Journal of Computational Physics 228 (2009) 7473–7491

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A massively parallel multi-block hybrid compact–WENO scheme for compressible flows

J. Chao, A. Haselbacher *, S. Balachandar

Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, United States

ARTICLE INFO

Article history:
Received 1 January 2009
Received in revised form 26 June 2009
Accepted 7 July 2009
Available online 15 July 2009

Keywords:
Compact methods
Hybrid methods
Shock-capturing methods
Finite-difference methods
Compressible flows
Parallel computing

ABSTRACT

A new multi-block hybrid compact–WENO finite-difference method for the massively parallel computation of compressible flows is presented. In contrast to earlier methods, our approach breaks the global dependence of compact methods by using explicit finite-difference methods at block interfaces and is fully conservative. The resulting method is fifth- and sixth-order accurate for the convective and diffusive fluxes, respectively. The impact of the explicit interface treatment on the stability and accuracy of the multi-block method is quantified for the advection and diffusion equations. Numerical errors increase slightly as the number of blocks is increased. It is also found that the maximum allowable time steps increase with the number of blocks. The method demonstrates excellent scalability on up to 1264 processors.

1. Introduction

Compact finite-difference methods, see, e.g. [12,5], have been widely used to simulate fluid flows because of their favorable resolution characteristics. In compact finite-difference methods, derivatives f' of a discrete function f are computed by solving a linear system $Af' = Bf$ where A and B are sparse, often banded or Toeplitz, matrices. In their original form, compact finite-difference methods are centered approximations and therefore lack the dissipation necessary to capture shock waves. Hybrid methods have been developed to address this deficiency by weighting a compact methods with a dissipative method depending on local flow conditions. An early hybrid method was developed by Adams and Shariff [1] for the computation of the interaction between a shock wave and a turbulent boundary layer. More recent examples include the methods of Pirozzoli [16], Ren et al. [18], Shen et al. [22], and Zhou et al. [35].

Despite the success of compact and hybrid methods in solving relatively simple problems, comparatively few applications have been reported for large-scale simulations of complex problems. A fundamental obstacle that prevents compact and hybrid methods from being applied to such problems is that the global nature of these methods incurs significant communication that makes efficient parallelization difficult. Prior work on the parallelization of compact methods is relatively limited. Sun and Moitra [25] developed the reduced parallel diagonal dominant (PDD) algorithm for the parallel inversion of tridiagonal linear systems. The reduced PDD algorithm avoids the performance bottlenecks of earlier algorithms by solving a perturbed linear system. The solution error induced by the perturbation was shown to be bounded. The parallel Thomas algorithm (PTA) proposed by Povitsky [17] avoids approximations, but suffers from a trade-off between computation and communication: either one obtains low idle time (good computational efficiency) and frequent sending of smaller messages

* Corresponding author. Address: Department of Mechanical and Aerospace Engineering, University of Florida, 222 MAE-B, P.O. Box 116300, Gainesville, FL 32611-6300, United States. Tel.: +1 (352) 392 9459; fax: +1 (352) 392 1071.
E-mail address: haselbac@ufl.edu (A. Haselbacher).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.005

(poor communication efficiency) or high idle time (poor computational efficiency) and less frequent sending of larger messages (good communication efficiency). The only systematic comparison of the reduced PDD and PTA methods for flow simulations appears to have been carried by Ladeinde et al. [10]. They concluded that the reduced PDD method performed much better than the PTA method and that the reduced PDD method produced large errors for fourth-order accurate approximations. One drawback of the study of Ladeinde et al. is that it was restricted to no more than 16 processors. Even so, typical speed-ups of the PDD method on 16 processors were smaller than 50%. Other methods for the parallel solution of tridiagonal systems have been developed, see, e.g. [14], but we are not aware of their use in flow simulations in the published literature.

The largest parallel computation to date involving compact finite-difference methods appears to have been carried out by Cook et al. [2]. They presented a parallel solution method for variable-density incompressible flows on the IBM BlueGene/L supercomputer. Using a transpose method, careful process-to-processor mapping, and tuning of critical code segments, Cook et al. were able to get excellent weak scalability on up to 65,536 processors. Their approach is very communication intensive—about 60% of the total time is due to communication—and the excellent performance may be critically dependent on the sophisticated communication networks of the IBM BlueGene/L. For this reason, we are interested in alternative parallelization strategies that perform well on less sophisticated and more widely available parallel machines such as commodity Linux clusters.

Based on the widespread use of parallel multi-block methods based on explicit finite-difference and finite-volume methods, see, e.g. [15], several authors have considered multi-block-type approaches with compact finite-difference methods. Early work on multi-domain compact methods was performed by Gaitonde and Visbal [6], who considered overlapping domains that were advanced independently using one-sided approximations near domain boundaries. In the overlap regions, two solutions existed that were reconciled by imposing the solution from one domain onto the other and vice versa. Gaitonde and Visbal found that this interface treatment could distort smooth flow structures unless higher-order filter formulations were used in the overlap regions. The difference between the solutions in the overlap regions was determined to be small. Ladeinde et al. [10] found the method to have good parallel scalability on up to 16 processors. This makes the approach of Gaitonde and Visbal attractive for subsonic flows, see, e.g. [34], where the lack of conservation is unlikely to be a serious problem. To deal with supersonic flows, Sengupta et al. [21] symmetrized the matrix $A^{-1}B$ to reduce the biasing effect of the one-sided stencils used at domain boundaries. They demonstrated the resulting method for a supersonic viscous flow about a cone-cylinder configuration. The parallelization strategy was similar to that used by Gaitonde and Visbal in that it used overlapping domains. The parallel performance of the method was only tested on up to six processors.

In the context of multi-domain methods, the approach of Costa and Don [3] also merits discussion. Costa and Don proposed a multi-domain hybrid spectral–WENO (weighted non-oscillatory, see [9]) scheme for hyperbolic conservative laws. In their approach, a spectral method or a WENO scheme is used in a given domain depending on whether high gradients and discontinuities are detected. This approach can capture flows with simple shock structures with high accuracy. However, if discontinuities are scattered throughout the entire solution domain, such that at least one discontinuity is in each subdomain, the scheme of Costa and Don reduces to the WENO scheme everywhere. At domain interfaces, they proposed spectral–spectral, spectral–WENO, WENO–WENO treatments. This logical and binary switching complicates the implementation of their approach.

Based on this short review of prior work, we identify two key areas in which further progress is needed. First, the parallel performance of compact methods should be improved. Second, an accurate and conservative block-interface treatment should be developed to allow the computation of moving shock waves. Therefore, our goal is to develop a hybrid method that is not only accurate enough for the computation of compressible turbulent flows but also exhibits good parallel performance on large numbers of processors. Our approach is based on a multi-block method that uses explicit approximations at block boundaries to break the global dependence of the underlying compact method. Because the primary focus of this article is on the block-interface treatment and the parallel performance of the resulting method, we build on the existing hybrid compact–WENO schemes of Pirozzoli [16] and Ren et al. [18]. The extension to turbulent flows will be addressed in future work. One key question that must be addressed is the stability and accuracy of our proposed approach. As shown recently by Larsson and Gustafsson [11], the stability of hybrid finite-difference methods is not guaranteed even on a single block. Accordingly, a thorough stability study was carried out for our method.

The salient features of our method can be described in a two-dimensional setting. Although our method has been programmed and tested in three dimensions, we thus restrict attention to two dimensions for brevity. The rest of this article is structured as follows. The governing equations are summarized in Section 2. The numerical method used in each block is presented in Section 3. The interface treatment of the multi-block numerical method is discussed in Section 4. The impact of the interface treatment on the stability and accuracy of the multi-block method is investigated in Section 5. Results for five test cases are shown in Section 6. The parallel performance of the multi-block method is assessed in Section 7. Conclusions are offered in Section 8.

2. Governing equations

The equations governing the viscous flow of a compressible fluid can be written in conservation form as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \frac{\partial \mathbf{f}^d}{\partial x} + \frac{\partial \mathbf{g}^d}{\partial y}, \tag{1}$$

where the conserved variable vector $\mathbf{u}$, the convective flux vectors $\mathbf{f}$ and $\mathbf{g}$, and the diffusive flux vectors $\mathbf{f}^d$ and $\mathbf{g}^d$ are defined as

$$\mathbf{u}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix},\quad \mathbf{f}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho Hu\end{bmatrix},\quad \mathbf{g}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho Hv\end{bmatrix},\quad \mathbf{f}^d=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ (\boldsymbol{\tau}\cdot\mathbf{v})_x-q_x\end{bmatrix},\quad \mathbf{g}^d=\begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ (\boldsymbol{\tau}\cdot\mathbf{v})_y-q_y\end{bmatrix}, \tag{2}$$

where $\rho,\ \mathbf{v}=\{u,v\}^T,\ p,\ E$, and $H=E+p/\rho$ are the density, the velocity vector, the pressure, the total energy, and the total enthalpy, respectively. The ideal-gas law is used,

$$p=(\gamma-1)\rho\left(E-\frac{1}{2}\|\mathbf{v}\|^2\right), \tag{3}$$

where γ is the ratio of specific heats. Assuming a Newtonian fluid, the stress tensor $\boldsymbol{\tau}$ is

$$\boldsymbol{\tau}=\mu[\nabla\mathbf{v}+(\nabla\mathbf{v})^T]-\frac{2}{3}\mu\nabla\cdot\mathbf{v}\mathbf{I}, \tag{4}$$

where μ is the dynamic viscosity, $\mathbf{I}$ is the identity tensor, and the heat flux is given by Fourier's law,

$$\mathbf{q}=-k\nabla T, \tag{5}$$

where k is the thermal conductivity. The dynamic viscosity is calculated from Sutherland's law. The conductivity is obtained from the Prandtl number $\mathrm{Pr}=\mu C_p/k$, where C_p is the specific heat at constant pressure.

3. Single-block numerical method

In this section, we summarize the numerical method used within each block. For simplicity, we assume the grid spacing h to be constant. Our method can be generalized to non-uniform and curvilinear grids in a straightforward manner using the approaches of Gamet et al. [7] and Visbal and Gaitonde [31]. The semi-discrete equations developed in this section are integrated in time using the fourth-order accurate Runge–Kutta method. For brevity, only the methods for computing the flux derivatives $\partial\mathbf{f}/\partial x$ and $\partial\mathbf{f}^d/\partial x$ are described. The flux derivatives $\partial\mathbf{g}/\partial y$ and $\partial\mathbf{g}^d/\partial y$ are treated analogously.

3.1. Discretization of convection terms

3.1.1. Hybrid compact–WENO scheme

Consider the semi-discrete conservative finite-difference scheme for the one-dimensional Euler equations on a grid with spacing $h=x_{i+1/2}-x_{i-1/2}$,

$$\frac{d\mathbf{u}_i}{dt}+\frac{\hat{\mathbf{f}}_{i+1/2}-\hat{\mathbf{f}}_{i-1/2}}{h}=\mathbf{0}, \tag{6}$$

where $\hat{\mathbf{f}}_{i+1/2}$ and $\hat{\mathbf{f}}_{i-1/2}$ are the numerical fluxes at $i+1/2$ and $i-1/2$. In the following, we assume for simplicity that the eigenvalues of the Jacobian matrix $\mathbf{A}=\partial\mathbf{f}/\partial\mathbf{q}$ are all positive. If the eigenvalues are all negative, analogous formulae apply. If positive and negative eigenvalues exist, characteristic decomposition is used as described in the next section.

The fifth-order hybrid compact–WENO scheme of Pirozzoli [16] is adopted for the computation of convective fluxes. In Pirozzoli's method, the numerical flux $\hat{\mathbf{f}}$ can be written as

$$\frac{1}{2}\sigma_{i+1/2}\hat{\mathbf{f}}_{i-1/2}+\hat{\mathbf{f}}_{i+1/2}+\frac{1}{6}\sigma_{i+1/2}\hat{\mathbf{f}}_{i+3/2}=\sigma_{i+1/2}\mathbf{f}^{\mathrm{compact}}_{i+1/2}+(1-\sigma_{i+1/2})\mathbf{f}^{\mathrm{weno}}_{i+1/2}, \tag{7}$$

where $\sigma_{i+1/2}$ is a weighting coefficient defined below, $\mathbf{f}^{\mathrm{compact}}_{i+1/2}$ is given by

$$\mathbf{f}^{\mathrm{compact}}_{i+1/2}=\frac{1}{18}\mathbf{f}_{i-1}+\frac{19}{18}\mathbf{f}_i+\frac{5}{9}\mathbf{f}_{i+1}, \tag{8}$$

and $\mathbf{f}^{\mathrm{weno}}_{i+1/2}$ is the numerical flux determined from the fifth-order WENO scheme of Jiang and Shu [9].

In Eq. (7), the coefficient $\sigma_{i+1/2}$ determines whether the compact or the WENO scheme is used locally. If $\sigma_{i+1/2}=1$, Eq. (7) reduces to the compact scheme; if $\sigma_{i+1/2}=0$, Eq. (7) reduces to the WENO scheme. The value of $\sigma_{i+1/2}$ is determined by the discontinuity sensor of Ren et al. [18],

$$\sigma_{i+1/2}=\min\left[1,\frac{\min(r_i,r_{i+1})}{r_c}\right], \tag{9}$$

where r_c is a user-defined constant and r_i is given by

$$r_i = \frac{\left|2\Delta f^m_{i+1/2}\Delta f^m_{i-1/2}\right| + \epsilon}{(f^m_{i+1/2})^2 + (f^m_{i-1/2})^2 + \epsilon}, \tag{10}$$

where $\Delta f^m_{i+1/2} = f^m_{i+1} - f^m_i$, f^m_i denotes the mth component of the flux vector $\mathbf{f}_i$, and $\epsilon = 0.9r_c\xi^2/(1-0.9r_c)$ is a small positive number used to avoid division by zero with ξ a user-defined positive number. We use $r_c = 0.3$ and $\xi = 10^{-3}$ throughout.

Fourth-order explicit boundary closures are given by Pirozzoli [16] as

$$\hat{\mathbf{f}}_{1/2} = \frac{1}{4}\mathbf{f}_0 + \frac{13}{12}\mathbf{f}_1 - \frac{5}{12}\mathbf{f}_2 + \frac{1}{12}\mathbf{f}_3, \tag{11}$$

$$\hat{\mathbf{f}}_{N+1/2} = \frac{25}{12}\mathbf{f}_N - \frac{23}{12}\mathbf{f}_{N-1} + \frac{13}{12}\mathbf{f}_{N-2} - \frac{1}{4}\mathbf{f}_{N-3}. \tag{12}$$

3.1.2. Characteristic decomposition and flux splitting

Characteristic decomposition and flux splitting, see [18], are used to extend the hybrid compact–WENO method described in Section 3.1.1 to the case when positive and negative eigenvalues exist. The characteristic decomposition is performed using the Jacobian matrix $\mathbf{A}_{i+1/2} = \partial\mathbf{f}_{i+1/2}/\partial\mathbf{q}$ where $\mathbf{f}_{i+1/2} = \frac{1}{2}(\mathbf{f}_i + \mathbf{f}_{i+1})$. With $\mathbf{L}_{i+1/2}$ denoting the matrix of left eigenvectors of $\mathbf{A}_{i+1/2}$, the convective fluxes and conservative variables can be projected to the eigenvector spaces $\mathbf{w}_i$ and $\mathbf{s}_i$,

$$\mathbf{w}_i = \mathbf{L}_{i+1/2}\mathbf{f}_i, \tag{13}$$

$$\mathbf{s}_i = \mathbf{L}_{i+1/2}\mathbf{u}_i. \tag{14}$$

The flux splitting is performed in eigenvector space. In this paper, the global Lax–Friedrichs flux splitting, see, e.g. [28], is used,

$$\mathbf{w}^{\pm}_i = \frac{1}{2}(\mathbf{w}_i \pm \lambda_{\max}\mathbf{s}_i), \tag{15}$$

where $\lambda_{\max} = \max_i\lambda_i$ denotes the maximum eigenvalue in the computational domain.

With these definitions, Pirozzoli's hybrid compact–WENO scheme can be extended to arbitrary eigenvalues as

$$\frac{1}{2}\sigma^+_{i+1/2}\hat{\mathbf{w}}^+_{i-1/2} + \hat{\mathbf{w}}^+_{i+1/2} + \frac{1}{6}\sigma^+_{i+1/2}\hat{\mathbf{w}}^+_{i+3/2} = \sigma^+_{i+1/2}\mathbf{w}^{+\text{compact}}_{i+1/2} + (1-\sigma^+_{i+1/2})\mathbf{w}^{+\text{weno}}_{i+1/2}, \tag{16}$$

$$\frac{1}{6}\sigma^-_{i+1/2}\hat{\mathbf{w}}^-_{i-1/2} + \hat{\mathbf{w}}^-_{i+1/2} + \frac{1}{2}\sigma^-_{i+1/2}\hat{\mathbf{w}}^-_{i+3/2} = \sigma^-_{i+1/2}\mathbf{w}^{-\text{compact}}_{i+1/2} + (1-\sigma^-_{i+1/2})\mathbf{w}^{-\text{weno}}_{i+1/2}, \tag{17}$$

where $\sigma^{\pm}_{i+1/2}$ are computed from $\mathbf{w}^+$ and $\mathbf{w}^-$ using Eqs. (9) and (10). Before solving Eqs. (16) and (17), they should be projected back to physical space because the numerical fluxes in the eigenvector space $\mathbf{w}^{\pm}_{i+1/2}$ are defined locally, see [18]. The corresponding equations are

$$\frac{1}{2}\sigma^+_{i+1/2}\hat{\mathbf{f}}^+_{i-1/2} + \hat{\mathbf{f}}^+_{i+1/2} + \frac{1}{6}\sigma^+_{i+1/2}\hat{\mathbf{f}}^+_{i+3/2} = \sigma^+_{i+1/2}\mathbf{f}^{+\text{compact}}_{i+1/2} + (1-\sigma^+_{i+1/2})\mathbf{f}^{+\text{weno}}_{i+1/2}, \tag{18}$$

$$\frac{1}{6}\sigma^-_{i+1/2}\hat{\mathbf{f}}^-_{i-1/2} + \hat{\mathbf{f}}^-_{i+1/2} + \frac{1}{2}\sigma^-_{i+1/2}\hat{\mathbf{f}}^-_{i+3/2} = \sigma^-_{i+1/2}\mathbf{f}^{-\text{compact}}_{i+1/2} + (1-\sigma^-_{i+1/2})\mathbf{f}^{-\text{weno}}_{i+1/2}, \tag{19}$$

where

$$\mathbf{f}^{\pm\text{compact}}_{i+1/2} = \mathbf{L}^{-1}_{i+1/2}\mathbf{w}^{\pm\text{compact}}_{i+1/2}, \tag{20}$$

$$\mathbf{f}^{\pm\text{weno}}_{i+1/2} = \mathbf{L}^{-1}_{i+1/2}\mathbf{w}^{\pm\ \text{weno}}_{i+1/2}. \tag{21}$$

Eqs. (18) and (19), supplemented with boundary closures such as those given by Eqs. (11) and (12), can be written concisely in matrix form as

$$\mathbf{A}^+\hat{\mathbf{f}}^+ = \mathbf{B}^+\mathbf{f}^+, \tag{22}$$

$$\mathbf{A}^-\hat{\mathbf{f}}^- = \mathbf{B}^-\mathbf{f}^-. \tag{23}$$

The numerical fluxes are obtained from the split numerical fluxes as $\hat{\mathbf{f}} = \hat{\mathbf{f}}^+ + \hat{\mathbf{f}}^-$ or

$$\hat{\mathbf{f}} = (\mathbf{A}^+)^{-1}\mathbf{B}^+\mathbf{f}^+ + (\mathbf{A}^-)^{-1}\mathbf{B}^-\mathbf{f}^-, \tag{24}$$

and therefore the semi-discrete finite-difference method given by Eq. (6) can be expressed as

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{h}\mathbf{C}[(\mathbf{A}^+)^{-1}\mathbf{B}^+\mathbf{f}^+ + (\mathbf{A}^-)^{-1}\mathbf{B}^-\mathbf{f}^-], \tag{25}$$

where $\mathbf{C}$ is the matrix representing the numerical-flux difference $\hat{\mathbf{f}}_{i+1/2} - \hat{\mathbf{f}}_{i-1/2}$. The tridiagonal matrices $\mathbf{A}^+$ and $\mathbf{A}^-$ are inverted using the Thomas algorithm.

3.2. Discretization of diffusion terms

The diffusion terms are computed using the tridiagonal sixth-order compact method, see [12,5]. For example,

$$\frac{1}{3}\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{i-1}+\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{i}+\frac{1}{3}\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{i+1}=\frac{14}{9}\frac{\mathbf{f}^d_{i+1}-\mathbf{f}^d_{i-1}}{2h}+\frac{1}{9}\frac{\mathbf{f}^d_{i+2}-\mathbf{f}^d_{i-2}}{4h}. \tag{26}$$

Near physical boundaries, the formulae derived by Gaitonde and Visbal [5] are used,

$$\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{1}+5\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{2}=\frac{1}{h}\left(-\frac{197}{60}\mathbf{f}^d_1-\frac{5}{12}\mathbf{f}^d_2+5\mathbf{f}^d_3-\frac{5}{3}\mathbf{f}^d_4+\frac{5}{12}\mathbf{f}^d_5-\frac{1}{20}\mathbf{f}^d_6\right), \tag{27}$$

$$\frac{2}{11}\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{1}+\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{2}+\frac{2}{11}\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_{3}=\frac{1}{h}\left(-\frac{20}{33}\mathbf{f}^d_1-\frac{35}{132}\mathbf{f}^d_2+\frac{34}{33}\mathbf{f}^d_3-\frac{7}{33}\mathbf{f}^d_4+\frac{2}{33}\mathbf{f}^d_5-\frac{1}{132}\mathbf{f}^d_6\right). \tag{28}$$

Eqs. (26)–(28) can be expressed as

$$\mathbf{A}^d\mathbf{f}^{d'}=\mathbf{B}^d\mathbf{f}^d. \tag{29}$$

The diffusive fluxes $\mathbf{f}^d$ appearing on the right-hand side of Eq. (29) are computed from derivatives of the velocity components and the temperature that are in turn calculated from Eq. (29).

4. Multi-block numerical method

As stated in the introduction, the main objective of this work is to improve the parallel performance of compact and hybrid compact–WENO methods. This requires that the global dependence of compact and hybrid methods be broken. To achieve this objective, we use a multi-block approach in which non-compact, i.e., explicit, methods are used at the block boundaries. As a result, it is not necessary to use parallel methods such as PTA or PDD for the solution of the tridiagonal systems given by Eqs. (22), (23) and (29).

Fig. 1 shows a schematic depiction of the block-interface treatment in the multi-block approach in one dimension. In our approach, we restrict the use of the hybrid compact–WENO scheme described in Section 3 to the interior nodes in each block and use a higher-order explicit method at the nodes shared by blocks. Compared to the work of Visbal and Gaitonde [29–31], Visbal and Gordnier [32], and Gaitonde and Visbal [6] on multi-block compact methods, our approach does not use overlapping blocks. The solution in the dummy nodes of each block is not computed, but received from neighboring blocks where the dummy nodes coincide with interior nodes. Therefore, our approach is conservative and can deal with discontinuities passing through block interfaces. Furthermore, compared to the approach of Costa and Don [3], our method does not require logical switching, which simplifies the implementation.

The computation of the convective and diffusive terms at block interfaces is presented in the following. Similar to the description of the single-block method, the presentation of the multi-block method is restricted to one dimension. The impact of the explicit block-interface treatment on the stability and accuracy of the multi-block hybrid method is quantified in Section 5.

The multi-block method described below was parallelized with the message-passing interface (MPI), see [8], to send/receive solution data to/from neighboring blocks. While the parallel implementation was designed carefully to allow overlapping of computation and communication, no dedicated optimization has been carried out yet. The parallel performance of the multi-block method is described in Section 7.

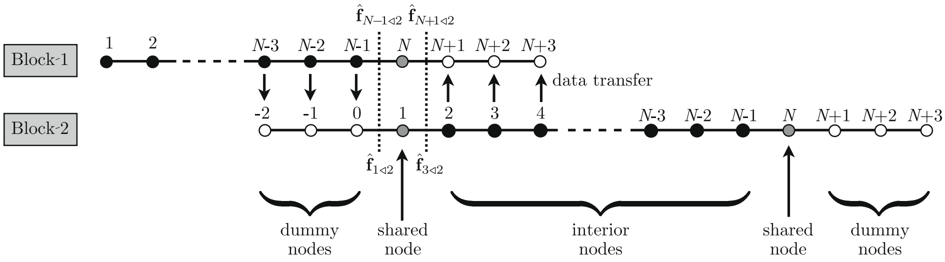

$\hat{\mathbf{f}}_{N-1\triangleleft 2}$
$\hat{\mathbf{f}}_{N+1\triangleleft 2}$
Block-1
1
2
N-3
N-2
N-1
N
N+1
N+2
N+3
data transfer
Block-2
-2
-1
0
1
2
3
4
N-3
N-2
N-1
N
N+1
N+2
N+3
$\hat{\mathbf{f}}_{1\triangleleft 2}$
$\hat{\mathbf{f}}_{3\triangleleft 2}$
dummy nodes
shared node
interior nodes
shared node
dummy nodes

Fig. 1. Schematic depiction of interface between two blocks, definition of dummy, shared, and interior nodes, and illustration of data transfer between blocks.

4.1. Computation of convective fluxes

For the convective fluxes, the hybrid compact–WENO scheme described in Sections 3.1.1 and 3.1.2 is used in each block on interior nodes. At shared nodes, the explicit WENO scheme or an explicit upstream central scheme is used, depending on the values of the discontinuity sensor given by Eq. (9). Referring to Fig. 1, the convective flux $\hat{\mathbf{f}}_{N+1/2}$ is computed as

$$\hat{\mathbf{f}}_{N+1/2} = \sigma_{N+1/2}\mathbf{f}^{\mathrm{uc}}_{N+1/2} + (1-\sigma_{N+1/2})\mathbf{f}^{\mathrm{weno}}_{N+1/2}, \tag{30}$$

where $\mathbf{f}^{\mathrm{uc}}_{N+1/2}$ is given by the fifth-order accurate approximation of Shu [23],

$$\mathbf{f}^{\mathrm{uc}}_{N+1/2} = \frac{1}{60}(2\mathbf{f}_{N-2} - 13\mathbf{f}_{N-1} + 47\mathbf{f}_N + 27\mathbf{f}_{N+1} - 3\mathbf{f}_{N+2}). \tag{31}$$

The flux $\hat{\mathbf{f}}_{N+1/2}$ is computed similarly. Because an explicit scheme is used at the shared nodes, the computational stencils for $\hat{\mathbf{f}}_{N-1/2}$ and $\hat{\mathbf{f}}_{N+1/2}$ in block 1 and $\hat{\mathbf{f}}_{1/2}$ and $\hat{\mathbf{f}}_{3/2}$ in block 2 are identical. Therefore the conservative variables and the fluxes computed at the block interface are identical for both blocks, ensuring conservation.

One difficulty arises on multi-block grids in the determination of $\lambda_{\max}$ as described in Section 3.1.2. If the global characteristic decomposition is used across all blocks, the maximum eigenvalue must be computed through reduction and broadcast operations. Such collective operations are typically expensive compared to non-collective send/receive operations and often do not scale well to large numbers of processors. This source of inefficiency can be avoided by using local characteristic decompositions, see [23]. The characteristic decomposition described in Section 3.1.2 is restricted to the interior nodes of each block. In other words, the characteristic decomposition in a given block is performed with the maximum eigenvalue within that block and therefore Eq. (15) is replaced by

$$\mathbf{w}^{\pm}_i = \frac{1}{2}(\mathbf{w}_i \pm \lambda_{\mathrm{local}}\mathbf{s}_i). \tag{32}$$

Furthermore, to preserve conservation at block interfaces, the maximum eigenvalue $\lambda_{\max}$ in Eq. (15) is determined by the local maximum over the computational stencil of the explicit scheme used at block interfaces.

4.2. Computation of diffusive fluxes

The diffusive fluxes at block interfaces are computed using the same procedure described in Section 4.1 for the convective fluxes. With three dummy nodes used for the convective fluxes, a sixth-order explicit central difference for the first derivatives in the diffusive fluxes can be used at the block interface, i.e.,

$$\left(\frac{\partial \mathbf{f}^d}{\partial x}\right)_N = \frac{3}{2}\frac{\mathbf{f}^d_{N+1} - \mathbf{f}^d_{N-1}}{2h} - \frac{3}{5}\frac{\mathbf{f}^d_{N+2} - \mathbf{f}^d_{N-2}}{4h} + \frac{1}{10}\frac{\mathbf{f}^d_{N+3} - \mathbf{f}^d_{N-3}}{6h}. \tag{33}$$

To ensure that the diffusive terms are discretized in a conservative manner, the diffusive fluxes are transferred between blocks prior to computing the derivatives using Eq. (33).

5. Stability and accuracy analysis

The hybrid compact–WENO scheme described in Sections 3 and 4 has been tested on several problems with single and multiple blocks. The results for some of these problems are presented in Section 6. The tests indicate that the hybrid compact–WENO scheme is stable, accurate, and robust. In this section, we present results from theoretical and numerical analyses to quantify the impact of the block-interface treatments on the stability and accuracy of our method for the advection and diffusion equations. Throughout the remainder of the paper, the number of blocks is denoted by N_b.

5.1. Advection equation

Consider the one-dimensional advection equation

$$\frac{\partial \phi}{\partial t} + a\frac{\partial \phi}{\partial x} = 0, \tag{34}$$

where $\phi(x,t)$ is a scalar variable and $a > 0$ is the constant advection speed. With the vector of unknowns denoted by $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_{N_p})^T$, where N_p is the total number of grid points in the solution domain (i.e., in all blocks), we can write the semi-discrete form of Eq. (34) as

$$\frac{d\boldsymbol{\phi}}{dt} = -\frac{a}{h}\mathbf{C}\mathbf{A}^{-1}(\mathbf{B}\boldsymbol{\phi} + \mathbf{b}), \tag{35}$$

where $h = 1/(N_p - 1)$, **A** and **B** are the coefficient matrices of the compact scheme in Eq. (22) for $\sigma_{i+1/2} = 1$ (i.e., without the WENO scheme), **b** is a vector incorporating boundary conditions, and the matrix **C** represents the conservative approximation of the first derivative. The explicit approximation given by Eq. (31) is used at block interfaces.

5.1.1. Stability analysis

An eigenvalue analysis states that the semi-discrete scheme given by Eq. (35) is stable if all eigenvalues of the matrix

$$\mathbf{D} = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}, \tag{36}$$

have non-positive real parts, and, when scaled by the timestep, fall within the stability region of the time-integration method. For a single block, Pirozzoli [16] found that his compact scheme with the boundary closures given by Eqs. (11) and (12) has only negative eigenvalues. We are interested in investigating the stability properties of our multi-block compact method for periodic and Dirichlet boundary conditions. With periodic boundary conditions, i.e., in the absence of physical boundary conditions, we can isolate the impact of the block-interface treatment on the stability of the multi-block compact method. Dirichlet boundary conditions represent the more physically realistic situation of inflow boundary conditions. With the assumption that $a > 0$, we impose the Dirichlet boundary condition at the left boundary.

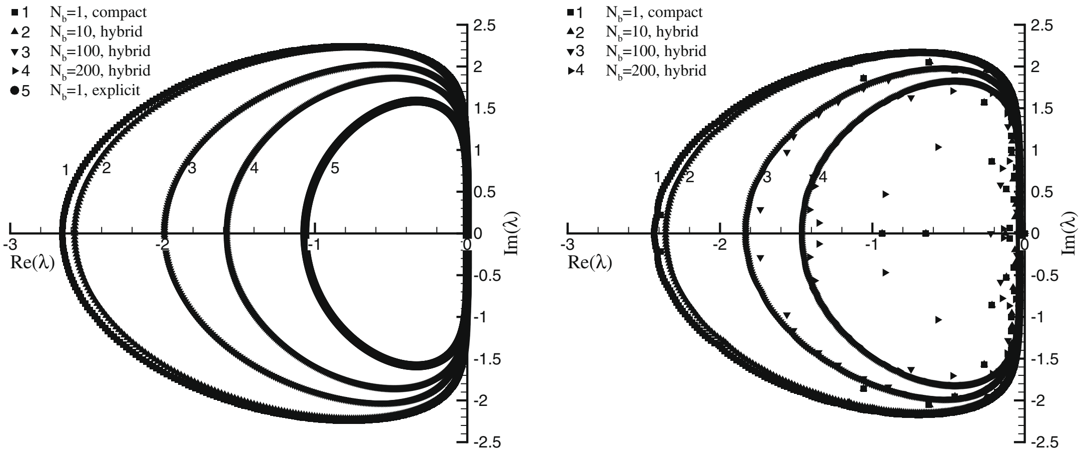
The eigenvalue spectra of **D** as a function of N_b and the boundary conditions are shown in Fig. 2 for $N_p = 1001$. For the maximum value of $N_b = 200$ considered, each block contains only four interior grid points. It can be seen that all eigenvalues have negative real parts irrespective of the number of blocks and the boundary conditions, indicating that the underlying multi-block scheme is linearly stable. Comparison of the eigenvalue spectra of the multi-block compact method with those of the single-block compact and explicit methods shows that the multi-block scheme becomes increasingly stable with increasing N_b (in the sense of the eigenvalues being closer to the origin) and therefore allowing a larger timestep. For example, with a Dirichlet boundary condition, the maximum CFL number $\mathrm{CFL} = a\Delta t/h$ for stable computations is about 1.037 for $N_b = 1$. However, with $N_b = 100$, the maximum CFL number for stable computations increases to 1.221. For periodic boundary conditions, the variation of maximum CFL with N_b/N_p (i.e., the inverse of the number of points per block) is shown in Fig. 3(a).

It should be noted that these conclusions are dependent on the use of the fourth-order accurate Runge–Kutta method used for time integration. With increasing N_b, the maximum extent of the eigenvalue spectra changes from the real to the imaginary component in such a way as to take advantage of the shape of the stability region, see Fig. 4. Similar increases of maximum allowable timesteps of the multi-block can thus be expected for other time-integration methods whose stability regions are similar in shape to that of the fourth-order Runge–Kutta method, such as the third-order Adams–Bashforth method, for example.

5.1.2. Accuracy study

To assess the impact of the block-interface treatment on the accuracy of the multi-block scheme, a grid-refinement study is performed for $a = 1$ on the domain $-1 \leqslant x \leqslant 1$ with periodic boundary conditions. The initial condition is $\phi(x, 0) = \sin(\pi x)$. The explicit approximation given by Eq. (31) is used at block interfaces and the periodic boundaries. To ensure that time-discretization errors do not dominate space discretization errors, the time step is determined from the relation $\mathrm{CFL} = 0.01N_p^{-1/4}$. Numerical errors are computed from the L_∞- and L_2-norms of the difference between the computed and exact solutions at $t = 0.25$.

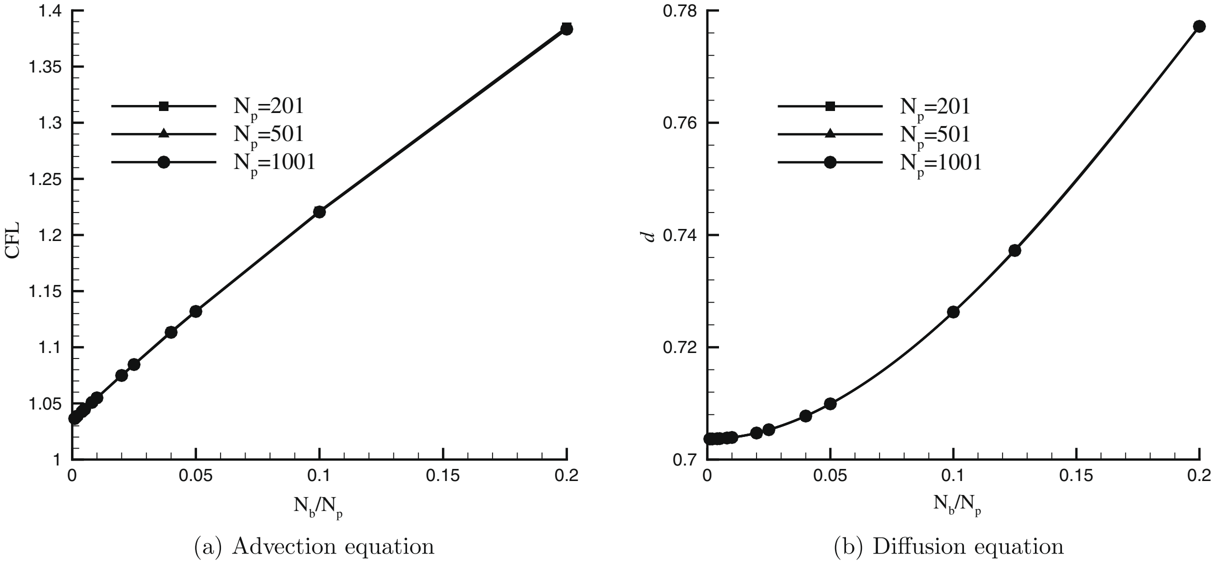
Fig. 5(a) shows the error behavior of the multi-block scheme as a function of the number of grid points for $N_b = 20$. It can be seen clearly that the multi-block scheme achieves the expected formal order of accuracy for a given number of blocks. Fig. 5(b) shows the error behavior as a function of N_b for $N_p = \{101, 801\}$. The results indicate that the block-interface



(a) Periodic boundary conditions

(b) Dirichlet boundary condition

Fig. 2. The eigenvalue spectra of the matrix **D** given by Eq. (36) for $N_p = 1001$ as a function of boundary condition and N_b.



(a) Advection equation

(b) Diffusion equation

Fig. 3. The variation of maximum non-dimensional timesteps with N_b/N_p for fourth-order Runge–Kutta time integration and periodic boundary conditions.

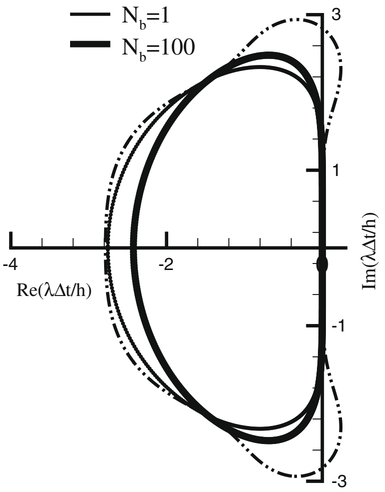


Fig. 4. Comparison of scaled eigenvalue spectra of the matrix **D** given by Eq. (36) at maximum allowable timestep for $N_b = 1$ and 100 superimposed on the stability region of the fourth-order Runge–Kutta method (indicated by dash-dotted line).

treatment increases the errors somewhat, but the increase is relatively small; after an initial increase from $N_b = 1–2$, the errors increase very slowly with N_b for $N_p = 101$. For $N_p = 801$, the error is approximately independent of N_b.

5.2. Diffusion equation

Consider the one-dimensional diffusion equation

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}, \tag{37}$$

where $\phi(x,t)$ is a scalar variable and α is the constant diffusivity coefficient. With $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_{N_p})^T$ denoting the vector of unknowns, the semi-discrete form of Eq. (37) can be written as

$$\frac{d\boldsymbol{\phi}}{dt} = \frac{\alpha}{h^2}\mathbf{D}\boldsymbol{\phi}, \tag{38}$$

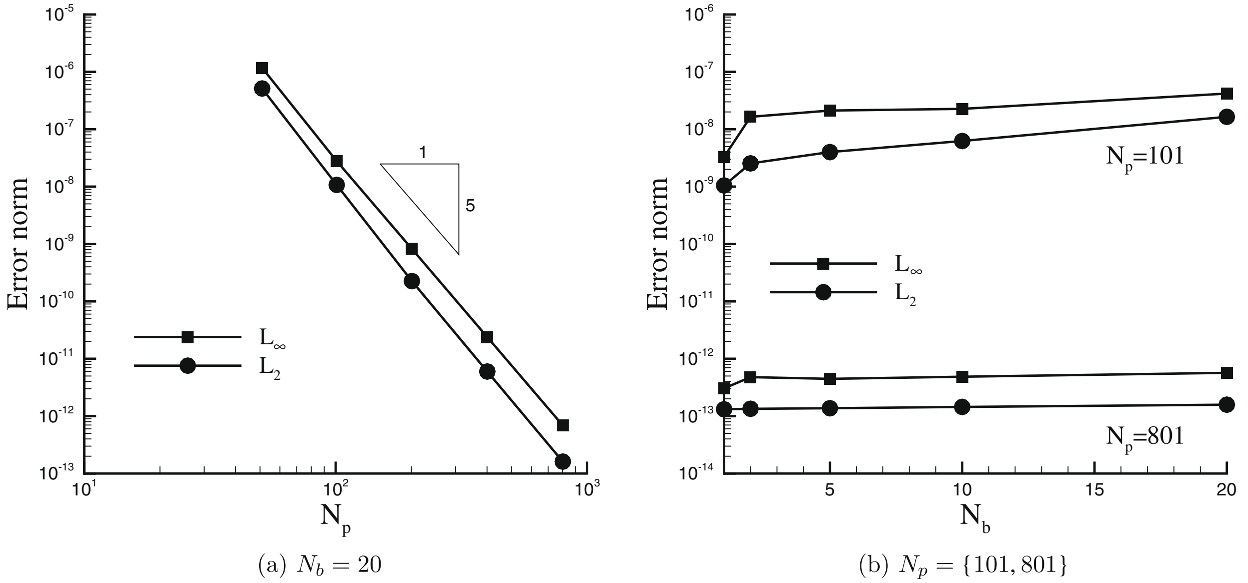


Fig. 5. Accuracy study of the multi-block compact scheme for the one-dimensional advection equation as a function of N_p at constant N_b and vice versa.

where **D** is a matrix representing a discrete approximation to the second-derivative operator. Following the discussion in Section 3.2, the matrix **D** can be expressed as

$$\mathbf{D} = [(\mathbf{A}^d)^{-1}\mathbf{B}^d]^2. \tag{39}$$

where $\mathbf{A}^d$ and $\mathbf{B}^d$ are the matrices in Eq. (29), modified through the use of the sixth-order explicit central difference given by Eq. (33) at block interfaces and periodic boundaries.

5.2.1. Stability analysis

The conditions for stability of Eq. (38) are identical to those stated in Section 5.1.1. The sixth-order compact finite-difference scheme with the boundary closures given by Eqs. (27) and (28) is known to be stable. As with the investigation of the stability of the multi-block compact method for the advection equation, we are interested in the influence of periodic and Dirichlet boundary conditions on the stability of the multi-block compact method for the diffusion equation.

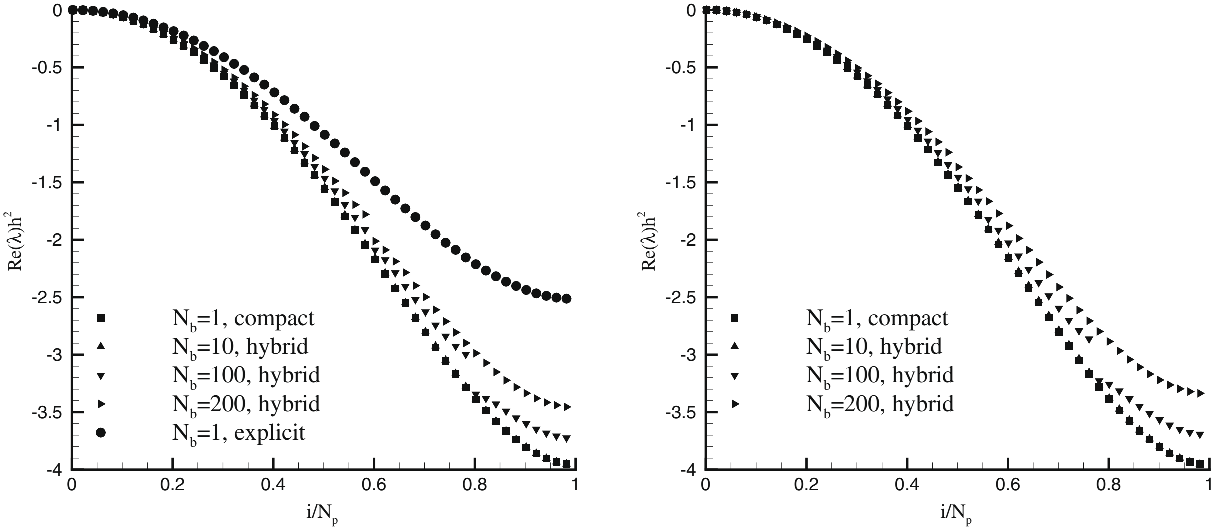
Fig. 6 shows the real parts of the eigenvalues of the matrix **D** multiplied by the square of grid spacing as a function of N_b and the boundary conditions for $N_p = 1001$. The real parts of the eigenvalues of the matrix **D** are seen to be negative irrespective of N_b, indicating that the multi-block scheme is stable for the diffusion terms. We note that the eigenvalue spectra for $N_b = 10$ follow very closely those for the single-block compact scheme. In other words, the impact of the block-interface treatment is relatively small; even with $N_b = 200$, the eigenvalues associated with the high-frequency modes are still closer to the single-block compact method than to the single-block explicit method. Finally, it can be observed that relatively large jumps exist in the eigenvalues for $N_b = 100$ and 200. The reason for these jumps is currently unknown. It should be noted that the eigenvalue spectra of single-block finite-difference differentiation matrices contain small jumps also, see [4]. These are not visible in Fig. 6 due to the scale of the figure. We conclude that the interface treatment does not cause, but that it may accentuate, the jumps. There appears to be no obvious relationship between the number of jumps and N_b and/or N_p.

Similar to the advection equation, the multi-block compact method also allows larger time steps compared to the single-block compact method for the one-dimensional diffusion equation. Fig. 3(b) shows that the maximum diffusion number $d = \alpha\Delta t/h^2$ increases with N_b/N_p.

In contrast to the behavior for the advection equation, the overall shape of the stability region of the time-integration method is immaterial in determining the maximum timestep for the diffusion equation. Because the eigenvalues of the matrix **D** are purely real, only the intercept of the stability region on the negative real axis matters. Thus the observed increase in the maximum timestep for the diffusion equation with the present multi-block method is likely to hold for any other time-integration method.

5.2.2. Accuracy analysis

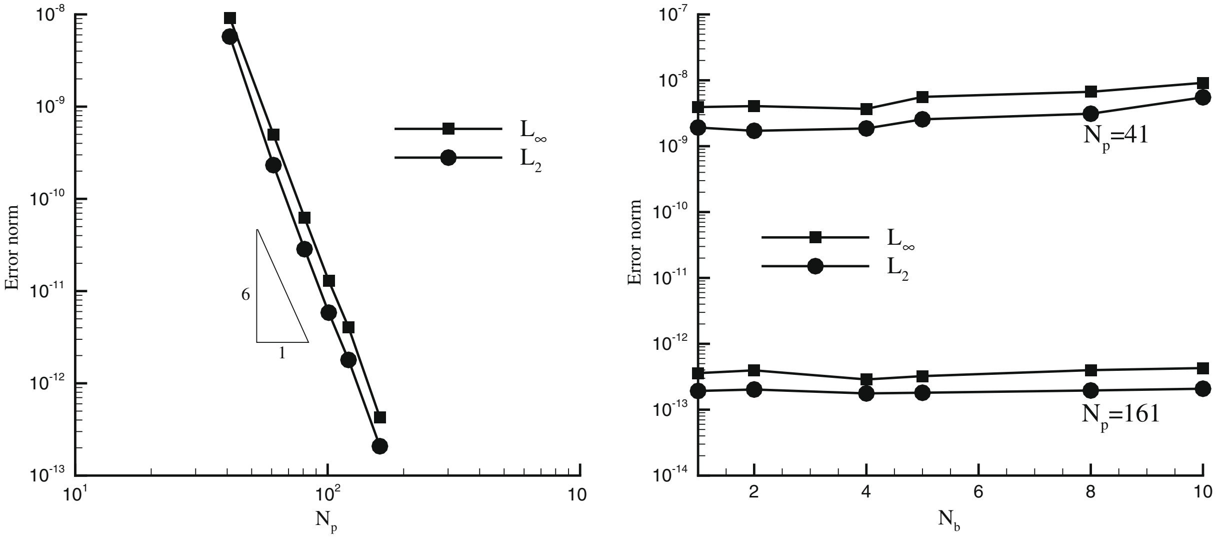
The impact of the block-interface treatment on the accuracy of the multi-block compact scheme is assessed for the diffusion equation by solving Eq. (37) for $\alpha = 1$ on the domain $0 \leqslant x \leqslant 1$. The boundary conditions are $\phi(0,t) = 0$ and $\phi\prime(1,t) = 0$ and the initial condition is $\phi(x,0) = 1$. The exact solution is



(a) Periodic boundary conditions

(b) Dirichlet boundary conditions

Fig. 6. The eigenvalues λ_i of the matrix **D** given by Eq. (39) for $N_p = 1001$ as a function of boundary condition and N_b. Every 20 points are skipped for distinguishing symbols.



(a) $N_b = 10$

(b) $N_p = \{41, 161\}$

Fig. 7. Accuracy study of the multi-block compact scheme for the one-dimensional diffusion equation as a function of N_p for constant N_b and vice versa.

$$\phi(x,t) = \frac{4}{\pi}\sum_{n=0}^{\infty}\frac{1}{2n+1}\exp\left[-\left(n+\frac{1}{2}\right)^2\pi^2 t\right]\sin\left[\left(n+\frac{1}{2}\right)\pi x\right]. \tag{40}$$

To ensure that time-discretization errors do not dominate spatial discretization errors, the time step is chosen according to $d = 0.01N_p^{1/2}$. Numerical errors are computed from the L_∞- and L_2-norms of the differences between the exact and computed solutions at $t = 0.5$.

Fig. 7(a) shows the behavior of the error norms as a function of N_p for $N_b = 10$. The results indicate that the multi-block compact method for the diffusion equation is indeed sixth-order accurate. Fig. 7(b) presents the behavior of the error norms as a function of N_b for $N_p = \{41, 161\}$. As for the advection equation, the interface treatment leads to a slight increase in the error norms, but the increase is small even for $N_b = 10$, where only three interior grid points exist for the grid with $N_p = 41$.

6. Results

In this section, results obtained with the multi-block method are presented for five well-known test cases. For each test case, the impact of the number of grid points N_p and/or the number of blocks N_b is assessed. The results presented below complement those discussed in Section 5. Throughout, we use $\gamma = 1.4$ and $C_p = 1004.5$ J/kg K.

6.1. Vortex convection

We consider the convection of a vortex to assess the accuracy of the method for smooth solutions of the Euler equations. The initial conditions are a uniform freestream aligned with the x-coordinate direction of density ρ_∞, velocity u_∞, and pressure p_∞, modified by a vortex,

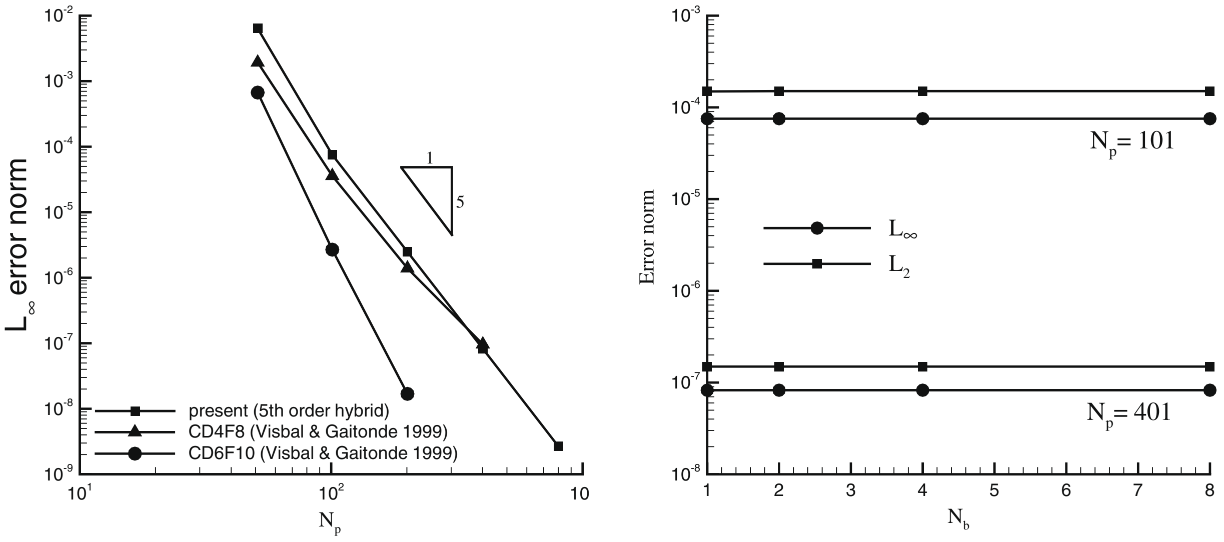
$$u(x,y) = u_\infty - \frac{C(y-y_0)}{r_c^2}\exp\left(-\frac{1}{2}\tau^2\right), \tag{41}$$

$$v(x,y) = -\frac{C(x-x_0)}{r_c^2}\exp\left(-\frac{1}{2}\tau^2\right), \tag{42}$$

$$p(x,y) = p_\infty - \frac{\rho C^2}{2r_c^2}\exp(-\tau^2), \tag{43}$$

for $\tau = r/r_c \leqslant 1$, where $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$, r_c is the vortex-core radius, (x_0, y_0) are the coordinates of the initial location of the vortex, and C is a dimensional constant. This constant is chosen such that the non-dimensional vortex-strength parameter $C/(u_\infty r_c) = 0.2$. The vortex-core radius is $r_c = 0.2$. The freestream Mach number $M_\infty = u_\infty/c_\infty$, where c_∞ is the freestream speed of sound, is chosen to be 0.1. A computational domain of $[0,4]\times[0,4]$ is used. The initial location of the vortex is $(x_0, y_0) = (2,2)$. Periodic conditions are set on all boundaries. A CFL number of 0.04 is used for all computations. (This value has been verified to be small enough to lead to negligible time-discretization errors.) Because of the low strength of the vortex and the low freestream Mach number, the vortex is simply convected with the flow. Therefore, errors can be computed by taking the initial condition as the exact solution, and the order of accuracy can be determined through a grid-refinement study. Uniform grids with 51×51 up to 801×801 points are used. On the coarsest grid, the grid spacing relative to the vortex-core radius is $h/r_c = 0.4$. The errors presented below are computed at the time when the vortex has returned to its initial location, i.e., when it has convected a distance of $8r_c$. The domain size, grid resolutions, and initial and boundary conditions are identical to those used by Visbal and Gaitonde [29] to allow a direct comparison. For multi-block computations, the computational domain is partitioned only in the streamwise direction.

Fig. 8(a) shows the behavior of the L_∞-norm of the error in the computed v-velocity along $y = 2$ as a function of N_p for $N_b = 4$. The results indicate the multi-block method attains fifth-order accuracy for smooth problems. Fig. 8(a) also shows a comparison between the errors incurred by the present method and the fourth- (CD4F8) and sixth-order (CD6F10) accurate centered methods with filtering of Visbal and Gaitonde [29]. The errors of the present method are comparable to those of



(a) $N_b = 4$

(b) $N_p = \{101, 401\}$

Fig. 8. Accuracy study of the multi-block scheme for the vortex-convection problem as a function of N_p at constant N_b and vice versa.

Visbal and Gaitonde's fourth-order method on the coarse grids, but become lower on finer grids. Fig. 8(b) shows the behaviors of the L_∞- and L_2-norms of the error in the v-velocity as a function of N_b for $N_p = \{101, 401\}$. It can be seen that the error norms are nearly independent of N_b. The vorticity contours as a function of N_b are presented in Fig. 9. The contour lines are indistinguishable to plotting accuracy and the vorticity extrema are identical to three significant digits, indicating that the block-interface treatment has only a minor impact on smooth solutions. These observations corroborate the findings of Section 5. (Determining the efficiency, i.e., the accuracy relative to the required computing time, of our and Visbal and Gaitonde's methods would require dedicated optimization and timing and was not the objective of this test problem.)

6.2. Shu–Osher problem

The classical one-dimensional problem of Shu and Osher [24] is used to assess the impact of the interface treatment on moving shock waves. The initial condition

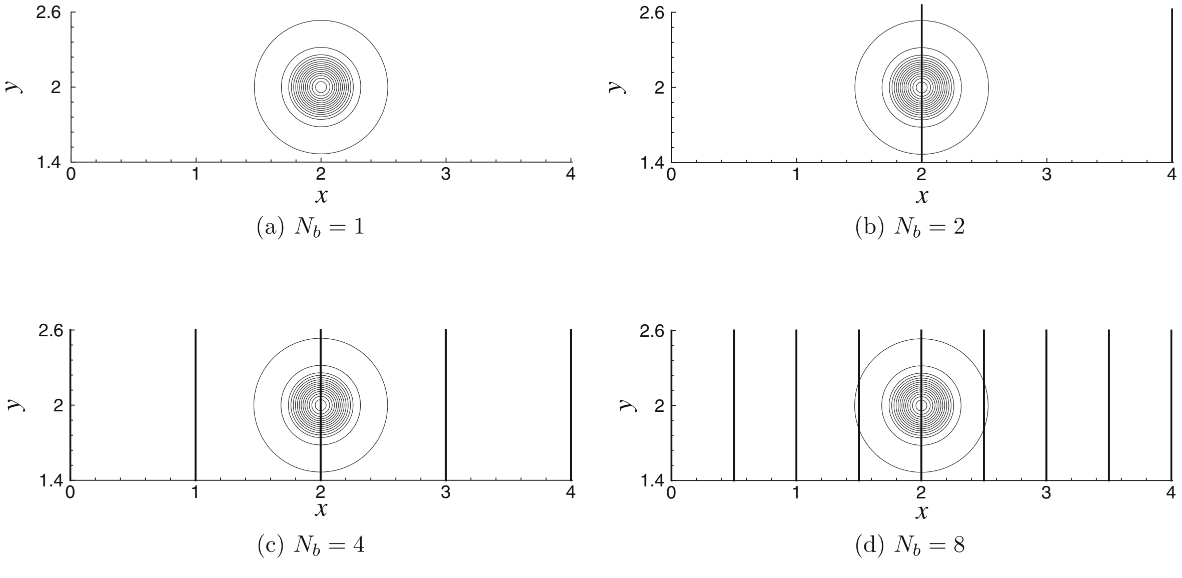
$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x < 1, \\ (1 + 0.2\sin[5(x-1)], 0, 1) & \text{if } x \geqslant 1, \end{cases} \tag{44}$$

represents a shock wave moving into a stagnant fluid with a sinusoidal density variation. Numerical solutions of the Euler equations with the multi-block hybrid compact–WENO method are computed on the domain $-5 \leqslant x \leqslant 5$ with 201 uniformly distributed grid points and CFL $= 0.3$. Because of the absence of an analytical solution, we follow other authors in using a numerical solution obtained with the pure WENO scheme on a (single-block) uniform grid with $N_p = 2001$ as a reference solution.

The numerical solutions obtained for various values of N_b are compared with the reference solution for the density at $t = 1.8$ in Fig. 10. Two main observations can be made. First, the multi-block hybrid compact–WENO method performs better than the pure WENO method for the same number of grid points. Second, the number of blocks affects the solution in the highly oscillatory region only immediately behind the shock, and even there the influence is minor.

6.3. Taylor–Sedov blast wave

To assess the ability of the method to capture strong shock waves, we solve the evolution of a blast wave generated by a point source of energy. A similarity solution valid for large times was derived independently by Taylor [26] and Sedov [20] under the assumption that the blast wave propagates into uniform surroundings. The Taylor–Sedov blast wave is a challenging test because of the strength of the shock wave, the use of a Cartesian grid that introduces a directional bias into what is a one-dimensional problem in the radial coordinate, and the need to represent the point source by a sufficient number of grid points. Representing the point source as a sphere of radius R_0 at the origin, the initial conditions are chosen to be $(\rho_\infty, u_\infty, p_\infty) = (1, 0, 10^{-6})$ for $r \geqslant R_0$ and $(\rho_0, u_0, p_0) = (1, 0, (\gamma - 1)\rho E_0/V_0)$ for $r < R_0$, where r is the radial position from the origin, E_0 is the energy contained in the point source, and $V_0 = \frac{4}{3}\pi R_0^3$ is the volume of the energy source at the origin. In the results presented below, $R_0 = 0.01$ and $E_0 = 1.0$. The computational domain is $[-0.4, 0.4] \times [-0.4, 0.4]$. Grid resolutions



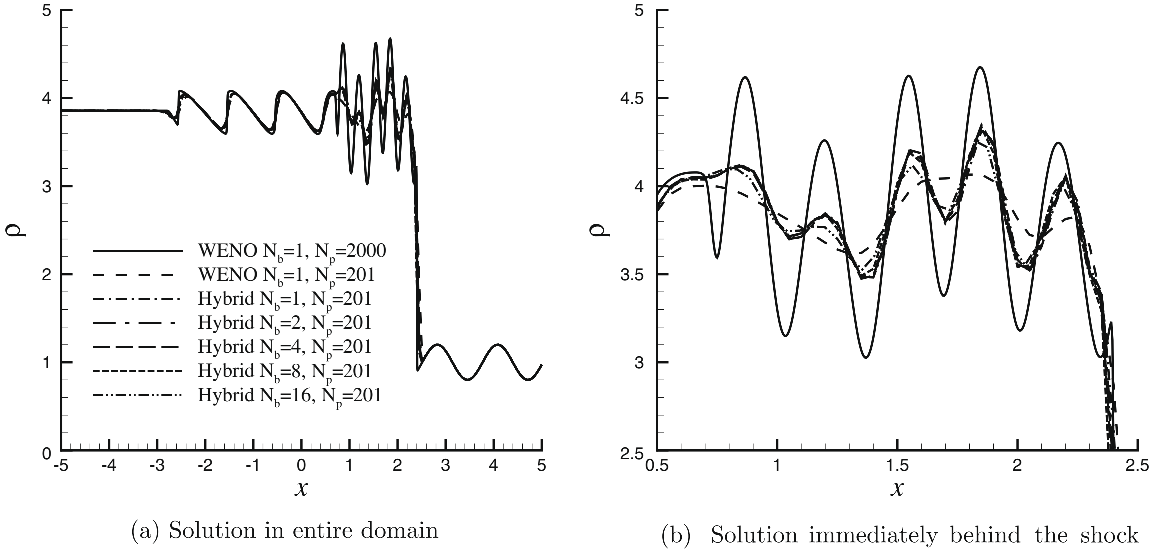
(a) $N_b = 1$

(b) $N_b = 2$

(c) $N_b = 4$

(d) $N_b = 8$

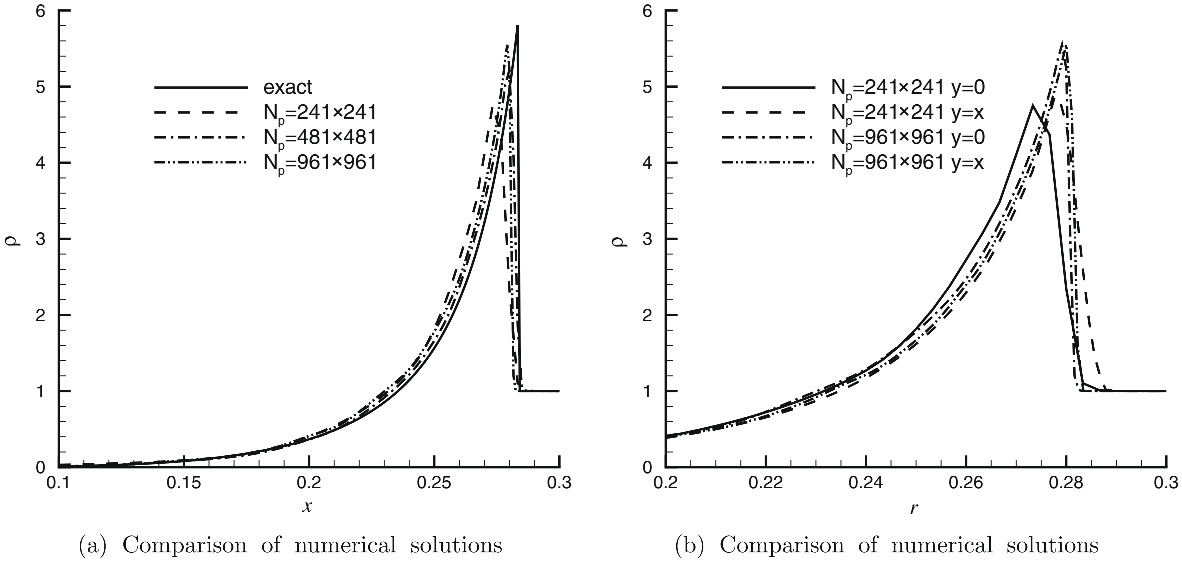
Fig. 9. Vorticity contours for the vortex-convection problem with $h/r_c = 0.1$. 14 equispaced contour levels are used. For all cases, the extrema are identical to three significant figures at $(\omega_{\min}, \omega_{\max}) = (-0.935, 6.839)$.



(a) Solution in entire domain

(b) Solution immediately behind the shock

Fig. 10. Density profiles for the Shu–Osher problem at $t = 1.8$.



(a) Comparison of numerical solutions

(b) Comparison of numerical solutions

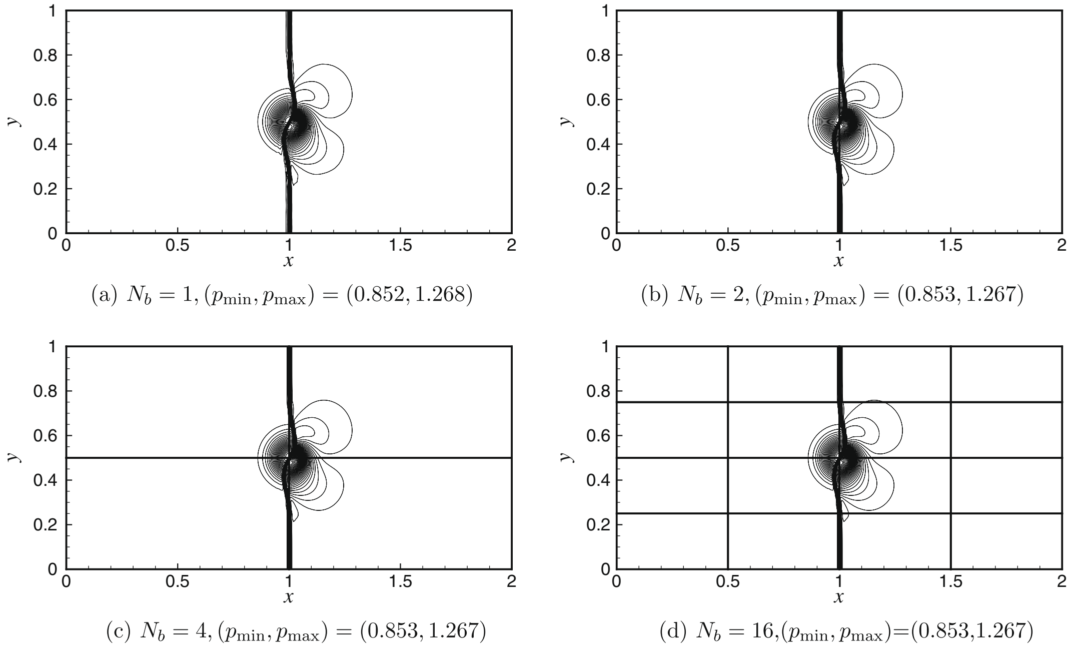
Fig. 11. Density profiles for the Taylor–Sedov blast-wave problem at $t = 0.08$ as a function of N_p.

of $N_p = \{241 \times 241, 481 \times 481, 961 \times 961\}$ are used with $N_b = \{16, 64, 256\}$, respectively. The numerical solutions for the density at $t = 0.08$ along $y = 0$ are compared to the similarity solution in Fig. 11(a). It can be seen that the numerical solutions agree well with the similarity solution with increasing grid refinement. To assess the impact of the Cartesian grid on the solution, Fig. 11(b) compares the density along $y = 0$ and $y = x$. The anisotropy in the numerical solutions reduces with increasing grid resolution. The sensitivity to grid resolution is well known for Cartesian grids. The results indicate that the block-interface treatment is stable and accurate for strong shock waves.

6.4. Shock–vortex interaction

We next consider a shock–vortex interaction problem, see [23], in which a moving vortex superimposed on an inviscid flow with Mach number 1.1 hits a stationary normal shock. The initial conditions are

$$(\rho, u, v, p) = \begin{cases} (1, 1.1\sqrt{\gamma}, 0, 1) & \text{if } x < 1, \\ (1.169, 1.1134, 0, 1.245) & \text{if } x \geqslant 1, \end{cases} \tag{45}$$



(a) $N_b = 1, (p_{\min}, p_{\max}) = (0.852, 1.268)$ (b) $N_b = 2, (p_{\min}, p_{\max}) = (0.853, 1.267)$

(c) $N_b = 4, (p_{\min}, p_{\max}) = (0.853, 1.267)$ (d) $N_b = 16,(p_{\min}, p_{\max})=(0.853,1.267)$

Fig. 12. Pressure contours for shock–vortex interaction at $t = 0.2$. Sixty equispaced contour levels are shown. The straight black lines indicate block interfaces.

modified by the velocity, temperature, and entropy perturbations induced by the vortex for $r \leqslant r_c$,

$$u'(r,\theta) = \epsilon\tau \exp[\alpha(1-\tau^2)] \sin\theta, \tag{46}$$

$$v'(r,\theta) = -\epsilon\tau \exp[\alpha(1-\tau^2)] \cos\theta, \tag{47}$$

$$T'(r,\theta) = \frac{\gamma-1}{\gamma}\frac{\epsilon^2}{4\alpha} \exp[2\alpha(1-\tau^2)], \tag{48}$$

$$s'(r,\theta) = 0, \tag{49}$$

where $\epsilon = 0.3,\ \tau = r/r_c,\ \alpha = 0.204,\ \theta = \tan^{-1}(y/x),\ r = \sqrt{(x-x_0)^2 + (y-y_0)^2},\ r_c = 0.05$, and (x_0, y_0) are the coordinates of the initial location of the vortex. We choose a computational domain of $[0,2] \times [0,1]$ with 401×101 uniformly distributed grid points. The initial location of the vortex is $(x_0, y_0) = (0.75, 0.5)$. Periodic boundary conditions are used at the top and bottom domain boundaries. A CFL number of 0.3 is used for all computations.

Figs. 12 and 13 show the pressure contours at $t = 0.2$ and 0.35 as a function of N_b. It can be seen that the overall solutions are visually indistinguishable from each other. Minor differences are restricted to the immediate vicinity of the straight part of the shock wave at $t = 0.2$. The pressure extrema listed in the captions are very similar to those reported by [13] using a characteristic filtering method on a single block. The pressure variation through the vortex center at $t = 0.35$ is plotted in Fig. 14. These observations indicate that the explicit treatment of the block interfaces has only a minimal impact on the solution.

6.5. Compressible mixing layer

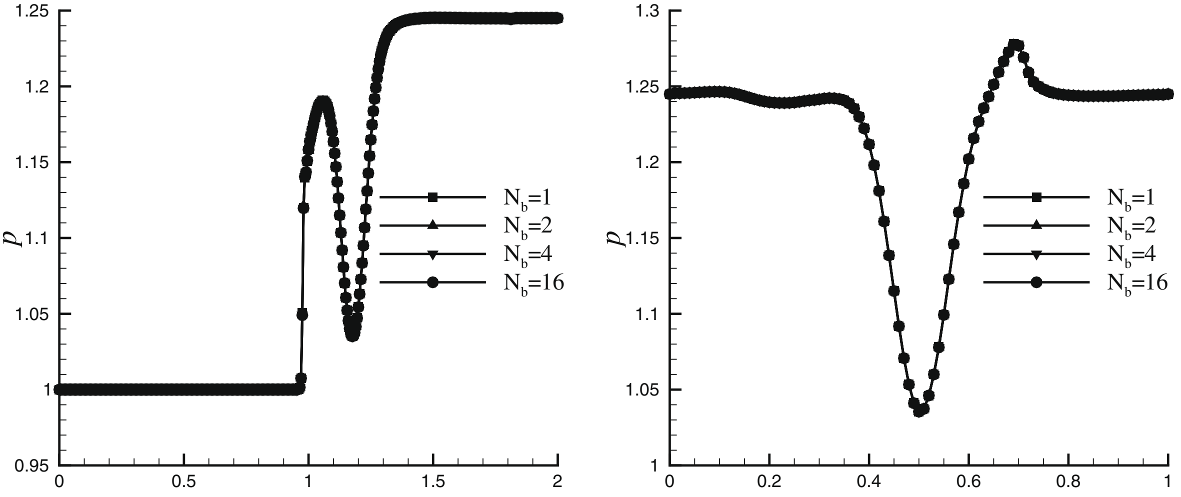
As the final test case we consider the compressible mixing layer, in which two streams with equal but opposite velocities excite the absolute instability of the shear layer developing between them. The objective is to establish that the multi-block compact–WENO method accurately computes viscous flows at relatively low Reynolds number.

The initial mean velocity field is

$$u(y) = \frac{1}{2}(u_1 - u_2)\tanh(2y), \tag{50}$$

$$v = 0, \tag{51}$$

where $u_1 > 0$ and $u_2 = -u_1$ are the freestream velocities. The initial mean velocity field is perturbed by a single mode, see [33],



(a) Along $y = 0.5$.

(b) Along $x = 0.35$.

Fig. 14. Pressure variation through vortex core for shock–vortex interaction problem at $t = 0.35$.

$$u'(x,y) = \frac{a}{\pi}\frac{L_x y}{b^2}\exp\left(-\frac{y^2}{b^2}\right)\sin\left(\frac{4\pi x}{L_x} - \frac{\pi}{2}\right)(u_1 - u_2), \tag{52}$$

$$v'(x,y) = a\exp\left(-\frac{y^2}{b^2}\right)\cos\left(\frac{4\pi x}{L_x} - \frac{\pi}{2}\right)(u_1 - u_2), \tag{53}$$

where a is the amplitude coefficient, L_x is the domain length (specified below), and b is the y-modulation of the perturbation. The local speed of sound is given by

$$c^2 = c_1^2 + \frac{\gamma - 1}{2}(u_1^2 - u^2). \tag{54}$$

Fig. 16. Variation of normalized vorticity thickness with non-dimensional time for the compressible mixing layer and comparison with the result of Sandham and Reynolds [19].

The freestream velocities and speeds of sound are related through the convective Mach number

$$M_c = \frac{u_1 - u_2}{c_1 + c_2}. \tag{55}$$

The extent of the mixing layer is characterized through its vorticity thickness, defined as

$$\delta_\omega(t) = \frac{|u_1 - u_2|}{\max_{x,y}|\partial u/\partial y|}. \tag{56}$$

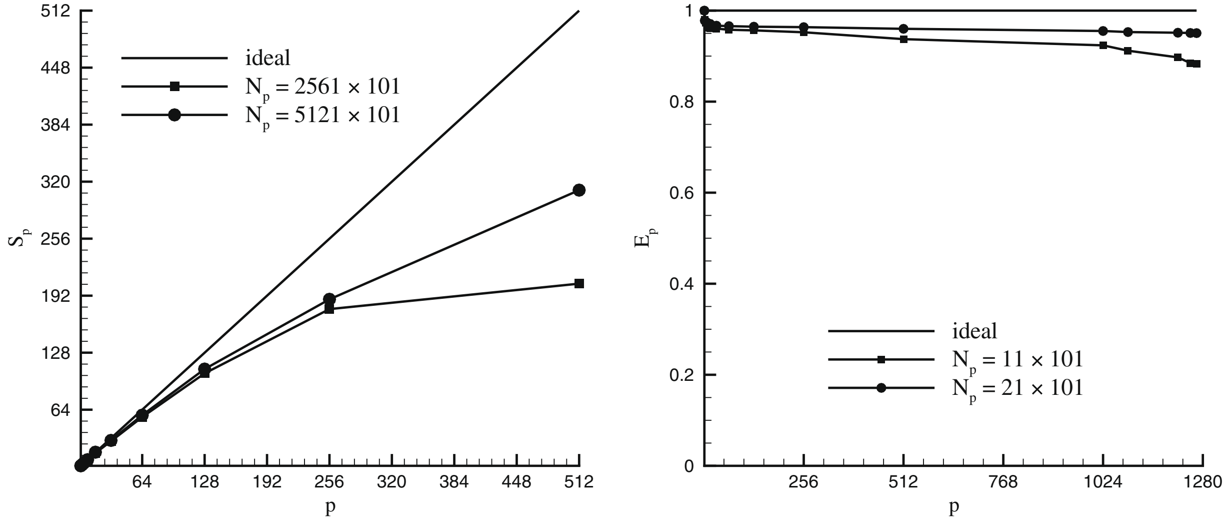
The initial vorticity thickness is used to define a Reynolds number as

$$\mathrm{Re} = \frac{\rho_1 u_1 \delta_\omega(0)}{\mu_1}, \tag{57}$$

where μ_1 is the dynamic viscosity associated with the freestream of velocity u_1. The dynamic viscosity is assumed to be given by Sutherland's law with reference viscosity μ_1 and reference temperature $T_1 = c_1^2/(\gamma - 1)C_p$. Here, we take $M_c = 0.4,\ \mathrm{Re} = 400,\ a = 0.05,\ b = \sqrt{10}$, and $\mathrm{Pr} = 1$.

The results presented below were obtained for a computational domain of $[0, 16.12] \times [-30, 30]$. Periodic boundary conditions were specified in the streamwise direction. The non-reflecting slip boundary conditions of Thompson [27] were imposed at the top and bottom boundaries. A refinement study with three uniform grids with $101 \times 257, 201 \times 513$, and 401×1025 points was carried out. The computational domain was decomposed into 16 equal-sized blocks (4 blocks in each direction). A CFL number of 0.3 was used for all computations.

Fig. 15 shows the vorticity contours of the mixing layer at the dimensionless times $t^* = u_1 t/\delta_\omega(0) = \{0, 5, 10, 15\}$. The vortex pairing generated by the instability of the mixing layer can be seen clearly. Fig. 16 shows the normalized vorticity thickness as a function of non-dimensional time and grid resolution. The grid-refinement study indicates that the solution obtained with the multi-block compact method is converged. Also shown in Fig. 16 is a comparison with the result of



(a) Strong scalability

(b) Weak scalability

Fig. 17. Results of parallel-performance study.

Sandham and Reynolds [19]. Sandham and Reynolds used a sixth-order accurate compact scheme in the streamwise direction and a spectral scheme in cross-stream-wise direction on a 61×81 grid clustered around $y = 0$. Due to the use of a spectral method and grid clustering, their grid resolution is necessarily lower. We can thus use the solution of Sandham and Reynolds as a highly accurate reference solution. It can be seen from Fig. 16 that the solutions computed with the present multi-block compact–WENO method compare very well with the solution of Sandham and Reynolds.

7. Parallel performance

The parallel performance of the multi-block hybrid compact–WENO method is assessed for the compressible mixing-layer problem described in Section 6.5. The number of grid points used depends on the type of performance study as described below. Irrespective of the type of study, the computational domain is split only in the x-coordinate direction, i.e., as depicted in Figs. 12(b) and 13(b). Together with the periodic boundary conditions used in the streamwise direction, this results in perfect load balancing in the sense that not only is the number of grid points per block identical, but also the number of variables to be sent and received per block. The results were obtained on a Linux cluster with AMD Opteron processors rated at 2.2 GHz and with a memory of at least 4 Gb.

7.1. Strong scalability

Strong scalability assesses the parallel performance of a method as an increasing number of processors is used to solve a problem of fixed overall size. The results from strong scalability studies are relevant when additional processors are used to reduce the run-time. As the number of processors increases, the number of points assigned to each processor decreases and therefore the communication cost increases relative to the cost of computation, leading to reduced performance. In our study, we use grids with overall sizes of 2561×101 and 5121×101 points and measure the performance through the speed-up on p processors, $S_p = T_1/T_p$, where T_1 is the time consumed on a single processor and T_p is the time consumed on p processors. The results presented in Fig. 17(a) indicate that speed-ups of about 185 and 205 are obtained with the coarse and fine grids on 256 processors, respectively. The corresponding speed-ups on 512 processors are 190 and 310. Considering that no (five) and five (15) interior points that are not sent or received exist in each block of the coarse (fine) grid on 256 and 512 processors, respectively, this represents excellent performance.

7.2. Weak scalability

Weak scalability studies assess the parallel performance of a method in terms of solving a problem of fixed size per processor as the number of processors increases. The results from such studies are relevant when additional processors are used to increase the overall resolution with the goal of keeping the total run-time constant. In our study, we used grids with 11×101 and 21×101 points per processor on up to 1264 processors. This leaves only nine and 17 interior grid points per block in the x-coordinate direction, respectively. Despite this unfavorable ratio of communication to computation, it can seen from Fig. 17(b) that the efficiency $E_p = T_1/T_p$, is about 95% on 256 processors and about 88% for 1264 processors for the grid with 11 × 101 points per processor. For the grid with 21 × 101 points per processor, the efficiency is nearly constant at 97% up to 1264 processors. These results are substantially better than those reported in the literature for other parallel compact approaches executed on commodity clusters. Because the code has not yet undergone dedicated optimization, it is possible that the parallel performance can be improved further.

8. Conclusions

The main contribution of this paper is the development and assessment of a new multi-block hybrid compact–WENO finite-difference method for the massively parallel simulation of compressible flows. In contrast to earlier efforts at developing parallel compact or hybrid methods, our approach avoids the global dependence of compact methods that adversely affects parallel performance. Instead, the present approach breaks the global dependence through the use of explicit finite-difference approximations at block interfaces. The resulting multi-block method is fifth- and sixth-order accurate for the convective and diffusive fluxes, respectively. A dedicated analysis of the multi-block method for the advection and diffusion equations quantified the impact of the block-interface treatment on stability and accuracy. It was found that the stability of the multi-block method actually improves with increasing number of blocks if the fourth-order accurate Runge–Kutta method is used for time integration. Increasing the number of blocks while keeping the number of grid points constant leads to a slight loss in accuracy. The parallel performance of the method was assessed through strong and weak scalability studies. The multi-block hybrid method attained an efficiency of 88% on 1264 processors of a commodity cluster at the first attempt without dedicated code tuning. This level of performance is much higher than that typically obtained with other strategies for the parallelization of compact methods. Therefore, the new method proposed here may make problems that require massive parallel computing power more amenable to compact and hybrid methods.

Being based on blocks, the new method is also suitable for the computations of flows in or around complex geometries. Overlapping blocks, which are sometimes advantageous for geometries involving moving boundaries, can in principle also be treated. For example, data in the dummy points could be obtained by interpolating rather than simply copying from neighboring blocks. The conservation and robustnaraere327(invould)-2770y(levels.)]TJ1to compac25(are)-230(someti-346(is)-302(347(th(

[28] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, second ed., Springer, 1999.
[29] M. Visbal, D. Gaitonde, High-order accurate methods for complex unsteady subsonic flows, AIAA J. 37 (1999) 1231–1239.
[30] M. Visbal, D. Gaitonde, Very high-order spatially implicit schemes for computational acoustics on curvilinear meshes, J. Comput. Acoust. 9 (2001) 1259–1286.
[31] M. Visbal, D. Gaitonde, On the use of higher-order finite difference schemes on curvilinear and deforming meshes, J. Comput. Phys. 181 (2002) 155–185.
[32] M. Visbal, R. Gordnier, Direct numerical simulation of the interaction of a boundary layer with a flexible panel, AIAA Paper 2001-2721, 2001.
[33] H. Yee, N. Sandham, M. Djomehri, An artificial nonlinear diffusivity method for supersonic reacting flows with shocks, J. Comput. Phys. 150 (1999) 199–239.
[34] X. Zhang, G. Blaisdell, S. Lyrintzis, High-order compact schemes with filters on multi-block domains, J. Sci. Comput. 21 (2004) 321–339.
[35] Q. Zhou, Z. Yao, F. He, M. Shen, A new family of high-order compact upwind difference schemes with good spectral resolution, J. Comput. Phys. 227 (2007) 1306–1339.