A central compact hybrid-variable method with spectral-like resolution: One-dimensional case

Md Mahmudul Hasan[a], Xianyi Zeng[a,b,*]

[a] *Computational Science Program, The University of Texas at El Paso, , El Paso, 79902, TX, United States*
[b] *Department of Mathematical Sciences, The University of Texas at El Paso, , El Paso, 79902, TX, United States*

Abstract

We present a central compact hybrid-variable method (CHVM) with spectral-like accuracy for first-order hyperbolic problems with moderate or less discontinuities. It incorporates the compact difference strategy and a recently proposed hybrid-variable discretization technique to achieve even higher accuracy on a given stencil of grid cells. The CHVM is first constructed for the one-dimensional (1D) model linear advection equations, in which case the accuracy and stability analysis are conducted; then it is extended to 1D nonlinear problems such as the Burgers' equation and the Euler equations. A novel Gauss-Seidel type low-pass high-order filter is constructed to suppress spurious oscillations near discontinuities. The performance of the proposed method is assessed by extensive benchmark tests.

Keywords: hybrid-variable methods, compact scheme, spectral-like accuracy, hyperbolic equations, unconditional stability, low-pass filter for hybrid data
2020 MSC: 65M06, 65M08, 65M22

1. Introduction

Multi-scale phenomenon plays a crucial role in many physical and engineering applications such as turbulent flow computations and aeroacoustics. Numerous efforts have been devoted to construct numerical methods that are high accurate in a

[*] Corresponding author

Email addresses: mhasan5@miners.utep.edu (Md Mahmudul Hasan), xyzeng@lehigh.edu (Xianyi Zeng) wide spectrum of wavelengths, for example, the methods with so-called spectral-like accuracy aim at accurately computing waves that are barely resolved by the mesh grids. A pioneer is the compact difference method proposed in the seminal work by S. K. Lele [1], which constructs finite difference approximations to one-dimensional hyperbolic problems with spectral-like accuracy. In particular, it splits the degrees of freedom in its coefficients between two objectives: achieving a certain formal order of accuracy and obtaining exact wavenumber at certain pre-selected frequencies in a formal Fourier analysis of the method.

Since then, the compact difference strategy has been adopted in many work to construct numerical methods with spectral-like accuracy [2, 3, 4, 5, 6, 7], and a recent one is by Liu et al. [8, 9]. This method differs from others in that it employs two types of unknowns per mesh cell, with one approximating grid-point values and the other one approximating the solution at cell centroids. Consequently, all degrees of freedom can be utilized to achieve the optimal order of accuracy and the resulting method shows good ability of computing the smallest wavelength that is resolvable by the computational grid. Like any other central methods, the compact scheme proposed by Liu et al. relies on the exact cancellation of Taylor series expansion terms of the numerical solutions in a centered stencil on a uniform grid; if the mesh is irregular, one expects the formal order of accuracy to be decreased.

In this work, we propose a central compact method that incorporates a recently developed superconvergent hybrid-variable (HV) discretization framework [10, 11], as a first step to build numerical method with spectral-like accuracy that is suitable for both uniform and non-uniform computational grids. Following the HV framework, we employ two types of variables: one approximates the nodal solutions at grid points and the other one approximates the averaged solutions for each mesh cell. Therefore, this method shares similar spectral property as the method by Liu et al., see also the formal Fourier analysis in Section 2.5. Unlike the other work, however, it is proved in [11] that approximating both nodal values and cell-averaged values lead to an inherent superconvergence property that is not possessed by any method that only takes nodal approximations; and such superconvergence is independent of the mesh uniformity, see for example its numerical examples on two-dimensional triangular grids in [10].

To this end, combining the compact difference strategy and the HV framework is a natural next step of constructing high-order and spectral-like accurate method for general grids and we name it the central compact hybrid-variable method or CHVM for short. In this work, we focus on the preliminary study of CHVM by limiting ourselves to one-dimensional problems and we construct optimally sixth-order accurate method using the smallest grid stencil containing two neighboring cells at each grid point. Furthermore, we shall assume uniform grids throughout the paper for simpler accuracy and stability analysis hence superconvergence will not be observed (see the analysis in [11]). Formal analysis on non-uniform grids and the superconvergence behavior will be presented in future work.

The rest of the paper is organized as follows. In Section 2, the construction of the compact hybrid-variable scheme is detailed for 1D advection equations with various boundary conditions. This section also contains the formal accuracy and stability analyses of the proposed method. Extension to nonlinear scalar problems and systems is provided in Section 3, demonstrated by its application to solve the Burgers' equation and the Euler equations. Since discontinuities generally appear for nonlinear hyperbolic problems even when the initial data is smooth, we also construct a Gauss-Seidel type low-pass filter to effectively remove node-to-node oscillations in this section. The numerical performance of the proposed method is assessed in Section 4, and lastly Section 5 concludes this paper.

2. A compact hybrid-variable method for 1D advection equations

In this work we focus on the 1D hyperbolic system:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{0}\,, \tag{2.1}$$

where $\boldsymbol{u}$ is a vector-valued function in $\mathbb{R}^d$ and $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ is a smooth flux function. For the construction of the new method, we assume that $\boldsymbol{u}$ has sufficient regularity so that all required spatial derivatives exist. In this section, we focus on the model linear advection equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0\,, \tag{2.2}$$

where c is the constant advection velocity; and then we extend the method to nonlinear scalar equations and systems in Section 3.

2.1. A brief review of the hybrid-variable (HV) discretization strategy

To simplify the situation, in Sections 2.1–2.3 we consider the Cauchy problem of (2.2) that is defined on the closed interval $x \in [0,\ 1]$ with appropriate initial condition and the periodic boundary condition $u(0,t) = u(1,t)$. The computational domain is divided uniformly into N cells $[jh,\ (j+1)h]$, $j = 0, \cdots, N-1$, where $h = 1/N$ is the cell size. The cell faces and the cell centers are denoted $x_j = jh$ and $x_{j+1/2} = (j+1/2)h$, respectively. In the HV discretization framework, instead of searching for only the nodal approximations (as in finite difference methods) or only the cell-averaged approximations (as in finite volume methods), we seek both approximations and evolve them in time. In particular, the semi-discretized variables are denoted:

$$u_j(t) \approx u(x_j, t)\,, \quad \overline{u}_{j+1/2}(t) \approx \frac{1}{h}\int_{jh}^{(j+1)h} u(x,t)dx\ , \tag{2.3}$$

for the nodal and cell-averaged approximations, respectively.

To derive the semi-discretized equations for the cell-averaged variables, we integrate (2.2) over $[jh, (j+1)h]$ to obtain:

$$\frac{d\overline{u}_{j+1/2}}{dt} + \frac{1}{h}\left[cu_{j+1} - cu_j\right] = 0\ , \tag{2.4a}$$

where the dependence on t is suppressed for simplicity. Similarly, the semi-discretized equation for the nodal variables is:

$$\frac{du_j}{dt} + c[\mathcal{D}_x u]_j = 0\ , \tag{2.4b}$$

where $[\mathcal{D}_x u]_j$ approximates $u_x(x_j, t)$ that characterizes the HV method.

In previous work [11], the hybrid-variable discrete differential operator (HV-DDO) $[\mathcal{D}_x]_j$ is constructed *explicitly* as a linear combination of close-by solutions. For example, a first-order upwind HV-DDO is given by:

$$[\mathcal{D}_x u]_j = \begin{cases} \frac{2}{h}(u_j - \overline{u}_{j-1/2}) & \text{if } c \geq 0 \\ \frac{2}{h}(\overline{u}_{j+1/2} - u_j) & \text{if } c < 0 \end{cases}\ . \tag{2.5}$$

In this work, however, we seek *implicit* construction of these operators following the compact difference strategy first proposed by Lele [1] in the context of finite difference approximation to acoustic problems, which is reviewed next.

2.2. A brief review of the central compact difference schemes

In the context of finite difference methods (hence we only consider u_j for the moment), one finds the approximations $\mathcal{D}_x u_j \approx u(x_j)$ (we do not use the brackets to distinguish it from the HV-DDOs), such that for a given stencil $s > 0$ and $\forall j$:

$$\sum_{k=-s}^{s} \alpha_k \mathcal{D}_x u_{j+k} = \frac{1}{h}\sum_{k=-s}^{s} \beta_k u_{j+k}\ , \tag{2.6}$$

where α_0 is normalized to 1 and the coefficients α_k and β_k are constructed so that a target number of leading terms in the Taylor series expansions about x_j of:

$$\sum_{k=-s}^{s} \alpha_k u_x(x_{j+k}) \quad \text{and} \quad \frac{1}{h}\sum_{k=-s}^{s} \beta_k u(x_{j+k}) \tag{2.7}$$

are matched, and the rest of degrees of freedom is usually utilized to make sure the numerical wavenumber equals the exact one at certain pre-selected frequencies in a formal Fourier analysis, see also the discussion in Section 2.5.

For example, using three grid points ($s = 1$) and aiming at optimal order of accuracy, the resulting compact discretization formula is given by:

$$\frac{1}{4}\mathcal{D}_x u_{j-1} + \mathcal{D}_x u_j + \frac{1}{4}\mathcal{D}_x u_{j+1} = -\frac{3}{4h}u_{j-1} + \frac{3}{4h}u_{j+1}\,, \quad \forall j\,, \tag{2.8}$$

which leads to a linear system of all approximations $\mathcal{D}_x u_j$. The coefficients in (2.8) are computed to match the four leading terms in the Taylor series expansions of (2.7), or alternatively:

$$\frac{1}{4}u_x(x_{j-1}) + u_x(x_j) + \frac{1}{4}u_x(x_{j+1}) = -\frac{3}{4h}u(x_{j-1}) + \frac{3}{4h}u(x_{j+1}) + O(h^4)\,, \quad \forall j$$

for sufficiently smooth u.

2.3. A sixth-order accurate compact hybrid-variable method (CHVM)

Combining the two strategies, given a stencil $s > 0$ we seek a formula that is similar to (2.6) but uses both u_j and $\overline{u}_{j+1/2}$ on the right hand side. In this paper, we consider the smallest stencil possible, i.e., $s = 1$, and will see that the CHVM strategy gives rise to a sixth-order method (c.f., fourth-order by (2.8)). To this end, the HV-DDOs are constructed by:

$$\alpha[\mathcal{D}_x u]_{j-1} + [\mathcal{D}_x u]_j + \alpha[\mathcal{D}_x u]_{j+1} = \frac{1}{h}\left[-\beta u_{j-1} - \gamma\overline{u}_{j-1/2} + \gamma\overline{u}_{j+1/2} + \beta u_{j+1}\right]\,, \tag{2.9}$$

where the symmetry of the coefficients come form the central nature of the operator and a uniform grid, and the coefficients α, β, γ are computed such that:

$$\begin{aligned} &\alpha u_x(x_{j-1}) + u_x(x_j) + \alpha u_x(x_{j+1}) \\ =\ & -\frac{\beta}{h}u(x_{j-1}) - \frac{\gamma}{h^2}\int_{x_{j-1}}^{x_j} u(x)dx + \frac{\gamma}{h^2}\int_{x_j}^{x_{j+1}} u(x)dx + \frac{\beta}{h}u(x_{j+1}) + O(h^p)\,, \end{aligned} \tag{2.10}$$

for some integer p as large as possible.

For simplicity, we write $u(x_j)$, $u_x(x_j)$, $u_{xx}(x_j)$ as u, u_x, $u^{(2)}$, etc.; that is, $u^{(k)}$ denotes the k^{th} derivative of u for $k \geq 2$. Then the Taylor series expansions of the terms in the left hand side of (2.10) are:

$$u_x(x_j) = u_x \,,$$
$$u_x(x_{j-1}) + u_x(x_{j+1}) = 2u_x + h^2u^{(3)} + \frac{1}{12}h^4u^{(5)} + O(h^6) \,;$$

and that of the nodal values in the right hand side are given by:

$$u(x_{j+1}) - u(x_{j-1}) = 2hu_x + \frac{1}{3}h^3u^{(3)} + \frac{1}{60}h^5u^{(5)} + O(h^7) \,,$$

whereas in the case of the cell-averaged values we have:

$$\begin{aligned}
&\frac{1}{h}\int_{x_{j-1}}^{x_j} u(x)dx = \frac{1}{h}\int_{-h}^{0} u(x_j + s)ds \\
&= \frac{1}{h}\int_{-h}^{0}\left(u + su_x + \frac{1}{2}s^2u^{(2)} + \frac{1}{6}s^3u^{(3)} + \frac{1}{24}s^4u^{(4)} + \frac{1}{120}s^5u^{(5)} + \frac{1}{720}s^6u^{(6)} + O(h^7)\right)ds \\
&= u - \frac{1}{2}hu_x + \frac{1}{6}h^2u^{(2)} - \frac{1}{24}h^3u^{(3)} + \frac{1}{120}h^4u^{(4)} - \frac{1}{720}h^5u^{(5)} + \frac{1}{5040}h^6u^{(6)} + O(h^7) \,,
\end{aligned}$$

and a similar formula for $\frac{1}{h}\int_{x_j}^{x_{j+1}} u(x)dx$, which eventually gives:

$$\frac{1}{h}\int_{x_j}^{x_{j+1}} u(x)dx - \frac{1}{h}\int_{x_{j-1}}^{x_j} u(x)dx = hu_x + \frac{1}{12}h^3u^{(3)} + \frac{1}{360}h^5u^{(5)} + O(h^7) \,.$$

Plugging them into (2.10) results in a linear system for α, β, and γ:

$$1 + 2\alpha = 2\beta + \gamma \,, \quad \alpha = \frac{1}{3}\beta + \frac{1}{12}\gamma \,, \quad \frac{1}{12}\alpha = \frac{1}{60}\beta + \frac{1}{360}\gamma \,,$$

with the leading error $O(h^6)$, i.e., $p = 6$. Solving the linear system gives:

$$\alpha = -\frac{1}{8}, \quad \beta = -\frac{9}{8}, \quad \gamma = 3 \,. \tag{2.11}$$

Denoting the solution vectors as well as the vector of the HV-DDOs by:

$$\overline{\boldsymbol{u}} = \begin{bmatrix} \overline{u}_{1/2} \\ \overline{u}_{3/2} \\ \vdots \\ \overline{u}_{N-1/2} \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \boldsymbol{d} = \begin{bmatrix} [\mathcal{D}_x u]_0 \\ [\mathcal{D}_x u]_1 \\ \vdots \\ [\mathcal{D}_x u]_{N-1} \end{bmatrix}, \tag{2.12}$$

where u_N and $[\mathcal{D}_x u]_N$ are omitted in $\boldsymbol{u}$ and $\boldsymbol{d}$, respectively, due to the periodic boundary conditions; we thus have:

$$\boldsymbol{P}\boldsymbol{d} = \frac{1}{h}\boldsymbol{Q}\overline{\boldsymbol{u}} + \frac{1}{h}\boldsymbol{R}\boldsymbol{u} \ , \tag{2.13}$$

with $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$ being circulant matrices: $\boldsymbol{P} = \mathrm{circ}_N(1,\, \alpha,\, 0,\, \cdots,\, 0,\, \alpha)$, $\boldsymbol{Q} = \mathrm{circ}_N(\gamma,\, 0,\, \cdots,\, 0,\, -\gamma)$, and $\boldsymbol{R} = \mathrm{circ}_N(0,\, \beta,\, 0,\, \cdots,\, 0,\, -\beta)$. Here we use the notation $\mathrm{circ}_N(c_0,\ c_1,\ \cdots,\ c_{N-1})$ to designate the matrix $\boldsymbol{C} = [c_{ij}]_{N\times N}$, whose entries are given by $c_{ij} = c_{(j-i)(\mathrm{mod}\ N)}$.

It is easy to see that $\boldsymbol{P}$ is non-singular for all N, hence we can write the semi-discretization (2.4) as a system of ODEs in the matrix form:

$$\frac{d}{dt}\begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} + \frac{c}{h}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{P}^{-1}\boldsymbol{Q} & \boldsymbol{P}^{-1}\boldsymbol{R} \end{bmatrix}\begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} , \tag{2.14}$$

where $\boldsymbol{A} = \mathrm{circ}_N(-1,\ 1,\ 0,\ \cdots,\ 0)$.

In this work, the ODE system (2.14) is integrated by the five-stage, fifth-order, L-stable, and stiffly-accurate Diagonally Implicit Runge-Kutta method [12, Table 24], denoted hereafter by DIRK5 in the rest of the paper. The details of DIRK5 are provided in Appendix A.

2.4. Accuracy analysis

Here we show that the method given by (2.14) is formally sixth-order accurate in space. The standard von Neumann analysis is utilized for this purpose, that is, we assume the initial condition is given by a simple wave:

$$u(x, 0) = e^{i\kappa x} \ , \tag{2.15}$$

where κ is an arbitrary constant wavenumber; then the exact solution is given by $u(x,t) = e^{i\kappa(x-ct)}$ and the exact nodal and cell-averaged solutions (designated by a superscript *) are:

$$u_j^\star(t) = e^{-ic\kappa t}e^{ij\theta} \ , \quad \overline{u}_{j+1/2}^\star(t) = \frac{e^{-ic\kappa t}}{i\theta}\left(e^{i(j+1)\theta} - e^{ij\theta}\right) \ , \tag{2.16}$$

with $\theta = \kappa h$ being the scaled wavenumber. It is not difficult to see that the solutions to the ODE system (2.4) as well as $[\mathcal{D}_x u]$ can be written as:

$$u_j(t) = N(t)e^{ij\theta},\ \overline{u}_{j+1/2}(t) = A(t)\frac{1}{i\theta}\left(e^{i(j+1)\theta} - e^{ij\theta}\right),\ [\mathcal{D}_x u]_j = \frac{i\theta}{h}D(t)e^{ij\theta}, \tag{2.17}$$

where the multipliers $N(t)$, $A(t)$, and $D(t)$ satisfy:

$$A'(t) + ic\kappa N(t) = 0 \,, \tag{2.18a}$$

$$N'(t) + ic\kappa D(t) = 0 \,, \tag{2.18b}$$

$$i\theta D(t)(1 + \alpha e^{i\theta} + \alpha e^{-i\theta}) = \beta N(t)(e^{i\theta} - e^{-i\theta}) + \frac{\gamma}{i\theta} A(t)(e^{i\theta} - 2 + e^{-i\theta}) \,. \tag{2.18c}$$

Plugging (2.18c) into (2.18b) gives a system of two ODEs for $A(t)$ and $N(t)$:

$$\frac{d}{dt}\begin{bmatrix} A(t) \\ N(t) \end{bmatrix} + ic\kappa \begin{bmatrix} 0 & 1 \\ \frac{2\gamma(1-\cos\theta)}{\theta^2(1+2\alpha\cos\theta)} & \frac{2\beta\sin\theta}{\theta(1+2\alpha\cos\theta)} \end{bmatrix}\begin{bmatrix} A(t) \\ N(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \,. \tag{2.19}$$

Denoting the coefficient matrix in the left hand side by $\mathbf{C}(\theta)$, the next step is to compute its two eigenvalues so that the solutions to (2.19) can be derived. Indeed, the eigenvalues $\lambda_{1,2}$ of $\mathbf{C}(\theta)$ are the roots of:

$$\lambda^2 - \frac{2\beta\sin\theta}{\theta(1 + 2\alpha\cos\theta)}\lambda - \frac{2\gamma(1 - \cos\theta)}{\theta^2(1 + 2\alpha\cos\theta)} = 0 \,,$$

or equivalently:

$$\lambda_{1,2} = \frac{\beta\sin\theta \pm \sqrt{\Delta}}{\theta(1 + 2\alpha\cos\theta)}, \quad \Delta = \beta^2\sin^2\theta + 2\gamma(1 - \cos\theta)(1 + 2\alpha\cos\theta) \,. \tag{2.20}$$

To proceed, the terms in Δ are reorganized:

$$\begin{aligned} \Delta &= \beta^2\sin^2\theta + 2\gamma(1 - \cos\theta)\left[2\beta\frac{\sin\theta}{\theta} + 2\gamma\frac{1 - \cos\theta}{\theta^2} + \Delta'\right] \\ &= \left(\beta\sin\theta + \frac{2\gamma(1 - \cos\theta)}{\theta}\right)^2 + 2\gamma(1 - \cos\theta)\Delta' \,, \end{aligned}$$

$$\text{where } \Delta' = 1 + 2\alpha\cos\theta - 2\beta\frac{\sin\theta}{\theta} - 2\gamma\frac{1 - \cos\theta}{\theta^2} = 1 - \frac{1}{4}(1 - \frac{1}{2}\theta^2 + \frac{1}{24}\theta^4) + \frac{9}{4}(1 - \frac{1}{6}\theta^2 + \frac{1}{120}\theta^4) - 6(\frac{1}{2} - \frac{1}{24}\theta^2 + \frac{1}{720}\theta^4) + O(\theta^6) = O(\theta^6) \,.$$

Hence:

$$\Delta = \left(\beta\sin\theta + \frac{2\gamma(1 - \cos\theta)}{\theta}\right)^2 + O(\theta^8) \;\Rightarrow\; \sqrt{\Delta} = \beta\sin\theta + \frac{2\gamma(1 - \cos\theta)}{\theta} + O(\theta^7),$$

where we used the fact that $\beta \sin\theta + \frac{2\gamma(1-\cos\theta)}{\theta} = (15/8)\theta + O(\theta^3)$. Without loss of generality, we may choose the branches of $\sqrt{\Delta}$ properly and plug them into (2.20) and obtain:

$$\begin{aligned}
\lambda_1 &= \frac{1}{\theta(1+2\alpha\cos\theta)}\left(\beta\sin\theta + \beta\sin\theta + \frac{2\gamma(1-\cos\theta)}{\theta} + O(\theta^7)\right) \\
&= \frac{-\frac{9}{4}\theta\sin\theta + 6(1-\cos\theta)}{\theta^2(1-\frac{1}{4}\cos\theta)} + O(\theta^6) = 1 + O(\theta^6)\ , \qquad (2.21) \\
\lambda_2 &= \frac{1}{\theta(1+2\alpha\cos\theta)}\left(\beta\sin\theta - \beta\sin\theta - \frac{2\gamma(1-\cos\theta)}{\theta} + O(\theta^7)\right) \\
&= -\frac{2\gamma(1-\cos\theta)}{\theta^2(1+2\alpha\cos\theta)} + O(\theta^6) = -4 + O(\theta^6)\ . \qquad (2.22)
\end{aligned}$$

Thus the solutions to (2.19) are:

$$\begin{aligned}
A(t) &= \frac{\lambda_2 - 1}{\lambda_2 - \lambda_1} e^{-ic\kappa\lambda_1 t} + \frac{1-\lambda_1}{\lambda_2 - \lambda_1} e^{-ic\kappa\lambda_2 t} \\
&= [1 + O(\theta^6)] e^{-ic\kappa t(1+O(\theta^6))} + O(\theta^6) e^{-ic\kappa t(-4+O(\theta^6))} = e^{-ic\kappa t} + O(\theta^6)\ ,
\end{aligned}$$

and similarly:

$$N(t) = \frac{\lambda_1(\lambda_2 - 1)}{\lambda_2 - \lambda_1} e^{-ic\kappa\lambda_1 t} + \frac{\lambda_2(1-\lambda_1)}{\lambda_2 - \lambda_1} e^{-ic\kappa\lambda_2 t} = e^{-ic\kappa t} + O(\theta^6)\ .$$

In comparison with the exact solutions (2.16), we conclude that the proposed central CHVM is formally sixth-order accurate in space.

2.5. Spectral-like accuracy

While there is no unambiguous definition of "spectral-like" accuracy in the literature, it usually refers to a scenario in which that the scaled modified wavenumber $\theta' = \kappa' h$ remains almost on top of the scaled wavenumber $\theta = \kappa h$ when θ gets close to π, that is, when each wavelength is resolved by exactly two mesh cells. In the original work by Lele [1], this is achieved by tuning the parameters in (2.6) such that in addition to the fourth-order accuracy, $\theta'(\theta) = \theta$ is enforced at $\theta = 2.2, 2.3, 2.4$; whereas in the recent work by Liu et al. [8] as well as in present work, spectral-like accuracy is mainly due to using more unknowns per mesh cell and higher-order of accuracy in space.

Computing the scaled modified wavenumber for CHVM is a little tricky as it involves taking the Fourier transforms of the integral of a function. To this end, let us ignore the time abscissa and consider a generic function $u(x)$ and its anti-derivative $U(x) = \int_0^x u(y)dy$. Then the cell-averaged value can be written $\overline{u}_{j+1/2} = (U(x_{j+1}) - U(x_j))/h$. Let us denote the Fourier transform of a scalar function u by $\mathcal{F}(u)$ as usual:

$$\mathcal{F}(u)(\kappa) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u(x) e^{-\imath\kappa x} dx \ ,$$

then it is elementary to have:

$$\mathcal{F}(u_x) = \imath\kappa\mathcal{F}(u) \ . \tag{2.23}$$

Now computing a function $[\mathcal{D}_x u] \approx u_x$ by (2.9) and replacing x_j by x, one has:

$$\begin{aligned} &\alpha[\mathcal{D}_x u](x-h) + [\mathcal{D}_x u](x) + \alpha[\mathcal{D}_x u](x+h) \\ = &\frac{1}{h}\left[-\beta u(x-h) - \frac{\gamma}{h}(U(x) - U(x-h)) + \frac{\gamma}{h}(U(x+h) - U(x)) + \beta u(x+h)\right]. \end{aligned}$$

Taking the Fourier transform on both sides and realizing that:

$$u = U_x \quad \Rightarrow \quad \mathcal{F}(U) = \frac{1}{\imath\kappa}\mathcal{F}(u) \ ,$$

one obtains:

$$\left[1 + \alpha(e^{-\imath\kappa h} + e^{\imath\kappa h})\right]\mathcal{F}([\mathcal{D}_x u]) = \frac{1}{h}\left[\beta(e^{\imath\kappa h} - e^{-\imath\kappa h}) + \frac{\gamma}{\imath\kappa h}(e^{\imath\kappa h} - 2 + e^{-\imath\kappa h})\right]\mathcal{F}(u) \ .$$

Hence $\mathcal{F}([\mathcal{D}_x u]) = \imath\kappa'\mathcal{F}(u)$, where κ' is the modified wave number given by:

$$\kappa' = \frac{1}{h}\frac{2\beta\sin(\kappa h) + 2\gamma(1 - \cos(\kappa h))/(\kappa h)}{1 + 2\alpha\cos(\kappa h)} \ , \tag{2.24}$$

or in terms of scaled modified wavenumber:

$$\theta' = \frac{2\beta\sin\theta + 2\gamma(1 - \cos\theta)/\theta}{1 + 2\alpha\cos\theta} \ . \tag{2.25}$$

In Figure 2.1, (2.25) is compared to the scaled modified wavenumber of several central methods. Particularly, we select methods that are comparable to (2.9) in either the grid-stencil[1] or the number of terms in both hand sides[2]. The methods

[1] That is, no more than two cells for a method centered at x_j.

[2] Three terms in the left-hand side and four terms in the right hand side.

Table 1: Coefficients of different schemes

Scheme	a	b	c	α	β	Order
SPS	$\frac{3}{2}$	0	0	$\frac{1}{4}$	0	4
CTS	$\frac{14}{9}$	$\frac{1}{9}$	0	$\frac{1}{3}$	0	4
LSPS	1.302	0.993	0.037	0.577	0.089	10
CCS-T6	$\frac{16}{9}$	$-\frac{17}{18}$	0	$-\frac{1}{12}$	0	6
CCS-T8	2	$-\frac{61}{50}$	$-\frac{2}{25}$	$-\frac{3}{20}$	0	8

sharing the same grid stencil are Standard Padé Scheme (SPS) and Compact Tridiagonal Scheme (CTS) from [13, Page 538]. The method that shares a similar number of terms is the sixth-order Compact Central Scheme (CCS-T6) by Liu et al. [8, Table 2]. In addition, we hand pick two methods that in our opinion have comparable spectral properties as CHVM, namely Lele's Spectral-like Pentadiagonal Scheme (LSPS) [1, Eq. (3.1.6)] and the eighth-order Compact Central Scheme (CCS-T8) [8, Table 2]. These two methods, however, require a larger stencil than CHVM and thus more computationally expensive. The modified wavenumber of these methods are listed below:

- Modified wavenumber for SPS, CTS and LSPS.

$$\theta' = \frac{a\ \sin(\theta) + (b/2)\ \sin(2\theta) + (c/3)\ \sin(3\theta)}{1 + 2\alpha\ \cos(\theta) + 2\beta\ \cos(2\theta)}\ . \tag{2.26}$$

- Modified wavenumber for CCS-T6 and CCS-T8.

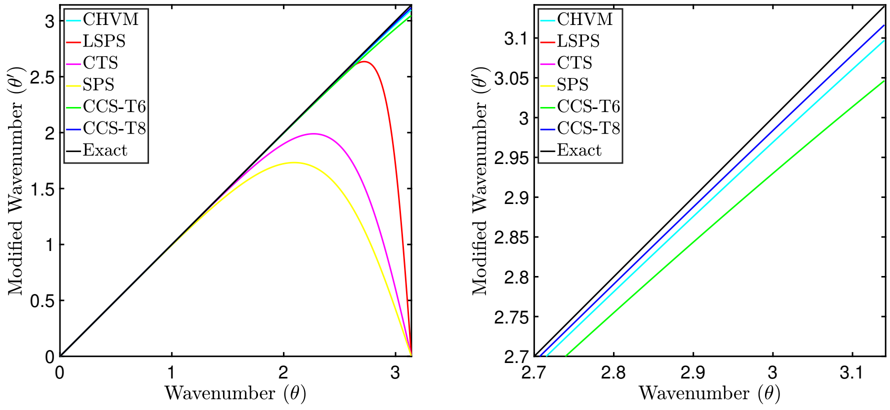
$$\theta' = 2\,\frac{a\ \sin(\theta/2) + (b/2)\ \sin(\theta) + (c/3)\ \sin(3\theta/2)}{1 + 2\alpha\ \cos(\theta) + 2\beta\ \cos(2\theta)}\ . \tag{2.27}$$

where the coefficients and the order of each scheme is summarized in Table 1.

Figure 2.1 shows that the modified wavenumber of CHVM has better resolution than SPS, CTS, LSPS, CCS-T6; whereas CCS-T8 achieved slightly better resolution than CHVM, with a larger stencil and thus higher computational cost comparing to the latter.

2.6. Stability analysis

In this part, we first prove the stability of the ODE system (2.14) and then discuss the stability of the fully-discretized methods. To this end, let us define the



(a) Global View

(b) Local View

Figure 2.1: Modified wavenumber of CHVM and comparison with other compact methods.

shifting matrix $\boldsymbol{S} = \text{circ}_N(0,\ 1,\ 0,\ \cdots,\ 0)$ and write the coefficients of (2.14) as:

$$\boldsymbol{A} = \boldsymbol{S} - \boldsymbol{I}\ , \quad \boldsymbol{P} = \boldsymbol{I} + \alpha\boldsymbol{S} + \alpha\boldsymbol{S}^t\ , \quad \boldsymbol{Q} = \gamma\boldsymbol{I} - \gamma\boldsymbol{S}^t\ , \quad \boldsymbol{R} = \beta\boldsymbol{S} - \beta\boldsymbol{S}^t\ , \tag{2.28}$$

where $\boldsymbol{I}$ is the $N \times N$ identity matrix and $\boldsymbol{S}^t$ is the transpose of $\boldsymbol{S}$; it is not difficult to check that $\boldsymbol{S}^t = \boldsymbol{S}^{-1}$. The stability result for the semi-discretized system is demonstrated by the next theorem, where we extend the discussion to complex vectors as customary in numerical stability analysis.

Theorem 2.1. *There exists a symmetric positive definite matrix $\boldsymbol{H} \in \mathbb{R}^{2N\times 2N}$, such that the $\boldsymbol{H}$-norm of the solution $\boldsymbol{w} = [\overline{\boldsymbol{u}}^t\ \boldsymbol{u}^t]^t$ of (2.14), i.e. $\|\boldsymbol{w}\|^2_{\boldsymbol{H}} = \overline{\boldsymbol{w}}^t\boldsymbol{H}\boldsymbol{w}$ where the overline denotes complex conjugate, is preserved for all $t > 0$. Furthermore, there exists a $\lambda_0 > 0$ that is independent of N, such that $\|\boldsymbol{w}\|_{\boldsymbol{H}} \geq \lambda_0\,\|\boldsymbol{w}\|$ for all $\boldsymbol{w} \in \mathbb{R}^{2N\times 1}$.*

Proof. Let us define:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \gamma^{-1}\boldsymbol{P} \end{bmatrix} \quad \text{and} \quad \boldsymbol{D} = \begin{bmatrix} 0 & \boldsymbol{A} \\ \boldsymbol{P}^{-1}\boldsymbol{Q} & \boldsymbol{P}^{-1}\boldsymbol{R} \end{bmatrix}, \tag{2.29}$$

then it is clear $\boldsymbol{H}$ is symmetric and (2.14) simplifies to:

$$\frac{d\boldsymbol{w}}{dt} = -\frac{c}{h}\boldsymbol{D}\boldsymbol{w}\ .$$

Using (2.28) we have:

$$\boldsymbol{M} \stackrel{\text{def}}{=} \boldsymbol{HD} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{S} - \boldsymbol{I} \\ \boldsymbol{I} - \boldsymbol{S}^t & (\beta/\gamma)(\boldsymbol{S} - \boldsymbol{S}^t) \end{bmatrix} , \tag{2.30}$$

which is clearly skew-symmetric. To this end:

$$\begin{aligned} \frac{d}{dt} \|\boldsymbol{w}\|_{\boldsymbol{H}}^2 = \left(\frac{d\overline{\boldsymbol{w}}}{dt} \right)^t \boldsymbol{Hw} + \overline{\boldsymbol{w}}^t \boldsymbol{H} \left(\frac{d\boldsymbol{w}}{dt} \right) = -\frac{c}{h} \overline{\boldsymbol{w}}^t \boldsymbol{D}^t \boldsymbol{H}^t \boldsymbol{w} - \frac{c}{h} \overline{\boldsymbol{w}}^t \boldsymbol{HDw} \\ = -\frac{c}{h} \overline{\boldsymbol{w}}^t (\boldsymbol{M}^t + \boldsymbol{M}) \boldsymbol{w} = 0 . \end{aligned}$$

It remains to show that $\boldsymbol{H}$ is positive definite and uniformly bounded away from zero for all N. Indeed, $\gamma^{-1}\boldsymbol{P}$ is a circulant matrix, whose eigenvalues are:

$$\frac{1}{\gamma} \left(1 + 2\alpha \cos \frac{k\pi}{N+1} \right) = \frac{1}{3} \left(1 - \frac{1}{4} \cos \frac{k\pi}{N+1} \right) \geq \frac{1}{4} , \quad 1 \leq k \leq N .$$

Hence we can choose $\lambda_0 = 1/2$ and complete the proof. □

As a direct consequence, all eigenvalues of the coefficient matrix $\boldsymbol{D}$ are pure imaginary. Indeed, let λ be an eigenvalue of $\boldsymbol{D}$ and $\boldsymbol{w}_0$ be a non-zero (possibly complex) eigenvector, then the solution of the ODE system with the initial data $\boldsymbol{w}(t) = \boldsymbol{w}_0$ is $\boldsymbol{w}(t) = e^{\lambda t}\boldsymbol{w}_0$. Hence following the theorem, $\|\boldsymbol{w}(t)\|_{\boldsymbol{H}}^2 = e^{2\mathrm{Re}\lambda t} \|\boldsymbol{w}_0\|_{\boldsymbol{H}}^2$ is independent of t. Since the $\boldsymbol{H}$-norm of the non-zero vector $\boldsymbol{w}_0$ is not zero, we must have $\mathrm{Re}\lambda = 0$.

Similarly, $\boldsymbol{D}$ must be diagonalizable for if not, there exists an eigenvalue λ of $\boldsymbol{D}$ and two non-zero vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, such that:

$$\boldsymbol{Dw}_1 = \lambda \boldsymbol{w}_1 , \quad \boldsymbol{Dw}_2 = \lambda \boldsymbol{w}_2 + \boldsymbol{w}_1 .$$

Hence the solution corresponding to the initial data $\boldsymbol{w}_2$ is given by $\boldsymbol{w}(t) = e^{\lambda t}(\boldsymbol{w}_2 + t\boldsymbol{w}_1)$. Thus $d(\|\boldsymbol{w}(t)\|_{\boldsymbol{H}}^2)/dt = 0$ requires $\overline{\boldsymbol{w}}_1^t \boldsymbol{H} \boldsymbol{w}_1 = 0$ and $\overline{\boldsymbol{w}}_2^t \boldsymbol{H} \boldsymbol{w}_1 = 0$; but the former contradicts the condition that $\boldsymbol{w}_1 \neq 0$ and $\boldsymbol{H}$ is symmetric positive definite.

Because $\boldsymbol{D}$ is diagonalizable with only pure imaginary eigenvalues, we can easily verify the stability of a fully-discretized method that combines the ODE system (2.14) and any ODE solver whose stability region contains the entire imaginary axis (such as the A-stable methods).

2.7. *Treatment of incoming flow boundaries*

In the case of non-periodic boundary conditions, the formula (2.9) remains valid for computing interior derivatives, i.e., $1 \leq j \leq N-1$; and we need two additional equations to determine all discrete derivatives $[\mathcal{D}_x u]_j,\ 0 \leq j \leq N$. To this end, we consider biased stencils at both ends and write:

$$[\mathcal{D}_x u]_0 + \alpha^l [\mathcal{D}_x u]_1 = \frac{1}{h}\left[\beta_0^l u_0 + \gamma^l \overline{u}_{1/2} + \beta_1^l u_1\right] \tag{2.31}$$

at the left boundary and:

$$\alpha^r [\mathcal{D}_x u]_{N-1} + [\mathcal{D}_x u]_N = \frac{1}{h}\left[\beta_{-1}^r u_{N-1} + \gamma^r \overline{u}_{N-1/2} + \beta_0^r u_N\right] \tag{2.32}$$

at the right boundary. The coefficients are again obtained by matching the leading coefficients of the Taylor series expansions and we have:

$$\alpha^l = -1\ , \quad \beta_0^l = -6\ , \quad \gamma^l = 12\ , \quad \beta_1^l = -6\ , \tag{2.33}$$

and

$$\alpha^r = -1\ , \quad \beta_{-1}^r = 6\ , \quad \gamma^r = -12\ , \quad \beta_0^r = 6\ , \tag{2.34}$$

so that (2.31) and (2.32) are $O(h^3)$ when the discrete approximations are replaced by exact values. The formula (2.31) and (2.32) are the foundation to imposing incoming flow boundary conditions. To fix the idea, let us suppose $c > 0$ and consider the initial boundary value problem (IBVP) given as below:

$$\begin{cases} \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0\ , & (x,\ t) \in [0,\ 1] \times [0,\ T]\ , \\ u(x, 0) = u_{\mathrm{init}}(x)\ , & x \in [0,\ 1]\ , \\ u(0, t) = g(t)\ , & t \in [0,\ T]\ . \end{cases} \quad , \tag{2.35}$$

where $T > 0$ denotes the terminal time and u_{init} and g are prescribed initial condition and boundary condition, respectively. The boundary condition at $x = 0$ is then enforced by setting $u_0(t) = g(t)$ and taking it into account while constructing $[\mathcal{D}_x u]_0$ and $[\mathcal{D}_x u]_1$. To this end, we denote (c.f., (2.12)):

$$\overline{\boldsymbol{u}} = \begin{bmatrix} \overline{u}_{1/2} \\ \overline{u}_{3/2} \\ \vdots \\ \overline{u}_{N-1/2} \end{bmatrix}\ , \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}\ , \quad \boldsymbol{d} = \begin{bmatrix} [\mathcal{D}_x u]_0 \\ [\mathcal{D}_x u]_1 \\ \vdots \\ [\mathcal{D}_x u]_N \end{bmatrix}\ , \tag{2.36}$$

which are then related by:

$$\boldsymbol{P}\boldsymbol{d} = \frac{1}{h}\boldsymbol{Q}\overline{\boldsymbol{u}} + \frac{1}{h}\boldsymbol{R}\boldsymbol{u} + \frac{g(t)}{h}(\beta_0^l \boldsymbol{e}_1^{N+1} - \beta \boldsymbol{e}_2^{N+1})\,, \tag{2.37}$$

where $\boldsymbol{e}_j^M$ denotes the j^{th} unit vector in $\mathbb{R}^M$, and the coefficient matrices $\boldsymbol{P} \in \mathbb{R}^{(N+1)\times(N+1)}$, $\boldsymbol{Q} \in \mathbb{R}^{(N+1)\times N}$, and $\boldsymbol{R} \in \mathbb{R}^{(N+1)\times N}$ are:

$$\boldsymbol{P} = \begin{bmatrix} 1 & \alpha^l & \cdots & 0 & 0 \\ \alpha & 1 & \alpha & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \alpha & 1 & \alpha \\ 0 & 0 & \cdots & \alpha^r & 1 \end{bmatrix}, \ \boldsymbol{Q} = \begin{bmatrix} \gamma^l & 0 & \cdots & 0 & 0 \\ -\gamma & \gamma & \cdots & 0 & 0 \\ 0 & -\gamma & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\gamma & \gamma \\ 0 & 0 & \cdots & 0 & \gamma^r \end{bmatrix}, \ \boldsymbol{R} = \begin{bmatrix} \beta_1^l & 0 & \cdots & 0 & 0 \\ 0 & \beta & \cdots & 0 & 0 \\ -\beta & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \beta \\ 0 & 0 & \cdots & \beta_1^r & \beta_0^r \end{bmatrix}. \tag{2.38}$$

To this end, the resulting ODE system reads:

$$\frac{d}{dt}\begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} + \frac{c}{h}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{S}\boldsymbol{P}^{-1}\boldsymbol{Q} & \boldsymbol{S}\boldsymbol{P}^{-1}\boldsymbol{R} \end{bmatrix}\begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} = \frac{cg(t)}{h}\begin{bmatrix} \boldsymbol{e}_1^N \\ \boldsymbol{S}\boldsymbol{P}^{-1}(-\beta_0^l \boldsymbol{e}_1^{N+1} + \beta \boldsymbol{e}_2^{N+1}) \end{bmatrix}, \tag{2.39}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{S} \in \mathbb{R}^{N\times(N+1)}$ are given by:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{2.40}$$

The incoming flow boundary condition at the right boundary in the case $c < 0$ can be enforced similarly.

At the end of this section, we numerically verify the stability of the method with previous boundary treatments. In particular, the eigenvalues of $\left[\begin{smallmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{S}\boldsymbol{P}^{-1}\boldsymbol{Q} & \boldsymbol{S}\boldsymbol{P}^{-1}\boldsymbol{R} \end{smallmatrix}\right]$ in the case $N = 1000$ are plotted in Figure 2.2. One observes that all eigenvalues are on the closed right complex plane thus the ODE system (2.39) is stable.

3. Extension to 1D nonlinear problems

In this section we provide the key ingredients extending CHVM to the general 1D problem (2.1), repeated here for convenience:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{0}\,, \tag{3.1}$$

as well as a Gauss-Seidel type low-pass filtering scheme to remove spurious oscillations near strong discontinuities.

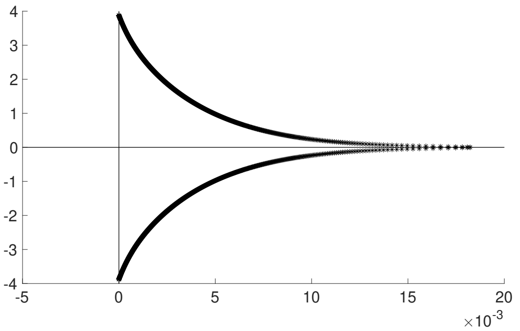


Figure 2.2: The eigenvalues of the matrix in the left hand side of (2.39) when $N = 1000$.

3.1. The CHVM: Semi-discretization in space

Denoting the nodal approximations and the cell-averaged approximations by $\boldsymbol{u}_j$ and $\overline{\boldsymbol{u}}_{j+1/2}$, the spatial discretization of CHVM gives rise to the ODE system:

$$\frac{d\overline{\boldsymbol{u}}_{j+1/2}}{dt} + \frac{1}{h}\left[\boldsymbol{f}(\boldsymbol{u}_{j+1}) - \boldsymbol{f}(\boldsymbol{u}_j)\right] = \boldsymbol{0} \ , \tag{3.2a}$$

$$\frac{d\boldsymbol{u}_j}{dt} + \boldsymbol{j}(\boldsymbol{u}_j)[\mathcal{D}_x\boldsymbol{u}]_j = \boldsymbol{0} \ , \tag{3.2b}$$

where $\boldsymbol{j}(\boldsymbol{u}) \overset{\text{def}}{=\!=} \partial\boldsymbol{f}(\boldsymbol{u})/\partial\boldsymbol{u} \in \mathbb{R}^{d\times d}$ is the Jacobian matrix of the flux function.

Assume periodic boundary condition for simplicity, the HV-DDOs $[\mathcal{D}_x\boldsymbol{u}]_j,\ 0 \le j \le N-1$ are obtained by a similar formula to (2.13). For later uses, we shall denote the entries of the matrices by $\boldsymbol{P}^{-1}\boldsymbol{Q} = [q_{ij}]_{0\le i,j\le N-1}$ and $\boldsymbol{P}^{-1}\boldsymbol{R} = [r_{ij}]_{0\le i,j\le N-1}$.

3.2. Newton-Raphson method for CHVM with DIRK time-integrators

Using the notation from Section 2.6, we denote the solution vector by $\boldsymbol{w}$ and write (3.2) compactly as:

$$\frac{d\boldsymbol{w}}{dt} = \boldsymbol{F}(\boldsymbol{w},\ t) \ , \tag{3.3}$$

where the generic dependence in t is added to account for potential inclusion of terms due to boundary data. Combining (3.3) with an implicit time-integrator requires solving nonlinear systems; and we derive the details in the case of Diagonally Implicit Runge-Kutta (DIRK) methods here. Let the Butcher tableau of a chosen DIRK be given in Table 2. Following the notation of Appendix A, the k^{th}-stage solution $\boldsymbol{w}^{(k)}$, $1 \le k \le s$ are computed consecutively by:

$$\boldsymbol{w}^{(k)} = \boldsymbol{z}^{(k)} + a_{kk}\Delta t\boldsymbol{F}(\boldsymbol{w}^{(k)},\ t_n + c_k\Delta t) \ , \tag{3.4}$$

Table 2: The Butcher tableau of a DIRK method.

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

where $\boldsymbol{z}^{(k)} = \boldsymbol{w}^n + \sum_{1 \le l < k} a_{kl} \Delta t \boldsymbol{F}(\boldsymbol{w}^{(l)},\ t_n + c_l \Delta t)$ is known while computing $\boldsymbol{w}^{(k)}$.

Each (3.4) is a nonlinear system that is solved by the Newton-Raphson method. Note that the Jacobian matrix involved in the Newton iterations can easily be assembled from the flux Jacobian $\boldsymbol{j}(\boldsymbol{u})$, as described briefly below. Rewriting (3.4) as finding the zero of $\mathscr{F}(\boldsymbol{w}) = \mathbf{0}$, where:

$$\mathscr{F}(\boldsymbol{w}) \overset{\text{def}}{=\!=} \boldsymbol{w} - \boldsymbol{z}^{(k)} - a_{kk} \Delta t \boldsymbol{F}(\boldsymbol{w},\ t_n + c_k \Delta t)\ ; \tag{3.5}$$

then the Jacobian of $\mathscr{F}$ is given by $\partial \mathscr{F} / \partial \boldsymbol{w} = \boldsymbol{I} - a_{kk} \Delta t \boldsymbol{J}(\boldsymbol{w}, t_n + c_k \Delta t)$.

To demonstrate the assembly of $\boldsymbol{J}$, let us assume periodic boundary conditions and write $\boldsymbol{w} = [\overline{\boldsymbol{u}}^t_{1/2} \ \cdots \ \overline{\boldsymbol{u}}^t_{N-1/2} \ \boldsymbol{u}^t_0 \ \cdots \ \boldsymbol{u}^t_{N-1}]^t \in \mathbb{R}^{2Nd}$ as before. Then $\boldsymbol{J} \in \mathbb{R}^{2Nd \times 2Nd}$ has the block structure:

$$\boldsymbol{J} = -\frac{1}{h} \begin{bmatrix} \mathbf{0} & \boldsymbol{J}_{cn} \\ \boldsymbol{J}_{nc} & \boldsymbol{J}_{nn} \end{bmatrix}, \tag{3.6}$$

where each block is $Nd \times Nd$ as given below.

Writing $\boldsymbol{j}(\boldsymbol{u}_j)$ as $\boldsymbol{j}_j$ for simplicity, then $\boldsymbol{J}_{cn}$ and $\boldsymbol{J}_{nc}$ are given by:

$$\boldsymbol{J}_{cn} = \begin{bmatrix} -\boldsymbol{j}_0 & \boldsymbol{j}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{j}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\boldsymbol{j}_{N-2} & \boldsymbol{j}_{N-1} \\ \boldsymbol{j}_0 & \mathbf{0} & \cdots & \mathbf{0} & -\boldsymbol{j}_{N-1} \end{bmatrix}, \ \boldsymbol{J}_{nc} = \begin{bmatrix} q_{0,0}\boldsymbol{j}_0 & \cdots & q_{0,N-1}\boldsymbol{j}_{N-1} \\ \vdots & \ddots & \vdots \\ q_{N-1,0}\boldsymbol{j}_0 & \cdots & q_{N-1,N-1}\boldsymbol{j}_{N-1} \end{bmatrix}, \tag{3.7}$$

for q_{ij} see the end of Section 3.1. Computing $\boldsymbol{J}_{nn}$ is slightly more complicated as $\boldsymbol{j}(\boldsymbol{u})$ is generally nonlinear:

$$\boldsymbol{J}_{nn} = \begin{bmatrix} r_{0,0}\boldsymbol{j}_0 & \cdots & r_{0,N-1}\boldsymbol{j}_{N-1} \\ \vdots & \ddots & \vdots \\ r_{N-1,0}\boldsymbol{j}_0 & \cdots & r_{N-1,N-1}\boldsymbol{j}_{N-1} \end{bmatrix} + h \begin{bmatrix} \frac{\partial \boldsymbol{j}(\boldsymbol{u}_0)}{\partial \boldsymbol{u}} : [\mathcal{D}_x \boldsymbol{u}]_0 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{\partial \boldsymbol{j}(\boldsymbol{u}_{N-1})}{\partial \boldsymbol{u}} : [\mathcal{D}_x \boldsymbol{u}]_{N-1} \end{bmatrix}. \tag{3.8}$$

Here $\bullet : \bullet$ denotes the contraction between the last index of a third-order tensor and a vector.

Remark. Modification for the case of Dirichlet boundary condition, such as $\boldsymbol{u}_0(t) \equiv \boldsymbol{g}(t)$ at the left boundary, is straightforward. To begin, we replace the corresponding d entries of the nonlinear function $\mathscr{F}$ with $\boldsymbol{u}_0 - \boldsymbol{g}(t_n + c_k\Delta t)$. To implement the matrix $\boldsymbol{J}$, we first replace the constants q_{ij} of $\boldsymbol{J}_{nc}$ in (3.7) and r_{ij} of $\boldsymbol{J}_{nn}$ in (3.8) by the entries of $\boldsymbol{SP}^{-1}\boldsymbol{Q}$ and $\boldsymbol{SP}^{-1}\boldsymbol{R}$, respectively, see (2.39); next the rows corresponding to $\boldsymbol{u}_0$ are zeroed out.

For nonlinear systems, occasionally only part of or a combination of $\boldsymbol{u}$ is enforced at a boundary, depending on the directions of the characteristics; we shall demonstrate the treatment of such boundaries in Section 3.3.2.

Lastly, as we adopt an L-stable time-integrators in this work, the stability of the method does not dependent on the size of Δt, at least in the linear case. Nevertheless, we quantify the time-step size as usual using the Courant number:

$$\Delta t = \frac{\alpha_{\text{cfl}} h}{\max\left(\max_j \lambda_{\max}(\boldsymbol{j}(\overline{\boldsymbol{u}}_{j+1/2})),\ \max_j \lambda_{\max}(\boldsymbol{j}(\boldsymbol{u}_j)),\ \varepsilon\right)} , \tag{3.9}$$

where $\lambda_{\max}$ denotes the largest absolute value of all eigenvalues of a matrix, α_{cfl} is a user specified parameter, and $\varepsilon > 0$ is a very small number that prevents division-by-zero.

3.3. *Examples of nonlinear problems and boundary conditions*

We illustrate CHVM for nonlinear equations with two examples in this section as well as the incorporation of various boundary conditions.

3.3.1. *The Burgers' equation*

The scalar Burgers' equation is given by:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0 , \quad (x,\ t) \in [0,\ 1] \times [0,\ T] . \tag{3.10}$$

Hence following the general notations given earlier:

$$d = 1 , \quad f(u) = \frac{1}{2}u^2 , \quad j(u) = u , \quad \frac{\partial j(u)}{\partial u} = 1 , \quad \lambda_{\max}(j(u)) = |u| .$$

Enforcing the Dirichlet boundary condition follows the same procedure as described in the remark near the end of Section 3.2.

3.3.2. The Euler equations

A model equation for nonlinear system of conservation laws is given by the Euler equations that describes the dynamics of inviscid compressible fluids:

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ (E + p)v \end{bmatrix}, \tag{3.11}$$

where ρ, v, E, and p are density, velocity, total energy density, and pressure, respectively. The system (3.11) has one more unknown than equations, and it is closed by the ideal gas equation of state:

$$p = (\gamma - 1)\rho e \,, \tag{3.12}$$

with γ being the constant specific heat capacity ratio and $e = (E - \rho v^2/2)/\rho$ the specific internal energy density. We take $\gamma = 1.4$ throughout the remainder of the paper. The flux Jacobian matrix and its largest eigenvalue are thus given by:

$$\boldsymbol{j}(\boldsymbol{u}) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}v^2 & (3-\gamma)v & \gamma - 1 \\ \frac{\gamma-1}{2}v^3 - v\frac{E+p}{\rho} & \frac{E+p}{\rho} - (\gamma-1)v^2 & \gamma v \end{bmatrix}, \quad \lambda_{\max}(\boldsymbol{j}(\boldsymbol{u})) = |v| + c_s \,,$$

where $c_s = \sqrt{\gamma p/\rho}$ is the speed of sound.

For non-periodic boundary conditions, it is usually the case that only part or a combination of the boundary data is enforced, and we demonstrate the procedure here using the example of a wall boundary condition at the left end point:

$$v(0,\ t) = 0 \,. \tag{3.13}$$

In this case, we set the second entry (momentum) of $\boldsymbol{u}_0$ to zero; and to this end will first construct $\boldsymbol{J}$ as described in the remark near the end of Section 3.2 and then zero out the row corresponding to the momentum of the first nodal value $\boldsymbol{u}_0$.

3.4. A Gauss-Seidel type low-pass filter

It is well-known that discontinuity generally occurs in finite time for nonlinear conservation laws, even if the initial data is smooth. In this section, we describe a Gauss-Seidel type low-pass filter for hybrid data. To this end, let us denote the filtered cell-average solutions and nodal solutions for a scalar problem[3] by $\hat{\bar{u}}_{j+1/2}$ and $\hat{u}_j$, respectively, then the Gauss-Seidel filter is given in matrix form by:

$$\begin{bmatrix} \boldsymbol{M}_{l,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{l,nc} & \boldsymbol{M}_{l,nn} \end{bmatrix} \begin{bmatrix} \hat{\overline{\boldsymbol{u}}} \\ \hat{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{r,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{r,nc} & \boldsymbol{M}_{r,nn} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} . \tag{3.14}$$

A thorough analysis of low-pass filters for hybrid-data is beyond the scope of this work; and we will present the result in another paper.

3.4.1. Periodic case

In the case of periodic problems, $\overline{\boldsymbol{u}}$ and $\boldsymbol{u}$ are given by (2.12) and all rows of matrices in (3.14) are determined from the filtering formula at local cells or grid points. Particular for the cell-averages, each row of $\boldsymbol{M}_{l,cc}\hat{\overline{\boldsymbol{u}}} = \boldsymbol{M}_{r,cc}\overline{\boldsymbol{u}}$ is obtained from a sixth-order compact-stencil filter that is motivated by Lele [Eqn. (C.2.8)][1]:

$$\begin{aligned} \hat{\bar{u}}_{j+1/2} + \frac{3}{10}\left(\hat{\bar{u}}_{j-3/2} + \hat{\bar{u}}_{j+5/2}\right) = \frac{1}{2}\bar{u}_{j+1/2} + \frac{3}{8}\left(\bar{u}_{j-1/2} + \bar{u}_{j+3/2}\right) + \\ \frac{3}{20}\left(\bar{u}_{j-3/2} + \bar{u}_{j+5/2}\right) + \frac{1}{40}\left(\bar{u}_{j-5/2} + \bar{u}_{j+7/2}\right) \ ; \end{aligned} \tag{3.15}$$

this is the highest-order that one can achieve with the given stencil. For the nodal variables, each row of $\boldsymbol{M}_{l,nc}\hat{\overline{\boldsymbol{u}}} + \boldsymbol{M}_{l,nn}\hat{\boldsymbol{u}} = \boldsymbol{M}_{r,nc}\overline{\boldsymbol{u}} + \boldsymbol{M}_{r,nn}\boldsymbol{u}$ is obtained from a one-parameter family of optimally eight-order filters[4]:

$$\begin{aligned} &\hat{u}_j - 105\mu\left(\hat{\bar{u}}_{j-1/2} + \hat{\bar{u}}_{j+1/2}\right) + \frac{210\mu - 1}{2}\left(\hat{u}_{j-1} + \hat{u}_{j+1}\right) \\ =\ &(1 - 108\mu)u_j - \frac{25\mu}{2}\left(\bar{u}_{j-1/2} + \bar{u}_{j+1/2}\right) + \frac{114\mu - 1}{2}\left(u_{j-1} + u_{j+1}\right) + \\ &\frac{25\mu}{2}\left(\bar{u}_{j-3/2} + \bar{u}_{j+3/2}\right) - 3\mu\left(u_{j-2} + u_{j+2}\right) \ , \end{aligned} \tag{3.16}$$

where $0 < \mu \leq 1/140$ is a tunable parameter; generally speaking, the larger the value of μ, the better suppression of high frequencies the filter achieves. In fact, the transfer function of (3.16) is given by:

$$\mathcal{F}(\theta) = \frac{1 - 108\mu + (114\mu - 1)\cos\theta - 6\mu\cos 2\theta - 25\mu\frac{\sin\theta}{\theta} + 25\mu\frac{\sin 2\theta - \sin\theta}{\theta}}{1 + (210\mu - 1)\cos\theta - 210\mu\frac{\sin\theta}{\theta}} \ . \tag{3.17}$$

[3] The system case is handled similarly.

[4] Requiring further accuracy on the same stencil reduces the filter to the identity one.

Note that $\mathcal{F}(\theta)$ is well-defined for all $0 < \theta \leq 2\pi$, where the range is $[0,\ 2\pi]$ instead of $[0,\ \pi]$ as in conventional low-pass filters, because there are two degrees of freedoms per grid cell; and one can easily see $\mathcal{F}(0) = 1$ by the L'Hôpital's rule. Sample $\mathcal{F}$-curves corresponding to different values of μ are given Figure 3.1. We adopt a conservative choice and set $\mu = 1/140$ in all numerical tests in this work.

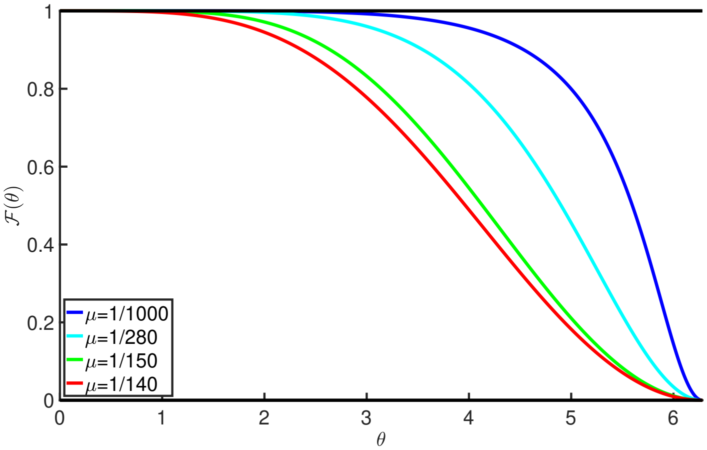


Figure 3.1: The transfer functions of the filter (3.17) given by different μ's in the range $(0,\ 1/140]$.

3.4.2. Non-periodic case

In the case of non-periodic boundary conditions, the filters for interior nodes and cells are the same as before; but modifications are needed near the two boundaries. Let us write $\overline{\boldsymbol{u}} = [\overline{u}_{1/2}\ \cdots\ \overline{u}_{N-1/2}]^t \in \mathbb{R}^N$ and $\boldsymbol{u} = [u_0\ u_1\ \cdots\ u_N]^t \in \mathbb{R}^{N+1}$, then the dimensions of the matrices in (3.14) are given by:

$$\boldsymbol{M}_{l,cc},\ \boldsymbol{M}_{r,cc} \in \mathbb{R}^{N\times N}\ ; \quad \boldsymbol{M}_{l,nc},\ \boldsymbol{M}_{r,nc} \in \mathbb{R}^{(N+1)\times N}\ ; \quad \boldsymbol{M}_{l,nn},\ \boldsymbol{M}_{r,nn} \in \mathbb{R}^{(N+1)\times(N+1)}\ .$$

The strategy in [1] is adopted to construct explicit fifth-order filters near the boundaries; particularly for the cell-averaged values at the left boundary, the first three rows of $\boldsymbol{M}_{l,cc}$ and $\boldsymbol{M}_{r,cc}$ are modified according to:

$$\hat{\overline{u}}_{1/2} = \frac{31}{32}\overline{u}_{1/2} + \frac{5}{32}\overline{u}_{3/2} - \frac{5}{16}\overline{u}_{5/2} + \frac{5}{16}\overline{u}_{7/2} - \frac{5}{32}\overline{u}_{9/2} + \frac{1}{32}\overline{u}_{11/2}\ , \tag{3.18a}$$

$$\hat{\overline{u}}_{3/2} = \frac{1}{32}\overline{u}_{1/2} + \frac{27}{32}\overline{u}_{3/2} + \frac{5}{16}\overline{u}_{5/2} - \frac{5}{16}\overline{u}_{7/2} + \frac{5}{32}\overline{u}_{9/2} - \frac{1}{32}\overline{u}_{11/2}\ , \tag{3.18b}$$

$$\hat{\overline{u}}_{5/2} = -\frac{1}{32}\overline{u}_{1/2} + \frac{5}{32}\overline{u}_{3/2} + \frac{11}{16}\overline{u}_{5/2} + \frac{5}{16}\overline{u}_{7/2} - \frac{5}{32}\overline{u}_{9/2} + \frac{1}{32}\overline{u}_{11/2}\ ; \tag{3.18c}$$

the last three rows corresponding to the right boundary are modified symmetrically. A two-step procedure is used to construct the filters at nodes near boundaries. Using the left boundary as an example, we first modify the top two rows of $\boldsymbol{M}_{l,nc}$, $\boldsymbol{M}_{l,nn}$, $\boldsymbol{M}_{r,nc}$, and $\boldsymbol{M}_{r,nn}$ according to:

$$\hat{u}_0 = \frac{9}{10}u_0 + \frac{1}{3}\overline{u}_{1/2} - \frac{3}{5}u_1 + \frac{19}{30}\overline{u}_{3/2} - \frac{3}{10}u_2 + \frac{1}{30}\overline{u}_{5/2}\ , \tag{3.19a}$$

$$\hat{u}_1 = -\frac{1}{10}u_0 + \frac{1}{3}\overline{u}_{1/2} + \frac{2}{5}u_1 + \frac{19}{30}\overline{u}_{3/2} - \frac{3}{10}u_2 + \frac{1}{30}\overline{u}_{5/2}\ , \tag{3.19b}$$

which are unique filters that are fifth-order accurate and satisfy $\mathrm{Re}\mathcal{F}(2\pi) = 0$ as suggested by [14, 15] for non-symmetric filters, given the stencil used.

In the second step, one replace (3.19a) with $\hat{u}_0 = u_0$ if a Dirichlet boundary condition is provided at this location. The nodes near the right boundary are handled symmetrically.

3.4.3. Selection of number of passes

In practice, one typically needs to decide on the number of passes (denoted by p) of the selected filter, that is, once the filtering matrices in (3.14) are computed, the filtered values are actually computed as:

$$\begin{bmatrix} \hat{\overline{\boldsymbol{u}}} \\ \hat{\boldsymbol{u}} \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{M}_{l,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{l,nc} & \boldsymbol{M}_{l,nn} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{M}_{r,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{r,nc} & \boldsymbol{M}_{r,nn} \end{bmatrix} \right)^p \begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} . \tag{3.20}$$

In our experience, $p = 1$ is sufficient for almost all computations unless the grid is extremely coarse. On the safe side we adopt $p = 4$ for all examples presented; note that the additional computational cost is negligible as one can compute the filtering matrix $\left(\begin{bmatrix} \boldsymbol{M}_{l,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{l,nc} & \boldsymbol{M}_{l,nn} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{M}_{r,cc} & \boldsymbol{0} \\ \boldsymbol{M}_{r,nc} & \boldsymbol{M}_{r,nn} \end{bmatrix} \right)^p$ once and store it for all later uses.

Remark. Like all linear filters, while our filter is effective in suppressing node-to-node oscillations it does not remove all spurious oscillations, especially when an overshoot or undershoot spreads across several cells. Nevertheless, it suffices our purpose for smooth or weakly discontinuous problems, like those in most acoustics applications. For hyperbolic conservation laws with strong discontinuities, numerous strategies have been proposed in the linterature to remove as much overshoot/undershoot as possible for a central compact scheme, such as blending the central differential operator with an upwind one [9] and using a WENO-type limiter [4, 16].

4. Numerical examples

In this section we present numerous tests to assess the numerical performance of the proposed method. Particularly, Section 4.1 focuses on linear advection problems with various boundary conditions and Section 4.2 concentrates on non-linear problems including benchmark shock tube tests for the Euler equations as well as a multi-scale shock/sinusoidal flow interaction problem. The Courant number $\alpha_{\text{cfl}} = 2.0$ is used for all computations unless otherwise noted; to our experience CHVM with DIRK5 is stable with any time step size.

4.1. Linear advection equations

In the first group of tests, we consider the linear advection equation in Section 2. Particularly, we verify the order of accuracy of CHVM when the solutions are smooth. For this purpose, the L_1-norm of the numerical errors at the terminal time $t = T$ is computed as:

$$\text{err}(u) = \frac{h}{2}\left|u_0^{n_T} - u_{\text{ref}}(x_0, T)\right| + \sum_{j=1}^{N-1} h\left|u_j^{n_T} - u_{\text{ref}}(x_j, T)\right| + \frac{h}{2}\left|u_N^{n_T} - u_{\text{ref}}(x_N, T)\right|, \quad (4.1)$$

$$\text{err}(\overline{u}) = \sum_{j=1}^{N} h\left|\overline{u}_{j-1/2}^{n_T} - \frac{1}{h}\int_{(j-1)h}^{jh} u_{\text{ref}}(x, T)dx\right|. \quad (4.2)$$

Here h is the cell size, u_{ref} is the reference solution, and n_T is the total number of time steps for the computation reaching $t = T$. The reference solution is determined either analytically or by solving the same problem on a much finer grid.

4.1.1. A Cauchy problem: Fixed Courant number computations

In this test and the next, we solve the advection problem (2.2) with $c = 1$ on the domain $(x, t) \in [0, 1]\times[0, 1]$, the periodic boundary condition $u(0, t) = u(1, t)$ for all $0 \le t \le 1$, and a composite wave initial condition:

$$u(x, 0) = \sin(2\pi x) + \cos(4\pi x), \quad 0 \le x \le 1\ . \quad (4.3)$$

The reference solution at $T = 1$ is the same as the initial condition $u_{\text{ref}}(x, 1) = u(x, 0)$, which is used to compute the L_1-errors reported in Table 3. Here we use a sequence of six uniform grids with the number of cells ranging from 8 to 256. As the spatial discretization method (CHVM) is sixth-order accurate and the time-integrator (DIRK5) is fifth-order, we expect overall fifth-order of convergence, which is indeed the case whether the filter is applied or not.

Table 3: The L_1-errors by CHVM with or without the filter to solve a Cauchy problem for advection equation with fixed Courant number in Section 4.1.1.

	CHVM without filter				CHVM with filter			
N	Error in u	Order	Error in $\overline{u}$	Order	Error in u	Order	Error in $\overline{u}$	Order
8	3.571e-1		2.607e-1		5.318e-1		6.366e-1	
16	2.561e-2	3.80	2.685e-2	3.28	1.109e-1	2.26	1.109e-1	2.52
32	1.132e-3	4.50	1.137e-3	4.56	3.582e-3	4.95	3.535e-3	4.97
64	3.850e-5	4.88	3.853e-5	4.88	1.124e-4	4.99	1.119e-4	4.98
128	1.229e-6	4.97	1.229e-6	4.97	3.516e-6	5.00	3.512e-6	4.99
256	3.861e-8	4.99	3.861e-8	4.99	1.099e-7	5.00	1.099e-7	5.00

4.1.2. A Cauchy problem: Fixed time step size computations

In the second test, we consider the same problem as before but fix the time step size to $\Delta t = 0.01$; to this end the reference solution is computed by using CHVM without filter on a grid of 1280 uniform cells. Using the same time step size and the same sequence of grids, the computed L_1-errors by CHVM with and without the filter are reported in Table 4.

Table 4: The L_1-errors by CHVM with or without the filter to solve a Cauchy problem for advection equation with fixed time step size in Section 4.1.2.

	CHVM without filter				CHVM with filter			
N	Error in u	Order	Error in $\overline{u}$	Order	Error in u	Order	Error in $\overline{u}$	Order
8	1.228e-3		1.627e-3		8.063e-1		7.084e-1	
16	2.294e-5	5.74	2.506e-5	6.02	5.270e-1	0.61	5.514e-1	0.36
32	3.662e-7	5.97	3.942e-7	5.99	1.540e-2	5.10	1.557e-2	5.15
64	5.871e-9	5.96	6.109e-9	6.01	2.319e-4	6.05	2.324e-4	6.07
128	9.397e-11	5.97	9.493e-11	6.01	3.579e-6	6.02	3.580e-6	6.02
256	1.482e-12	5.99	1.483e-12	6.00	5.573e-8	6.00	5.573e-8	6.01

4.1.3. An initial boundary value problem.

The last accuracy test for the advection equation concerns the enforcement of boundary conditions as well as the boundary filters. Particularly we consider essentially the same problem as before, except that instead of the periodic boundary condition we enforce the following Dirichlet condition at the left boundary:

$$u(0,t) = \sin(2\pi t) + \cos(4\pi t)\ . \tag{4.4}$$

The exact solution at $T = 1$ remains the same as the initial condition; and the L_1-errors are reported in Table 5 for CHVM both with and without the filters.

Table 5: The L_1-errors by CHVM with or without the filter to solve an initial boundary value problem for the advection equation with fixed Courant number in Section 4.1.3.

	CHVM without filter				CHVM with filter			
N	Error in u	Order	Error in $\overline{u}$	Order	Error in u	Order	Error in $\overline{u}$	Order
8	2.028e-1		1.254e-1		2.690e-1		2.796e-1	
16	1.289e-2	3.98	1.203e-2	3.38	4.742e-2	2.50	4.874e-2	2.52
32	5.514e-4	4.55	5.333e-4	4.50	1.771e-3	4.74	1.785e-3	4.77
64	1.914e-5	4.85	1.871e-5	4.83	5.578e-5	4.99	5.570e-5	5.00
128	6.197e-7	4.95	5.945e-7	4.98	1.620e-6	5.11	1.614e-6	5.11
256	1.939e-8	5.00	1.789e-8	5.05	5.799e-8	4.80	5.786e-8	4.80

From Tables 3– 5, we verify the numerical order of accuracy of the proposed method, and observe that the application of filter leads to slightly larger errors but it does not deteriorate the formal order of accuracy.

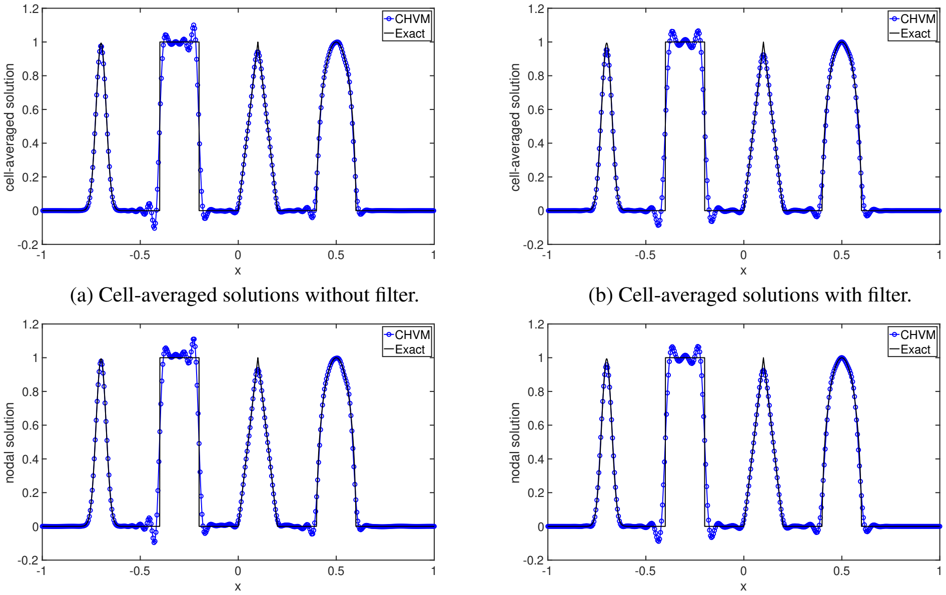
4.1.4. Advection of Gaussian, square, sharp triangle, and half ellipse waves

In the last test of linear problems, we consider the advection of an initial wave that is composed of a smooth but narrow Gaussians, a square wave, a sharp triangle wave, and a half ellipse [17]. The governing equation is again given by (2.2) with $c = 1$ on the computational domain $-1 \le x \le 1$ and the time interval $t \in [0,\ 8]$, with the periodic boundary condition and the initial data given below:

$$u(x,0) = \begin{cases} \frac{1}{6}\left[G_{\beta,z-\delta}(x) + 4G_{\beta,z}(x) + G_{\beta,z+\delta}(x)\right], & -0.8 \le x \le -0.6\ ; \\ 1, & -0.4 \le x \le -0.2\ ; \\ 1 - |10(x-0.1)|\ , & 0 \le x \le 0.2\ ; \\ \frac{1}{6}\left[L_{\alpha,a-\delta}(x) + 4L_{\alpha,a}(x) + L_{\alpha,a+\delta}(x)\right], & 0.4 \le x \le 0.6\ ; \\ 0, & \text{otherwise}\ . \end{cases}, \tag{4.5}$$

where $G_{\beta,z}(x) \overset{\text{def}}{=\!=} e^{-\beta(x-z)^2}$ and $L_{\alpha,a}(x) \overset{\text{def}}{=\!=} \sqrt{\max(1-\alpha^2(x-a)^2, 0)}$, and the constants are taken as $a = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \log(2)/(36\delta^2)$.

Using a uniform grid with 400 cells, the CHVM solutions without and with the filter are plotted in the left and the right columns of Figure 4.1, respectively.



(a) Cell-averaged solutions without filter.

(b) Cell-averaged solutions with filter.

(c) Nodal solutions without filter.

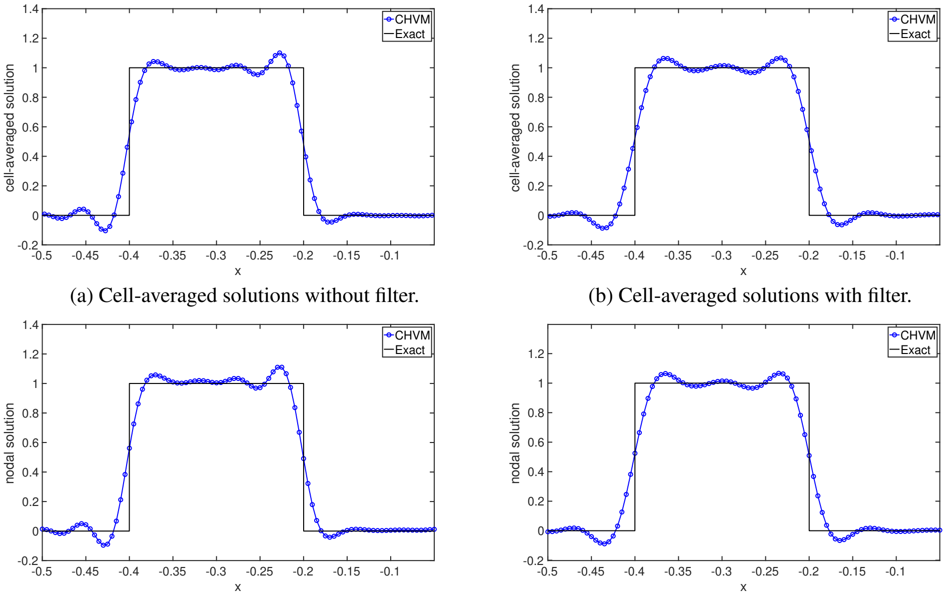
(d) Nodal solutions with filter.

Figure 4.1: Numerical solutions at $T = 8.0$ obtained by CHVM to solve (4.5).

We can see that using the filter slightly decreases the magnitudes of the oscillations in the upstream direction of discontinuities and it produces a more symmetric profile comparing to the CHVM solution without a filter, particularly as seen in the local view provided in Figure 4.2.

To compare the performance of CHVM with existing compact schemes, we solve the same problem with LSPS and CCS-T6 (see Section 2.5) and plot the numerical solutions in Figure 4.3. A uniform grid with 200 cells are used for all computations and only nodal values are plotted, that is, cell averages by CHVM and mid-cell solutions by CCS-T6 are omitted from the figures. Furthermore, all three methods are used without application of any filter. From the plots, it seems that all three methods lead to Gibbs type oscillation with similar magnitudes, but CHVM introduces much less phase error than the others.

A closer investigation of the solutions to the left of the Gaussians (see the dashed box region of Figure 4.3a as well as Figure 4.4) demonstrates that CHVM and CCS-T6 provide better stability than LSPS, as the latter causes node-to-node oscillations in this region.



(a) Cell-averaged solutions without filter.

(b) Cell-averaged solutions with filter.

(c) Nodal solutions without filter.

(d) Nodal solutions with filter.

Figure 4.2: Local view around the square wave of the CHVM solutions at $T = 8.0$ for (4.5).

4.2. Nonlinear problems

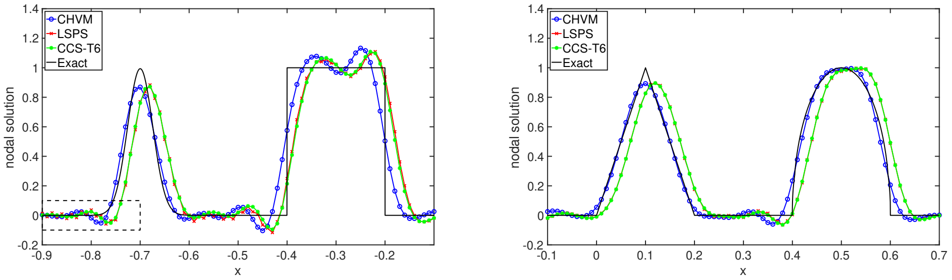
The performance of CHVM for solving nonlinear problems is assessed in this section and we consider the scalar Burgers' equation and the system of Euler equations. In particular, we will verify the order of accuracy for nonlinear systems and will evaluate the impact of the wall boundary condition on the formal order of accuracy. Furthermore, since discontinuities generally develop for nonlinear conservation laws even when the initial data is smooth, we shall illustrate the necessity of applying the filters by various tests given below.

4.2.1. The Burgers' equation: Development of an N-wave

First we consider a nonlinear scalar problem governed by the 1D Burgers' equation (3.10). The computational domain is [0, 1] and the periodic boundary condition and initial data are given by:

$$\begin{cases} u(0,t) = u(1,t), & t \geq 0\ , \\ u(x,0) = u_0(x) = \sin(2\pi x) + 1, & x \in [0,\ 1]\ . \end{cases} \tag{4.6}$$

This problem admits a smooth solution until $t = 1/(2\pi) \approx 0.1592$, at which moment a shock wave originates at $x = (1+\pi)/(2\pi)$; the location of the shock wave



(a) Numerical solutions for the left two waveforms.

(b) Numerical solutions for the right two waveforms.

Figure 4.3: Nodal solution comparison among CHVM without filter, LSPS, and CCS-T6 for (4.5).

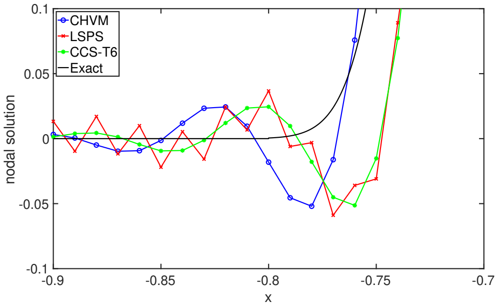


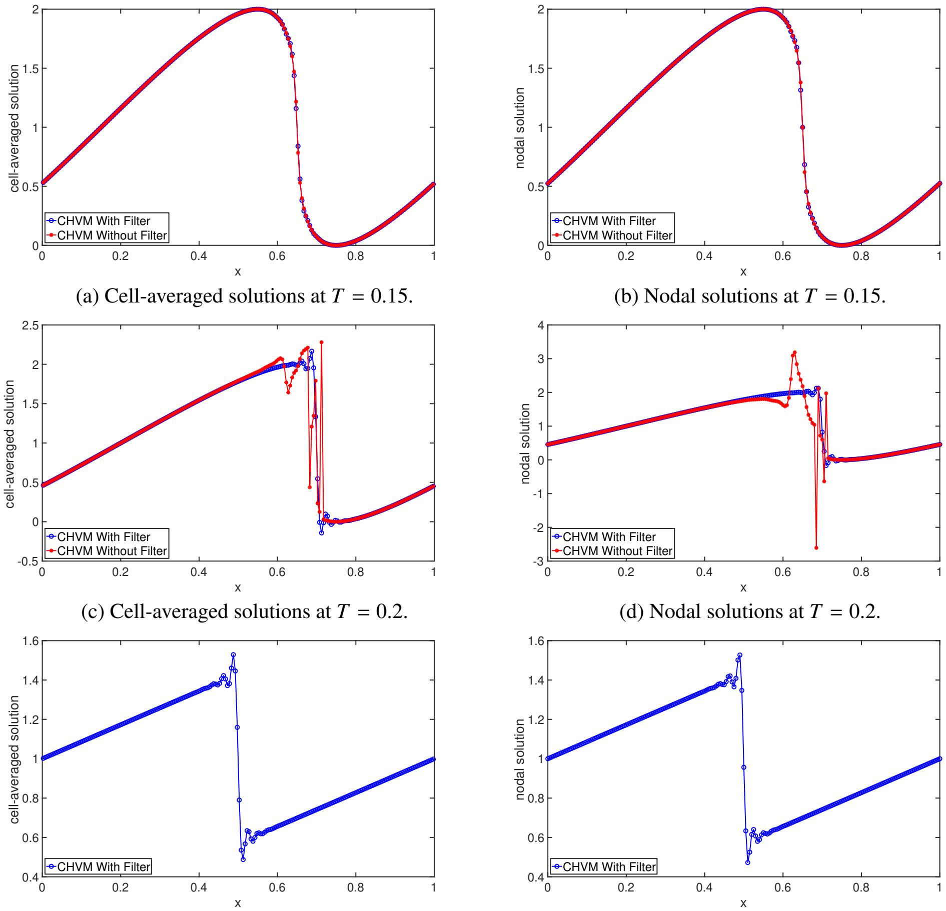
Figure 4.4: Local view of the solutions by CHVM without filter, LSPS, and CCS-T6 for (4.5).

at $t > 1/(2\pi)$ is $(t + 1/2)$. The numerical solutions are consistent with the analysis; indeed, using a 200-cell grid, the CHVM solutions at $T = 0.15$ (Figures 4.5a and 4.5b) demonstrate a smooth profile, whereas we start to see the shock wave development in the solutions at $T = 0.2$ (Figures 4.5c and 4.5d) and the formation of an N-wave is clearly observed at $T = 1.0$ (Figures 4.5e and 4.5f), at least when the filter is applied.

Furthermore, we see from Figure 4.5 that applying the filter or not has almost negligible impact on the quality of the numerical solutions when the curves are smooth ($T = 0.15$); but not applying the filter leads to inconsistent numerical solution shortly after the shock develops ($T = 0.2$) and eventually breaks down the computation in the long run ($T = 1.0$).

4.2.2. The Euler equations: Wave collision with periodic boundary conditions.

Here we consider the nonlinear Euler equations (3.11) with specific heat ratio $\gamma = 1.4$ on the computational domain $\Omega = [-2, 2]$ and periodic boundary condi-



(a) Cell-averaged solutions at $T = 0.15$.

(b) Nodal solutions at $T = 0.15$.

(c) Cell-averaged solutions at $T = 0.2$.

(d) Nodal solutions at $T = 0.2$.

(e) Cell-averaged solutions at $T = 1.0$.

(f) Nodal solutions at $T = 1.0$.

Figure 4.5: CHVM solutions to the Burgers' equation in Section 4.2.1 using 200 cells.

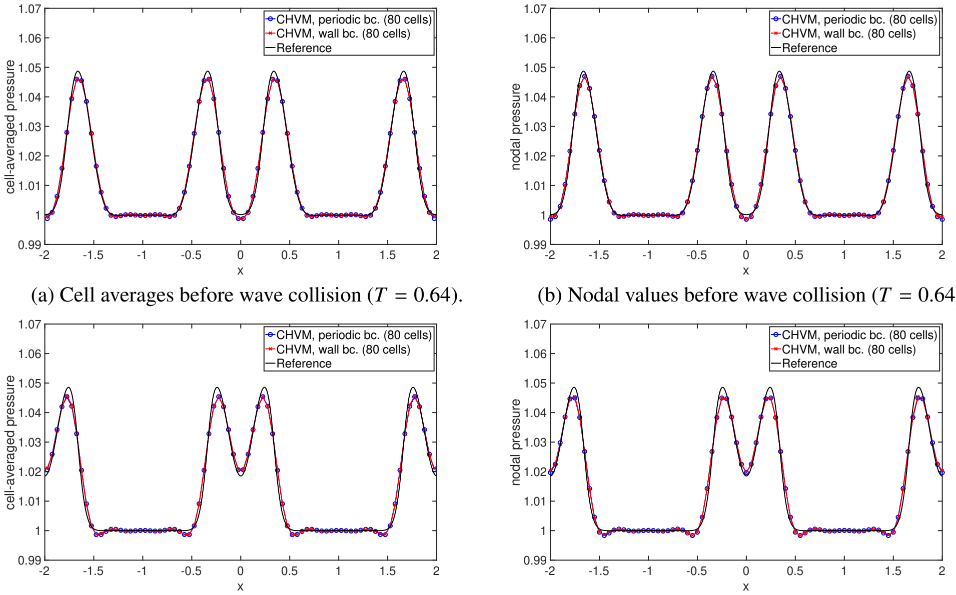
tions $\boldsymbol{u}(-2,t) = \boldsymbol{u}(2,t)$ and the following initial data:

$$\begin{cases} \rho(x,0) = 1.4 + 1.4\varepsilon B(x)\ , \\ v(x,0) = 0.0\ , \qquad\qquad\qquad \forall x \in [-2,\ 2]\ . \\ p(x,0) = 1 + \varepsilon B(x)\ , \end{cases} \tag{4.7}$$

Here $\varepsilon = 0.1$ and $B(x)$ describes two symmetric bumps and it is given by:

$$B(x) = \begin{cases} \left[\frac{1}{2}(1 - \cos(2\pi(x + 0.5)))\right]^4 , & -1.5 \le x \le -0.5 \\ \left[\frac{1}{2}(1 - \cos(2\pi(x - 0.5)))\right]^4 , & 0.5 \le x \le 1.5 \\ 0.0 , & \text{otherwise.} \end{cases} \tag{4.8}$$

This test is inspired by a similar problem in [18, 19], where we modified the initial waveform to allow accuracy analysis beyond second-order. Due to the symmetry of the profiles in $B(x)$, the solution consists of two symmetric waves moving in opposite directions originating from the bump centered $x = -0.5$, and two similar waves originating from the bump centered at $x = 0.5$. The two outward moving waves will then run into each other at $x = -2$ (or equivalently, $x = 2$) when $T \approx 0.64$, see Figures 4.6a and 4.6b; and the two inward moving waves will run into each other at $x = 0$ about the same time. Hence we refer to this test as a "wave collision" problem; and it admits a smooth solution until $T = 1.2$, see sample solution curves in Figures 4.6c and 4.6d.



(a) Cell averages before wave collision ($T = 0.64$).

(b) Nodal values before wave collision ($T = 0.64$).

(c) Cell averages at $T = 1.2$.

(d) Nodal values at $T = 1.2$.

Figure 4.6: CHVM solutions to the wave collision problem in Sections 4.2.2 and 4.2.3.

We compute the numerical solutions using a sequence of five uniform grids with the number of cells ranging from 40 to 640; and a reference solution is computed using a finer grid with 2560 cells. All computations are performed with the filter enabled. Sample solution curves at different times are provided in Figure 4.6; note that we only plot solutions computed using 80 cells, as the those computed with 160 cells or more are visually indistinguishable from the reference curves. To assess the numerical convergence, the L_1-errors for the primitive variables (density, velocity, and pressure) are reported in Table 6. Note that the numerical error for all nodal variables and the cell-averaged density can be computed using formula that are similar to (4.1) and (4.2), respectively; whereas for the “cell-averaged” velocity and pressure (denoted $\overline{v}$ and $\overline{p}$ for simplicity), they are first computed from the cell-averaged conservative variables as shown below:

$$\overline{v} = \frac{\overline{\rho v}}{\overline{\rho}}\ , \quad \overline{p} = (\gamma - 1)\left(\overline{E} - \frac{1}{2}\frac{\overline{\rho v}^2}{\overline{\rho}}\right)\ , \tag{4.9}$$

and then compared to the reference values to calculate the numerical errors.

Table 6: The L_1-errors by CHVM with filter to solve a smooth wave collision problem with periodic boundary conditions for the Euler equation in Section 4.2.2.

Cell-averaged values						
	Density		Velocity		Pressure	
N	Error in $\overline{\rho}$	Order	Error in $\overline{v}$	Order	Error in $\overline{p}$	Order
40	2.199e-2		1.127e-2		2.224e-2	
80	5.335e-3	2.04	3.264e-3	1.79	4.310e-3	2.37
160	4.426e-4	3.59	2.632e-4	3.63	3.968e-4	3.44
320	1.929e-5	4.52	1.234e-5	4.41	1.803e-5	4.46
640	5.937e-7	5.02	3.821e-7	5.01	5.567e-7	5.02

Nodal values						
	Density		Velocity		Pressure	
N	Error in ρ	Order	Error in v	Order	Error in p	Order
40	2.617e-2		1.260e-2		2.578e-2	
80	5.600e-3	2.22	3.361e-3	1.91	4.712e-3	2.45
160	4.533e-4	3.63	2.717e-4	3.63	4.080e-4	3.53
320	1.896e-5	4.58	1.213e-5	4.49	1.770e-5	4.53
640	5.937e-7	5.00	3.822e-7	4.99	5.567e-7	4.99

From Table 6 we clearly observe that when the grid is sufficiently refined, the fifth-order accuracy is obtained as expected and the usage of the filter does not impact the formal order of accuracy as in the linear case.

4.2.3. The Euler equations: Wave collision with wall boundary conditions.

In the second test for Euler equations, we consider essentially the same test as in Section 4.2.2, but apply the wall boundary conditions at both $x = -2$ and $x = 2$ instead of the periodic boundary condition. Note that due to symmetry, the two problems are mathematically equivalent to each other, see Figure 4.6; and we use the same reference solution as before to compute the L_1-errors for the primitive variables and convergence rates, as reported in Table 7. From the table, we see that our wall boundary condition enforcement does not affect the formal order of accuracy of the method.

Table 7: The L_1-errors by CHVM with filter to solve a smooth wave collision problem with wall boundary conditions for the Euler equation in Section 4.2.3.

Cell-averaged values

	Density		Velocity		Pressure	
N	Error in $\overline{\rho}$	Order	Error in $\overline{v}$	Order	Error in $\overline{p}$	Order
40	2.018e-2		9.776e-3		1.910e-2	
80	5.060e-3	2.00	3.010e-3	1.70	4.034e-3	2.24
160	4.247e-4	3.57	2.537e-4	3.57	3.788e-4	3.41
320	1.907e-5	4.48	1.218e-5	4.38	1.780e-5	4.41
640	5.930e-7	5.01	3.816e-7	5.00	5.560e-7	5.00

Nodal values

	Density		Velocity		Pressure	
N	Error in ρ	Order	Error in v	Order	Error in p	Order
40	2.488e-2		1.122e-2		2.315e-2	
80	5.308e-3	2.23	3.094e-3	1.86	4.392e-3	2.40
160	4.353e-4	3.61	2.621e-4	3.56	3.897e-4	3.49
320	1.876e-5	4.54	1.199e-5	4.45	1.750e-5	4.48
640	5.931e-7	4.98	3.818e-7	4.97	5.560e-7	4.98

4.2.4. The Euler equations: The Sod shock tube problem.

In this example and the next, we consider two Riemann problems for the Euler equations with moderate discontinuities. The current one is the well-known Sod shock tube test [20], where we consider the computational domain $\Omega = [-5,\ 5]$ and set the initial data as:

$$\begin{cases} \rho(x,0) = 1.0\ , \\ v(x,0) = 0.0\ , \quad x \in [-5,\ 0]\ ; \\ p(x,0) = 1.0 \end{cases} \qquad \begin{cases} \rho(x,0) = 0.125\ , \\ v(x,0) = 0.0\ , \qquad x \in (0,\ 5]\ . \\ p(x,0) = 0.1 \end{cases} \tag{4.10}$$

The admissible solution consists of a left-going rarefaction, a middle contact discontinuity, and a right-going shock wave. By the termination time $T = 2.0$ neither the rarefaction nor the shock wave hit the boundaries, thus the Dirichlet boundary condition is applied at both end points throughout the computations.

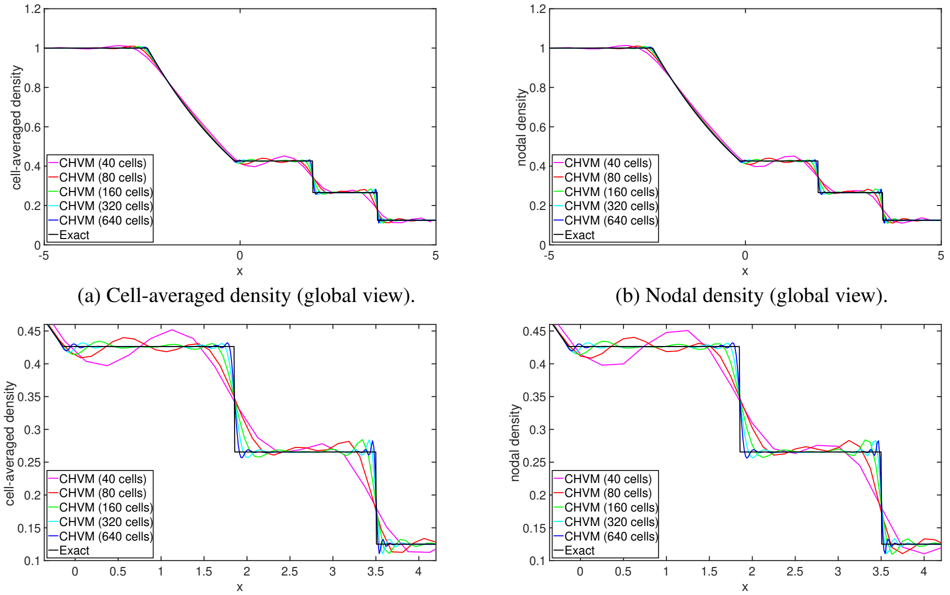
In the top row of Figure 4.7, we present the density plots computed by CHVM with filter on a sequence of four grids with number of cells ranging from 40 to 640, and compare these curves to the exact one. In the bottom row of the same figure, the zoomed-in plots around the two discontinuities are shown and we observe the overall convergence of numerical solutions towards the analytical one. Particularly, the L_1-errors are computed as before and reported in Table 8, where we observe 1st order of convergence as expected due to the discontinuities.

4.2.5. The Euler equations: The Lax shock tube problem.

Next, we consider the Lax shock tube problem [21] that contains a stronger shock than the previous one. The computational domain is again set to $\Omega = [-5,\ 5]$ and the initial condition is given by:

$$\begin{cases} \rho(x,0) = 0.445\ , \\ v(x,0) = 0.698\ , \quad x \in [-5,\ 0]\ ; \\ p(x,0) = 3.528 \end{cases} \qquad \begin{cases} \rho(x,0) = 0.5\ , \\ v(x,0) = 0.0\ , \qquad x \in (0,\ 5]\ . \\ p(x,0) = 0.571 \end{cases} \tag{4.11}$$

The problem is solved until $T = 1.5$ with Dirichlet boundary condition applied at both ends of the domain. In Figure 4.8, the density curves computed by CHVM on the same sequence of grids are plotted against the exact one, with the top row and bottom row demonstrating the global view and local view around the discontinuities, respectively. Finally, the L_1-errors are computed and provided in Table 9, from which we observe the 1st order of convergence as expected.



(a) Cell-averaged density (global view).

(b) Nodal density (global view).

(c) Cell-averaged density (local view).

(d) Nodal density (local view).

Figure 4.7: CHVM solutions (density) to the Sod shock tube problem at $T = 2.0$.

4.2.6. The Euler equations: A small-scale smooth flow interacting with a shock.

Lastly, we consider a model problem for multi-scale shock-turbulence interaction, which is motivated by the Shu-Osher test [22, 9]. In particular, a Mach 2 shock wave is moving into a smooth flow field with sinusoidal density disturbance. The computational domain is $\Omega = [-0.2,\ 1.8]$ and the initial data is:

$$\begin{cases} \rho(x,0) = 3.733333\ , \\ v(x,0) = 1.25\ , & x \leq 0\ ; \\ p(x,0) = 4.5 \end{cases} \qquad \begin{cases} \rho(x,0) = \frac{7+\sin(5\pi x)}{5}\ , \\ v(x,0) = 0.0\ , & x > 0\ . \\ p(x,0) = 1.0 \end{cases} \tag{4.12}$$

The problem is solved until $T = 0.79$, which allows Dirichlet boundary condition to be enforced at both ends of Ω with data given by their initial values.

Since no analytical solution is available, we compute a reference one on a very fine grid with 12,000 uniform cells using the second-order MUSCL method [23] built on the Roe flux [24] and the slope limiter by van Albada [25]. In Figure 4.9, the cell-averaged density computed by CHVM on a uniform 200-cell grid is plotted against the reference one. For comparison, we also plot the MUSCL solutions using either the same grid as CHVM (i.e., 200 cells) or the same number of unknowns as the CHVM computation (i.e., 400 cells). From the plots we see that

Table 8: The L_1-errors by CHVM with filter to solve the Sod shock tube problem in Section 4.2.4.

Cell-averaged values						
	Density		Velocity		Pressure	
N	Error in $\overline{\rho}$	Order	Error in $\overline{v}$	Order	Error in $\overline{p}$	Order
40	1.328e-1		4.004e-1		1.531e-1	
80	7.808e-2	0.77	2.256e-1	0.83	7.509e-2	1.03
160	3.917e-2	0.96	1.099e-1	1.04	3.535e-2	1.09
320	2.088e-2	0.90	5.395e-2	1.03	1.722e-2	1.04
640	1.042e-2	1.00	2.775e-2	0.96	8.485e-3	1.02

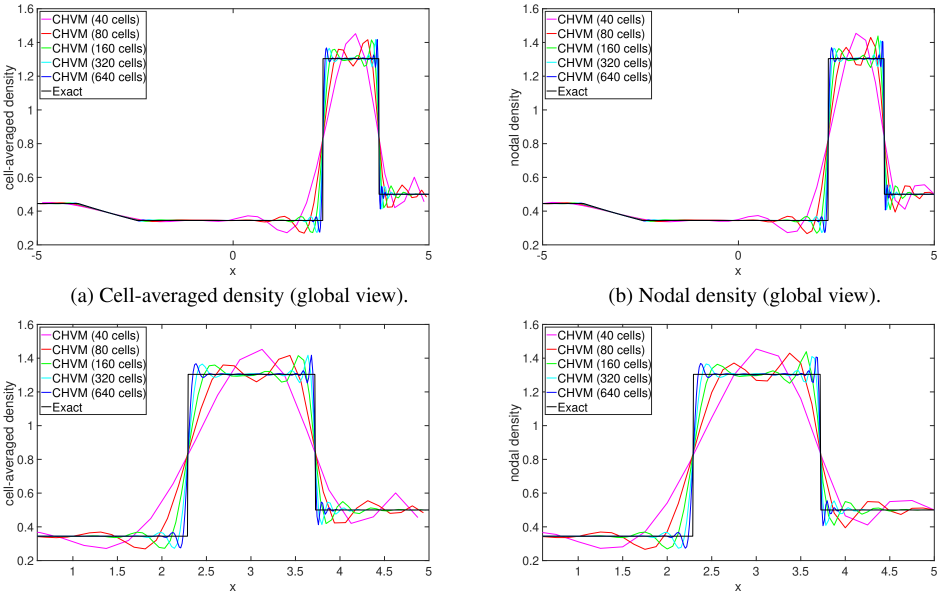
Nodal values						
	Density		Velocity		Pressure	
N	Error in ρ	Order	Error in v	Order	Error in p	Order
40	1.436e-1		3.683e-1		1.461e-1	
80	8.283e-2	0.79	2.151e-1	0.78	7.310e-2	1.00
160	4.520e-2	0.87	1.264e-1	0.77	3.840e-2	0.93
320	2.381e-2	0.92	6.304e-2	1.00	1.919e-2	1.00
640	1.278e-2	0.90	3.427e-2	0.88	1.013e-2	0.92

the CHVM solution is almost on top of the reference one in the small-scale post-shock density disturbances, while the MUSCL solution is much more diffusive at these locations whether the same grid or the same number of unknowns is used. The spurious oscillations computed by CHVM near the shock fronts are expected as our method is linear; suppressing these wiggles would require a nonlinear filter, which will be left for future investigation.

5. Conclusions

In this work we present a new central compact hybrid-variable method (CHVM) with spectral-like accuracy for one-dimensional first-order hyperbolic problems with no more than moderate discontinuities. While we do not foresee any difficulty extending the methodology to multiple space dimensions, the technical details and analysis will be left for future work.

The proposed CHVM incorporates the compact difference strategy [1] and a



(a) Cell-averaged density (global view).

(b) Nodal density (global view).

(c) Cell-averaged density (local view).

(d) Nodal density (local view).

Figure 4.8: CHVM solutions (density) to the Lax shock tube problem at $T = 1.5$.

recently developed hybrid-variable discretization framework [11] to achieve the optimally sixth-order accuracy in space using a small stencil of two neighboring cells at each grid point. In particular, the numerical solutions approximate both nodal values and cell-averaged values of the conservative variable and are advanced in time using method of lines. The spatial discretization of cell-averaged variables is derived from the weak form of the governing hyperbolic equation whereas the spatial discretization of nodal variables builds on a discrete approximation to the nodal derivatives. The discrete nodal derivatives are computed similarly as in the compact difference method and it takes into account of both nearby nodal and cell-averaged solutions. We proved that the semi-discretized method is formally sixth-order accurate in space and neutrally stable when it is applied to linear problems. To this end, we pair the CHVM discretization with an L-stable, five-stage, and fifth-order accurate diagonally implicit Runge-Kutta method for integration in time; and the resulting method is unconditionally stable.

As discontinuity usually occurs for nonlinear hyperbolic problems even when the initial data is smooth, we design a Gauss-Seidel type low-pass filter to remove node-to-node oscillations. Since the filter is linear, it does not remove spurious oscillations completely but we demonstrate that it is valuable for problems with no more than moderate discontinuities, which usually occur in acoustic applications

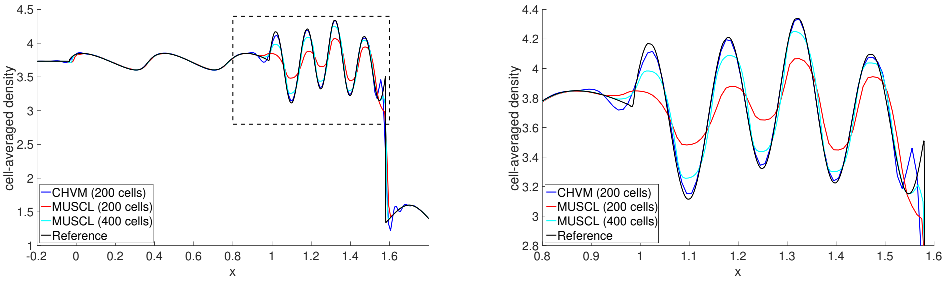
Table 9: The L_1-errors by CHVM with filter to solve the Lax shock tube problem in Section 4.2.5.

Cell-averaged values						
	Density		Velocity		Pressure	
N	Error in $\overline{\rho}$	Order	Error in $\overline{v}$	Order	Error in $\overline{p}$	Order
40	4.856e-1		7.501e-1		8.219e-1	
80	2.683e-1	0.86	3.933e-1	0.93	4.441e-1	0.89
160	1.455e-1	0.88	1.965e-1	1.00	2.072e-1	1.10
320	9.087e-2	0.68	1.098e-1	0.84	1.227e-1	0.76
640	4.774e-2	0.93	5.649e-2	0.96	6.323e-2	0.96

Nodal values						
	Density		Velocity		Pressure	
N	Error in ρ	Order	Error in v	Order	Error in p	Order
40	5.401e-1		7.548e-1		8.141e-1	
80	3.247e-1	0.73	4.385e-1	0.78	4.847e-1	0.75
160	1.785e-1	0.86	2.067e-1	1.08	2.323e-1	1.06
320	1.015e-1	0.81	1.216e-1	0.77	1.279e-1	0.86
640	5.555e-2	0.87	6.225e-2	0.97	6.549e-2	0.97

and turbulent flow computations.

The numerical performance of the proposed method as well as the impact of the filter is assessed by extensive benchmark tests. In particular, we verify the theoretical order of accuracy by solving smooth problems governed by either linear advection equations or the nonlinear system of Euler equations, with periodic boundary condition, Dirichlet boundary condition, or wall boundary condition in the case of Euler equations. In these accuracy tests, we also show that the filter does not affect the formal order of accuracy of the method. Next, we apply CHVM to solve benchmark nonlinear problems with discontinuities, including the development of N-wave by the scalar Burgers' equation and the Sod and Lax shock tube tests governed by the Euler equations; and it is shown that the low-pass filter significantly reduce the impact of Gibbs type oscillation near discontinuities. Finally, CHVM is shown to compute much more accurate solution than the second-order MUSCL scheme in solving the interaction between a Mach 2 shock wave and a smooth flow with small-scale density disturbances.



(a) Comparison of $\overline{\rho}$, global view.

(b) Comparison of $\overline{\rho}$, local view.

Figure 4.9: Comparison of $\overline{\rho}$ computed by CHVM (200 cells) and MUSCL (200 cells, 400 cells) against a reference solution.

Acknowledgements

The second author is supported by the U. S. National Science Foundation under the Grant Number 2137934.

Appendix A. An L-stable five-stage and fifth-order DIRK method

We review the time-integrator that is used in the numerical examples of this paper. To fix the idea, let us consider integrating the ODE system:

$$\boldsymbol{w}' = \boldsymbol{F}(\boldsymbol{w},\ t) \ , \tag{A.1}$$

where $\boldsymbol{w}(t)$ is the solution vector at time t and $\boldsymbol{F}(\boldsymbol{w},\ t)$ is obtained from the semi-discretization. For example, corresponding to the ODE system (2.14) we have:

$$\boldsymbol{w} = \begin{bmatrix} \overline{\boldsymbol{u}} \\ \boldsymbol{u} \end{bmatrix} \quad \text{and} \quad \boldsymbol{F}(\boldsymbol{w}, t) = -\frac{c}{h}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{P}^{-1}\boldsymbol{Q} & \boldsymbol{P}^{-1}\boldsymbol{R} \end{bmatrix}\boldsymbol{w} \ .$$

The purpose is to integration (A.1) from time t_n to $t_{n+1} = t_n + \Delta t$, or updating the discrete solutions from $\boldsymbol{w}^n$ to $\boldsymbol{w}^{n+1}$.

The DIRK5 method is a five-stage, fifth-order, L-stable, and stiffly-accurate scheme [12, Table 24] that first computes the intermediate solutions $\boldsymbol{w}^{(k)}$, $1 \leq k \leq 5$, such that:

$$\boldsymbol{w}^{(k)} = \boldsymbol{w}^n + \sum_{l=1}^{k} a_{kl}\Delta t \boldsymbol{F}(\boldsymbol{w}^{(l)}, t_n + c_l\Delta t) \ , \quad 1 \leq k \leq 5 \ ,$$

and then compute $\boldsymbol{w}^{n+1}$ by:

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^{n} + \sum_{k=1}^{5} b_k \Delta t \boldsymbol{F}(\boldsymbol{w}^{(k)}, t_n + c_k \Delta t) \ . \tag{A.2}$$

The values of $a_{kl}, b_k,$ and c_k are summarized by the classical Butcher tableau in Table A.10. In particular, the left-most column contains the values c_k, $1 \leq k \leq 5$ from top to bottom; the bottom row provides b_l, $1 \leq l \leq 5$ from left to right; and the upper-right block contains a_{kl}, $1 \leq l \leq k \leq 5$, where k and l are the row number and column number, respectively.

Table A.10: The Butcher tableau for the DIRK5 method.

$\frac{4024571134387}{14474071345096}$	$\frac{4024571134387}{14474071345096}$				
$\frac{5555633399575}{5431021154178}$	$\frac{9365021263232}{12572342979331}$	$\frac{4024571134387}{14474071345096}$			
$\frac{5255299487392}{12852514622453}$	$\frac{2144716224527}{9320917548702}$	$\frac{-397905335951}{4008788611757}$	$\frac{4024571134387}{14474071345096}$		
$\frac{3}{20}$	$\frac{-291541413000}{6267936762551}$	$\frac{226761949132}{4473940808273}$	$\frac{-1282248297070}{9697416712681}$	$\frac{4024571134387}{14474071345096}$	
$\frac{10449500210709}{14474071345096}$	$\frac{-2481679516057}{4626464057815}$	$\frac{-197112422687}{6604378783090}$	$\frac{3952887910906}{9713059315593}$	$\frac{4906835613583}{8134926921134}$	$\frac{4024571134387}{14474071345096}$
	$\frac{-2522702558582}{12162329469185}$	$\frac{1018267903655}{12907234417901}$	$\frac{4542392826351}{13702606430957}$	$\frac{5001116467727}{12224457745473}$	$\frac{1509636094297}{3891594770934}$

References

[1] S. K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1) (1992) 16–42.

[2] H. C. Yee, Explicit and implicit multidimensional compact high-resolution shock-capturing methods: Formulation, J. Comput. Phys. 131 (1) (1997) 216–232.

[3] J. M. C. Pereira, M. H. Kobayashi, J. C. F. Pereira, A fourth-order-accurate finite volume compact method for the incompressible Navier–Stokes solutions, J. Comput. Phys. 167 (1) (2001) 217–243.

[4] S. Zhang, S. Jiang, C.-W. Shu, Development of nonlinear weighted compact schemes with increasingly higher order accuracy, J. Comput. Phys. 227 (15) (2008) 7294–7321.

[5] S. Laizet, E. Lamballais, High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy, J. Comput. Phys. 228 (16) (2009) 5989–6015.

[6] Z. Wang, J. Li, B. Wang, Y. Xu, X. Chen, A new central compact finite difference scheme with high spectral resolution for acoustic wave equation, J. Comput. Phys. 336 (2018) 191–206.

[7] L. Caban, A. Tyliszczak, High-order compact difference schemes on wide computational stencils with a spectral-like accuracy, Comput. Math. Appl. 108 (2022) 123–140.

[8] X. Liu, S. Zhang, H. Zhang, C.-W. Shu, A new class of central compact schemes with spectral-like resolution I: Linear schemes, J. Comput. Phys. 248 (2013) 235–256.

[9] X. Liu, S. Zhang, H. Zhang, C.-W. Shu, A new class of central compact schemes with spectral-like resolution II: Hybrid weighted nonlinear schemes, J. Comput. Phys. 284 (2015) 133–154.

[10] X. Zeng, A high-order hybrid finite difference-finite volume approach with application to inviscid compressible flow problems: A preliminary study, Comput. Fluids 98 (2014) 91–110.

[11] X. Zeng, Linear hybrid-variable methods for advection equations, Adv. Comput. Math. 45 (2) (2019) 929–980.

[12] C. A. Kennedy, M. H. Carpenter, Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review, NASA Technical Memorandum, TM-2016-219173 (March 2016).

[13] L. Collatz, The numerical treatment of differential equations, Vol. 60, Springer New York, 1966.

[14] D. V. Gaitonde, M. R. Visbal, Further development of a Navier-Stokes solution procedure based on higher-order formulas, in: 37th AIAA Aerospace Sciences Meeting & Exhibit, 1999, reno, NV.

[15] M. R. Visbal, D. V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, J. Comput. Phys. 181 (1) (2002) 155–185.

[16] X. Li, X. Shen, C. Chen, J. Tang, F. Xiao, A note on non-negativity correction for a multimoment finite-volume transport model with WENO limiter, Q. J. Roy. Meteor. Soc. 146 (726) (2020) 546–556, notes and correspondence.

[17] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.

[18] R. Klein, Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I - One-dimensional flow, J. Comput. Phys. 121 (2) (1995) 213–237.

[19] F. Cordier, P. Degond, A. Kumbaro, An asymptotic-preserving all-speed scheme for the Euler and Navier–Stokes equations, J. Comput. Phys. 231 (17) (2012) 5685–5704.

[20] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1) (1978) 1–31.

[21] P. D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Commun. Pure Appl. Math. 7 (1) (1954) 159–193.

[22] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1) (1989) 32–78.

[23] B. van Leer, Towards the ultimate conservative difference scheme IV. A new approach to numerical convection, J. Comput. Phys. 23 (3) (1977) 276–299.

[24] P. L. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (2) (1981) 357–372.

[25] G. D. van Albada, B. van Leer, J. W. W. Roberts, A comparative study of computational methods in cosmic gas dynamics, Astron. Astrophys. 108 (1) (1982) 76–84.