_4^3	C_5^3	SV scheme	σ_{max}
std RK2	1	0	$\frac{1}{2}$	1	–	–	–	SV1U	1
opt RK2	1	0	$\frac{1}{4}$	1	–	–	–	SV1U	2
std RK3	1	0	$\frac{1}{3}$	$\frac{1}{2}$	1	–	–	SV2U	0.595
opt RK3	1	0	$\frac{1}{5}$	$\frac{1}{2}$	1	–	–	SV2U	0.820
std RK5	1	0	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	1	SV4D08U	0.245
	$\underline{m=1}$:	$\underline{m=1}$:							
opt RK5	1	0	$\frac{85}{1300}$	$\frac{1}{10}$	$\frac{9}{50}$	$\frac{1}{4}$	$\frac{132}{300}$	SV4D08U	0.510
	$\underline{m\neq 1}$:	$\underline{m\neq 1}$:							
	$1-C_m^3$	C_m^3							

standard Runge–Kutta scheme will be unstable for the chosen value of σ. The optimized scheme is stable however, and a large part of the Fourier footprint lies in a zone with high damping. It is thus concluded that both the stability domain $|G| < 1$ and the zones with high damping can be significantly enlarged by a suitable choice of the Runge–Kutta scheme coefficients. The coefficients of the RK-schemes in Fig. 10 are included in Table 3, along with the stability limit σ_{max} when the SV4D08U-scheme is used. Also, the coefficients of a two-stage (RK2) and a three-stage (RK3) RK scheme are given, with σ_{max} when combined with the spatial scheme mentioned in the table. The SV1U- and SV2U-schemes are the first- and second-order accurate upwind SV schemes respectively. The combinations of spatial schemes and optimized RK-schemes in Table 3 correspond to the p-multigrid levels that will be used in the test cases in Section 6.

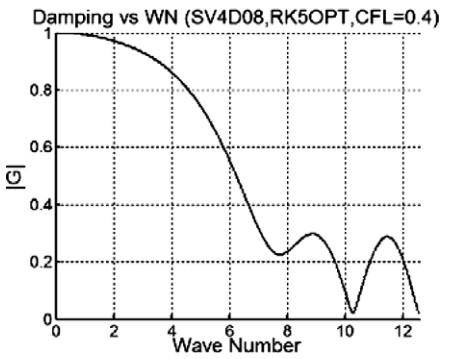

Fig. 11. Amplification factor amplitude $|G|$ versus wavenumber K for the SV4D08U-scheme and the optimized five-stage RK scheme with $\sigma = 0.4$.

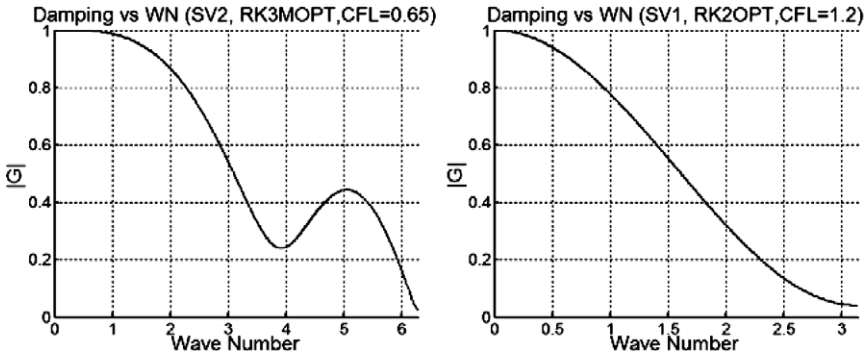

Fig. 12. Amplification factor amplitude $|G|$ versus wavenumber K for the SV2U-scheme and the optimized RK3-scheme with $\sigma = 0.65$ (left) and the SV1U-scheme and the optimized RK2-scheme with $\sigma = 1.2$ (right).

In order to see more clearly how much each wavenumber is damped, Fig. 11 shows the amplification factor amplitude $|G|$ as a function of the wavenumber K for the optimized RK5-scheme and the SV4D08U-scheme, with $\sigma = 0.4$. It can be seen that for $K > K_{\max}/2$, $|G|$ is smaller than 0.5 and that for the biggest part of this high-frequency range, $|G|$ is smaller than 0.3. The high-frequency waves will thus be damped quite efficiently. In Fig. 12, analogous plots are shown for the SV2U-scheme with the optimized RK3-scheme and $\sigma = 0.65$, and for the SV1U-scheme with the optimized RK2-scheme and $\sigma = 1.2$. The same conclusion can be drawn.

6. Numerical tests

6.1. 1D linear advection

To verify the results on the wave propagation characteristics of the SV method, the linear advection of a Gaussian profile is considered. The equation to be solved is (7), with $a = 1$, on the following computational domain: $0 \leqslant x \leqslant 1$. The initial solution is given by

$$v(x,0) = \exp\left(-\left(\frac{x-0.5}{0.1}\right)^2\right) \tag{29}$$

The cell size is $\Delta x = 0.1$, which is quite large, so that the dispersive and dissipative behaviour of the schemes can be well observed. Time integration was performed with a four-stage fourth-order accurate Runge–Kutta algorithm and a time step of $\Delta t = 0.005$, resulting in a CFL-number $\sigma = 0.05$. The latter is sufficiently small to ensure that the error is mainly due to the spatial discretization. A long integration time of 50 is chosen to clearly distinguish the dissipation and the dispersion errors.

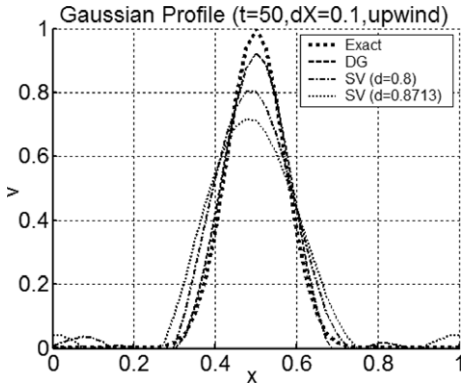

Fig. 13. Comparison of Gaussian profiles at $t = 50$ computed with the fourth-order DG scheme and with the fourth-order SV schemes with $d = 0.8$ and $d = 0.8713$, using an upwind Riemann flux ($\alpha = 1$).

In Fig. 13, the profiles at $t = 50$ obtained with different upwind schemes are shown. At this time, the wave has travelled through the domain fifty times. The figure clearly illustrates that the fourth-order SV scheme with $d = 0.8$ is more accurate than the scheme with $d = 0.8713$. Also, the discontinuous Galerkin scheme produces a better result than either of the SV schemes. These results are in accordance with the analysis made in Section 3.

The instability of the upwind scheme based on the Gauss–Lobatto points was also verified numerically, with the third-order TVD Runge–Kutta scheme that was used in Wang [1]. The mesh size was set to $\Delta x = 0.01$ and the CFL-number was $\sigma = 0.125$. To illustrate the weakness of this instability, the solution is plotted at $t = 30$ and at $t = 45$ in Fig. 14. Despite the fact that at $t = 30$, the wave has travelled through the computational domain 30 times, there is no indication of any instability yet. However, at $t = 45$ the instability has clearly started developing. For $\sigma = 0.2$, the computation was stable and for $\sigma = 0.25$, it immediately became unstable, which confirms the results found in Section 3.

Fig. 15 shows the advected waves at $t = 50$, obtained with different central schemes. As was predicted in Section 3, the fourth-order SV scheme with $d = 0.6380$ performs significantly better than the other SV schemes, and it slightly outperforms the discontinuous Galerkin scheme. When compared with Fig. 13, one could conclude that the central schemes give better results than the upwind schemes. One should keep in mind however that with these central schemes, the numerical group velocity is not defined for certain wave numbers. Especially for non-linear problems, where energy can be transferred between wavenumbers, this could pose problems. Consequently, the usefulness of these schemes is limited in practice.

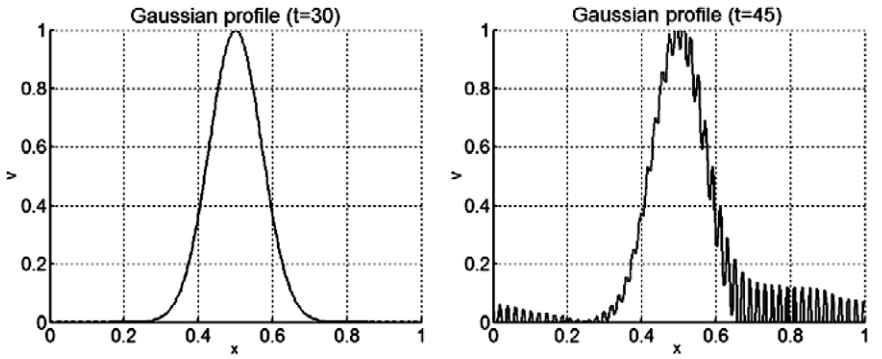

Fig. 14. 1D advection of a Gaussian wave with the fourth-order upwind SV scheme based on the Gauss–Lobatto points, for CFL = 0.125. $t = 30$ (left) and $t = 45$ (right).

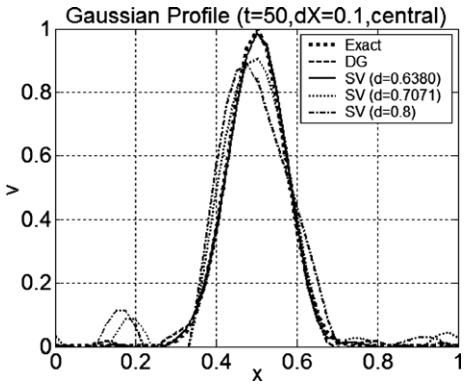

Fig. 15. Comparison of Gaussian profiles at $t = 50$ computed with the fourth-order DG scheme and the fourth-order SV schemes with $d = 0.6380$, $d = 0.7071$ and $d = 0.8$, using a central Riemann flux ($\alpha = 0$).

6.2. 1D linear advection with source term

To test the performance of the p-multigrid algorithm, the steady state solution of the following model problem is computed.

$$\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} = n x^{n-1} \tag{30}$$

on the domain $0 \leqslant x \leqslant 1$, with $v = 0$ at the inflow. The steady state solution of (30) is

$$v = x^n \tag{31}$$

The steady state solution of (30) for $n = 4$ was computed with the SV4D08U-scheme, on a uniform mesh containing 400 cells. A full p-multigrid algorithm combined with the optimized Runge–Kutta smoothers that were discussed in Section 5 was used, starting from a uniformely zero initial solution. The selected values for the CFL-numbers σ are those which were determined in the previous section and which ensured an efficient damping of the high-frequency waves. The parameter η_{switch}, used to determine when to start computing on a finer level, was set to 0.2. The same computation was done using the third-order accurate TVD RK scheme as a smoother. The CFL-numbers for this case were determined in the same way as for the optimized RK-schemes, to obtain an efficient damping of the high-frequency waves. As a comparison, two single grid computations on the same mesh, using the optimized five-stage RK and the third-order TVD RK smoother respectively, were performed. The parameters of these computations are summarized in Table 4, where v_1 and v_2 are the number of sweeps in respectively the descending and the ascending part of a V-cycle.

Furthermore, the full p-multigrid computations were repeated on a non-uniform mesh with 400 cells. This mesh was obtained by randomly perturbing the node positions of the uniform mesh, with a maximum deviation of 25% of the uniform mesh cell size. A local time stepping technique was used on this perturbed mesh, meaning that the time steps were chosen for each cell individually such that the local CFL-number corresponded to the prescribed value. The single grid computations were also done on the perturbed mesh, but the residual history was nearly indistinguishable from the computations on the uniform mesh.

Fig. 16 shows the residual history of these computations. Notice the little peaks in the curves corresponding to the full multigrid computations, where the switch to a finer level is made. It is clearly observed that significantly less iterations are needed with the full multigrid algorithm than with the single grid algorithm, for both the optimized RK and the TVD RK smoothers. For the full p-multigrid computations on the uniform mesh, the optimized RK smoothers required 25% less V-cycles to converge than the TVD RK smoother. On the perturbed mesh, the number of V-cycles required by the optimized RK smoothers to converge the solution is the same as on the uniform mesh. For the TVD RK smoother however, a significant increase in the number of cycles needed for convergence was observed.

To make a fair comparison between the computations, the computational times should be considered, since a single grid iteration takes much less time than a V-multigrid cycle. Furthermore, the smoother also has a significant influence on the CPU-time needed to perform an iteration. The measured CPU-times on an Intel Pentium Centrino^{TM} processor with a clockspeed of 1.73 GHz are included in Table 5. The V-cycles VMG1, VMG2 and VMG3 are illustrated in Fig. 17.

Table 4
Computation parameters of the 1D linear advection model problem computation

Algorithm	p-MG level	Spatial scheme	Smoother	CFL(σ)	v_1	v_2
FMG	3	SV4D08U	Optimized RK5	0.40	2	2
FMG	2	SV2U	Optimized RK3	0.65	3	3
FMG	1	SV1U	Optimized RK2	1.20	16	–
FMG	3	SV4D08U	TVD RK3	0.15	2	2
FMG	2	SV2U	TVD RK3	0.40	3	3
FMG	1	SV1U	TVD RK3	1.00	16	–
SG	–	SV4D08U	Optimized RK5	0.40	1	–
SG	–	SV4D08U	TVD RK3	0.15	1	–

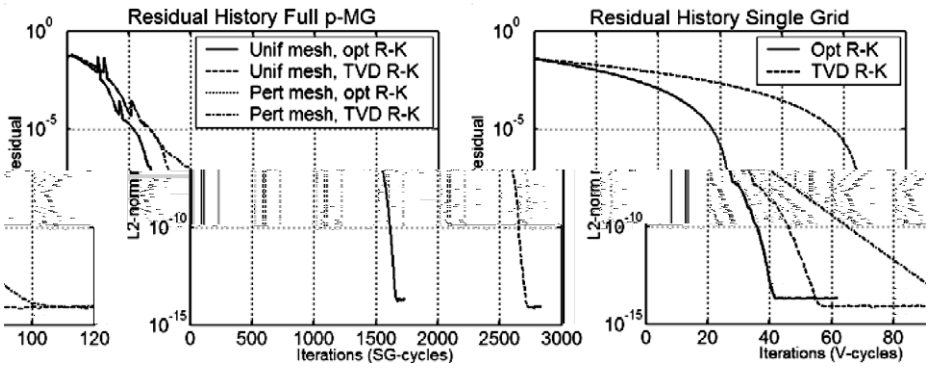

Fig. 16. Residual history of the 1D linear advection model problem computation, using the full multigrid algorithm (left) and a single grid algorithm (right).

Table 5
CPU-time (s) per cycle for the 1D linear advection model problem, on a mesh with 400 cells, for two different smoothers and for different cycles

	SG ($p = 3$)	VMG1	VMG2	VMG3
Optimized RK	2.41	2.28	4.71	15.27
Third-order TVD RK	1.44	3.39	5.63	12.28

Measured on an Intel Pentium Centrino^{TM} processor with a clockspeed of 1.73 GHz. The programming was done in Matlab.

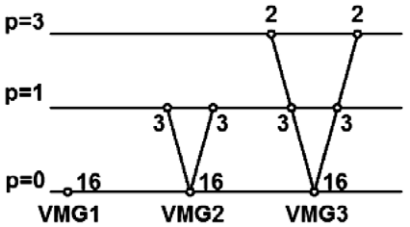

Fig. 17. Used V-cycles, with number of sweeps on each multigrid level.

For the L2-norm of the residual to drop 11 orders of magnitude, 1652 single grid cycles with the optimized RK smoother are needed, amounting to an estimated total CPU-time of 1652×2.41 s ≈ 3981 s. With the full multigrid algorithm and the optimized RK smoothers, 10 VMG1 cycles, 7 VMG2 cycles and 23 VMG3 cycles are needed, resulting in an estimated total time of 10×2.28 s $+ 7 \times 4.71$ s $+ 23 \times 15.27$ s ≈ 407 s for the computation. The full multigrid algorithm thus required about 9.8 times less computational time than the single grid algorithm. The total CPU-times for the other computations can be estimated in the same way, as summarized in Table 6. From this table, it can be seen that although the optimized five-stage RK smoother needed more time for the single grid computation than the TVD RK smoother, the optimized RK smoothers were

Table 6
Estimated total CPU-times for the residual L2-norm to drop 11 orders of magnitude, for the linear advection model problem

Computation	#SG	#VMG1	#VMG2	#VMG3	Total CPU-time
SG, unif mesh, opt RK	1652	–	–	–	≈3981 s
SG, unif mesh, TVD RK	2694	–	–	–	≈3879 s
FMG, unif mesh, opt RK	–	10	7	23	≈407 s
FMG, unif mesh, TVD RK	–	12	9	32	≈484 s
FMG, pert mesh, opt RK	–	10	7	23	≈407 s
FMG, pert mesh, TVD RK	–	12	9	69	≈939 s

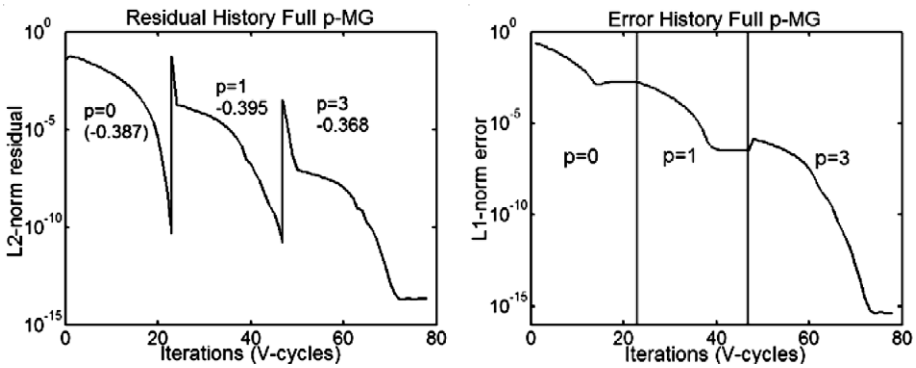

Fig. 18. Residual L2-norm (left) and error L1-norm (right) history for the 1D linear advection model problem, using the full multigrid algorithm, for a mesh containing 400 cells, with full convergence to 10^{-10} on each level.

significantly faster for the multigrid computations. Especially for the full p-multigrid computation on the perturbed mesh, the optimized RK smoothers converged the solution in less than half the time needed by the TVD RK smoother. It can thus be concluded that the optimized RK smoothers are more suited for the present full p-multigrid algorithm for the SV method than the TVD RK scheme.

Some tests to investigate the dependence upon p and h were done as well. These tests are analogous to the tests done in Fidkowski et al. [15]. Fig. 18 shows the residual history (left) and the error history (right) of a full p-multigrid computation on the mesh containing 400 cells, where the residual was converged to 10^{-10} on each level instead of using the residual based switching criterion. The average rate of convergence, which is indicated in the figure, varies only slightly with increasing p. From the right figure one can conclude that, for the coarser levels, truncation error accuracy is reached long before the residual is converged. This justifies the use of the residual based switching criterion. On the finest multigrid level however, the error does not reach the truncation error level before the residuals reach machine zero, for this computation. Fig. 19 shows the residual histories of four full p-multigrid computations (left) and of four single grid computations (right), on meshes of different sizes. It is clearly seen that for the full p-multigrid algorithm, the dependence upon h of the number of iterations needed to achieve convergence is small for large h. For meshes with more cells, this dependence is more important. However, from the right figure one can conclude that for the single grid algorithm, the required number of iterations increases much faster with decreasing h.

6.3. Steady quasi-1D Euler flow through convergent nozzle

This test case is taken from Wang et al. [3]. The quasi-1D Euler equations are given by

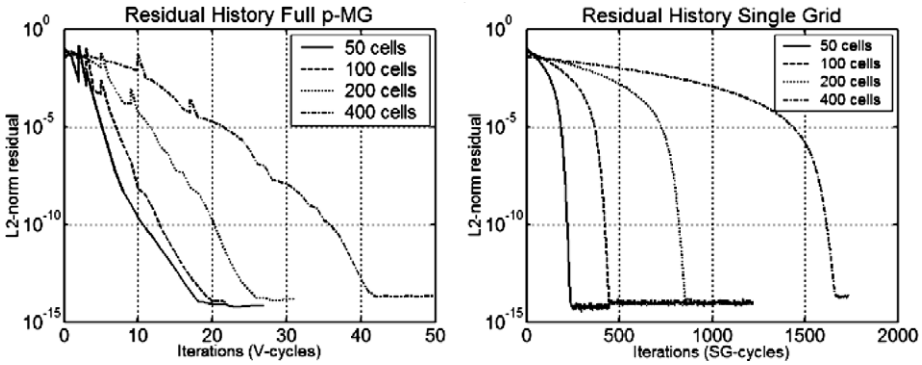

Fig. 19. Residual history for the 1D linear advection model problem, for different meshes, using the full multigrid algorithm (left) and a single grid algorithm (right).

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho A\\ \rho u A\\ \rho E A\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u A\\ (\rho u^2+p)A\\ (\rho u E+up)A\end{bmatrix}=\begin{bmatrix}0\\ p\frac{\partial A}{\partial x}\\ 0\end{bmatrix} \tag{32}$$

where the energy E is defined by

$$E=\frac{1}{\gamma-1}\frac{p}{\rho}+\frac{1}{2}u^2 \tag{33}$$

and the area of the convergent nozzle is given by

$$A(x)=1.5-0.5\tanh(x),\quad -5\leqslant x\leqslant 5 \tag{34}$$

The inflow and outflow conditions are

$$\begin{aligned}\{\rho_{\text{in}},u_{\text{in}},p_{\text{in}}\}&=\{1.2949245,0.30891936,1.0256854\}\\ \{\rho_{\text{out}},u_{\text{out}},p_{\text{out}}\}&=\{1,0.8,0.71428571\}\end{aligned} \tag{35}$$

An analytical solution of this problem is available. The steady state solution of (32) was computed using the SV scheme with $d=0.8$ combined with a Flux Difference Splitting (FDS) Riemann flux, on both a uniform and a non-uniform mesh with 100 cells. The non-uniform mesh was again obtained by randomly perturbing the node positions of the uniform mesh, by a maximum amount of 25% of the uniform mesh cell size. A full p-multigrid algorithm combined with the optimized RK smoothers was used, starting from an initial solution which varies linearly between the inlet and the outlet conditions. To ensure an efficient damping, a local time stepping technique was used. The time steps were determined using the following formula:

$$\Delta t_i=\frac{2\sigma\Delta x_i}{(|u|+c)_i^{\xi=-1}+(|u|+c)_i^{\xi=1}},\quad i=1,\ldots,N_{\text{SV}} \tag{36}$$

where the CFL-number σ has a prescribed value, which ensures an efficient damping of the high-frequency waves. In (36), the maximum wave propagation speed in the SV with index i is estimated by the average of the largest eigenvalues on the boundaries of the SV. The speed of sound c is defined by $c^2=\gamma p/\rho$. The multigrid algorithm parameters are the same as for the first test case, except for the CFL-number at the finest multigrid level, which was set to 0.1. This smaller value was required there for the solution to converge rapidly. The reason for this is probably that the smoothing efficiency analysis of Section 5 is not entirely valid anymore, because an upwind Riemann flux was assumed for the calculation of the Fourier footprints (see Section 3) and for the present computation, a FDS flux is used. As a comparison, single grid computations were performed on both meshes, using the optimized five-stage RK smoother with CFL-number $\sigma=0.4$. The residual history of the single grid computation on the perturbed mesh was again hardly distinguishable from that of the computation on the uniform mesh.

Fig. 20 shows the obtained residual histories. Again, significantly less iterations are needed with the full multigrid algorithm than with the single grid algorithm. On the perturbed mesh, the algorithm needed only a few iterations more to converge the solution than on the uniform mesh.

The measured CPU-times per iteration are listed in Table 7. For the L2-norm of all the residuals to drop 11 orders of magnitude, 4801 single grid cycles were needed, amounting to an estimated total CPU-time of $4801\times 1.47\text{ s}\approx 7057\text{ s}$. With the full multigrid algorithm, 7 VMG1 cycles, 8 VMG2 cycles and 86 VMG3 cycles were needed, resulting in an estimated total time of $7\times 1.45\text{ s}+8\times 3.05\text{ s}+86\times 9.47\text{ s}\approx 849\text{ s}$ for the computation. The time needed for the multigrid algorithm on the perturbed mesh is estimated in the same way. These total times have been summarized in Table 8. It is seen that on the uniform grid, the full p-multigrid algorithm required about 8.3 times less CPU-time than the single grid algorithm, while on the perturbed mesh, it was 7.8 times faster.

Fig. 21 shows the residual history (left) and the mass density error history (right) of a full p-multigrid computation on the mesh containing 100 cells, where the residuals were converged to 10^{-10} on each level. For this case, there is a more significant dependence of the convergence rate on p. The average convergence rate on the level with $p=3$ is just over half of the convergence rate on the level with $p=1$. For this computation, the error reaches the truncation error level before the residuals are converged, on each of the multigrid levels, which again

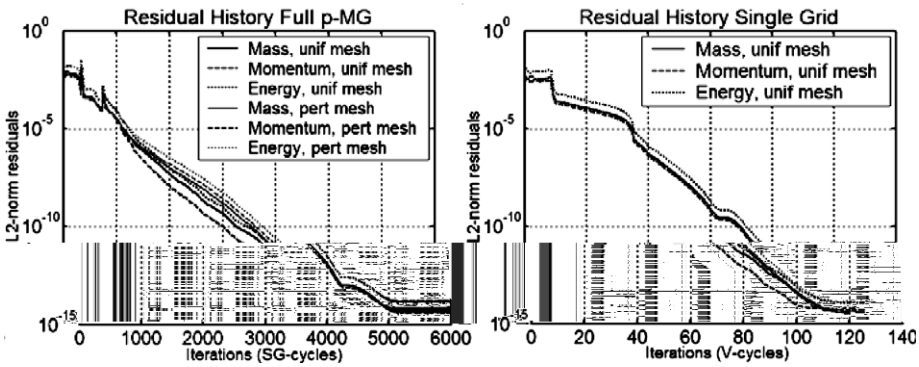

Fig. 20. Residual history of the quasi-1D Euler flow computation, using the full multigrid algorithm (left) and a single grid algorithm (right).

Table 7
CPU-time (s) per cycle for the quasi-1D Euler flow problem, on a mesh with 100 cells, for the optimized RK smoother and for different cycles

	SG ($p = 3$)	VMG1	VMG2	VMG3
Optimized RK	1.47	1.45	3.05	9.47

Measured on an Intel Pentium Centrino^{TM} processor with a clockspeed of 1.73 GHz. The programming was done in Matlab.

Table 8
Estimated total CPU-times for the residual L2-norm to drop 11 orders of magnitude, for the quasi-1D Euler flow problem

Computation	#SG	#VMG1	#VMG2	#VMG3	Total CPU-time
SG, unif mesh, opt RK	4801	–	–	–	≈7057 s
FMG, unif mesh, opt RK	–	7	8	86	≈849 s
FMG, pert mesh, opt RK	–	7	8	91	≈896 s

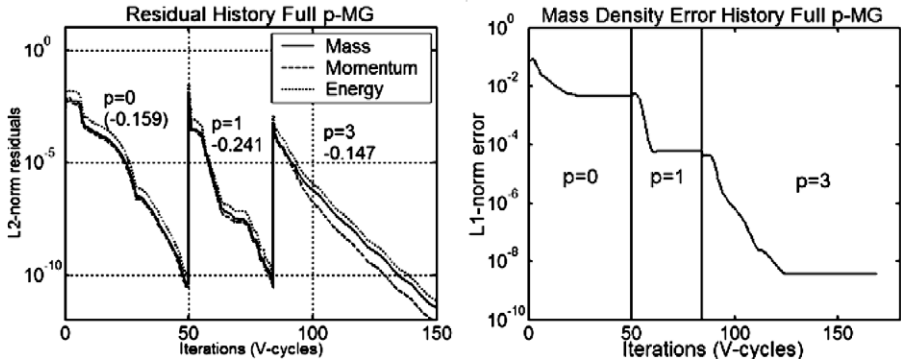

Fig. 21. Residual L2-norm (left) and mass density error L1-norm (right) history for the quasi-1D Euler flow problem, using the full multigrid algorithm, for a mesh containing 100 cells, with full convergence to 10^{-10} on each level.

justifies the use of the residual based switching criterion. Fig. 22 shows the residual histories of four full p-multigrid computations (left) and of four single grid computations (right), on meshes of different sizes. It is clearly seen that for the full p-multigrid algorithm, the dependence on h of the number of iterations needed to achieve convergence is very small. Only the computation on the finest mesh required slightly more iterations than the computations on the coarser meshes. In the right figure one can see that the number of iterations needed by the single grid algorithm with the optimized five-stage RK smoother does have an important h-dependence.

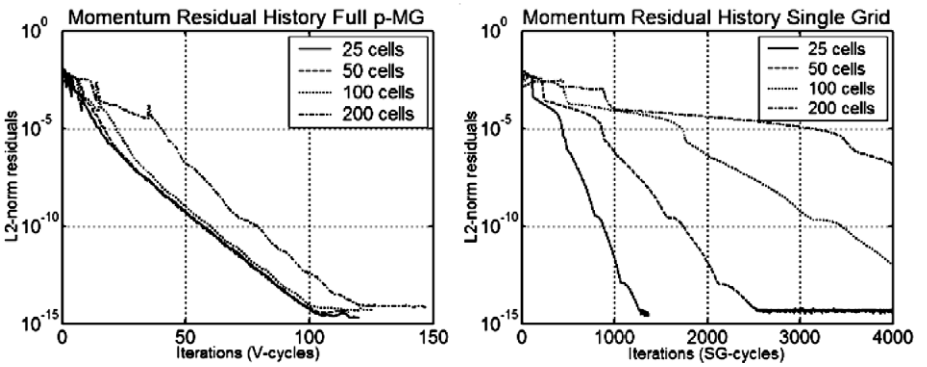

Fig. 22. Residual history for the quasi-1D Euler flow problem, for different meshes, using the full multigrid algorithm (left) and a single grid algorithm (right).

7. Conclusions

The wave propagation properties of the spectral volume method have been analyzed and the influence of the partitioning into control volumes was studied. Based on this analysis, a 1D SV scheme with good wave propagation properties was proposed. A comparison was made with the dispersion and dissipation properties of the discontinuous Galerkin method. It was also shown that the third and fourth-order accurate SV schemes based on the Gauss–Lobatto distributions are weakly unstable for the 1D linear advection equation, when combined with a simple upwind flux. The results of this analysis have been numerically verified by the 1D linear advection of a Gaussian profile. It may be concluded that the dispersion and dissipation properties, which also take into account the influence of the Riemann flux, are a better measure for the quality of the spectral volume partitions than the Lebesgue number. Furthermore, a *p*-multigrid algorithm for the SV method was implemented. A definition for the restriction and prolongation operators was formulated, based on the operator definitions for the discontinuous Galerkin method. Optimized explicit Runge–Kutta smoothing operators were developed and their efficiency was analyzed. The *p*-multigrid algorithm was tested on a 1D linear advection equation with a source term and on the quasi-1D Euler flow through a convergent nozzle. In both cases, a significant decrease in the necessary number of iterations and computational time was observed. Implementation of the *p*-multigrid algorithm for 2D is underway and will be reported in a future publication.

Acknowledgement

Part of this research was funded by IWT under project SBO 050163. This funding is gratefully acknowledged.

References

[1] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, Journal of Computational Physics 178 (2002) 210–251.
[2] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: Extension to two-dimensional scalar equation, Journal of Computational Physics 179 (2002) 665–697.
[3] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: One dimensional systems and partition optimization, Journal of Scientific Computing 20 (2004) 137–157.
[4] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: Extension to two-dimensional Euler equations, Journal of Computational Physics 194 (2) (2004) 716–741.
[5] Y. Liu, M. Vinokur, Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: Extension to three-dimensional systems, Journal of Computational Physics 212 (2006) 454–472.
[6] Y. Sun, Z.Y. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow, Journal of Computational Physics 215 (2006) 41–58.
[7] Y. Sun, Z.J. Wang, Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids, International Journal for Numerical Methods in Fluids 45 (2004) 819–838.

[8] M. Zhang, C.W. Shu, An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods, Computers and Fluids 34 (2005) 581–592.
[9] Fang Q. Hu, M.Y. Hussaini, Patrick Rasetarinera, An analysis of the discontinuous Galerkin method for wave propagation problems, Journal of Computational Physics 151 (1999) 921–946.
[10] B.T. Helenbrook, H.L. Atkins, Application of "*p*"-multigrid to discontinuous Galerkin formulations of the Poisson equation, AIAA Journal 44 (3) (2005) 566–575.
[11] E.M. Ronquist, A.T. Patera, Spectral element multigrid I. Formulation and numerical results, Journal of Scientific Computing 2 (4) (1987) 389406.
[12] B.T. Helenbrook, D.J. Mavriplis, H.A. Atkins, Analysis of *p*-multigrid for continuous and discontinuous finite element discretizations, AIAA Paper 2003-3989, 2003.
[13] F. Bassi, S. Rebay, Numerical solution of the Euler equations with a multiorder discontinuous finite element method, in: Second International Conference on Computational Fluid Dynamics, Sydney, Australia, 2002.
[14] Krzysztof J. Fidkowski, A high-order discontinuous Galerkin multigrid solver for aerodynamic applications, Master's Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.
[15] Krysztof J. Fidkowski, Todd A. Oliver, James Lu, David L. Darmofal, *p*-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, Journal of Computational Physics 207 (2005) 92–113.
[16] A. Brandt, Guide to Multigrid Development, Springer, Berlin, 1982.
[17] C.-W. Shu, Total variation diminishing Runge–Kutta schemes, SIAM Journal of Scientific and Statistical Computing 9 (1988) 1073–1084.
[18] C. Lacor, S. Smirnov, M. Baelmans, A finite volume formulation of compact central schemes on arbitrary structured grids, Journal of Computational Physics 198 (2004) 535–566.
[19] J. Ramboer, T. Broeckhoven, S. Smirnov, C. Lacor, Optimization of time integration schemes coupled to spatial discretization for use in CAA applications, Journal of Computational Physics 213 (2006) 777–802.