Computers and Fluids 159 (2017) 327–337

Contents lists available at ScienceDirect

Computers and Fluids

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

A conservative overlap method for multi-block parallelization of compact finite-volume schemes

F. Capuano[a,*], A. Mastellone[b], E.M. De Angelis[a]

[a] Dipartimento di Ingegneria Industriale (DII), Università di Napoli "Federico II", Napoli, 80125, Italy
[b] Centro Italiano Ricerche Aerospaziali (CIRA), Capua, 81043, Italy

ARTICLE INFO

Article history:
Received 8 April 2017
Revised 18 August 2017
Accepted 11 October 2017
Available online 20 October 2017

Keywords:
Compact schemes
Finite-volume method
Local conservation
MPI Parallel computing
Turbulent flows

ABSTRACT

A conservative approach for MPI-based parallelization of tridiagonal compact schemes is developed in the context of multi-block finite-volume methods. For each block, an enlarged linear system is solved by *overlapping* a certain number of neighbour cells from adjacent sub-domains. The values at block-to-block boundary faces are evaluated by a high-order centered approximation formula. Unlike previous methods, conservation is retained by properly re-computing the common interface value between two neighbouring blocks. Numerical tests show that parallelization artifacts decrease significantly as the number of overlapping cells is increased, at some expense of parallel efficiency. A reasonable trade-off between accuracy and performances is discussed in the paper with reference to both the spectral properties of the method and the results of fully turbulent numerical simulations.

1. Introduction

Compact schemes are widely employed in numerical simulations of turbulent flows, because of their beneficial resolution properties and high overall accuracy [1]. In contrast to classical *explicit* schemes, compact methods are global and require the solution of a tri- or penta-diagonal linear system to provide the desired derivatives or interpolated values [2]. The implicit nature of such methods leads to a difficult implementation in a parallel framework, especially when a MPI-based domain decomposition technique is employed. This issue may preclude the potential benefits of compact methods for direct (DNS) or large-eddy simulations (LES) of turbulence, for which massively parallel computations are mandatory. Several algorithms have been developed over the years to tackle this problem efficiently, each with pros and cons; the approaches fall mainly into three categories, which are briefly reviewed as follows.

The first category is typically referred to as *transpose* methods, and is popular in the pseudo-spectral community. In this case, the computational domain is partitioned along one (or two) dimensions at a time (*slabs* or *pencils* respectively), so that the application of the global scheme can be performed exactly along the remaining direction(s). Then, the computational space is transposed by means of *all-to-all* communications to allow completion of the algorithm [3–5]. Although in this case the parallelization is free of errors (i.e., the solution is identical to the serial one), the resulting approach is very communication intensive, as the collective communications require to exchange a volume of data which increases in size as N^3, being N the number of unknowns of each one-dimensional computation - in contrast with the N^2 scaling of conventional techniques. On the other hand, upon use of optimized libraries (such as MPI_Alltoall and FFTW3) and careful processor mapping, some authors were able to obtain excellent scalability on up to $\mathcal{O}(10^5)$ processors [3,6]. By construction, the transpose method is especially suited for simple cartesian domains, while an appropriate generalization to complex geometries may require involved computer programming to properly take into account the domain topology. Also, the number of processors is limited to no more than N (or N^2) when slabs (or pencils) are used.

Alternatives to the transpose method are represented by algorithmic approaches, i.e., methods that aim to parallelize the solution of the banded linear system. Notable examples include the pipelined Thomas [7], the parallel diagonal dominant [8] (which are limited to tridiagonal systems), and the SPIKE algorithms [9]. These methods are powerful and provide exact (pipelined Thomas, SPIKE) or bounded (parallel diagonal dominant) parallelizations errors, but are usually susceptible to penalties in efficiency due to idle times. Also, the computational complexity of the computer code is generally highly increased. To the authors' knowledge, the use of such methods in computational fluid dynamics (CFD) applications is relatively limited. For a comparison of the pipelined Thomas and parallel diagonal dominant algorithms for flow simu-

* Corresponding author.
E-mail address: francesco.capuano@unina.it (F. Capuano).

https://doi.org/10.1016/j.compfluid.2017.10.017
0045-7930/

lations, see [10]. The SPIKE algorithm has been recently applied to large-eddy simulation in [11].

The third family of methods is constituted by the so-called boundary approximation approach (BAA). In this case, the original linear system is split into disjoint matrix systems that can be solved independently for each sub-domain by exchanging a (small) number of halo cells. The method is thus naturally suited to the widespread MPI-based multi-block partitioning technique, which is at the base of many finite-difference and finite-volume CFD codes. The major drawback of this procedure is that the global dependence of the compact scheme is broken at block-to-block interfaces, leading to a certain degree of deterioration similar to the one occurring at regular boundaries. These effects are mainly attributed to the altered dissipation and dispersion properties due to the boundary closure [12]. However, it is well known that the spectral characteristics of the discretization are crucial for an accurate numerical simulation of multi-scale and acoustics phenomena, such as compressible turbulence [13]. Several boundary approximations have been developed in recent years to reduce parallelization artifacts. These mainly rely on overlapping grids [14,15] or halo points [16] with proper boundary closures at adjacent sub-domain interfaces. In some cases, the resulting schemes are optimized in wavenumber space for accurate acoustic computations, or used in conjunction with suitable filtering operators to remove high-frequency errors [17]. Particularly Gaitonde & Visbal [15] suggested to use a region of overlap between adjacent subdomains aimed to reduce parallelization artifacts in their finite-difference method; boundary closure was achieved by high-order one-sided formulas. However, their approach is not locally conservative, which might be troublesome for flows with shocks [18].

The present work falls within the BAA category and focuses on tridiagonal compact schemes. An overlapping strategy is developed in order to preserve accuracy on the interior points of each sub-domain; however, boundary closure is achieved by explicit centered formulas. Unlike previous studies, which have been concerned with the finite-difference method, a finite-volume (FV) discretization is employed here. A straightforward implementation of the overlapping method in a FV framework leads to the formal loss of local conservation at block-to-block interfaces. In this work, a method aimed to overcome this issue is developed so to retain the inherent conservation properties of the finite-volume method. The proposed algorithm is thus suitable for high-fidelity, parallel computations of compressible shock-free turbulent flows, and could serve as a building block for flows with discontinuities.

The paper is organized as follows. In Section 2, the relevant governing equations are introduced; then, the employed finite-volume discretization and serial numerical method are described. Section 3 presents the parallel method. A modified wavenumber analysis is reported in Section 4, while Section 5 reports a series of numerical tests aimed to characterize the accuracy of the proposed approach. Section 6 focuses on efficiency and parallel performances. Concluding remarks are given in Section 7.

2. Governing equations and serial numerical method

The fully compressible Navier-Stokes equations are considered in this work. In a three-dimensional Cartesian coordinate frame (x, y, z), the motion of a gas with density ρ, velocity $\mathbf{u} = (u, v, w)$, pressure p, temperature T and total energy E is governed by the system:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} + \frac{\partial \mathbf{F}_z}{\partial z} = 0, \tag{1}$$

where $\mathbf{U}$ is the vector of conservative variables, $\mathbf{U} = (\rho, \rho u, \rho v, \rho w, E)$. The total energy E is defined as the sum of internal and kinetic energy, $E = \rho e + \rho |\mathbf{u}|^2/2$. The vectors $\mathbf{F}_x = \mathbf{F}_x^c - \mathbf{F}_x^d$, $\mathbf{F}_y = \mathbf{F}_y^c - \mathbf{F}_y^d$ and $\mathbf{F}_z = \mathbf{F}_z^c - \mathbf{F}_z^d$ represent the fluxes along the three components. The inviscid (convective) fluxes are defined as:

$$\mathbf{F}_x^c = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{F}_y^c = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix},$$

$$\mathbf{F}_z^c = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \tag{2}$$

whereas the diffusive fluxes are

$$\mathbf{F}_x^d = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ \tau_{13} \\ (\tau \mathbf{u})_1 - q_1 \end{pmatrix}, \quad \mathbf{F}_y^d = \begin{pmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ \tau_{23} \\ (\tau \mathbf{u})_2 - q_2 \end{pmatrix},$$

$$\mathbf{F}_z^d = \begin{pmatrix} 0 \\ \tau_{31} \\ \tau_{32} \\ \tau_{33} \\ (\tau \mathbf{u})_3 - q_3 \end{pmatrix}. \tag{3}$$

The stress tensor τ_{ij} and the conductive heat flux q_i are expressed by the usual Newton's and Fourier's laws, respectively $\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)$ and $q_i = -\lambda \frac{\partial T}{\partial x_i}$, where μ is the molecular viscosity and λ the thermal conductivity of the fluid. Closure is achieved by means of the ideal-gas equation of state, $p = \rho RT$.

2.1. Finite-volume discretization

The computational domain is partitioned into a structured grid of hexahedrons indexed by (i, j, k). The finite-volume method is based upon integration of Eq. (1) over a generic control volume Ω, yielding

$$V_{ijk}\frac{\partial \tilde{\mathbf{U}}_{ijk}}{\partial t} + \int_{\partial\Omega} (\mathbf{F}_x n_x + \mathbf{F}_y n_y + \mathbf{F}_z n_z) d\sigma = 0, \tag{4}$$

where $V_{ijk} = |\Omega|$ is the volume of the region and

$$\tilde{\mathbf{U}}_{ijk} = \frac{1}{V_{ijk}} \int_\Omega \mathbf{U} d\Omega. \tag{5}$$

The integral in Eq. (4) applies to each control-volume; as a consequence, surface integrals over inner cell faces cancel out and discrete global conservation of primary unknowns is guaranteed through the telescopic property. The built-in global conservation feature is indeed one of the major advantages of finite-volume methods.

The meaning of the cell integral in Eq. (5) deserves further discussion. In a so-called *pointwise* approach, nodal values $\mathbf{U}_P$ are usually supposed to be known at the cell-center, resulting in a second-order accurate average, i.e., $\tilde{\mathbf{U}} = \mathbf{U}_P + \mathcal{O}(\Delta^2)$, with Δ being a relevant grid spacing. Therefore, if high-order accuracy is seeked, correspondent high-order formulas are needed for the evaluation of

Table 1
Interpolation schemes considered in the work.

Scheme	α	L	γ_1	γ_2	Order
2E	0	1	1/2	0	$\mathcal{O}(\Delta^2)$
4E	0	2	7/12	−1/12	$\mathcal{O}(\Delta^4)$
4C	1/4	1	3/4	0	$\mathcal{O}(\Delta^4)$
6C	1/3	2	29/36	1/36	$\mathcal{O}(\Delta^6)$

the integral [19]. However, a more careful look at Eq. (4) suggests to store and advance in time the cell-averaged values themselves. In this approach, the integral has not to be evaluated and the accuracy of the method depends solely on the fluxes reconstruction [20]. The cell-averaged approach is adopted here and will be used throughout the paper.

The solution algorithm follows that of any typical semi-discrete method, in which Eq. (4) is first semi-discretized in space and then advanced in time. The surface fluxes appearing in Eq. (4) need to be properly approximated. For simplicity, the cell interface $S_{i-1/2}$ along a generic (j, k) line and between cells (i) and $(i-1)$ is considered. The corresponding flux is defined by:

$$\tilde{\mathbf{F}}_{i-1/2} = \int_{S_{i-1/2}} (\mathbf{F}_x n_x + \mathbf{F}_y n_y + \mathbf{F}_z n_z) d\sigma \approx S\big(\tilde{\mathbf{F}}_x n_x + \tilde{\mathbf{F}}_y n_y + \tilde{\mathbf{F}}_z n_z\big), \tag{6}$$

where $S = \left|S_{i-1/2}\right|$. It is assumed that the normal vector is constant along the interface, while the operator $\tilde{(\cdot)}$ denotes surface averaging. Starting from Eq. (6), the discretizations of convective and diffusive fluxes will be described in the next sections.

2.1.1. Convective fluxes

The convective fluxes are evaluated using face-averaged values of conservative variables:

$$\tilde{\mathbf{F}}^c_{i-1/2} = S\big(\mathbf{F}^c_x(\tilde{\mathbf{U}}_{i-1/2})n_x + \mathbf{F}^c_y(\tilde{\mathbf{U}}_{i-1/2})n_y + \mathbf{F}^c_z(\tilde{\mathbf{U}}_{i-1/2})n_z\big). \tag{7}$$

The core of the method, as well as most of the code overall accuracy, lies on a proper interpolation of conservative variables to cell faces. Following [21], face-averaged values are obtained by means of a *cartesian-like* approach, i.e., $\mathbf{U}$ is assumed to depend exclusively on the curvilinear abscissa along each line of the grid. Under this hypothesis, the face-averaged value $\tilde{\mathbf{U}}$ at the interface $(i-1/2, j, k)$ is expressed by the following interpolation formula:

$$\alpha\tilde{\mathbf{U}}_{i-3/2} + \tilde{\mathbf{U}}_{i-1/2} + \alpha\tilde{\mathbf{U}}_{i+1/2} = \sum_{l=1}^{L} \gamma_l\big(\tilde{\mathbf{U}}_{i-l} + \tilde{\mathbf{U}}_{i+l-1}\big). \tag{8}$$

Eq. (8) represents a generic family of explicit ($\alpha = 0$) or implicit ($\alpha \neq 0$) centered interpolation formulas based on cell-averaged values. The coefficients α, L and γ_l determine the order of accuracy and the spectral properties of the scheme. In Table 1, a number of possibilities are given, assuming uniform grid-spacing Δ, along with the corresponding acronym used in the work. The present work will particularly focus on the sixth-order compact method (6C). Note that in the case of non-uniform (Cartesian) grids, the order of accuracy can be preserved by computing the coefficients on the basis of a Taylor series expansion that takes into account the metrics of the mesh [22].

It is worth to remark that, in the context of explicit schemes, a proper computation of convective fluxes can lead to discrete energy-conservation properties, i.e., the convective term does not spuriously contribute to the kinetic energy balance in the inviscid limit [23]. This property guarantees well-known benefits, especially for turbulent flows [24]. However, energy-conserving fluxes for compact schemes in conservation (divergence) form have not yet been found; therefore, the use of Eq. (7) appears to be adequate.

2.1.2. Diffusive fluxes

Unlike convective fluxes, diffusive terms are linear and typically do not raise substantial difficulties. In many CFD applications, they are calculated by a second-order centered formula. The application of a compact scheme to the computation of diffusive fluxes can be carried out by using a scheme similar to Eq. (8), with face-averaged derivatives appearing on the left-hand side. Indeed, diffusive fluxes depend on face-averaged derivatives of velocity and temperature. The use of a compact scheme for the diffusive fluxes did not yield relevant benefits for the tests reported in this work. Therefore, a second-order centred formula has been employed for simplicity. However, it is worth to emphasize that the parallelization algorithm proposed here can be extended in a straightforward manner to the compact computation of diffusive fluxes as well.

2.2. Time advancement

Once the convective and diffusive fluxes have been discretized, Eq. (4) reduces to a system of ordinary differential equations:

$$V_{ijk}\frac{d\bar{\mathbf{U}}_{ijk}}{dt} = \mathbf{R}_{ijk}, \tag{9}$$

where $\mathbf{R}_{ijk}$ is the *residual* resulting from spatial discretization. Time integration from t^n to t^{n+1} is achieved by standard explicit Runge–Kutta schemes:

$$\bar{\mathbf{U}}^{n+1} = \bar{\mathbf{U}}^n + \Delta t \sum_{l=1}^{s} b_p \mathbf{R}\big(\bar{\mathbf{U}}^p\big), \tag{10}$$

$$\bar{\mathbf{U}}^p = \bar{\mathbf{U}}^n + \Delta t \sum_{q=1}^{p-1} a_{pq} \mathbf{R}\big(\bar{\mathbf{U}}^q\big), \tag{11}$$

where s is the number of stages, b_p and a_{pq} are the Runge–Kutta coefficients and the subscript ijk has been omitted for clarity. Runge–Kutta methods are widely used in numerical simulations of both incompressible and compressible turbulence, due to their ease of implementation and relatively large stability footprint. They also have the advantage over multistep methods of being self-starting, i.e., information from previous time-levels is not required. Furthermore, the Runge–Kutta coefficients can be optimized for multiple purposes, e.g., to improve dissipation and dispersion properties for computational acoustics [25], or to enhance the accuracy of the discrete kinetic energy conservation [26,27]. For standard explicit schemes, the temporal order of accuracy is typically equal to the number of stages, at least for $s \leq 4$. The algorithms described in this work are compatible with explicit schemes of arbitrary order; in Section 5, a three-stage, third-order method is employed.

It is worth to remind the reader that the overall linear stability characteristics of the algorithm result from the coupling between the Runge–Kutta method and the spatial scheme used for convective and diffusive terms. For convection-dominated flows, the constraint is usually dictated by the convective operator, yielding $\frac{(|\mathbf{u}|+c)\Delta t}{\Delta x} \leq \frac{\sigma_i}{w'_m}$, where c is the speed of sound, σ_i depends on the stability segment of the time-advancement method on the imaginary axis, and w'_m is the maximum value of the modified wavenumber of the spatial operator [1]. To give an example, third-order Runge–Kutta methods used in conjunction with the schemes 4E, 4C and 6C yield an acoustic CFL limit of 1.262, 1 and 0.871 respectively.

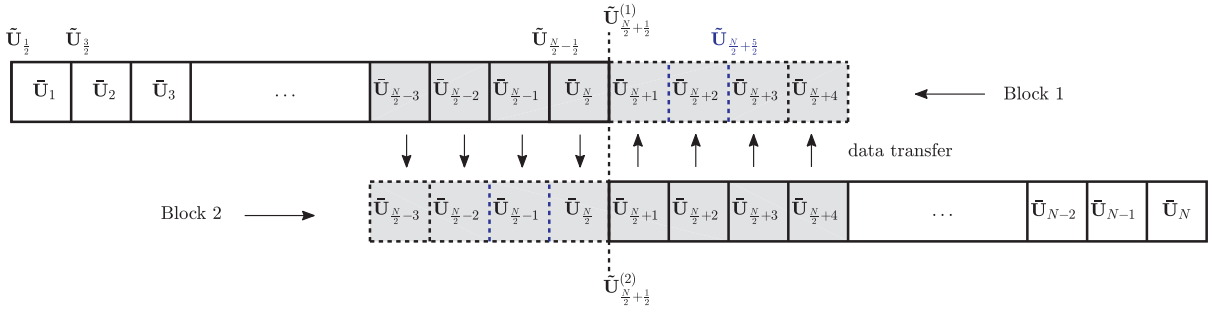

Fig. 1. Diagram describing the interface between two blocks and the transfer of data involved in MPI communications. Overlapping interfaces (here for $M = 2$) are shown in blue, while halo cells in grey. The data transfer involves $N_{ex} = M + 2$ cells per side. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Parallel multi-block method

The objective of this work is to parallelize the algorithm presented in Section 2, and particularly the computation of convective (and/or diffusive) fluxes via the compact formula in Eq. (8). The paradigm considered here is based on a multi-block partitioning technique: the computational domain is split into non-overlapping blocks which are assigned to different processors. The parallelization is achieved by means of conventional MPI communications.

For the sake of clarity, a uniform one-dimensional domain is considered in the following, with N computational cells of spacing Δ and $N + 1$ cell interfaces, labeled with subscripts ranging from $1/2$ to $N/2 + 1/2$. The extension to multi-dimensional computations is straightforward.

3.1. Overlap method

For simplicity, the one-dimensional domain is equally split into two sub-domains, each having $N/2$ cells and $N/2 + 1$ interfaces (see Fig. 1). The objective is to apply the compact scheme of Eq. (8) in such a way that the two sub-domains can be computed independently, subject only to a conventional data exchange. In general, this can be achieved by applying Eq. (8) to each block and treating the shared interface as a regular domain boundary, to be computed by a suitable (explicit or compact) scheme.

In the present work, the compact scheme is applied in each sub-domain to an enlarged system that includes a certain number of *overlapping* interfaces and cells from the neighbour blocks. Only centered formulas are used, and the last interface value is computed by means of a high-order explicit scheme. As a result of the overlapping, the border approximation effects are decreased and the accuracy for the interior points is preserved. From a mathematical point of view, the beneficial effects of the overlapping technique can be attributed to the exponential decay rate of the inverse of tridiagonal banded matrices away from the diagonal [28]. It is also worth to mention that an overlapping-like technique has been rigorously shown to reduce the parallelization error within a given tolerance for tridiagonal Toeplitz matrices [29].

In the following, with reference to Eq. (8), the overlap method is presented for a generic tridiagonal scheme ($\alpha \neq 0$) with $L = 2$, therefore accommodating up to sixth-order accuracy (C6). Of course lower-order accurate cases can be easily recovered, e.g. by setting $\gamma_2 = 0$.

The compact scheme for the first block reads

$$\begin{bmatrix} b.c. & & & & & & & \\ & \ddots & & & & & & \\ & \alpha & 1 & \alpha & & & & \\ & & \alpha & 1 & \alpha & & & \\ & & & \alpha & 1 & \alpha & & \\ & & & & & \ddots & & \\ & & & & & & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{U}t_{1/2} \\ \vdots \\ \\ \tilde{\mathbf{U}}^{(1)}_{N/2+1/2} \\ \mathbf{U}t_{N/2+3/2} \\ \vdots \\ \mathbf{U}t_{N/2+M+1/2} \end{bmatrix}$$

$$= \begin{bmatrix} b.c. & & & & & & & \\ & \ddots & & & & & & \\ \gamma_2 & \gamma_1 & \gamma_1 & \gamma_2 & & & & \\ & \gamma_2 & \gamma_1 & \gamma_1 & \gamma_2 & & & \\ & & & & & \ddots & & \\ & & & \gamma_2 & \gamma_1 & \gamma_1 & \gamma_2 & \\ & & & & b & a & a & b \end{bmatrix} \begin{bmatrix} \mathbf{U}m_1 \\ \vdots \\ \mathbf{U}m_{N/2} \\ \mathbf{U}m_{N/2+1} \\ \vdots \\ \tilde{\mathbf{U}}_{N/2+M+1} \\ \mathbf{U}m_{N/2+M+2} \end{bmatrix},$$

where M is an integer that determines the size of the overlapping region, i.e., the number of additional interface values to be computed (cf. Fig. 1, in which a case with $M = 2$ is shown). Note that at the far left and right sides of the computational domain, some (physical) boundary condition (b.c.) is assumed, whose details are not essential to the purpose of the present discussion. In this case, the left-hand side matrix is square with $(N/2 + M + 1)$ rows, while the right-hand side matrix has dimensions $(N/2 + M + 1) \times (N/2 + M + 2)$. The scheme of Eq. (8) is applied up to $i = N/2 + M$, while the total number of computed interface values is $N/2 + M + 1$, of which only the first $N/2 + 1$ are retained. The last calculated interface value is $N/2 + M + 1/2$, that is evaluated by means of the centered formula

$$\tilde{\mathbf{U}}_{N/2+M+1/2} = a\left(\tilde{\mathbf{U}}_{N/2+M+1} + \tilde{\mathbf{U}}_{N/2+M}\right) + b\left(\tilde{\mathbf{U}}_{N/2+M-1} + \tilde{\mathbf{U}}_{N/2+M+2}\right), \tag{12}$$

where the values a and b can be chosen to provide (at maximum) fourth-order accuracy, which is accomplished with $a = 7/12$ and $b = -1/12$. This boundary closure requires a total number of $N_{ex} = M + 2$ solution data layers to be sent/received to/from each neighbouring block. This can be accomplished by a standard MPI technique. Regarding the formal order of accuracy of the method, it has been shown theoretically in [30] that, for Padé-like finite-volume schemes, the order of accuracy of the global scheme reduces to the one of the boundary closure. Therefore, both the C4 and C6 schemes are expected to be formally fourth-order accurate when used in conjunction with the overlap method presented here. The use of Eq. (12) has been selected among various attempts

aimed to improve the spectral properties of the resulting scheme. Other boundary closures, both symmetric or one-sided, have been tested (but are not shown here). The results were either unsuccessful, or did not provide any relevant advantages, on equal computational cost, over the use of the highest-order centered formula. It is worth to mention that a dedicated optimization in wavenumber space could be carried out to further enhance the scheme properties (see, e.g., [16]). However, this issue has not been taken into account in the present work.

Note that for $M=0$ the two values $\tilde{\mathbf{U}}^{(1)}_{N/2+1/2}$ and $\tilde{\mathbf{U}}^{(2)}_{N/2+1/2}$ are equal, since they are calculated by the explicit formula Eq. (12) based on the same values; as a consequence, the fluxes that are computed from these values are also equal for the two blocks, and local conservation is retained. The resulting scheme is equivalent to the one proposed in [21], and the approach is analogous to the conservative finite-difference compact method developed in [18], in which the interface flux is computed on the basis of the fifth-order accurate approximation of Shu [31]. For $M>0$, the shared interface values are calculated by different schemes, and the method formally loses local conservation. Therefore, without proper remedies, the choice $M\neq 0$ might not be suitable for flows with shocks or discontinuities. The problem could be solved by taking a proper average of $\tilde{\mathbf{U}}^{(1)}_{N/2+1/2}$ and $\tilde{\mathbf{U}}^{(2)}_{N/2+1/2}$ (e.g., a Roe-like average, as proposed in [21]), but this would require an additional communication step between the processors. In this work, the following strategy is proposed to restore the local and global conservation properties of the method.

3.2. Enforcing local conservation (ELC)

The local (and global) conservation condition for the method is given by

$$\tilde{\mathbf{U}}^{(1)}_{N/2+1/2} = \tilde{\mathbf{U}}^{(2)}_{N/2+1/2}. \tag{13}$$

Clearly, Eq. (13) cannot be directly satisfied if $M\neq 0$. Therefore, the two interface values have to be re-computed in each block by a suitable scheme that is based on the same values. A way to achieve this goal is to perform a further application of the same compact formula that involves the maximum possible number of available data. With reference to Fig. 1, the compact scheme extends to all the available (grey) cells $2N_{\mathrm{ex}} = 2M+4$, and reads (for the first block),

$$\begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & \alpha & 1 & \alpha & & & & \\ & & \alpha & 1 & \alpha & & & \\ & & & \alpha & 1 & \alpha & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_{N/2-M+1/2} \\ \vdots \\ \\ \tilde{\mathbf{U}}^{(1)}_{N/2+1/2} \\ \tilde{\mathbf{U}}_{N/2+3/2} \\ \vdots \\ \tilde{\mathbf{U}}_{N/2+M+1/2} \end{bmatrix}$$

$$= \begin{bmatrix} b & a & a & b & & & & \\ & \gamma_2 & \gamma_1 & \gamma_1 & \gamma_2 & & & \\ & & & \ddots & & & & \\ & & \gamma_2 & \gamma_1 & \gamma_1 & \gamma_2 & & \\ & & & b & a & a & b & \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_{N/2-M-1} \\ \vdots \\ \tilde{\mathbf{U}}_{N/2} \\ \tilde{\mathbf{U}}_{N/2+1} \\ \vdots \\ \tilde{\mathbf{U}}_{N/2+M+2} \end{bmatrix}.$$

Note that the matrix coefficient is square of size $(2M+1)$, while the matrix on the right-hand side has size $(2M+1)\times(2M+4)$. The application of this scheme for Block 1 and Block 2 guarantees that Eq. (13) is satisfied. Again, this method proved to be the most effective among a number of possible strategies that have been tested throughout this study. This approach avoids a further exchange of data between processors (with associated risk of idle times) and allows each processor to proceed independently. On the other hand, a small amount of additional computational cost is required to solve the restricted tridiagonal system based on the halo cells.

The ELC method is clearly to be applied for $M\geq 1$.

Table 2
Values of $\|\mathrm{Im}(k'/k)\|_\infty$ for the first two points, for schemes C4 and C6 and for various numbers of overlap layers M.

M	C4		C6	
	$i=1$	$i=2$	$i=1$	$i=2$
0	7.87×10^{-2}	4.12×10^{-2}	1.25×10^{-1}	9.97×10^{-2}
1	2.53×10^{-2}	1.06×10^{-2}	6.32×10^{-2}	3.95×10^{-2}
2	8.84×10^{-3}	2.45×10^{-3}	2.50×10^{-2}	1.35×10^{-2}
3	2.72×10^{-3}	7.81×10^{-4}	1.23×10^{-2}	5.32×10^{-3}
4	6.30×10^{-4}	2.16×10^{-4}	4.45×10^{-3}	2.20×10^{-3}

4. Spectral analysis

The accuracy of the overlapping method is preliminarily assessed by performing a spectral resolution analysis. The scheme is applied to evaluate the derivative of the complex wave e^{ikx}; the numerical solution can be written as $ik'e^{ikx}$, where k' is the so-called *modified* wavenumber. The real and imaginary parts of k' modify the speed and amplitude of the wave, leading to dispersive and diffusive numerical errors respectively. It is well known that these spurious effects have to be minimized as much as possible in problems involving multi-scale phenomena and wave propagation, such as turbulence and acoustics [13].

Results are shown in Figs. 2 and 3 for the C4 and C6 schemes respectively, and for three values of M. Since each partition is fully decoupled from the others, a single block with $N=40$ is considered, with the boundary closure in Eq. (12) applied to both sides of the domain. In contrast to the serial scheme, the modified wavenumbers vary from node to node due to the use of a different scheme at boundaries [21,32]. Most importantly, diffusion and anti-diffusion effects appear at the boundary nodes, although the net dissipation is zero (for linear problems), since only centered schemes are used. Anti-diffusion is especially troublesome as it can lead to artificial amplification of high-frequency modes near block boundaries and thus to a spurious contamination of the computation. Note that the dispersive errors of symmetric nodes coincide (e.g. for $i=1$ and $i=N$), and hence only the curves for $i=1$ and $i=2$ are shown in Figs. 2(a) and 3(a). On the other hand, diffusive errors at such nodes are equal and opposite.

As M is increased, the errors decrease significantly and the solution converges to the serial one, which is purely real and coincides with its finite-difference counterpart [1]

$$k' = \frac{1}{\Delta}\frac{\tilde{\gamma}_1\sin(k\Delta)+\tilde{\gamma}_2\sin(2k\Delta)}{1+2\tilde{\alpha}\cos(k\Delta)}, \tag{14}$$

where $\tilde{\gamma}_{1,2}$ and $\tilde{\alpha}$ are the finite-difference coefficients for fourth- and sixth-order tridiagonal compact schemes. For $M=2$, some degree of anti-diffusion is still present at medium-high wavenumbers, while the errors for $M=4$ are lowered by two orders of magnitude with respect to the $M=0$ case. The ELC condition was applied in all the cases shown in this section, which resulted in a slight improvement of the results with respect to the corresponding non-conservative method. A summary of the maximum errors for the imaginary part of the modified wavenumber is reported in Table 2. It is interesting to note that the errors for the C4 scheme are much lower than those occurring for C6. This is attributed to

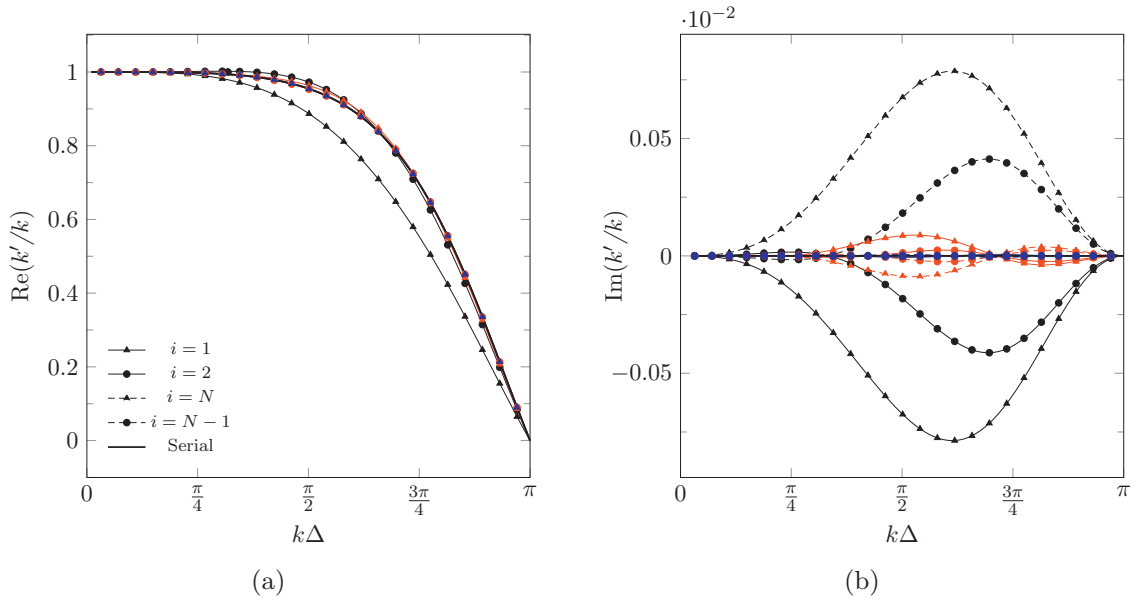

Fig. 2. Modified wavenumber of C4 scheme for the first and last two boundary points, for $M = 0$ (black), $M = 2$ (red) and $M = 4$ (blue). The serial curve is also shown for comparison. (a) real part of the modified wavenumber; (b) imaginary part of the modified wavenumber. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

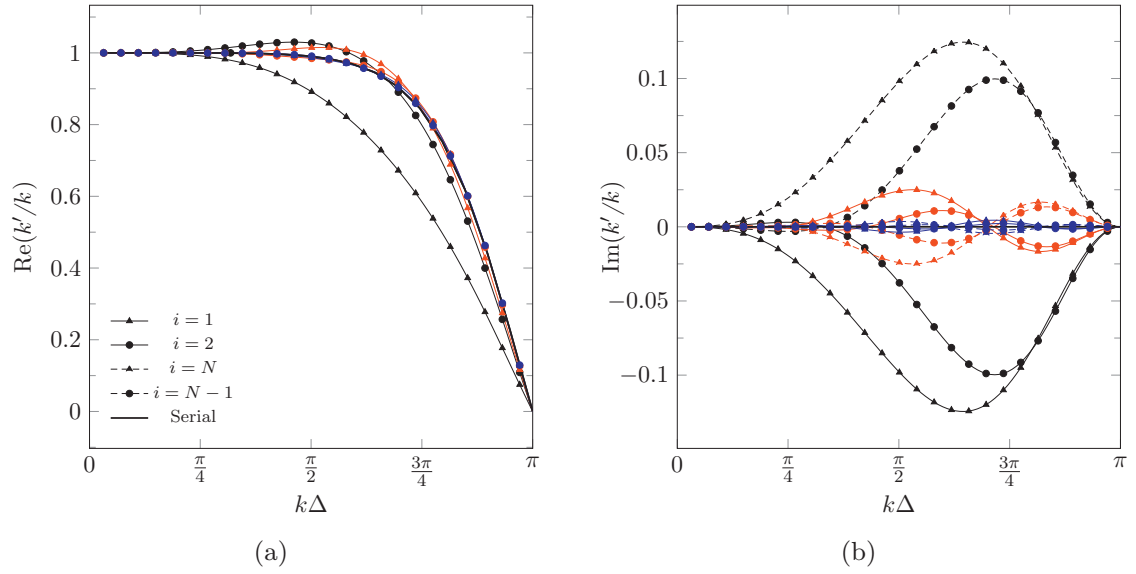

Fig. 3. Modified wavenumber of C6 scheme for the first and last two boundary points, for $M = 0$ (black), $M = 2$ (red) and $M = 4$ (blue). The serial curve is also shown for comparison. (a) real part of the modified wavenumber; (b) imaginary part of the modified wavenumber. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the fact that the *symmetry breaking* effect of the interface treatment is more accentuated in higher-order schemes than in low-order ones. The C6 scheme is thus more challenging to parallelize with a high degree of accuracy with respect to C4. The tests presented in the next section will focus on C6.

The computed spectra of the semi-discretized operators show that the schemes are linearly stable on a uniform mesh, with all the eigenvalues lying in the negative half plane.

From this preliminary (linear) analysis it can be concluded that the overlap method is very effective in reducing the spurious effects that occur as a consequence of breaking the symmetry of the compact difference operator at boundaries. As a rule of thumb, it was found that the maximum error on diffusion decreases of one order of magnitude every 2 added overlap layers.

5. Numerical results

The method presented in the previous sections has been implemented into the CIRA code SPARK-LES, a finite-volume solver that integrates the fully compressible Navier–Stokes equations on multi-block structured grids [33]. Numerical tests are presented in this section to assess the accuracy and performances of the developed technique. The results will be systematically compared to the serial solution. The use of low-pass filters, which has been extensively analyzed in previous studies to suppress anti-diffusion at high frequencies [16,17], has purposely not been considered in this study with the aim of isolating spurious parallelization artifacts. Nonetheless, all the tests presented in the following were stable without the use of any filtering operator.

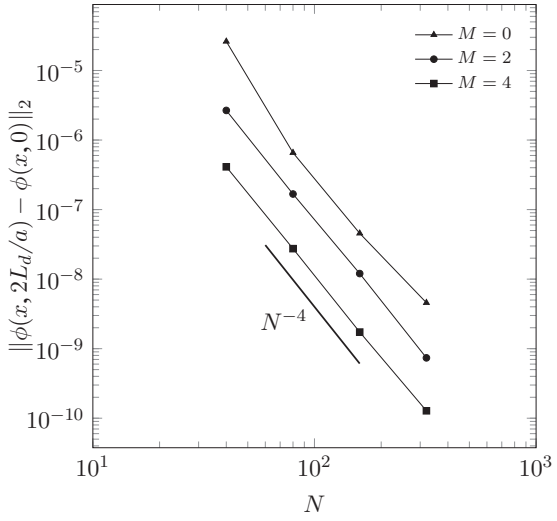

Fig. 4. Order of accuracy study of the multi-block overlap method for the one-dimensional advection equation discretized on 4 blocks, for three values of M.

5.1. Order of accuracy study

The first test aims to verify that the formal order of accuracy is correctly recovered. A grid-refinement study is performed for the one-dimensional advection equation

$$\frac{\partial \phi}{\partial t} + a\frac{\partial \phi}{\partial x}, \tag{15}$$

discretized on a domain $-L_d \leq x \leq L_d$. Periodic boundary conditions are applied to the sides of the domain. The initial condition is a sinusoidal wave $\phi(x, 0) = \sin(\pi x)$. The error is computed from the L_2 norm of the difference between the computed and exact solutions as the wave has travelled a distance of $2L_d/a$. Time advancement is carried out by means of a three-stage, third-order Runge–Kutta method with a CFL number equal to 0.1, which was verified to be sufficiently small to keep time integrations errors negligible with respect to spatial errors.

The domain is split into 4 blocks of equal dimensions and the multi-block treatment is applied at each block-to-block interface as well as to the sides of the computational domain. Results are shown in Fig. 4 for the C6 scheme. In all cases the expected fourth-order accuracy is correctly recovered. Note that the error decreases monotonically as the overlap layer is increased. Other tests (not shown here) performed with a higher number of blocks yielded similar results, with the error norm increasing very slightly with the number of partitions, as also reported in [18].

5.2. Convection of a vortex

The convection of an isentropic vortex is usually employed to test the numerical properties of parallel compact schemes, see for instance [12,16]. In particular, the test is employed here to analyze the effect of spurious acoustic reflections induced by the multi-block treatment as a vortical structure crosses a sub-domain boundary. The Euler equations are integrated over a periodic rectangular domain $x \in [-0.5L_d, 0.5L_d]$ and $y \in [-0.5L_d, 0.5L_d]$ of 320×320 cells. The initial condition is given as follows

$$u(x, y) = U_0 - \frac{\Gamma (y - y_0)}{R^2} \exp\left(-\frac{1}{2} r^2\right), \tag{16a}$$

$$v(x, y) = \frac{\Gamma (x - x_0)}{R^2} \exp\left(-\frac{1}{2} r^2\right), \tag{16b}$$

$$T(x, y) = T_0 - \frac{\Gamma^2}{2R^2 C_p} \exp\left(-r^2\right), \tag{16c}$$

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}/R$, the parameters Γ and R control the strength and the size of the vortex respectively, while C_p is the heat capacity at constant pressure of the gas. In the following, $\Gamma = 1.66$ and $R = L_d/50$ are chosen. The density is computed from the temperature field via isentropic relations; then, pressure is obtained by the equation of state. The dimensional parameters U_0 and T_0 are set to yield a Mach number equal to 0.5. Again, time advancement is obtained by a three-stage, third-order Runge–Kutta scheme with a CFL number equal to 0.5. The vortex is expected to travel without deformation advected by U_0.

It is effective to analyze the behaviour of the vortex as it crosses the interface between two blocks. To this aim, the domain is split in the middle into two sub-domains that are solved in parallel; then, the solution is compared to the serial one in terms of the quantity

$$\varepsilon_p = \frac{|p - p_{\text{serial}}|}{p_\infty - p_{\min}}, \tag{17}$$

that represents the parallelization error normalized by the pressure drop at the vortex core [12]. The vortex starts from $(x, y) = (-0.125L, 0)$ and is stopped at $u_\infty t/L = 0.25$, after travelling a quarter of the computational domain and having crossed the blocks boundary. Results are shown in Fig. 5 for three values of M and for two different ranges of logarithmic contour levels. Parallelization errors are evident for $M = 0$, especially in the upper range, and occupy the entire domain. However, they progressively disappear as M is increased and become more concentrated at the vortex core. Small residual effects are still present at the lowest error levels, but can be considered as negligible, in particular for the case $M = 4$.

In summary, the results confirm the beneficial effects of the overlapping in terms of reducing acoustic reflections at block-to-block interfaces. It appears that, for acoustic reflections to be reduced at negligible levels, a value of $M = 4$ is required.

The local conservation statement of the proposed method has also been assessed. The integral average over the computational domain of the x-momentum

$$\tilde{M}_x(t) = \frac{1}{|\Omega|} \int_\Omega \rho u \, \mathrm{d}\Omega \tag{18}$$

has been computed for one complete turnover of the vortex. Of course, this quantity should be conserved in time due to the periodic nature of the problem and the absence of viscosity. In this case, the domain has been split into 4×4 blocks along the x and y directions. A high value of the non-dimensional vortex strength $\frac{\Gamma}{U_0 R} = 0.2$ has been selected, while the vortex radius is $R = L_d/20$ and the Mach number is again equal to 0.5. Results are shown in Fig. 6 for $M = 2$, with and without the enforcement of local conservation. Without application of the remedy proposed in Section 3.2, non-negligible errors in the conservation of global momentum occur. The time evolution of the error follows a periodic pattern which is repeated four times, due to the presence of four blocks along the x direction. The enforcement of local conservation gives full (to machine accuracy) global momentum preservation.

5.3. Direct numerical simulation of the Taylor–Green Vortex

The three-dimensional Taylor–Green Vortex (TGV) is an excellent benchmark to assess the properties of numerical schemes in situations involving creation of small scales, transition and turbulence, which are the main target applications of the proposed method. This challenging case has been continuously listed among

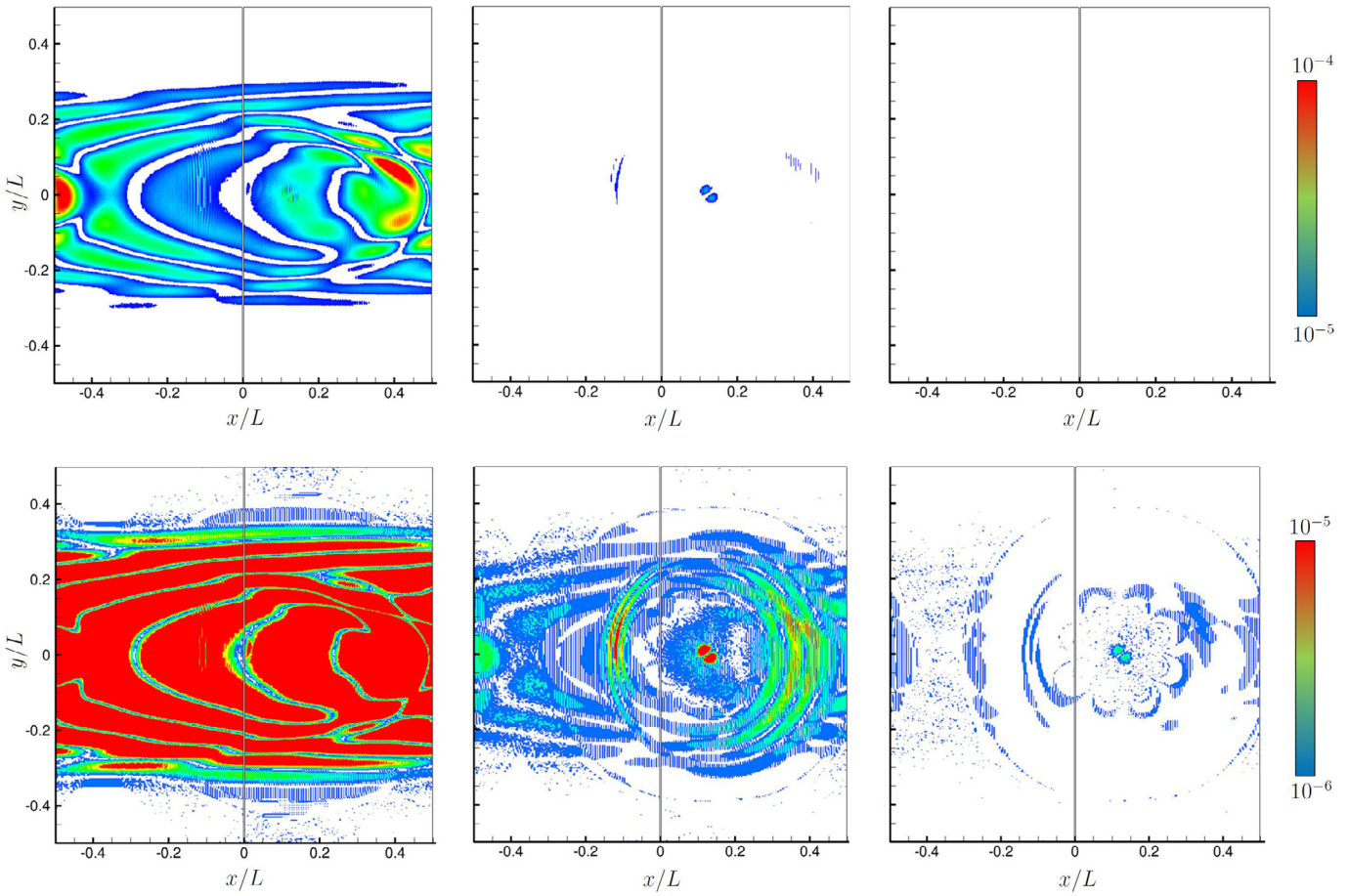

Fig. 5. Contours of ε_p for the isentropic vortex in logarithmic scales, for two different error levels (top and bottom rows) and for $M = 0$ (left column), $M = 2$ (middle column) and $M = 4$ (right column).

the proposed tests in all the editions of the International Workshop for High-Order CFD Methods (see [34] for a review). The following initial conditions are given:

$$u(x, y, z) = U_0 \sin(x/L_d) \cos(y/L_d) \cos(z/L_d), \tag{19a}$$

$$v(x, y, z) = -U_0 \cos(x/L_d) \sin(y/L_d) \cos(z/L_d), \tag{19b}$$

$$w(x, y, z) = 0, \tag{19c}$$

$$p(x, y, z) = p_0 + \frac{\rho_0 U_0^2}{16} [\cos(2x/L_d) + \cos(2y/L_d)] \times [\cos(2z/L_d) + 2], \tag{19d}$$

where the dimensional variables U_0, p_0 and ρ_0, together with the viscosity coefficient, are set to yield a Reynolds number equal to 1600 and a Mach number equal to 0.1. The domain consists of a periodic cube with side of $2\pi L_d$. Two grid resolutions have been analyzed; one is an under-resolved case with 128^3 cells, the other is a marginally resolved direct numerical simulation with 256^3 cells. The ratio of the grid spacing to the estimated Kolmogorov length scale is 4.06 and 2.03 respectively. In both cases, no subgrid-scale model is employed, so that any possible numerical artifact coming from the parallelization strategy is emphasized. All the simulations are time-advanced by means of a three-stage, third-order Runge–Kutta method and a CFL number equal to 0.3. The computational domain has been split uniformly into 8^3 and 16^3 blocks. It has to be emphasized that no filters of any type have been employed in the simulations, which were stable in all the cases analyzed. This is remarkable, since boundary closures analyzed in some previous studies which were stable from a linear analysis, turned out to be unstable in turbulent simulations without the use of proper filtering operators [17].

The first diagnostic parameter is the evolution of the global kinetic energy dissipation rate, integrated over the computational domain

$$\varepsilon = -\frac{d}{dt} \frac{1}{\rho_0 \Omega} \int_\Omega \rho \frac{u_i u_i}{2} d\Omega, \tag{20}$$

as a function of the non-dimensional time $t^* = tU_0/L_d$. Results are shown in Fig. 7. For both resolutions, the parallel simulations show no significant deviations from the serial data in the first part of the time evolution. This is reasonable since the early evolution of the Taylor-Green vortex is rather smooth and a relatively low range of wavenumbers is excited. However, after transition to turbulence ($t^* \approx 9$), the range of scales becomes increasingly wider and spurious numerical effects at medium-high wavenumbers (cf. Fig. 3) cause the $M = 0$ case to deviate from the serial solution, especially for the coarser grid. In both cases, the $M = 2$ case is close to the serial simulation. As the resolution is increased, a broader range of scales is well resolved and the curves tend to collapse. Results obtained with quasi-DNS resolution provide parallel results practically identical to the serial solution and in good agreement with the reference spectral data. The solution with 16^3 blocks also shows remarkably good behavior, despite the computational domain is in this case split into 4096 sub-domains. A reference so-

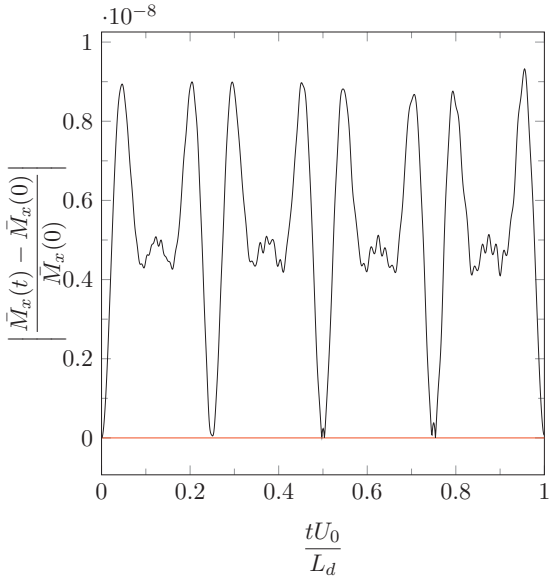

Fig. 6. Time evolution of the relative error on global x-momentum conservation for the inviscid vortex case, with (red line) or without (black line) enforcement of local conservation at block-to-block boundaries. In this case, $M = 2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

lution obtained with a pseudo-spectral method and full DNS accuracy is also reported [35]. It is worth to mention that the deviation from the spectral solution has mainly to be attributed to the marginal resolution of the two grids and to the incomplete resolving efficiency of compact schemes.

Additional insights into the quality of the local flow field can be gained by looking at a snapshot of the instantaneous vorticity magnitude in the early transition phase. In particular, contours of the vorticity norm on a subset of the plane $x = -\pi L_d$ at $t^* = 8$ are shown in Fig. 8. A correct representation of the concentration of vorticity and the development of the shear layer proved to be very challenging for the accuracy and resolution of numerical methods [34]. However, significant deviations from the serial solution can be here observed only for the $M = 0$ case on the coarsest resolution. The results obtained on the quasi-DNS grid are practically identical to plotting accuracy, giving additional fidelity that the local flow field is not spuriously contaminated by parallelization errors.

Table 3
Hardware configurations of Galileo and Marconi.

Item	Galileo	Marconi (Broadwell)
Nodes	516	1512
Cores	8256	54,432
Processor	Intel Haswell	Intel Xeon E5-2697
Frequency	2.40 GHz	2.30 GHz
Memory/node	128GB	128GB
Internal network	Infiniband	Intel OmniPath

6. Parallel performances

The impact of the overlapping technique on the parallel performances has been analyzed in terms of parallel efficiency and CPU time, based on the results obtained for the TGV test case presented in Section 5.3. The tests have been carried out on the CINECA supercomputing facilities Galileo [36] and Marconi (Broadwell partition) [37], whose main characteristics are reported in Table 3. The two architectures mainly differ in terms of number of nodes and interconnection protocol. For the parallel runs, two configurations typical of nowadays fully-resolved direct numerical simulations have been considered, with 512^3 and 1024^3 cells, distributed in 4096 equally-balanced blocks, and with three different values of M equal to 0, 2, 4.

Results are reported in Fig. 9. Specfically, the run time (in seconds) per time-step is shown as a function of the number of processors; note that the run time includes the entire three-step Runge–Kutta time-integration procedure. This so-called strong scaling test is used to evaluate the gain in elapsed time of a parallel run with respect to a serial computation, and is most relevant when one is interested in reducing the computational time by increasing the number of processors, which is typically the case of production CFD runs. Due to memory requirements, the 512^3 and 1024^3 tests could not be run on less than 64 and 256 cores respectively, on both Galileo and Marconi. Good scaling is found

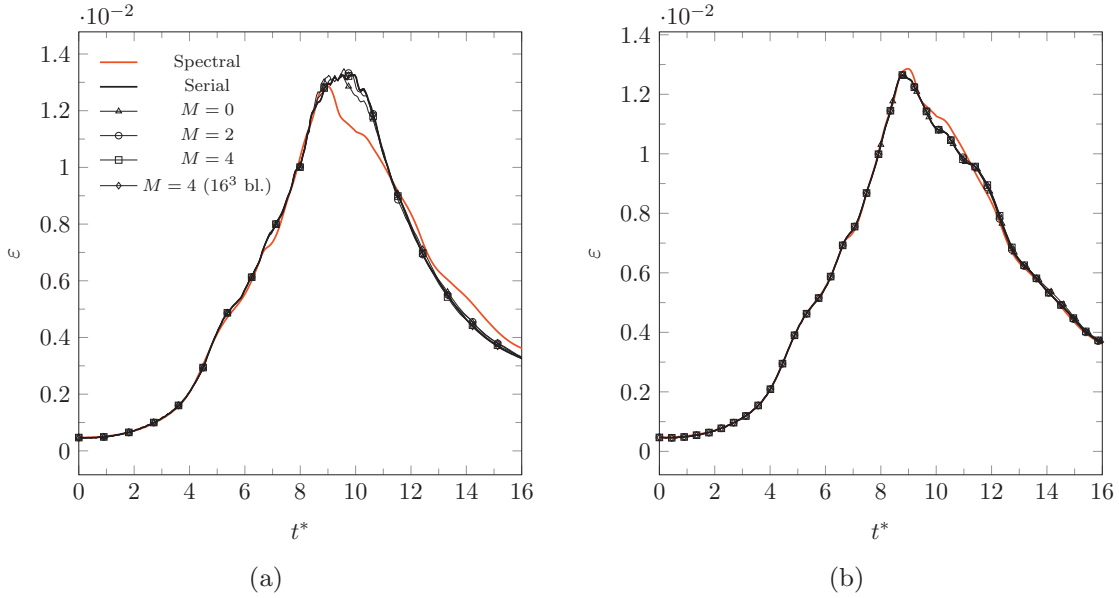

Fig. 7. Time evolution of the energy dissipation rate of the Taylor–Green vortex. In all cases the domain is divided into 8^3 blocks, except where specified. (a) 128^3 cells; (b) 256^3 cells.

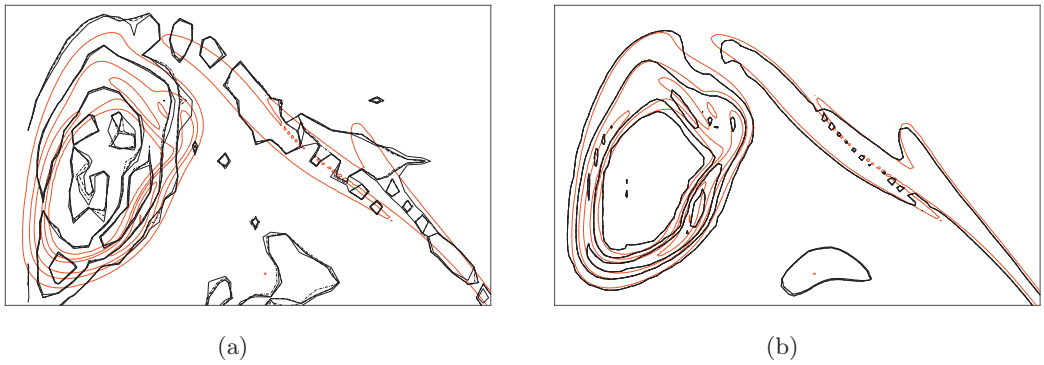

Fig. 8. Contour of dimensionless vorticity norm $\frac{L_d}{U_0}\|\omega\| = 1, 5, 10, 20, 30$ in a subset of the periodic face $\frac{x}{L_d} = \pi$ at time $t^* = 8$ for various values of M. Shown are the spectral solution (red line), the parallel solutions for $M = 0$ (thin black line), $M = 2$ (dash-dotted), $M = 4$ (dotted), $M = 4$ and 16^3 blocks (dashed), and the serial solution (thick black line). (a) 128^3 grid; (b) 256^3 grid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

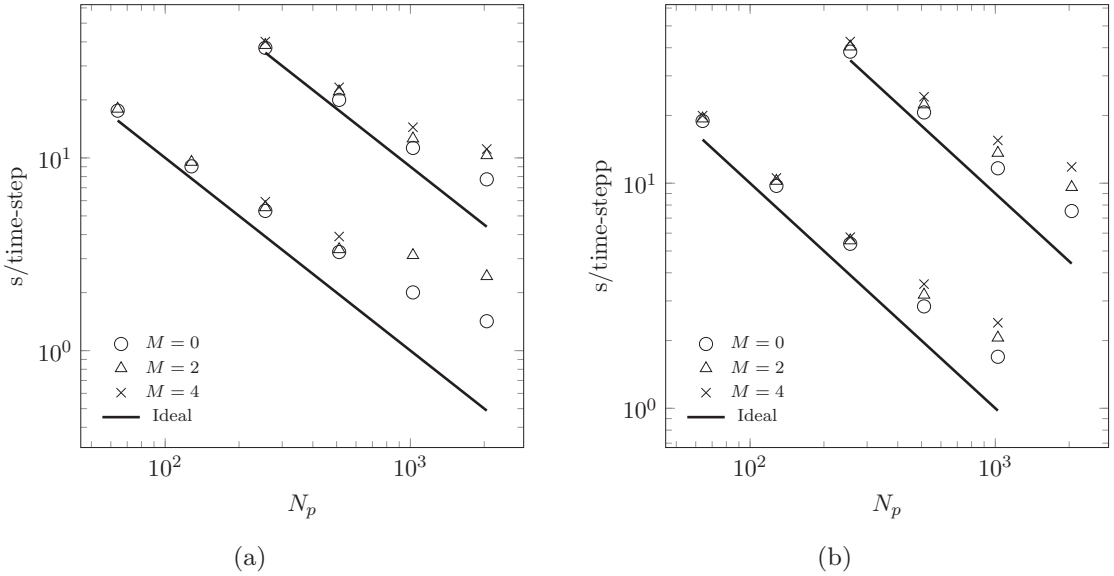

Fig. 9. Parallel performances of the overlapping approach for the 512^3 and 1024^3 configurations. (a) Galileo (b) Marconi.

on up to 1024 processors for both grids and on both platforms, with Marconi showing slightly superior speedups with respect to Galileo. For instance, in the baseline $M = 0$ case the parallel efficiency (defined as the ratio between the actual speedup and the ideal speedup) for the 512^3 configuration was 70% on Marconi and 54% on Galileo; on the other hand, the largest grid achieved a remarkable 83% for 1024 CPU cores on both architectures. Increasing the number of overlap layers has obviously a detrimental effect on parallel performances, due to the additional amount of data to be exchanged among the processors; in this regard, it is worth to recall here that the number of exchanged cell layers vary as $N_{ex} = M + 2$. The observed reduction in parallel efficiency, with respect to the $M = 0$ case, ranges on average from 15% to 25% for the 512^3 grid and from 10% to 15% for the 1024^3 one. In practice, the $M = 2$ overlap appears to be as a reasonable trade-off between accuracy and performances, leading to parallel efficiencies as high as 60% and 75% on 1024 processors for the small and the large grid respectively. When a number of processors higher than 2048 was attempted, the network latency started to become dominant, leading to performances flattening (especially for higher values of M), as a consequence of assigning a too small workload to each process. The number of CPU cores at which the speedup saturates is highly case- and machine-dependent [3] and might be increased by a proper code-platform tuning, which has not been pursued here.

The total run time is also affected by the overlapping, as a consequence of the over-sized linear system to be solved by each processor, and most importantly, of the MPI communication overhead. However, the increase in run times, on equal number of CPU cores, is moderate and ranges from about 5% to 15% when going from $M = 0$ to, e.g., $M = 2$.

It is worth to note that the above results were obtained without any dedicated code optimization; therefore, it is possible that the parallel performances shown in this section could be further enhanced. Also, analogous results were obtained at the first attempt on another Linux cluster owned by CIRA and equipped with a similar hardware as Marconi, showing that the BAA method can offer good performances also on unsophisticated commodity clusters.

7. Conclusions

A systematic method for the parallelization of tridiagonal compact finite-volume schemes has been developed. The algorithm is based on an overlapping technique, i.e., an over-extended linear system is solved for each sub-domain, with the aim of preserving the accuracy of the interior solution. Boundary closure is achieved by a centered fourth-order formula. In contrast to previous strategies based on overlap-like techniques, the method retains local and global conservation of primary invariants by properly re-computing the shared boundary values without losing accuracy and with

little additional computational cost. The inherent local conservation properties of the finite-volume method are thus retained by the present approach. The proposed idea of enforcing local conservation can be equally applied to any finite-difference method written in conservation form. The method can be implemented straightforwardly and is especially suited for multi-block-based parallel CFD codes, thus being applicable for the simulation of flows in complex geometries or with moving boundaries. The method has been proposed and tested on uniform cartesian meshes but can be readily applied to curvilinear meshes as well.

A linear Fourier analysis has shown that the developed method is able to preserve the spectral properties of the original scheme within an acceptable tolerance, with the errors rapidly decreasing as the overlap region M, i.e., the number of additional interfaces computed for each side of the block, is increased. Numerical tests showed that the method achieves the theoretically expected order of accuracy; in terms of spurious acoustic reflections at block-to-block interfaces, an isentropic vortex convection test demonstrated that a significant reduction can be accomplished already for $M = 2$, while $M = 4$ led to almost negligible parallelization errors. The method has finally been tested on a challenging turbulent case, the three-dimensional Taylor-Green vortex, involving transition to turbulence, creation of small scales and turbulent decay. Results were remarakbly satisfactory, yielding solutions very close to the serial one both in terms of integral quantities (kinetic energy dissipation rate) and local flow field analysis (vorticity magnitude prior to transition). Significant deviations from the serial solution were observed only for the case $M = 0$ on the coarsest grid resolution, but disappeared for $M = 2$. The schemes were stable in all cases without using any filtering operator.

From a point of view of computational performances, the choice of the overlapping region size is a trade-off between accuracy and parallel efficiency: numerical results suggest that $M = 2$ might be a reasonable choice, although the value of M could be selected depending on the specific problem under study. Without any dedicated optimization, the case with $M = 2$ attained a parallel efficiency in a strong scaling test as high as 75% on 1024 processors, thus making it appealing for efficient computations on commodity clusters. Accurate computational aeroacoustics analyses might require higher values of M, with unavoidable penalties in parallel performances. In this regard, it is worth to remark that dedicated optimizations in wavenumber space, such as those carried out in [16] for pentadiagonal schemes, could be developed to further improve the performances of the schemes. However, these optimizations have not been undertaken in this study and are left for future work.

Acknowledgements

This research work was partially funded by the HYPROB Program of the Italian Minister of Research. The authors gratefully acknowledge the members of the CFD laboratory of the HYPROB team. F. Capuano is grateful to G. Coppola and L. de Luca for useful comments and discussions on the manuscript. Part of this work was supported by a grant of HPC time from CINECA under the ISCRA project TESLAP.

References

[1] Lele SK. Compact finite difference schemes with spectral–like resolution. J Comput Phys 1992;103:16–42.
[2] Coppola G, Meola C. Generalization of the spline interpolation based on the principle of the compact schemes. J Sci Comput 2002;17:747–60.
[3] Laizet S, Li N. Incompact3d: a powerful tool to tackle turbulence problems with up to $O(10^5)$ computational cores. Int J Numer Meth Fluids 2011;67:1735–57.
[4] 2Decomp & FFT. http://www.2decomp.org/.
[5] FFTW. http://www.fftw.org/fftw2_doc/fftw_4.html/.
[6] Cook AW, Cabot WH, Williams PL, Miller BJ, de Supinski BR, Yates RK, et al. Tera-scalable algorithms for variable-density elliptic hydrodynamics with spectral accuracy. In: Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 conference. IEEE; 2005. p. 60.
[7] Povitsky A, Morris PJ. A higher-order compact method in space and time based on parallel implementation of the thomas algorithm. J Comput Phys 2000;161(1):182–203.
[8] Sun X-H, Moitra S. A fast parallel tridiagonal algorithm for a class of CFD applications, 3585. NASA technical paper; 1996.
[9] Polizzi E, Sameh AH. A parallel hybrid banded system solver: the spike algorithm. Parallel Comput 2006;32(2):177–94.
[10] Ladeinde F, Cai X, Visbal M, Gaitonde D. Parallel implementation of curvilinear high-order formulas. Int J Comput Fluid D 2003;17(6):467–85.
[11] Situ Y, Martha CS, Louis ME, Li Z, Sameh AH, Blaisdell GA, et al. Petascale large eddy simulation of jet engine noise based on the truncated spike algorithm. Parallel Comput 2014;40(9):496–511.
[12] Kim JW. Quasi-disjoint pentadiagonal matrix systems for the parallelization of compact finite-difference schemes and filters. J Comput Phys 2013;241:168–94.
[13] Colonius T, Lele SK. Computational aeroacoustics: progress on nonlinear problems of sound generation. Prog Aerosp Sci 2004;40:345–416.
[14] Sengupta TK, Dipankar A, Rao AK. A new compact scheme for parallel computing using domain decomposition. J Comput Phys 2007;220:654–77.
[15] Gaitonde DV, Visbal MR. Padé-type higher-order boundary filters for the Navier–Stokes equations. AIAA J 2000;38:2103–12.
[16] Kim JW, Sandberg RD. Efficient parallel computing with a compact finite difference scheme. Comp Fluids 2012;58:70–87.
[17] Zhang X, Blaisdell GA, Lyrintzis AS. High-order compact schemes with filters on multi-block domains. J Sci Comput 2004;21(3):321–39.
[18] Chao J, Haselbacher A, Balachandar S. A massively parallel multi-block hybrid compact–WENO scheme for compressible flows. J Comput Phys 2009;228(19):7473–91.
[19] Ferziger JH, Perić M. Computational methods for fluid dynamics. Springer; 2013.
[20] Lacor C, Smirnov S, Baelmans M. A finite volume formulation of compact central schemes on arbitrary structured grids. J Comput Phys 2004;198:535–66.
[21] Fosso P, Deniau H, Sicot F, Sagaut P. Curvilinear finite-volume schemes using high-order compact interpolation. J Comput Phys 2010;229:5090–122.
[22] Gamet L, Ducros F, Nicoud F, Poinsot T, et al. Compact finite difference schemes on non-uniform meshes. application to direct numerical simulations of compressible flows. Int J Numer Methods Fluids 1999;29(2):159–91.
[23] Ducros F, Laporte F, Soulères T, Guinot V, Moinat P, Caruelle B. High-order fluxes for conservative skew-symmetric-like schemes in structured meshes: application to compressible flows. J Comput Phys 2000;161:114–39.
[24] Capuano F, Coppola G, Balarac G, de Luca L. Energy preserving turbulent simulations at a reduced computational cost. J Comput Phys 2015a;298:480–94.
[25] Hu F, Hussaini M, Manthey J. Low-dissipation and low-dispersion Runge–Kutta schemes for computational acoustics. J Comput Phys 1996;124:177–91.
[26] Capuano F, Coppola G, de Luca L. An efficient time advancing strategy for energy-preserving simulations. J Comput Phys 2015b;295:209–29.
[27] Capuano F, Coppola G, Rández L, de Luca L. Explicit Runge–Kutta schemes for incompressible flow with improved energy-conservation properties. J Comput Phys 2017;328:86–94.
[28] Nabben R. Decay rates of the inverse of nonsymmetric tridiagonal and band matrices. SIAM J Matrix Anal Appl 1999;20:820–37.
[29] McNally JM, Garey LE, Shaw RE. A communication-less parallel algorithm for tridiagonal toeplitz systems. J Comput Appl Math 2008;212:260–71.
[30] Kobayashi MH. On a class of Padé finite volume methods. J Comput Phys 1999;156(1):137–80.
[31] Shu C-W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Advanced numerical approximation of nonlinear hyperbolic equations. Springer; 1998. p. 325–432.
[32] Sengupta TK, Ganeriwal G, De S. Analysis of central and upwind compact schemes. J Comput Phys 2003;192(2):677–94.
[33] Capuano F, Mastellone A, Di Benedetto S, Cutrone L, Schettino A. Preliminary developments towards a high-order and efficient LES code for propulsion applications. In: Proceedings of the jointly organized WCCM XI-ECCM V-ECFD VI; 2014. p. 7569–80.
[34] Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, et al. High-order CFD methods: current status and perspective. Int J Numer Methods Fluids 2013;72(8):811–45.
[35] van Rees WM, Leonard A, Pullin D, Koumoutsakos P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high reynolds numbers. J Comput Phys 2011;230:2794–805.
[36] Galileo. http://www.hpc.cineca.it/hardware/galileo.
[37] Marconi. http://www.hpc.cineca.it/hardware/marconi.