Preprint ANL/MCS-P5121-0414

EFFICIENT IMPLEMENTATION OF NONLINEAR COMPACT SCHEMES ON MASSIVELY PARALLEL PLATFORMS*

DEBOJYOTI GHOSH[†‡], EMIL M. CONSTANTINESCU[†§], AND JED BROWN[†¶]

Abstract. Weighted nonlinear compact schemes are ideal for simulating compressible, turbulent flows because of their nonoscillatory nature and high spectral resolution. However, they require the solution to banded systems of equations at each time-integration step or stage. We focus on tridiagonal compact schemes in this paper. We propose an efficient implementation of such schemes on massively parallel computing platforms through an iterative substructuring algorithm to solve the tridiagonal system of equations. The key features of our implementation are that it does not introduce any parallelization-based approximations or errors and it involves minimal neighbor-to-neighbor communications. We demonstrate the performance and scalability of our approach on the IBM Blue Gene/Q platform and show that the compact schemes are efficient and have performance comparable to that of standard noncompact finite-difference methods on large numbers of processors ($\sim 500,000$) and small subdomain sizes (4 points per dimension per processor).

Key words. compact schemes, WENO, CRWENO, high-performance computing, compressible flows

AMS subject classifications. 65M-04, 65F-04, 76N-04

1. Introduction.

Weighted, nonlinear compact schemes use the adaptive stencil selection of the weighted, essentially nonoscillatory (WENO) [27, 44] schemes to yield essentially nonoscillatory solutions with high spectral resolution; they are thus ideal for simulating compressible, turbulent flows. Notable efforts include weighted compact nonlinear schemes (WCNS) [11, 12, 50, 48], hybrid compact-ENO/WENO schemes [4, 3, 36, 40], weighted compact schemes (WCS) [28, 31, 49], compact-reconstruction WENO (CRWENO) schemes [17, 20, 16, 18], and finite-volume compact-WENO (FVCW) schemes [23]. These schemes show a significant improvement in the resolution of moderate- and small-length scales compared with the resolution of the standard WENO schemes of the same (or higher) order and were applied to the simulation of compressible, turbulent flows. The WCNS schemes [12, 50, 48] result in a system of equations with a linear left-hand side that can be prefactored. This is a substantial advantage; however, the spectral resolution of these schemes is only marginally higher than that of the WENO scheme. The hybrid compact-WENO, WCS, CRWENO, and FVCW schemes have a significantly higher spectral resolution, as demonstrated by both linear and nonlinear spectral analyses [36, 18]. They result in solution-dependent systems of equations at each time-integration step or stage. Tests have shown that on a single processor the additional cost of solving the tridiagonal system of equations is justified by the lower absolute errors and higher resolution of small-length scales and discontinuities [17, 16]; moreover, the CRWENO schemes are less expensive than

*This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research program, under contract DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357. We acknowledge Dr. Paul Fischer (Argonne National Laboratory) for his help and guidance in the parallel implementation of the tridiagonal solver.

[†]Mathematics & Computer Science Division, Argonne National Laboratory, Argonne, IL 60439

[‡]ghosh@mcs.anl.gov

[§]emconsta@mcs.anl.gov

[¶]jedbrown@mcs.anl.gov

the WENO schemes when comparing solutions of comparable accuracy or resolution. A quantitative analysis of the numerical cost of hybrid compact-WENO has not yet been presented in the literature; however, the conclusions regarding the CRWENO scheme are also expected to hold true for the hybrid schemes, since the computational complexity is similar.

An efficient, parallel solver for banded systems is thus a crucial issue in the implementation of nonlinear compact schemes on distributed-memory platforms. Past attempts have followed three approaches. One approach is to decouple the global system of equations into separate systems inside each subdomain by applying the noncompact WENO scheme [7] or biased compact schemes [29] at the interior (parallel subdomain) boundaries. This decoupling causes the global numerical properties to be a function of the number of processors. Specifically, the spectral properties of the compact scheme get compromised [7] as the number of processors increases for a given problem size, and numerical errors are observed [29]. A second approach is a parallel implementation of the tridiagonal solver, such as the pipelined Thomas algorithm [39] in which the idle time of the processors during the forward and backward solves is used to carry out nonlocal data-independent computations or local data-dependent Runge-Kutta step completion calculations. This algorithm requires a complicated static schedule of communications and computations, however, resulting in a trade-off between communication and computation efficiencies. A reduced parallel diagonally dominant [46] algorithm solves a perturbed linear system, introducing an error because of an assumption of diagonal dominance that is bounded. A third approach, involving data transposition [10, 21], collects the entire system of equations on one processor and solves it sequentially. This requires the transposition of "pencils" of data between processors. The approach is communication intensive; indeed, a large fraction of the total execution time is spent in the data transposition operations. Because of these drawbacks, massively parallel simulations of turbulent flows, such as [5], have been limited to using standard (noncompact) finite-difference methods with limited spectral resolution. We note that several implementations of a parallel tridiagonal solver [45, 26, 34, 47, 8, 13, 37, 38, 35, 15] have been proposed, although they have not been applied specifically to compact finite-difference schemes.

This paper presents a parallel implementation of nonlinear, tridiagonal compact schemes with the following aims that address the drawbacks of past approaches: the overall algorithm does not suffer from parallelization-related approximations or errors that are larger than the discretization errors, the implementation does not require complicated scheduling, and the overall scheme is computationally more efficient (compared with a standard finite-difference scheme) at subdomain sizes (points per processor) of practical relevance. This implementation will make the compact schemes viable for simulations such as that presented in [5]. The tridiagonal system is solved on multiple processors by using a substructuring approach [47, 13, 38, 35, 15], and an iterative method is used for the reduced system [38]. Arguably, this approach may not perform well in general, since several iterations may be required for an accurate solution [38]. We show here, however, that the reduced systems resulting from a compact finite-difference discretization are characterized by strong diagonal dominance, and thus one can obtain solutions of sufficient accuracy with few iterations. We stress here that "sufficient accuracy" implies that the error in the solution of the reduced system is insignificant compared with the discretization errors; in other words, multiprocessor and serial solutions are identical. We also show that one can specify a priori the number of Jacobi iterations for a given problem based on the number of

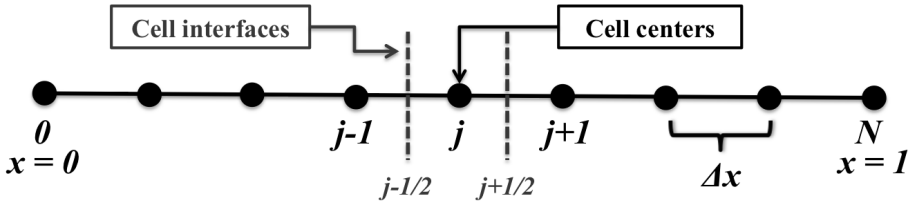


FIG. 2.1. *Discretized one-dimensional domain.*

grid points and number of processors and can avoid a norm-based exit criterion. We demonstrate the performance and scalability of our parallel compact scheme based on the iterative substructuring approach. Results are presented with the fifth-order CRWENO scheme [17]; however, this approach is also applicable to other nonlinear compact (hybrid compact-WENO, WCS, and FVCW) schemes that result in tridiagonal systems of equations.

The paper is organized as follows. Section 2 describes the numerical method and the CRWENO scheme, as well as the numerical properties that motivate its use. Section 3 describes our parallel implementation of these algorithms and presents a scalability analysis of the proposed method for some representative, manufactured problems. Large processor-count simulations of physically relevant problems are reported in Section 4. Conclusions are drawn in Section 5.

2. Numerical Method. A hyperbolic partial differential equation (PDE) can be expressed as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}\left(\mathbf{u}\right)}{\partial \mathbf{x}} = 0, \ \mathbf{x} \in \Omega, \tag{2.1}$$

$$\mathbf{u}\left(\mathbf{x}, t\right) = \mathbf{u}_{\Gamma}\left(t\right), \ \mathbf{x} \in \Gamma, \tag{2.2}$$

$$\mathbf{u}\left(\mathbf{x}, 0\right) = \mathbf{u}_{0}\left(\mathbf{x}\right), \ \mathbf{x} \in \Omega, \tag{2.3}$$

where $\mathbf{x}$ is the position vector in space, t is time, $\mathbf{u}$ is the conserved solution, and $\mathbf{f}\left(\mathbf{u}\right)$ is the hyperbolic flux function. The domain is given by Ω with the boundary as Γ. The boundary conditions and the initial solution are specified by $\mathbf{u}_{\Gamma}\left(t\right)$ and $\mathbf{u}_{0}\left(\mathbf{x}\right)$, respectively. A conservative, finite-difference discretization of (2.1) in space results in an ordinary differential equation (ODE) in time. Figure 2.1 shows an example of a one-dimensional domain of unit length discretized by a grid with $(N+1)$ points. The corresponding semi-discrete ODE for this domain is given by

$$\frac{d\mathbf{u}_j}{dt} + \frac{1}{\Delta x}\left(\mathbf{h}_{j+1/2} - \mathbf{h}_{j-1/2}\right) = 0, \tag{2.4}$$

where $\mathbf{u}_j = \mathbf{u}(x_j)$; $x_j = j\Delta x$ is the cell-centered solution and $\mathbf{h}_{j\pm1/2}$ is the numerical flux at the interface. The numerical flux function $\mathbf{h}(x)$ is required to satisfy exactly

$$\left.\frac{\partial \mathbf{f}}{\partial x}\right|_{x=x_j} = \frac{1}{\Delta x}[\mathbf{h}(x_{j+1/2}, t) - \mathbf{h}(x_{j-1/2}, t)], \tag{2.5}$$

where $\mathbf{f}(\mathbf{u})$ is the flux function and can thus be defined implicitly as

$$\mathbf{f}(x) = \frac{1}{\Delta x}\int_{x-\Delta x/2}^{x+\Delta x/2} \mathbf{h}(\xi) d\xi. \tag{2.6}$$

Equation (2.4) is numerically integrated in time by the classical fourth-order, four-stage or the strong stability-preserving three-stage, third-order Runge-Kutta schemes [22]. The reconstruction of the numerical flux at the interface ($\hat{f}_{j+1/2} \approx \mathbf{h}_{j+1/2}$) from the cell-centered flux ($\mathbf{f}_j = \mathbf{f}(\mathbf{u}_j)$) with the fifth-order WENO and CRWENO schemes is summarized in the following paragraphs. The left-biased reconstruction for a scalar quantity is described, and the corresponding expressions for the right-biased reconstruction are trivially obtained. Vector quantities are reconstructed by applying the scalar reconstruction method to each component.

2.1. WENO Schemes. The WENO schemes [32, 27] use adaptive stenciling to achieve high-order accuracy when the solution is smooth, and they yield nonoscillatory solutions across discontinuities. At a given interface, there are r candidate stencils of rth-order accuracy, with optimal coefficients such that the weighted sum results in a $(2r-1)$th-order interpolation. The optimal weights are scaled by the local smoothness of the solution to obtain the nonlinear WENO weights. The final scheme is the weighted sum of the rth-order stencils with the nonlinear weights. The fifth-order WENO scheme is constructed by three third-order schemes:

$$\hat{f}^1_{j+1/2} = \frac{1}{3}f_{j-2} - \frac{7}{6}f_{j-1} + \frac{11}{6}f_j,\ c_1 = \frac{1}{10}, \tag{2.7}$$

$$\hat{f}^2_{j+1/2} = -\frac{1}{6}f_{j-1} + \frac{5}{6}f_j + \frac{1}{3}f_{j+1},\ c_2 = \frac{6}{10}, \tag{2.8}$$

$$\hat{f}^3_{j+1/2} = \frac{1}{3}f_j + \frac{5}{6}f_{j+1} - \frac{1}{6}f_{j+2},\ c_3 = \frac{3}{10}. \tag{2.9}$$

By multiplying each of (2.7)–(2.9) with their optimal coefficient $c_k,\ k = 1, 2, 3$, and then adding the three, we obtain a linear, fifth-order interpolation scheme:

$$\hat{f}_{j+1/2} = \frac{1}{30}f_{j-2} - \frac{13}{60}f_{j-1} + \frac{47}{60}f_j + \frac{27}{60}f_{j+1} - \frac{1}{20}f_{j+2}. \tag{2.10}$$

The nonlinear weights are computed from the optimal coefficients and local solution smoothness as [27]

$$\omega_k = \frac{\alpha_k}{\sum_k \alpha_k};\ \alpha_k = \frac{c_k}{(\epsilon + \beta_k)^p};\ k = 1, 2, 3, \tag{2.11}$$

where $\epsilon = 10^{-6}$ is a small number to prevent division by zero. The smoothness indicators (β_k) for the stencils are given by

$$\beta_1 = \frac{13}{12}(f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4}(f_{j-2} - 4f_{j-1} + 3f_j)^2, \tag{2.12}$$

$$\beta_2 = \frac{13}{12}(f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4}(f_{j-1} - f_{j+1})^2, \tag{2.13}$$

$$\text{and } \beta_3 = \frac{13}{12}(f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4}(3f_j - 4f_{j+1} + f_{j+2})^2. \tag{2.14}$$

A mapping function was proposed for these weights [24] to address the drawbacks of this definition of the weights, and this approach is adopted here. By multiplying (2.7)–(2.9) by the nonlinear weights (instead of the optimal coefficients c_k) and then adding the three, we obtain the fifth-order WENO (WENO5) scheme:

$$\begin{aligned}\hat{f}_{j+1/2} = \frac{\omega_1}{3}f_{j-2} - \frac{1}{6}(7\omega_1 + \omega_2)f_{j-1} + \frac{1}{6}(11\omega_1 + 5\omega_2 + 2\omega_3)f_j \\ + \frac{1}{6}(2\omega_2 + 5\omega_3)f_{j+1} - \frac{\omega_3}{6}f_{j+2}.\end{aligned} \tag{2.15}$$

When the solution is smooth, $\omega_k \to c_k$, and (2.15) reduces to (2.10). An elaborate description of the WENO5 scheme, including discussion of the value of ϵ, is available in [27, 24].

2.2. CRWENO Schemes. Compact schemes use an implicitly-defined function to compute the flux at the interfaces; the numerical flux at an interface depends on the numerical flux at neighboring interfaces (as well as the known flux at the cell centers). Therefore, they require the solution to a system of equations. This dependence results in higher spectral resolution and lower absolute errors compared with standard finite-difference schemes of the same order of convergence. The CRWENO scheme applies solution-dependent weights to compact candidate stencils. A fifth-order CRWENO scheme [17, 16] is constructed by considering three third-order compact interpolation schemes:

$$\frac{2}{3}\hat{f}_{j-1/2} + \frac{1}{3}\hat{f}_{j+1/2} = \frac{1}{6}\left(f_{j-1} + 5f_j\right);\ c_1 = \frac{2}{10}, \tag{2.16}$$

$$\frac{1}{3}\hat{f}_{j-1/2} + \frac{2}{3}\hat{f}_{j+1/2} = \frac{1}{6}\left(5f_j + f_{j+1}\right);\ c_2 = \frac{5}{10}, \tag{2.17}$$

$$\frac{2}{3}\hat{f}_{j+1/2} + \frac{1}{3}\hat{f}_{j+3/2} = \frac{1}{6}\left(f_j + 5f_{j+1}\right);\ c_3 = \frac{3}{10}. \tag{2.18}$$

By multiplying (2.16)–(2.18) with their optimal coefficients ($c_k,\ k = 1, 2, 3$) and then adding the three, we obtain a linear, fifth-order compact scheme:

$$\frac{3}{10}\hat{f}_{j-1/2} + \frac{6}{10}\hat{f}_{j+1/2} + \frac{1}{10}\hat{f}_{j+3/2} = \frac{1}{30}f_{j-1} + \frac{19}{30}f_j + \frac{1}{3}f_{j+1}. \tag{2.19}$$

The optimal coefficients c_k are replaced with the nonlinear weights ω_k, and we get the fifth-order CRWENO scheme (CRWENO5):

$$\left(\frac{2}{3}\omega_1 + \frac{1}{3}\omega_2\right)\hat{f}_{j-1/2} + \left[\frac{1}{3}\omega_1 + \frac{2}{3}(\omega_2 + \omega_3)\right]\hat{f}_{j+1/2} + \frac{1}{3}\omega_3\hat{f}_{j+3/2} = \frac{\omega_1}{6}f_{j-1} + \frac{5(\omega_1 + \omega_2) + \omega_3}{6}f_j + \frac{\omega_2 + 5\omega_3}{6}f_{j+1}. \tag{2.20}$$

The weights ω_k are computed by (2.11) and (2.12)–(2.14). When the solution is smooth, $\omega_k \to c_k$, and (2.20) reduces to (2.19). The present implementation of the CRWENO schemes uses the nonlinear weights defined for the WENO scheme, an approach justified previously [17, 18]. A detailed description of the CRWENO5 scheme is available in [17], and an analysis of its sensitivity to ϵ and the behavior of the nonlinear weights is presented in [18].

The primary difference between the WENO scheme (an example of a standard finite-difference scheme) and the CRWENO scheme (an example of a compact scheme) is as follows. The WENO scheme, given by (2.15), expresses the unknown interface flux $\hat{f}_{j+1/2}$ as an explicit function of the known flux at the cell centers f_j. It is thus straightforward to compute the numerical flux at the interfaces. The CRWENO scheme, given by (2.20), defines the unknown interface flux as an implicit function – the flux at an interface $\hat{f}_{j+1/2}$ depends on the flux at neighboring interfaces ($\hat{f}_{j-1/2}$, $\hat{f}_{j+3/2}$). Thus, it requires the solution to a tridiagonal system of equations. Moreover, the weights ω_k are solution-dependent and the system of equations has to be solved along each grid line at every time-integration step or stage.

2.3. Boundary Treatment. The physical domain is extended by using "ghost" points, and the dependent variables at the ghost points are set such that the interface flux is consistent with the physical boundary conditions. The CRWENO5 scheme [17, 18] uses the WENO5 scheme at the boundary interfaces, and a numerical analysis of the overall discretization [16] showed that this boundary treatment was numerically stable. The overall scheme needs three ghost points at physical boundaries, and the resulting tridiagonal system of equations along a grid line has the first and last diagonal elements (corresponding to the physical boundary interfaces) as one and off-diagonal elements as zero.

2.4. Numerical Properties. The numerical properties of the CRWENO5 scheme are summarized in this section to explain the motivation behind its use. More detailed discussions were previously presented [17, 16, 18] that demonstrate the superior numerical properties of the CRWENO5 scheme compared with the WENO5 scheme. A Taylor series analysis of (2.19) (the linear fifth-order compact scheme underlying the CRWENO5 scheme) shows

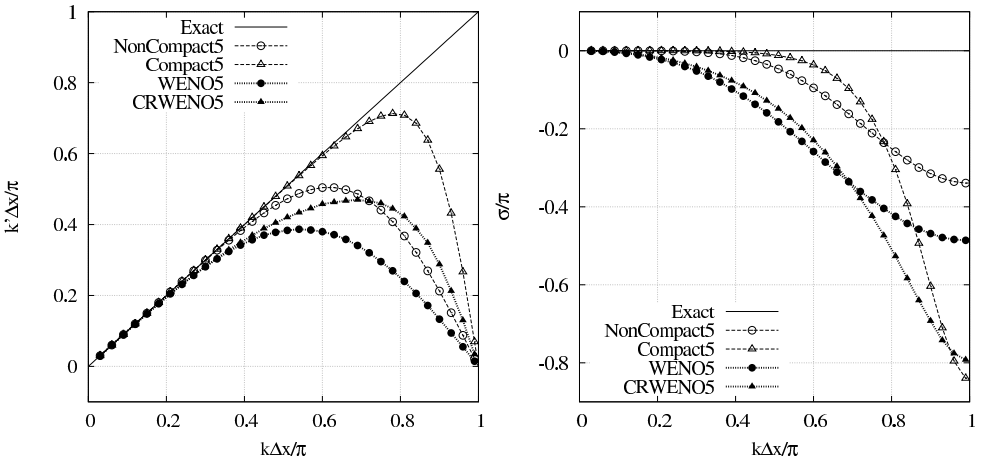
$$\begin{aligned}
&\frac{3}{10}f_{x,j-1} + \frac{6}{10}f_{x,j} + \frac{1}{10}f_{x,j+1} = \frac{1}{\Delta x}\left(\frac{-1}{30}f_{j-2} - \frac{18}{30}f_{j-1} + \frac{9}{30}f_j + \frac{10}{30}f_{j+1}\right) \\
&\Rightarrow f_{x,j} = f_{\Delta,j} + \frac{1}{600}\left.\frac{\partial^6 f}{\partial x^6}\right|_j \Delta x^5 + \frac{1}{2100}\left.\frac{\partial^7 f}{\partial x^7}\right|_j \Delta x^6 + O(\Delta x^7), \qquad (2.21)
\end{aligned}$$

where the term f_Δ denotes the finite-difference approximation to the first derivative. The corresponding expression for (2.10) (the underlying linear interpolation for the WENO5 scheme) is

$$\begin{aligned}
f_{x,j} = &\frac{1}{\Delta x}\left(\frac{-1}{30}f_{j-3} + \frac{1}{4}f_{j-2} - f_{j-1} + \frac{1}{3}f_j + \frac{1}{2}f_{j+1} - \frac{1}{20}f_{j+2}\right) \\
&+\frac{1}{60}\left.\frac{\partial^6 f}{\partial x^6}\right|_j \Delta x^5 + \frac{1}{140}\left.\frac{\partial^7 f}{\partial x^7}\right|_j \Delta x^6 + O(\Delta x^7). \qquad (2.22)
\end{aligned}$$

The leading-order dissipation and dispersion error terms show that the compact interpolation scheme yields solutions with 1/10 the dissipation error and 1/15 the dispersion error of the solutions obtained by the noncompact scheme, for the same order of convergence. Consequently, the fifth-order WENO scheme requires $10^{1/5} \approx 1.5$ times more grid points per dimension to yield a smooth solution with accuracy comparable to that of the fifth-order CRWENO scheme.

The primary motivation for the use of compact schemes is their high spectral resolution; they are thus well suited for applications with a large range of length scales. Detailed linear and nonlinear spectral analyses of the CRWENO5 scheme were presented in [17, 18] and are briefly discussed here. Figure 2.2 shows the dispersion and dissipation properties of the CRWENO5, WENO5, and their underlying fifth-order linear schemes, (2.10) and (2.19) (henceforth referred to as the "NonCompact5" and "Compact5" schemes, respectively). The nonlinear spectral properties of the WENO5 and CRWENO5 schemes are obtained by a statistical analysis [14, 18] of the schemes. The linear fifth-order compact and the CRWENO5 schemes have significantly higher spectral resolution than do the corresponding standard fifth-order and WENO5 schemes, respectively. The compact schemes also exhibit lower dissipation for the low and moderate wavenumbers that are accurately modeled, while they show higher dissipation for very high wavenumbers that are incorrectly aliased to lower



(a) Dispersion (b) Dissipation

FIG. 2.2. *Spectral properties of the linear and nonlinear schemes: "Compact5" refers to (2.19); "NonCompact5" refers to (2.10); CRWENO5 and WENO5 refer to (2.20) and (2.15), respectively.*

wavenumbers. Higher dissipation at very small length scales is advantageous because it diffuses the small length-scale errors.

2.5. Time Integration and Linear Stability. The classical fourth-order, four-stage and the strong stability-preserving third-order three-stage Runge-Kutta schemes are used to integrate (2.4) in time for the numerical examples presented in this paper. We thus briefly analyze and compare the linear stability restrictions on the time-step size for the WENO5 and CRWENO5 schemes. Figure 2.3 shows the stability regions of the three-stage, third-order ("RK3") and the four-stage, fourth-order ("RK4") Runge-Kutta schemes. The eigenvalues of the fifth-order standard (2.10) ("NonCompact5") and the fifth-order compact (2.19) ("Compact5") finite-difference schemes are also shown. The CRWENO scheme suffers from a smaller time-step limit than does the WENO scheme. This situation is verified through numerical experiments on the linear advection equation on a periodic domain. The WENO scheme yields stable solutions until CFL numbers of ~ 1.4 and ~ 1.5 for the third- and fourth-order Runge-Kutta schemes, respectively; the corresponding CFL limits for the CRWENO scheme are ~ 0.9 and ~ 1.0. Thus, from a linear stability perspective, the WENO scheme may use time-step sizes that are ~ 1.5 larger than those used by the CRWENO scheme, for the third- and fourth-order

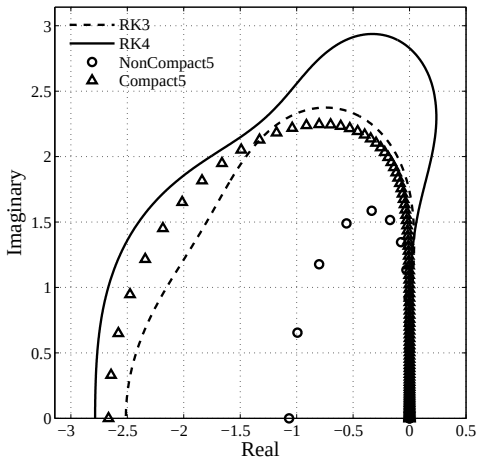


FIG. 2.3. *Eigenvalues of the standard and compact fifth-order finite-difference schemes, and the stability regions of three- and four-stage Runge-Kutta time-integration schemes.*

TABLE 2.1
Factors relating the grid size, CFL, and time-step size of a WENO5 solution to those of a CRWENO5 solution for a fair comparison with explicit time-integration. (D is the number of spatial dimensions.)

Type of Problem	Grid Size	CFL	Time-Step Size
Smooth	$\sim 1.5^D$	~ 1.5	~ 1
Nonsmooth	$\sim 1.25^D$ [1]	~ 1.5	~ 1.25

Runge-Kutta schemes.

2.6. Comparisons between the WENO5 and CRWENO5 Schemes. Table 2.1 summarizes the implications of the numerical properties and linear stability limits discussed in Sections 2.4 and 2.5, comparing the computational costs of the WENO5 and CRWENO5 schemes. Numerical analysis of the linear schemes shows that the WENO5 scheme yields comparable solutions on grids that have ~ 1.5 times more points in each dimension than does the grid used with the CRWENO5 scheme for a smooth, well-resolved solution. Numerical experiments in Sections 3.3 and 4 show this factor to be ~ 1.25 for solutions with unresolved length scales where the nonlinear weights are not optimal. It is thus appropriate to compare the computational cost of the CRWENO5 scheme on a given grid with that of the WENO5 scheme on a grid that has ~ 1.25 or ~ 1.5 times as many points along each dimension, depending on the problem type. Solutions obtained with explicit time-integration schemes often use the maximum time-step size allowed by the linear stability limit. The WENO5 scheme has a stability limit that is ~ 1.5 times higher than that of the CRWENO5 scheme, and the wall times for the WENO5 and CRWENO5 schemes are measured with the WENO5 solution obtained at a CFL number that is ~ 1.5 times higher. Thus, for a smooth, well-resolved problem, we use the same time-step size such that the CFL number of the WENO5 scheme is ~ 1.5 times higher. The time-step size for the WENO5 solution is ~ 1.25 times higher than that for the CRWENO5 solution for problems with unresolved scales.

3. Parallel Implementation. The fifth-order CRWENO scheme (2.20) results in a solution-dependent, tridiagonal system of equations of the form

$$\mathrm{A}\hat{\mathbf{f}} = \mathbf{r};\ \text{where } \mathbf{r} = \mathrm{B}\mathbf{f} + \mathbf{b}. \tag{3.1}$$

The tridiagonal, left-hand side matrix A is of size $(N+1)^2$, $\hat{\mathbf{f}} = [\hat{f}_{j+1/2}; j = 0, \cdots, N]^{\mathrm{T}}$ is the $(N+1)$-dimensional vector of unknown flux at the interfaces, B is a $(N+1) \times N$ matrix representing the right-hand side operator, and $\mathbf{f} = [f_j; j = 1, \cdots, N]^{\mathrm{T}}$ is the vector of (known) flux at the cell centers (N is the number of interior points in the grid). The boundary terms are represented by $\mathbf{b}$. Parallel implementations of the CRWENO scheme (as well as the hybrid compact-ENO/WENO schemes) depend on the efficient, multiprocessor solution of (3.1). We focus on a distributed-memory parallel algorithm in this study; shared-memory thread-based parallelization is beyond the scope of this paper and will be investigated in future studies. This section describes the parallel tridiagonal solver and demonstrates its performance and scalability.

3.1. Tridiagonal Solver. We require a parallel tridiagonal solver that solves the system to sufficient accuracy (so as to ensure that no parallelization errors exist

[1] This factor is determined by numerical experiments in Sections 3.3 and 4.

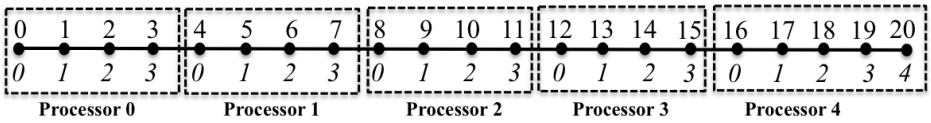


FIG. 3.1. *Example of a partitioned grid:* 21 *points distributed among five processors, with the global and local numbering of points.*

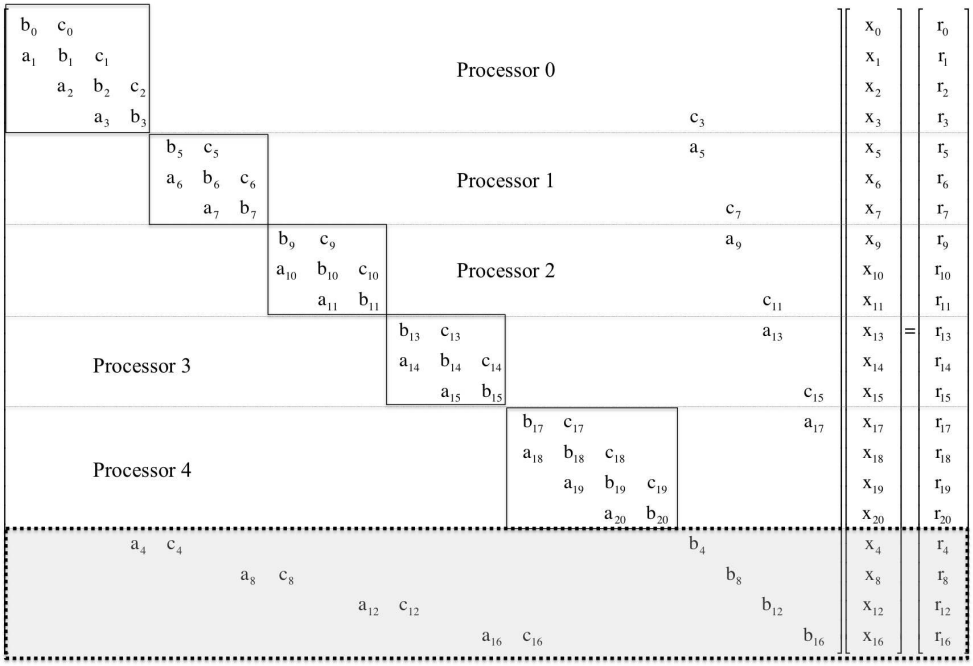


FIG. 3.2. *Reordered tridiagonal system of equations for a multiprocessor algorithm. The lines divide the data stored on each processor; the solid boxes show the decoupled tridiagonal blocks on each processor; the rows inside the dashed box at the bottom are the first point in each subdomain (except the first), and each row resides on a different processor.*

in the overall scheme), whose mathematical complexity is comparable to that of the serial Thomas algorithm and does not involve any collective communications. Figure 3.1 shows an example of a system with $N = 21$, distributed on five processors. We use the substructuring or partitioning approach [47, 38] that is explained by reordering the system as shown in Fig. 3.2. The first row on each subdomain (except the global first that is a physical boundary point) is placed at the bottom of the matrix in the order of the processor rank on which it resides (marked by the dotted box in Fig. 3.2). The decoupled tridiagonal blocks on each processor are marked by the solid boxes. Independent elimination of these decoupled blocks followed by the elimination of the reordered rows results in the reduced tridiagonal system of size $p - 1$ (p being the number of processors), shown in Fig. 3.3. A more detailed description of our implementation of this algorithm is provided in [19].

Several strategies exist for solving the reduced system [37, 38]; however, a scalable and efficient method is challenging because each row of the reduced system resides on a different processor. We choose to solve the reduced system iteratively using the Jacobi method. Although this approach may not work well for general systems

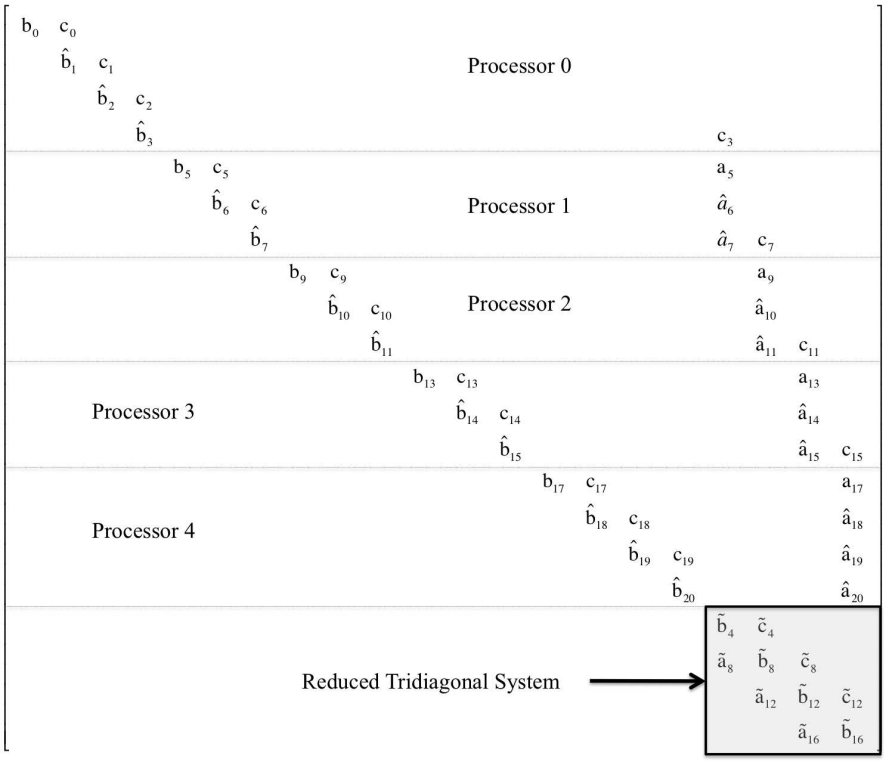


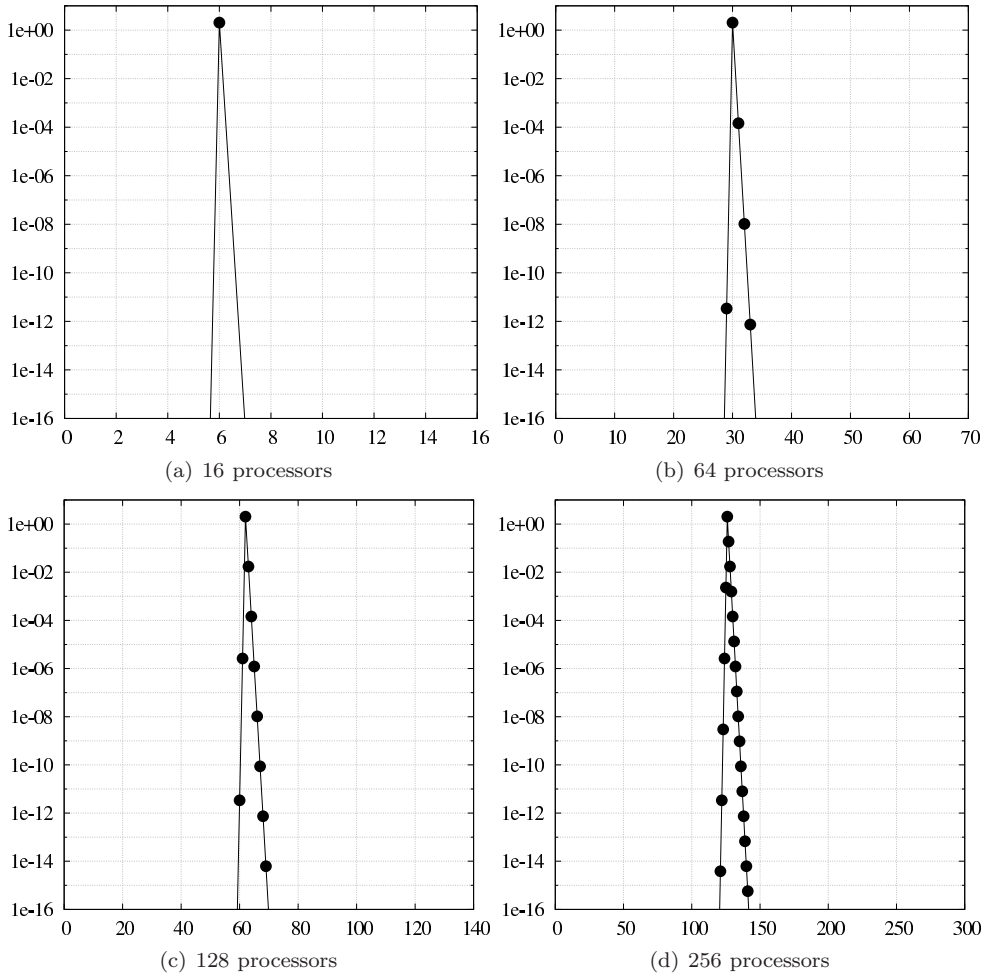
FIG. 3.3. *Elimination of all the rows results in a reduced tridiagonal equation (shown by the box) with each row residing on a different processor.*

[38], we specifically solve tridiagonal systems that result from the compact finite-difference discretization of a hyperbolic flux, such as (2.20). The reduced system represents the coupling between the first interfaces on each subdomain, separated by the interior grid points on each processor. Therefore, this system has a strong diagonal dominance for $p \ll N$; as $p \to O(N)$, the diagonal dominance decreases. We consider as an example the tridiagonal system (2.19) with $N = 1024$ and a random right-hand side. Figure 3.4 shows the elements of the $p/2$th column (an arbitrary choice) of the inverse of the reduced system for 16, 64, 128, and 256 processors (where p is the number of processors), corresponding to subdomain sizes of 64, 16, 8, and 4 points, respectively. Elements larger than machine zero (10^{-16}) are shown. We observe that for a subdomain size of 64 points (16 processors), only the diagonal element has a value higher than machine zero; the reduced system is effectively a diagonal matrix, and the solution is trivial. The number of non-machine-zero elements grows, and the diagonal dominance decreases as the subdomain size decreases.

We analyze the properties of the reduced system as a function of the global problem size and the number of processors, and we estimate the number of Jacobi iterations needed to achieve converged solutions. The reduced system of equations is expressed as

$$\mathrm{R}\hat{\mathbf{f}}_p = \tilde{\mathbf{r}}, \tag{3.2}$$

where R represents the tridiagonal matrix of size $p-1$ (inside the box in Fig. 3.3), $\tilde{\mathbf{r}}$ is



(a) 16 processors (b) 64 processors

(c) 128 processors (d) 256 processors

FIG. 3.4. *Elements of the $p/2$th column of the inverse of the reduced system for various numbers of processors (p).*

the corresponding right-hand side obtained from $\mathbf{r}$ in (3.1) by applying the elimination steps, and $\hat{\mathbf{f}}_p$ is a portion of the interface flux vector $\hat{\mathbf{f}}$ in (3.1) constituting the first interface of each subdomain (except the physical boundary interface). The Jacobi method is expressed as [42]

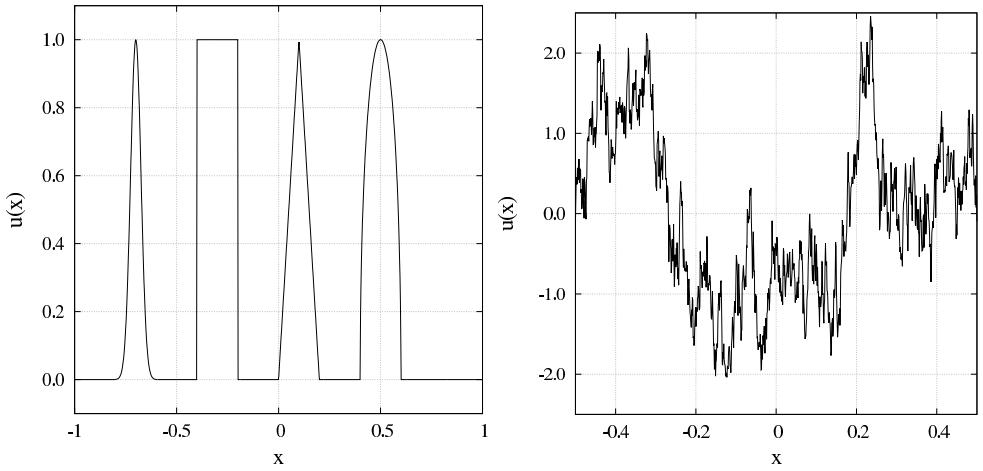
$$\hat{\mathbf{f}}_p^{k+1} = \mathrm{D}^{-1}\left(\mathrm{I} - \mathrm{R}\right)\hat{\mathbf{f}}_p^{k} + \mathrm{D}^{-1}\tilde{\mathbf{r}}, \tag{3.3}$$

where $\hat{\mathbf{f}}_p^k$ is the kth guess for $\hat{\mathbf{f}}_p$, D is the diagonal of R, and I is the identity matrix. The initial guess is taken as

$$\hat{\mathbf{f}}_p^0 = \mathrm{D}^{-1}\tilde{\mathbf{r}}. \tag{3.4}$$

The convergence of the Jacobi method is determined by the spectral radius of the iteration matrix; we thus require

$$\rho\left[\mathrm{D}^{-1}\left(\mathrm{I} - \mathrm{R}\right)\right] < 1. \tag{3.5}$$



(a) Case 2 (b) Case 3

FIG. 3.5. *Solutions ($N = 1024$) by which ω_i in (2.20) are computed for the analysis of the reduced system.*

We estimate the number of iterations for a converged solution from the convergence rate $\phi = -\log \rho$ [42] as

$$N_{Jacobi} = \frac{\log C}{\phi}, \tag{3.6}$$

where C is the convergence criterion or tolerance.

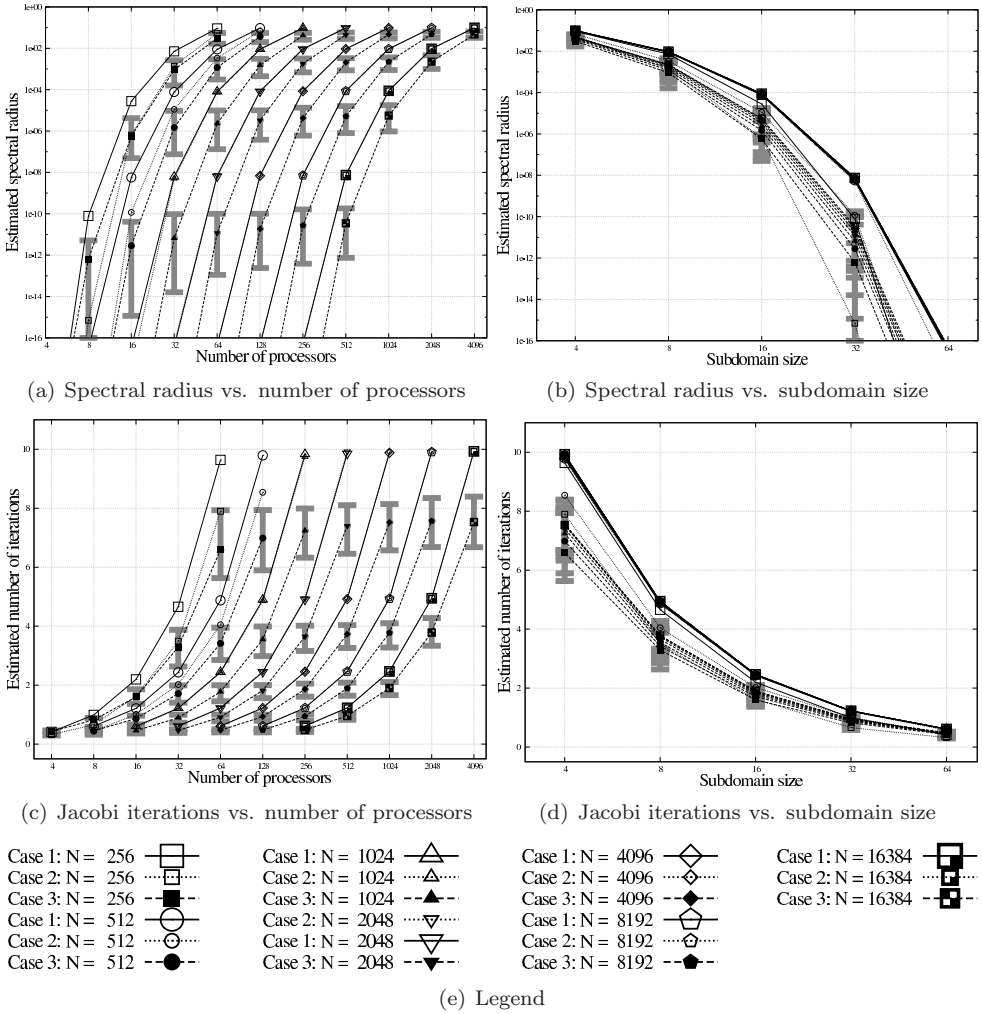
We evaluate the spectral radius of the Jacobi iteration matrix, $\left[\mathrm{D}^{-1}\left(\mathrm{I}-\mathrm{R}\right)\right]$, for several global problem sizes and number of processors. We also analyze the effect of nonlinear weights in (2.20) on the reduced system. We consider three cases. "Case 1" represents a smooth solution for which the nonlinear weights are optimal ($\omega_i = c_i$), and (2.20) is essentially (2.19). "Case 2" represents the tridiagonal system (2.20) with the weights computed for the solution shown in Fig. 3.5(a), given by

$$u(x) = \begin{cases} \exp\left(-\log(2)\frac{(x+7)^2}{0.0009}\right) & -0.8 \leq x \leq -0.6, \\ 1 & -0.4 \leq x \leq -0.2, \\ 1 - |10(x-0.1)| & 0 \leq x \leq 0.2, \\ \left[1 - 100(x-0.5)^2\right]^{1/2} & 0.4 \leq x \leq 0.6, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

This is representative of smooth solutions with isolated discontinuities. "Case 3" represents the tridiagonal system (2.20) with weights computed for a solution representative of turbulent flows, given by

$$u(x) = \sum_{k}^{N/2} A\left(k\right) \cos\left(2\pi k x + \phi\left(k\right)\right), \tag{3.8}$$

where $A(k) = k^{-5/3}$ is the amplitude and the phase $\phi(k) \in [-\pi, \pi]$ is a randomly chosen for each wavenumber k. Figure 3.5(b) shows one realization of (3.8). Although the



(a) Spectral radius vs. number of processors

(b) Spectral radius vs. subdomain size

(c) Jacobi iterations vs. number of processors

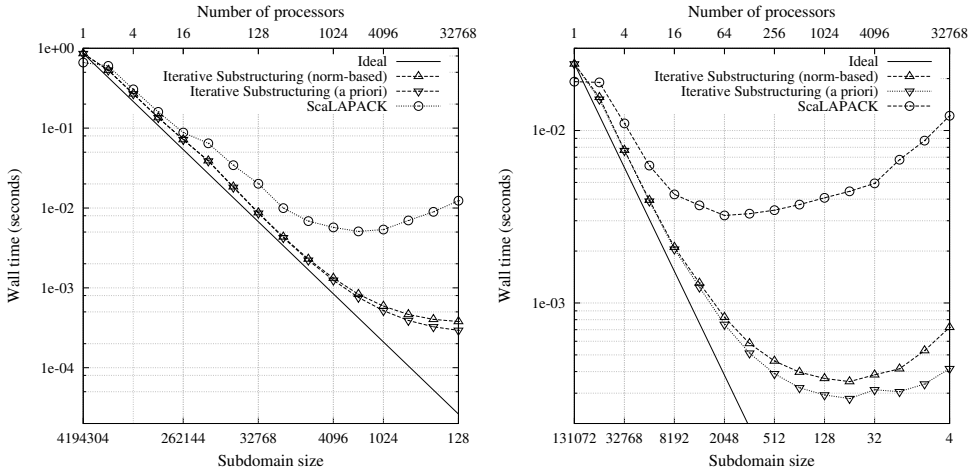
(d) Jacobi iterations vs. subdomain size

(e) Legend

FIG. 3.6. *Analysis of the iterative solution to the reduced system with the Jacobi method.*

solution is theoretically smooth, the presence of small length scales (high wavenumbers) results in highly nonlinear behavior [18]. Thus, our choice of cases includes all possible solution features of compressible, turbulent flows.

Figure 3.6(a) shows the spectral radius of the Jacobi iteration matrix as a function of the number of processors for the three cases described above and for several values of the global system size ($N = 256, 512, \ldots, 16384$). The largest number of processors for a given global system size corresponds to a subdomain size of 4 points per processor. Note that for "Case 3" the data points represent the average, and the gray error bars represent the maximum and minimum values over $10,000$ realizations of (3.8) (this analysis is similar to the nonlinear spectral analysis of WENO [14] and CRWENO [18] schemes). The spectral radius increases as the number of processors increases for a given global problem size. This result is expected because the reduced system represents the compact finite-difference coupling between the first point on each processor; as the number of processors increases, these points come closer, and



(a) $N = 4,194,304$ (b) $N = 131,072$

FIG. 3.7. *Performance of the parallel tridiagonal solver, and comparison with ScaLAPACK (*`pddtsv`*): Wall time as a function of subdomain size and number of processors for the solution of the tridiagonal system given by (2.19) with N grid points and a random right-hand side.*

the coupling is stronger. The spectral radius for the system with optimal weights ("Case 1") is the largest for a given global system size and number of processors; the spectral radius corresponding to the other two cases is either lower or similar. This result is again expected because the optimal weights result in the widest stencil with highest-order accuracy, while nonoptimal weights reduce the accuracy by biasing the stencil in one direction and make it more local. Thus, the spectral radius corresponding to the linear scheme (2.19) is an upper bound, and nonoptimal weights will result in an iteration matrix with a smaller or equal spectral radius. Figure 3.6(b) shows the spectral radius as a function of the subdomain size for the same systems and cases. The spectral radii for the systems corresponding to "Case 1" are insensitive to the global system size (and thus the size of the reduced system) and are a function of the subdomain size only.

Figure 3.6(c) shows the number of Jacobi iterations required for a tolerance of $C = 10^{-10}$ estimated by using (3.6) and the spectral radii reported in Fig. 3.6(a). The number of Jacobi iterations increases with the number of processors for a given global system size (N), as expected. The number of iterations required by the optimal case ("Case 1") is an upper bound on the number of iterations required by the other two cases for a given global system size and number of processors. Figure 3.6(d) shows the number of Jacobi iterations as a function of the subdomain size. We observe that the required number of Jacobi iterations for "Case 1" is a function of the subdomain size only, and not of the global system size or number of processors. We show results for subdomain sizes from 4 points to 64 points per processor; for subdomain sizes larger than 64 points per processor, no Jacobi iterations are needed if the initial guess is (3.4) since the reduced system is essentially a diagonal matrix (this is verified in subsequent sections). We do not consider subdomains smaller than 4 points per processor because this is almost the smallest practical subdomain size for fifth-order finite-difference methods.

The analysis presented leads to the following conclusions regarding the a priori

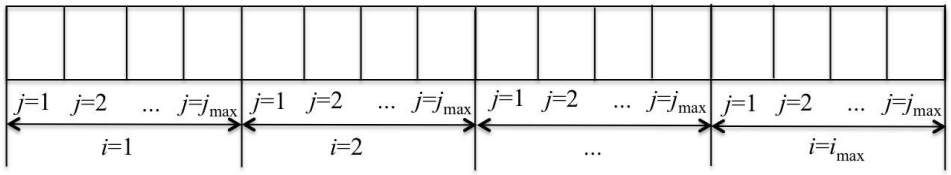


FIG. 3.8. *Array arrangement for the tridiagonal solver: two-dimensional problem with i and j as the grid indices in x and y, respectively, while reconstructing the x-derivatives.*

specification of the number of Jacobi iterations to solve the reduced system. When the solution is smooth and when the CRWENO5 scheme (2.20) is equivalent to the optimal fifth-order compact scheme (2.19), the number of iterations required is an upper bound for a given problem size and number of processors; any other solution resulting in nonoptimal weights will require a smaller or equal number of Jacobi iterations. This number of Jacobi iterations for the optimal case is a function of the subdomain size and not of the global problem size. Thus, for an arbitrary problem with a given grid size and number of processors, specifying the number of Jacobi iterations as the one required for a smooth solution with the corresponding subdomain size ensures that the solution to the reduced system is converged. This allows us to avoid a norm-based exit criterion and, consequently, the requirement of collective communications.

The performance of the parallel tridiagonal solver is demonstrated by solving a tridiagonal system of equations given by (2.19) with a random right-hand side, and compared with that of the ScaLAPACK [6, 9] routine `pddtsv`—a parallel tridiagonal solver for diagonally-dominant systems. Appendix A describes the computing platform and the software environment used. Figure 3.7(a) shows the wall time as a function of the number of processors and subdomain sizes for a system of size $N = 4,194,304$. The solutions are obtained using the iterative substructuring-based tridiagonal solver with a norm-based exit criterion ($C = 10^{-10}$) as well as a priori specification of the number of Jacobi iterations. The smallest subdomain size in this example is 128, and thus, both approaches (norm-based exit, and a priori specification) do not perform any Jacobi iterations. The initial guess, given by (3.4), is sufficiently accurate. The norm-based exit is slightly more expensive due to the collective communications in the calculation of the global norm. Solutions are also obtained using ScaLAPACK; however, the ScaLAPACK solver scales poorly for the tridiagonal system considered here when subdomain sizes are smaller than $4,096$ points per processor.

Figure 3.7(b) shows the wall times for a smaller system with $N = 131,072$ points, and smallest subdomain size is 4 points per processor. The iterative substructuring-based tridiagonal solver with a norm-based exit criterion scales well till a subdomain size of 128 points per processor; at smaller subdomain sizes, the cost of Jacobi iterations and global norm calculations increase significantly. A priori specification of the number of Jacobi iterations results in a similar performance; however, avoiding the calculation of the global norm results in a significant reduction of the cost. At subdomain sizes smaller than $\sim 10,000$ points per processor, the scalability and performance of the iterative substructured tridiagonal solver are superior to those of the ScaLAPACK routine for the systems considered here because the former exploits the strong diagonal dominance of the reduced system.

3.2. Extension to Multidimensions. Solutions to multidimensional problems using a compact finite-difference scheme require solving several tridiagonal systems of equations—one system along each grid line in each dimension. We extend our parallel tridiagonal solver to multiple dimensions by solving the systems in one function call. The data is stored in arrays with the innermost loop representing distinct systems; that is, the arrays containing the diagonals and the right-hand side vectors of all the systems have a given row element of all the systems in consecutive memory locations. Figure 3.8 shows this arrangement for a two-dimensional problem where the x-derivative (corresponding to grid index i) is being reconstructed. Each operation on an element of a single tridiagonal system is carried out on the corresponding elements of multiple tridiagonal systems, thus increasing the arithmetic density. In addition, messages that contained the relevant elements of one tridiagonal system contain elements of multiple systems, thus increasing communication efficiency; that is, the size of the messages increases while their number stays the same. Therefore, the cost of the tridiagonal solver is initially sublinear in the number of systems. We thus expect the proposed implementation of the tridiagonal solver to be more efficient for multidimensional simulations.

Reconstruction of the interface fluxes with the CRWENO5 scheme (2.20) is carried out independently along each dimension for multidimensional problems. The analysis presented in the previous section can thus be used to specify the number of Jacobi iterations based on the number of grid points and the number of processors along each dimension.

3.3. Performance Analysis. We analyze the performance of our parallel implementation of the CRWENO5 scheme by applying it to the inviscid Euler equations [30]. We consider smooth, well-resolved solutions as well as solutions with unresolved length scales. The software environment and hardware details of the computing platforms used in this study are summarized in Appendix A. The fourth-order four-stage Runge-Kutta (RK4) scheme is used for time integration. The scalar reconstruction schemes are applied to each component of the vector quantities. The CRWENO5 scheme was previously demonstrated [17] to be computationally more efficient than the WENO5 scheme (for both smooth and discontinuous problems) on a single processor with time-step sizes dictated by the linear stability limits of each scheme. The cost of the parallel tridiagonal solver described above increases with the number of processors for the same domain size because of the larger number of Jacobi iterations needed to solve the reduced system. We investigate the efficiency of the CRWENO5 scheme (relative to that of the WENO5 scheme) as the number of processors increases (i.e., the subdomain size becomes smaller). We expect CRWENO5 to be less efficient than the WENO5 scheme for subdomains smaller than a critical size; however, we show that it retains its higher efficiency for subdomain sizes of practical relevance.

We note that we use a modified definition of the parallel efficiency when comparing the two schemes. Although generally one defines the parallel efficiency of a scheme based on its own wall time on the smallest number of processors, we calculate the efficiencies of both schemes based on the wall time of the CRWENO5 scheme on the lowest number of processors considered. The modified parallel efficiency is given by

$$\text{Efficiency} = \frac{\tau_{0,\text{CRWENO5}}\ p_{0,\text{CRWENO5}}}{\tau p}, \tag{3.9}$$

where $\tau_{0,\text{CRWENO5}}$ is the wall time of the CRWENO5 solution on $p_{0,\text{CRWENO5}}$ number of processors (which is the minimum number for a given case), and τ is the wall time

of the CRWENO5 or WENO5 solution on p number of processors. This definition allows for the computational efficiencies of the two schemes (based on accuracy and wall time) to be reflected in our comparisons. The traditional definition results in a starting value of one for both and does not provide any information on the relationship between the wall times of the two schemes. Our definition results in a starting value of one for the CRWENO5 scheme and a starting value less than one for the WENO5 scheme, thus reflecting that the WENO5 scheme requires a larger wall time to yield a solution of desired accuracy on the smallest number of processors considered. The modified parallel efficiency shows the scalability of each scheme (through its slope), as well as the relative costs to compute similar solutions (through its absolute value).

We start with the one-dimensional advection of a sinusoidal density wave—an example of a smooth, well-resolved solution for which the Taylor series analysis (Section 2.4) holds. The initial density, velocity, and pressure are specified as

$$\begin{aligned} &\rho = \rho_0 + \tilde{\rho} \sin\left(2\pi x\right), \\ &u = 1 \; p = 1/\gamma, \end{aligned} \tag{3.10}$$

respectively, on a unit periodic domain $x \in [0, 1]$. The specific heat ratio is $\gamma = 1.4$. The mean density ρ_0 is 1, and the amplitude of the wave is $\tilde{\rho} = 0.1$. Solutions are obtained with the CRWENO5 scheme on grids with 64, 128, and 256 points (baseline grids), and with the WENO5 scheme on grids with 1.5 times as many points (96, 192, and 384). The solutions are obtained after one cycle over the periodic domain. A small time-step size of 10^{-4} is used such that the numerical time integration errors are negligible (relative to those from the spatial discretization).

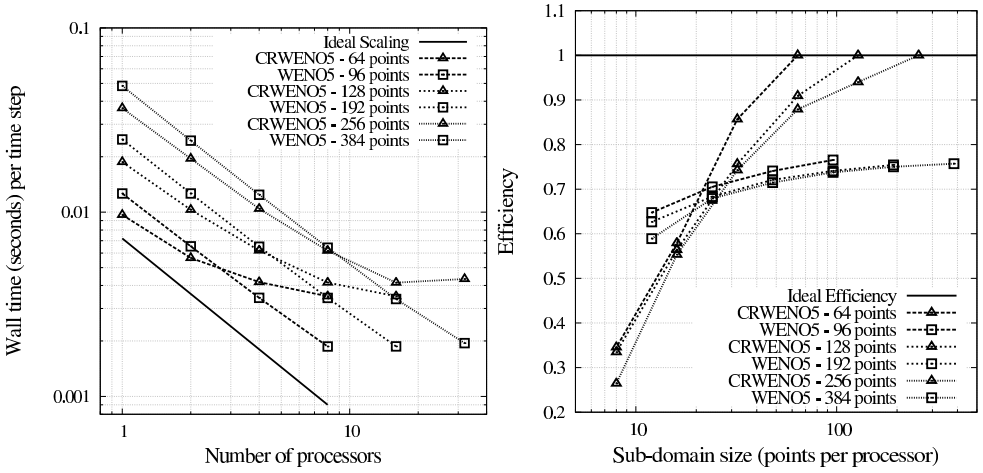
Table 3.1 shows the L_2 norm of the numerical errors and the wall times for different grid sizes (N_{global}). The number of Jacobi iterations (N_{Jac}) is specified based on Fig. 3.6(d). Both schemes show fifth-order convergence, and the errors for the WENO5 solutions on grids with 96, 192, and 384 points are comparable to those for the CRWENO5 solutions on grids with 64, 128, and 384 points, respectively. The numerical errors for the CRWENO5 scheme are identical for varying numbers of processors on a given grid, thus demonstrating that our algorithm does not introduce any parallelization-based errors. All the cases considered use the same time-step size; therefore the WENO5 solutions are obtained at a CFL number that is ~ 1.5 times higher than that of the CRWENO5 solutions (Section 2.6). The WENO5 cases on the finer grids are run on the same number of processors; in other words, with a given number of processors, we investigate whether the CRWENO5 scheme is more efficient than the WENO5 scheme.

The solutions obtained on one processor show that the CRWENO5 scheme is more efficient; the wall times of the CRWENO5 scheme are lower on the 64, 128, and 256 point grids than those of the WENO5 scheme on the 96, 192, and 384 point grids, respectively. These results agree with previous studies [17]. As we reduce the subdomain sizes for a given case of grid sizes (e.g., CRWENO5 on the 64-points grid and WENO5 on the 96-points grid), the relative cost of the CRWENO5 scheme increases because of the increasing number of Jacobi iterations. As a result, the WENO5 scheme is more efficient at smaller subdomain sizes. We observe from Table 3.1 that the CRWENO5 scheme is less expensive for subdomain sizes larger than 64 points, whereas for smaller subdomains the WENO5 scheme is less expensive. Figure 3.9(a) shows the wall time per time step as a function of the number of processors. The CRWENO5 scheme does not scale as well as the WENO5 scheme for larger numbers of processors; the wall time for the CRWENO5 scheme is initially lower than

TABLE 3.1

Errors (L_2) and wall times (in seconds) for the one-dimensional advection of a sinusoidal density wave.

WENO5			CRWENO5			
N_{global}	Error	Wall Time	N_{global}	Error	Wall Time	N_{Jac}
1 processor						
96	1.3332E-08	126.33	64	1.1561E-08	96.69	0
192	4.1680E-10	248.20	128	3.3927E-10	187.26	0
384	1.3024E-11	484.51	256	1.0253E-11	366.87	0
2 processors						
96	1.3332E-08	65.25	64	1.1561E-08	56.43	0
192	4.1680E-10	126.35	128	3.3927E-10	103.00	0
384	1.3024E-11	244.52	256	1.0253E-11	195.15	0
4 processors						
96	1.3332E-08	34.27	64	1.1561E-08	41.72	2
192	4.1680E-10	64.95	128	3.3927E-10	61.91	1
384	1.3024E-11	124.36	256	1.0253E-11	104.34	0
8 processors						
96	1.3332E-08	18.66	64	1.1561E-08	34.97	4
192	4.1680E-10	34.25	128	3.3927E-10	41.51	2
384	1.3024E-11	64.17	256	1.0253E-11	61.75	1
16 processors						
192	4.1680E-10	18.68	128	3.3927E-10	34.97	4
384	1.3024E-11	33.76	256	1.0253E-11	41.44	2
32 processors						
384	1.3024E-11	19.46	256	1.0253E-11	43.37	6



(a) Wall time vs. number of processors

(b) Modified parallel efficiency vs. subdomain size

FIG. 3.9. *One-dimensional advection of density sine wave: wall times and efficiencies for the CRWENO5 on grids with* 64, 128, *and* 256 *points, and the WENO5 scheme on grids with* 96, 192, *and* 384 *points (data in Table 3.1).*

TABLE 3.2
Errors (L_2) and wall times (in seconds) for the three-dimensional advection of a sinusoidal density wave.

WENO5			CRWENO5			
N_{global}	Error	Wall time	N_{global}	Error	Wall time	N_{Jac}
64 (4^3) processors						
48^3	5.3211E-07	12960	32^3	5.4544E-07	5058	10
512 (8^3) processors						
48^3	5.3211E-07	1818	32^3	5.4544E-07	933	10
96^3	1.6660E-08	12884	64^3	1.4849E-08	4985	10
4096 (16^3) processors						
96^3	1.6660E-08	1803	64^3	1.4849E-08	936	10
192^3	5.2096E-10	12819	128^3	4.3038E-10	4929	10
32768 (32^3) processors						
192^3	5.2096E-10	1953	128^3	4.3038E-10	941	10

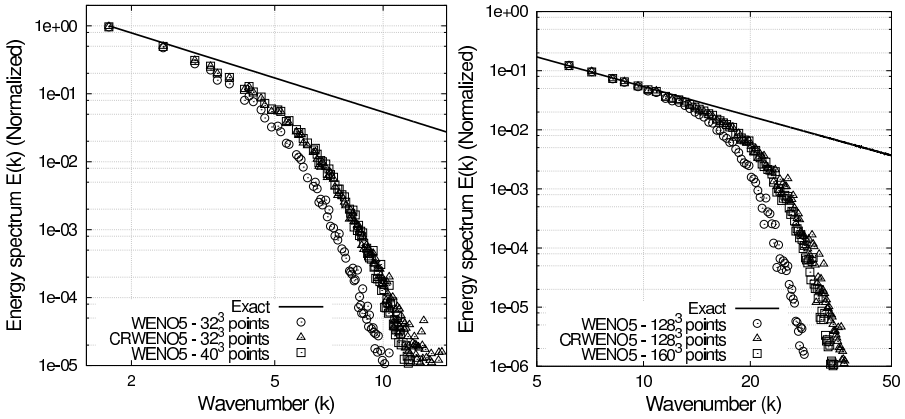
that of the WENO5 scheme, but as the number of processors increase, the cost of the CRWENO5 scheme exceeds that of the WENO5 scheme. Figure 3.9(b) shows the modified parallel efficiency as a function of the subdomain size and reiterates this result; the CRWENO5 scheme is more efficient for larger subdomains, but the efficiency decreases rapidly as subdomains grow smaller, and the WENO5 scheme is more efficient at the smallest subdomains considered.

We next consider the three-dimensional smooth, well-resolved solution. The dimensionality of the problem affects the efficiency and relative computational cost of the CRWENO5 scheme in two ways. The first effect is that the WENO5 scheme requires grids with ~ 1.25–$\sim 1.5^D$ (D being the number of dimensions) more points than that required by the CRWENO5 scheme to yield comparable solutions, and this factor increases by the Dth power. Thus, in two and three dimensions, this factor is ~ 2.25 and ~ 3.375 respectively. The other effect of dimensionality is on the efficiency of the tridiagonal solver, as discussed in Section 3.2. The solution to multidimensional problems requires solving several systems, thus increasing the arithmetic density and communication efficiency of the tridiagonal solver. These two factors indicate that our implementation of the CRWENO5 scheme is expected to be more efficient in higher dimensions.

The initial density, velocity, and pressure for the three-dimensional advection of a sinusoidal density wave are specified as

$$\begin{aligned} &\rho = \rho_0 + \tilde{\rho} \sin(2\pi x) \sin(2\pi y) \sin(2\pi z), \\ &u = v = w = 1 \; p = 1/\gamma, \end{aligned} \tag{3.11}$$

on a unit periodic domain $[0,1]^3$. The specific heat ratio is $\gamma = 1.4$. The mean density ρ_0 is 1, and the amplitude of the wave is $\tilde{\rho} = 0.1$. We use a time-step size of 10^{-4} for all the cases. Table 3.2 shows the errors (L_2 norm) and wall times for the grid sizes (N_{global}) considered. Fifth-order convergence is verified; and, the errors of the CRWENO5 scheme on the grids with 32^3, 64^3, and 128^3 points are comparable to those of the WENO scheme on grids with 1.5^3 times more points (48^3, 96^3, and 192^3). The CRWENO5 scheme is less expensive than the WENO5 scheme for all the cases considered, including the smallest subdomain size of 4^3 points per processor. The number of Jacobi iterations (N_{Jac}) in Table 3.2 is identical (10) for all



(a) 32^3 grid points (b) 128^3 grid points

FIG. 3.10. *Energy spectrum of the numerical solutions to the three-dimensional advection of density fluctuations.*

the cases. Although ten iterations are more than what is required for convergence for subdomains larger than 4 points per dimension per processor (see Fig. 3.6(d)), the CRWENO5 scheme is less expensive by a relatively large margin (because of the effect of dimensionality), allowing us to specify a more-than-adequate number of iterations. Thus, all the cases reported carried out with 10 Jacobi iterations to solve the reduced tridiagonal system.

We now consider the three-dimensional advection of density waves comprising all grid-supported wavenumbers—an example of a solution with unresolved length scales for which the WENO5 and CRWENO5 schemes show nonlinear behavior. The initial density fluctuations is prescribed in the Fourier space as

$$\hat{\rho}(k_x, k_y, k_z) = \frac{\tilde{\rho}\,|\mathbf{k}|^{-5/6}}{\sqrt{2}}\,(1+i)\,;|\mathbf{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2};$$
$$1 \leq k_x, k_y, k_z \leq N/2, \tag{3.12}$$

where N is the number of points per dimension on a square grid and the complex conjugates are taken in the remainder of the wavenumber domain. The amplitude decay is such that the fluctuation energy spectrum is representative of the kinetic energy spectrum of turbulent flows. The initial density is then specified in the physical space as

$$\rho = \rho_0 + \delta\rho \tag{3.13}$$

where $\delta\rho(x, y, z)$ is the inverse Fourier transform of $\hat{\rho}(k_x, k_y, k_z)$. The maximum amplitude of fluctuations $\tilde{\rho}$ is taken as 10^{-5} to ensure that the total density is non-negative. Uniform unit velocity (in all dimensions) and pressure ($p = 1/\gamma$) are specified where $\gamma = 1.4$ is the ratio of specific heats. A periodic domain of unit length in each dimension is taken.

We solve the problem on two grid sizes: CRWENO5 and WENO5 solutions are obtained on grids with 32^3 and 128^3 points, and the WENO5 solutions are obtained on grids with 1.25^3 times more points (40^3 and 160^3 points) as well. Figure 3.10 shows the density fluctuations spectrum for the solutions after one time period. The

TABLE 3.3
Wall times (in seconds) for the three-dimensional advection of density fluctuations.

	WENO5		CRWENO5		
N_{proc}	N_{local}	Wall Times	N_{local}	Wall Times	N_{Jac}
40^3 (WENO5) and 32^3 (CRWENO5) grid points					
8	20^3	4516.9	16^3	3343.9	10
64	10^3	655.3	8^3	520.1	10
512	5^3	93.4	4^3	92.8	10
160^3 (WENO5) and 128^3 (CRWENO5) grid points					
8	80^3	1004960	64^3	795420	10
64	40^3	137864	32^3	103310	10
512	20^3	18791	16^3	14651	10
4096	10^3	2616	8^3	2084	10
32768	5^3	376	4^3	378	10

spectral resolutions of the CRWENO5 scheme on the grids with 32^3 and 128^3 points are comparable to those of the WENO5 scheme on grids with 40^3 and 160^3 points, respectively. The WENO5 solutions are obtained at a CFL number ~ 1.5 times higher than that for the CRWENO5 solutions (Section 2.6); the time-step sizes are 1×10^{-3} (CRWENO5 on 32^3 points), 1.25×10^{-3} (WENO5 on 40^3 points), 2.5×10^{-4} (CRWENO5 on 128^3 points), and 3.125×10^{-5} (WENO5 on 160^3 points). We compare the wall times of these cases in Table 3.3 for solutions obtained on N_{proc} processors (N_{local} is the subdomain size). The number of Jacobi iterations (N_{Jac}) is specified as 8 for all the cases and is more than adequate to ensure convergence of the reduced system, as shown in our analysis (Figs. 3.6(c) and 3.6(d), "Case 3"). The CRWENO5 scheme is less expensive than the WENO5 scheme for all except the smallest subdomain sizes (4^3 points per processor) considered; the costs are similar at this subdomain size.

The numerical experiments presented in this section indicate that our implementation of the parallel tridiagonal solver does not introduce any parallelization-related errors. We analyze the computational cost of our implementation as a function of grid size and number of processors. In one spatial dimensional, a critical subdomain size is observed, and the CRWENO5 has a lower time to solution for subdomain sizes larger than this. The increasing cost of solving the tridiagonal system renders it less efficient at smaller subdomain sizes. In three dimensions, however, the WENO5 scheme requires several times more grid points to yield comparable solutions, and the CRWENO5 scheme is computationally less expensive for solutions of comparable accuracy for most subdomain sizes considered. The two schemes have comparable expense for the smallest practical subdomain size (4^3 points per processor). These results imply that our implementation of the CRWENO5 scheme is a viable alternative to standard, noncompact schemes even for massively parallel simulations.

4. Results. The performance of the CRWENO5 scheme is evaluated on benchmark flow problems. Previous studies [17, 20, 16, 18] demonstrated two desirable properties of the CRWENO5 scheme: accurate preservation of flow features as they convect large distances and improved resolution of a larger range of relevant length scales for turbulent flows. The two flow problems in this section—the long-term convection of an isentropic vortex and the decay of isotropic turbulence—illustrate these properties, and the computational efficiency of the CRWENO5 scheme on multiple

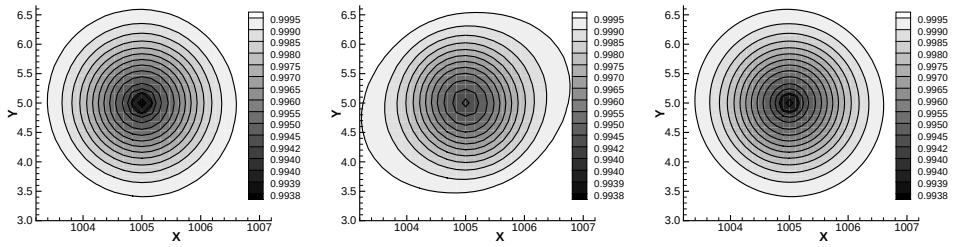
processors is demonstrated for these flows.

4.1. Isentropic Vortex Convection. The long-term convection of an isentropic vortex [43] tests the ability of the algorithm to preserve a flow feature for large simulation times. The vortex is a two-dimensional flow; however, we solve this flow over a three-dimensional domain in order to demonstrate the computational cost and efficiency of the three-dimensional solver. The CRWENO5 scheme shows a significant improvement in the preservation of the strength and shape of the vortex as it convects over a large distance [17]. In this study, we consider a large domain along the direction of vortex convection in order to evaluate the strong and weak scaling of the parallel algorithm for a large number of grid points and correspondingly large number of processors. The freestream flow is specified as $\rho_\infty = 1$ (density), $u_\infty = 0.5$ (x-velocity), $v_\infty = w_\infty = 0$ (y and z velocities), and $p_\infty = 1$ (pressure). The initial vortex is specified as

$$\begin{aligned} \rho &= \left[1 - \frac{(\gamma-1)b^2}{8\gamma\pi^2}\mathrm{e}^{1-r^2}\right]^{\frac{1}{\gamma-1}}, \\ \delta u &= -\frac{b}{2\pi}\mathrm{e}^{\frac{1-r^2}{2}}(y-y_c), \\ \delta v &= \frac{b}{2\pi}\mathrm{e}^{\frac{1-r^2}{2}}(x-x_c), \\ \delta w &= 0, \; p = \rho^\gamma, \end{aligned} \tag{4.1}$$

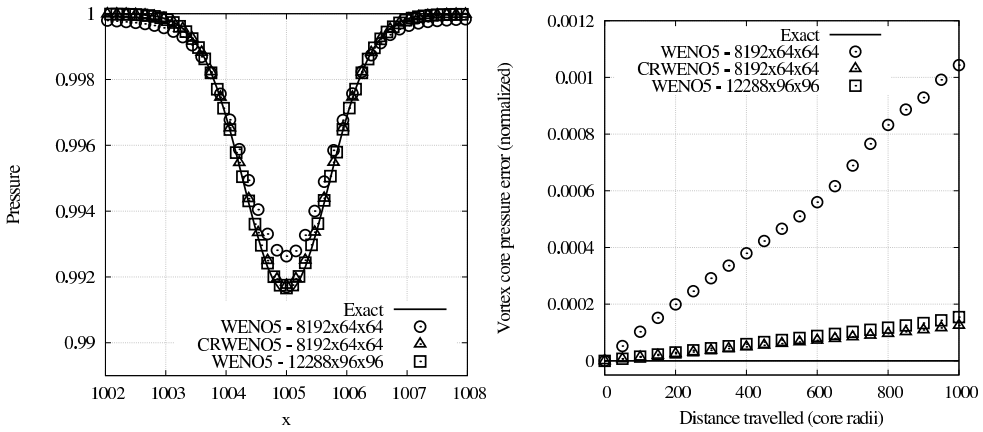
where δu, δv, and δw are the velocity perturbations, $(x_c, y_c) = (5, 5)$ is the initial location of the vortex center, $r = (x^2 + y^2)^{1/2}$ is the radial distance from the vortex center, $\gamma = 1.4$ is a ratio of specific heats, and $b = 0.5$ is the vortex strength. The vortex has a unit core radius. The flow is uniform along the z dimension. Periodic boundary conditions are specified on all boundaries.

The strong scaling of the algorithm is evaluated by solving the flow on a domain of length $1,280 \times 10 \times 10$. Solutions are obtained with the CRWENO5 and WENO5 schemes on a grid with $8,192 \times 64 \times 64$ points. Solutions are also obtained with the WENO5 scheme on a grid with 1.5^3 times more points ($12,288 \times 96 \times 96$ points). The vortex convects a distance of $1,000$ times the core radius. The solution is integrated in time by using the third-order-accurate strong-stability-preserving Runge-Kutta (SSPRK3) scheme [22] with a time-step size of 0.025 on both grids. Figure 4.1 shows the density contours of the flow for the solutions obtained with the WENO5 and CRWENO5 schemes. The solution obtained by the WENO5 scheme on the $8,192 \times 64 \times 64$ points grid is significantly dissipated, whereas the CRWENO5 scheme on this grid yields a solution comparable to the one obtained by the WENO5 scheme on the $12,288 \times 96 \times 96$ points grid. This is reiterated through Figs. 4.2(a) and 4.2(b), which show the cross-sectional pressure through the vortex core and the time history of the vortex core pressure error, respectively. We compare the wall times of the CRWENO5 scheme on the $8,192 \times 64 \times 64$ points grid with those of the WENO5 scheme on the $12,288 \times 96 \times 96$ points grid. The number of Jacobi iterations for the CRWENO5 scheme is fixed at 10, irrespective of the subdomain size. The domain is partitioned along all three dimensions. Figure 4.3(a) shows the wall times per core-radius distance traveled by the vortex (80 time steps) as a function of the number of processors. The subdomain sizes range from 4^3 (6^3 for WENO5) for $524,288$ ($2,048 \times 16 \times 16$) processors to 16^3 (24^3 for WENO5) for $8,192$ ($512 \times 4 \times 4$) processors. Although the CRWENO5 scheme does not scale as well as the WENO5 scheme



(a) 8192 × 64 × 64 points— CRWENO5 (b) 8192 × 64 × 64 points— WENO5 (c) 12288 × 96 × 96 points— WENO5

FIG. 4.1. *Isentropic vortex convection: density contours after vortex has traveled a distance* 1,000 *times its core radius.*

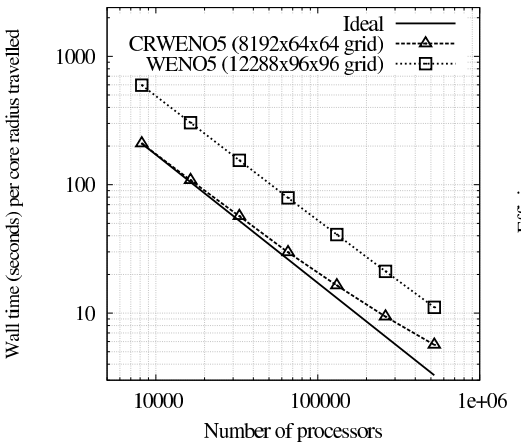


(a) Cross-sectional pressure (b) Core pressure error

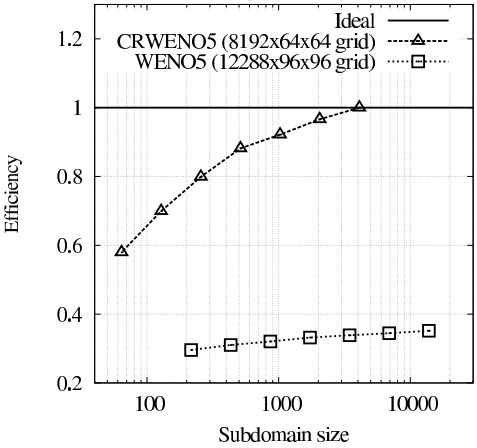
FIG. 4.2. *Isentropic vortex convection: cross-sectional pressure after vortex has traveled a distance* 1,000 *times its core radius, and pressure error at vortex center as a function of time.*

for larger numbers of processors, the absolute wall time is significantly lower. Figure 4.3(b) compares the modified parallel efficiencies of the two schemes as a function of the subdomain size. The efficiency of the CRWENO5 scheme decreases rapidly as the subdomain size decreases; however, in absolute terms, the CRWENO5 scheme is significantly more efficient than the WENO5 scheme even for the smallest subdomain size.

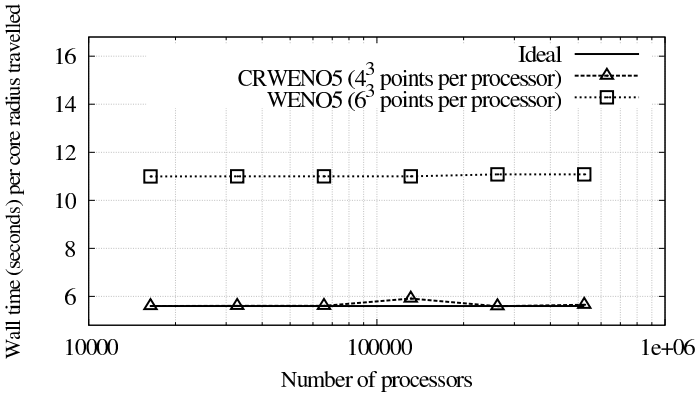
Figure 4.3(c) shows the wall times per core-radius distance traveled by the vortex of the CRWENO5 and WENO5 scheme for constant subdomain sizes of 4^3 and 6^3 points, respectively (the number of grid points and the number of processors are increased by the same factors). These results are obtained by varying the physical length, number of points, and number of processors along the direction of vortex convection while keeping these quantities along the other two dimensions constant. We initially start with a domain of size $40 \times 10 \times 10$ length units, discretized by a grid with $256 \times 64 \times 64$ points ($384 \times 96 \times 96$ points for WENO5) on $16,384$ ($64 \times 16 \times 16$) processors and increase the quantities in the x-dimension by a factor of two until a domain of size $1280 \times 10 \times 10$ length units, discretized by a grid with $8,192 \times 64 \times 64$



(a) Strong Scaling: Wall times



(b) Strong Scaling: Modified parallel efficiency

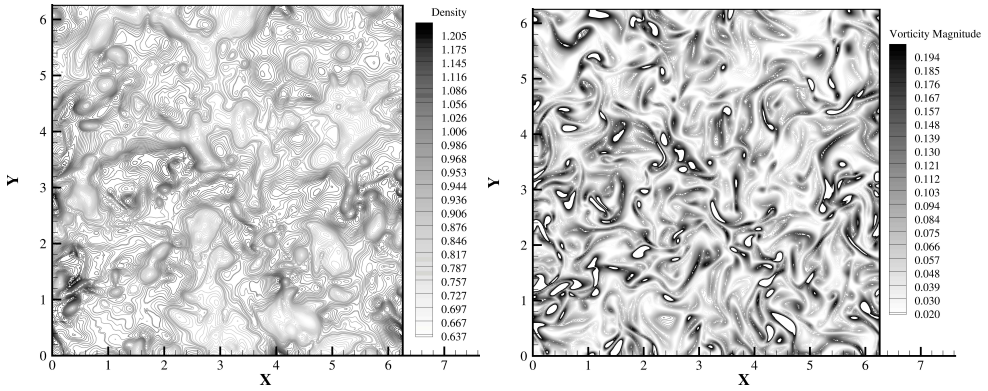


(c) Weak Scaling

FIG. 4.3. *Isentropic vortex convection: wall times and parallel efficiencies for the CRWENO5 and WENO5 schemes. the number of processors varies from 8,192 to 524,288.*

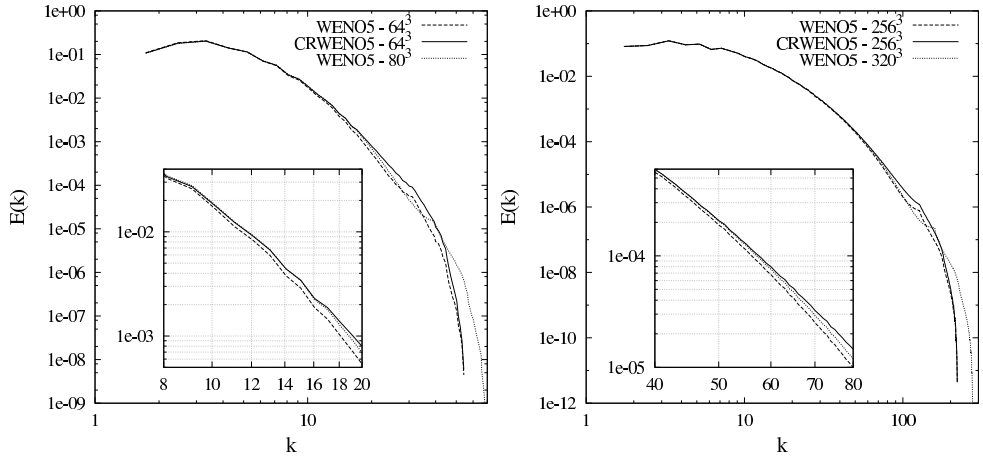
points ($12288\times96\times96$ points for WENO5) on $524,288$ ($2,048\times16\times16$) processors. The wall times for CRWENO5 scheme are significantly lower than those of the WENO5 scheme. The parallel implementation of the tridiagonal solver involves only point-to-point communications between processors, and thus an excellent weak scaling is observed. We therefore predict that the CRWENO5 scheme will remain more efficient than the WENO5 scheme as the problem sizes and the number of processors increase further.

4.2. Isotropic Turbulence Decay. The decay of an isotropic turbulent flow-field [41, 33] is a canonical turbulent flow problem and is characterized by a transfer of energy from larger to smaller length scales. The flow is compressible for higher values of the turbulent fluctuations, and a nonoscillatory scheme is required. Previous studies [16, 18] have demonstrated through direct numerical simulation that the CRWENO5 scheme yields solutions with higher resolution of moderate and high wavenumbers when compared with the WENO5 scheme on the same grid. A similar problem is solved in this paper to compare the computational costs of the two schemes for solutions of comparable resolution. The three-dimensional Navier-Stokes



(a) Density (b) Vorticity Magnitude

FIG. 4.4. *Isotropic turbulence decay: Solution at $Re_\lambda = 200$, obtained at $t/\tau = 3.0$ by CRWENO5 scheme on a grid with 256^3 points.*



(a) 64^3 grid, $Re_\lambda = 50$ (b) 256^3 grid, $Re_\lambda = 200$

FIG. 4.5. *Isotropic turbulence decay: energy spectrum at $t/\tau = 3.0$ for solutions obtained by the WENO5 and CRWENO5 schemes (inset figures are zoomed-in portions showing intermediate and small length scales).*

equations [25] are solved without a turbulence model; in addition to the numerical method described in Section 2, the viscous terms are discretized using fourth-order central differences. An initial solenoidal velocity field is specified that satisfies the following prescribed kinetic energy spectrum:

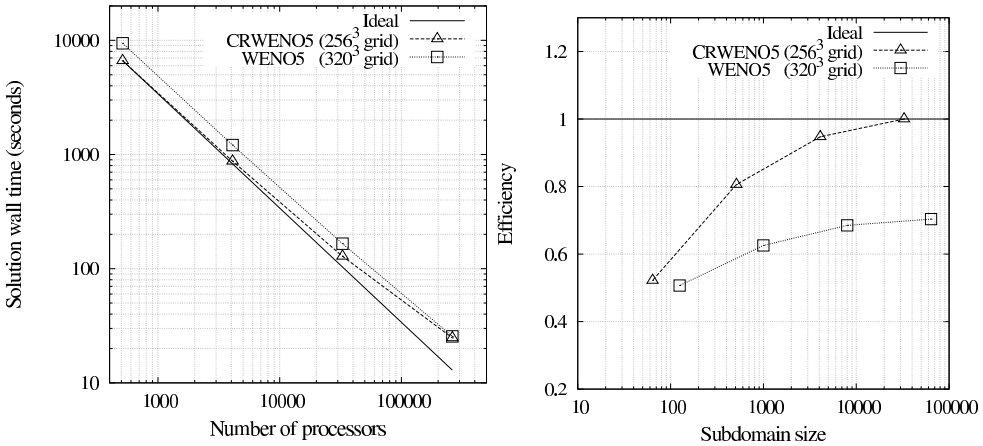
$$E(k) = 16\sqrt{\frac{2}{\pi}}\frac{u_0^2}{k_0}\left(\frac{k}{k_0}\right)^4 \exp\left[-2\left(\frac{k}{k_0}\right)^2\right], \tag{4.2}$$

where E is the kinetic energy, k is the wavenumber, $k_0 = 4$ is the wavenumber corresponding to the maximum kinetic energy, and $u_0 = 0.3$ is the RMS turbulence intensity. Constant initial density and pressure are specified ($\rho = 1$ and $p = 1/\gamma$, where $\gamma = 1.4$ is a ratio of specific heats). The procedure to specify the complete

initial solution is described in [41]. A periodic cubic domain is taken with edge length 2π. The problem is solved with the WENO5 and CRWENO5 schemes on two grids—64^3 and 256^3 points; in addition, solutions are obtained with the WENO5 scheme on grids that are 1.25 times as fine in each dimension (80^3 and 320^3 points). The flow is solved at initial Taylor microscale-based Reynolds numbers ($Re_\lambda = \rho u_0 \lambda/\mu$, where λ is the Taylor microscale and μ is the coefficient of viscosity) of 50 on the grids with $64^3$and 80^3 points, and 200 on the grids with 256^3 and 320^3 points. Solutions are obtained at a normalized time (t/τ, where $\tau = \lambda/u_0$ is the turbulent time scale based on the initial flow) of 3.0. The solutions are integrated in time with the four-stage fourth-order Runge-Kutta scheme, and the following time-step sizes are specified: 0.02 (CRWENO5 on 64^3 points), 0.03125 (WENO5 on 64^3 points), 0.025 (WENO5 on 80^3 points), 0.005 (CRWENO5 on 256^3 points), 0.008 (WENO5 on 256^3 points), and 0.00625 (WENO5 on 320^3 points). These values ensure that a fair comparison of the computational cost is made based on the linear stability limits of the two schemes. The initial turbulence intensity (u_0) results in a smooth, turbulent flow; however, the flow is characterized by severe gradients. These can be observed in Fig. 4.4, which shows the density and vorticity magnitude contours for the CRWENO5 solution obtained on the grid with 256^3 points. The number of Jacobi iterations for the tridiagonal solver in the CRWENO5 scheme is fixed at 10. Figure 4.5 shows the kinetic energy spectrum for the solutions obtained, with the inset in each figure showing the moderate and small length scales. The CRWENO5 scheme yields solutions with higher resolution than that of the WENO5 scheme on the same grid (64^3 and 256^3 points). At moderate length scales, the resolution of the CRWENO5 scheme on grids with 64^3 and 256^3 points is comparable to that of the WENO5 scheme on 80^3 and 320^3 points, respectively, whereas at smaller length scales, the CRWENO5 solutions have the highest resolution.

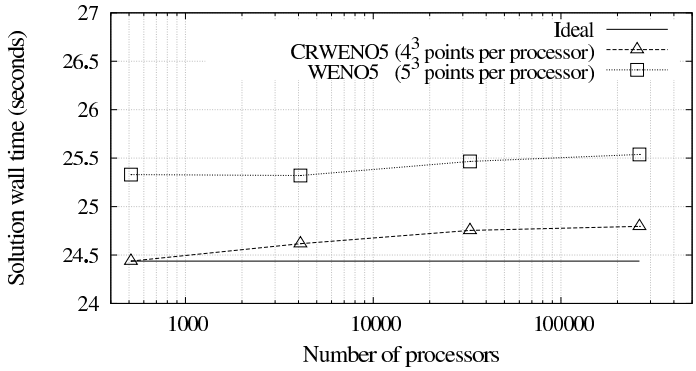
Figure 4.6(a) shows the solution wall times for the CRWENO5 scheme on the 256^3 points grid and the WENO5 scheme on the 320^3 points grid. The subdomain sizes vary from 4^3 (5^3 for WENO5) points for $262,144$ (64^3) processors to 32^3 (40^3 for WENO5) points on 512 (8^4) processors. Figure 4.6(b) shows the modified parallel efficiencies of the CRWENO5 and WENO5 schemes as a function of the subdomain sizes. Both figures show that the CRWENO5 scheme does not scale well at small subdomain sizes; however, it is more efficient than the WENO5 scheme for subdomain sizes larger than 4^3 points per processor, and the performances are similar at this subdomain size. Figure 4.6(c) shows the solution wall times of the CRWENO5 and WENO5 schemes with constant subdomain sizes of 4^3 and 5^3 points per processor, respectively. The problem sizes vary from 32^3 (40^3 for WENO5) points on 8^3 processors to 256^3 (320^3 for WENO5) points on 64^3 processors. The CRWENO5 scheme is observed to scale well and remains less expensive than the WENO5 scheme as the problem size increases.

5. Conclusions. We present an efficient parallel implementation of nonlinear compact schemes by applying the iterative substructuring approach to the solution of the tridiagonal system of equations. The diagonal dominance of the reduced system allows it to be solved iteratively to sufficient accuracy with few iterations, whose number is specified a priori. Collective communications, data transposition across processors, and complicated scheduling of computation and communications are avoided; minimal point-to-point communications between neighboring processors are required. Solutions on multiple processors are identical to those on a single processor; thus, parallelization does not affect the numerical properties (accuracy and resolution) of the compact schemes.



(a) Strong Scaling: Wall times

(b) Strong Scaling: Modified parallel efficiency



(c) Weak Scaling

FIG. 4.6. *Isotropic turbulence decay: wall times and parallel efficiencies for the CRWENO5 and WENO5 schemes. The number of processors varies from 512 to 262,144.*

In this paper we consider the CRWENO scheme as an example of a nonlinear compact scheme. The performance of this algorithm is demonstrated on manufactured solutions as well as physically relevant flow problems. We compare the computational cost of the CRWENO and WENO schemes as a function of the number of processors for comparable solutions. The effect of the increasing cost of the tridiagonal solver on the performance of the CRWENO scheme is demonstrated in one spatial dimension: it is computationally more efficient for larger subdomains; for smaller subdomains, the increasing cost of the tridiagonal solver renders it more expensive than the WENO scheme. The difference in the computational efficiencies of the CRWENO and WENO schemes is larger for three-dimensional problems, and the parallel tridiagonal solver achieves higher communication efficiency and arithmetic intensity. Our analysis on the IBM Blue Gene/Q architecture shows that the CRWENO scheme has a higher computational efficiency until very small subdomain sizes; at the smallest subdomain size considered (4 points per dimension), the efficiencies are similar. Our parallel implementation of the CRWENO scheme shows excellent weak scaling, compared with the noncompact WENO scheme. We demonstrate these properties on up to $\sim 500,000$ processors.

This study used the three- and four-stage explicit Runge-Kutta schemes for time integration, and the wall times for the CRWENO and WENO schemes are compared by allowing a larger CFL for the WENO scheme because of its higher linear stability limit. This is relevant for simulations where the solution is obtained with the largest possible stable time-step. Derivation and implementation of an optimal time-integration scheme for the CRWENO scheme (and other nonlinear compact schemes) are subjects of future research. Although this paper presents results for the CRWENO schemes, the implementation can be applied to other nonlinear compact schemes as well, such as the hybrid compact-WENO, WCS, and FVCW schemes.

The analysis presented in this paper and the conclusions drawn are based on the performance of our algorithm on the IBM BG/Q architecture, which is characterized by an excellent communication network. The scalability and parallel efficiency of our approach on other high-performance computing platforms will be investigated in the near future. In addition, performance improvements with alternative, platform-specific compilers will be explored.

Appendix A. Hardware and Software Details.

The computations presented in this study are carried out on the IBM Blue Gene/Q architecture. Smaller cases are solved on *Vesta*, a small development rack [2], while larger cases (including those presented in Section 4) are solved on *Mira* [1]. The two machines have identical hardware and software environments but differ by the number of racks—*Vesta* comprises two racks while *Mira* comprises 49, 152 racks. One rack of either system has 1024 compute nodes, each having a 1600 MHz PowerPC A2 processor with a 16-core chip and 16 GB RAM. Each core supports 4 hardware threads. A 17th core is available for the communication library. *Vesta* thus has 32, 768 cores with a peak performance of 419.44 teraflops, while *Mira* has 805, 306, 368 cores with a peak performance of 10 petaflops. The nodes are connected by a 5D torus network with 2 `GB/s` links.

The algorithm is coded in the `C` programming language, and the `GNU C` compiler suite is used to compile it. The `-O3` optimization flag is used. The Message Passing Interface (`MPI`) library is used to implement the parallel functions. We do not use any thread-based parallelism in our algorithm in this study. The performance tests are carried out by running 32 processes on each node of our platforms, or 2 processes per core to use the resources efficiently.

REFERENCES

[1] *ALCF/Mira Web page*, 2013. https://www.alcf.anl.gov/mira.
[2] *ALCF/Vesta Web page*, 2013. https://www.alcf.anl.gov/vesta.
[3] N. A. Adams, *Direct numerical simulation of turbulent compression ramp flow*, Theoretical and Computational Fluid Dynamics, 12 (1998), pp. 109–129.
[4] N. A. Adams and K. Shariff, *A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems*, Journal of Computational Physics, 127 (1996), pp. 27–51.
[5] I. Bermejo-Moreno, J. Bodart, J. Larsson, B. M. Barney, J. W. Nichols, and S. Jones, *Solving the compressible Navier-Stokes equations on up to 1.97 million cores and 4.1 trillion grid points*, in Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, SC '13, New York, NY, 2013, ACM, pp. 62:1–62:10.
[6] L. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. Whaley, *ScaLAPACK Users' Guide*, Society for Industrial and Applied Mathematics, 1997.

[7] J. Chao, A. Haselbacher, and S. Balachandar, *A massively parallel multi-block hybrid compact-WENO scheme for compressible flows*, Journal of Computational Physics, 228 (2009), pp. 7473–7491.

[8] S. C. Chen, D. J. Kuck, and A. H. Sameh, *Practical parallel band triangular systems solvers*, ACM Transactions on Mathematical Software, 4 (1978), pp. 270–277.

[9] A. Cleary and J. Dongarra, *Implementation in ScaLAPACK of divide-and-conquer algorithms for banded and tridiagonal linear systems*, Tech. Report UT-CS-97-358, University of Tennessee, Knoxville, TN, USA, 1997.

[10] A. W. Cook, W. H. Cabot, P. L. Williams, B. J. Miller, B. R. de Supinski, R. K. Yates, and M. L. Welcome, *Tera-scalable algorithms for variable-density elliptic hydrodynamics with spectral accuracy*, in Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, SC '05, Washington, DC, 2005, IEEE Computer Society, pp. 60–60.

[11] X. Deng and H. Maekawa, *Compact high-order accurate nonlinear schemes*, Journal of Computational Physics, 130 (1997), pp. 77–91.

[12] X. Deng and H. Zhang, *Developing high-order weighted compact nonlinear schemes*, Journal of Computational Physics, 165 (2000), pp. 22–44.

[13] J. J. Dongarra and A. H. Sameh, *On some parallel banded system solvers*, Parallel Computing, 1 (1984), pp. 223–235.

[14] D. Fauconnier and E. Dick, *Spectral analysis of nonlinear finite difference discretizations*, Journal of Computational and Applied Mathematics, 246 (2013), pp. 113–121. Fifth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2011).

[15] P. F. Fischer, F. P. Preparata, and J. E. Savage, *Generalized scans and tridiagonal systems*, Theoretical Computer Science, 255 (2001), pp. 423–436.

[16] D. Ghosh, *Compact-reconstruction weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, PhD thesis, University of Maryland, College Park, MD, 2013.

[17] D. Ghosh and J. D. Baeder, *Compact reconstruction schemes with weighted ENO limiting for hyperbolic conservation laws*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1678–A1706.

[18] ———, *Weighted non-linear compact schemes for the direct numerical simulation of compressible, turbulent flows*, Journal of Scientific Computing, 61 (2014), pp. 61–89.

[19] D. Ghosh, E. M. Constantinescu, and J. Brown, *Scalable nonlinear compact schemes*, Tech. Report ANL/MCS-TM-340, Argonne National Laboratory, Argonne, IL, April 2014.

[20] D. Ghosh, S. Medida, and J. D. Baeder, *Application of compact-reconstruction weighted essentially nonoscillatory schemes to compressible aerodynamic flows*, AIAA Journal, 52 (2014), pp. 1858–1870.

[21] S. Ghosh, *Direct numerical simulation of the interaction of a laser-induced plasma with isotropic turbulence*, PhD thesis, University of Minnesota, Minneapolis, MN, September 2008.

[22] S. Gottlieb, D. I. Ketcheson, and C.-W Shu, *High order strong stability preserving time discretizations*, Journal of Scientific Computing, 38 (2009), pp. 251–289.

[23] Y. Guo, T. Xiong, and Y. Shi, *A positivity-preserving high order finite volume compact-WENO scheme for compressible Euler equations*, Journal of Computational Physics, 274 (2014), pp. 505–523.

[24] A. K. Henrick, T. D. Aslam, and J. M. Powers, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, Journal of Computational Physics, 207 (2005), pp. 542–567.

[25] C. Hirsch, *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics: The Fundamentals of Computational Fluid Dynamics*, vol. 1 & 2, Elsevier Science, 2007.

[26] J. Hofhaus and E. Van de Velde, *Alternating-direction line-relaxation methods on multicomputers*, SIAM Journal on Scientific Computing, 17 (1996), pp. 454–478.

[27] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), pp. 202–228.

[28] L. Jiang, H. Shan, and C. Liu, *Weighted compact scheme for shock capturing*, International Journal of Computational Fluid Dynamics, 15 (2001), pp. 147–155.

[29] J. W. Kim and R. D. Sandberg, *Efficient parallel computing with a compact finite difference scheme*, Computers & Fluids, 58 (2012), pp. 70–87.

[30] C. B. Laney, *Computational Gasdynamics*, Cambridge University Press, 1998.

[31] C. Liu, L. Jiang, M. Visbal, and P. Xie, *Smart weighted compact scheme for shock tube and shock-entropy interaction*, in 36th AIAA Fluid Dynamics Conference and Exhibit, American Institute of Aeronautics and Astronautics, 2014/03/20 2006.

[32] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), pp. 200–212.
[33] N. N. Mansour and A. A. Wray, *Decay of isotropic turbulence at low Reynolds number*, Physics of Fluids (1994-present), 6 (1994), pp. 808–814.
[34] N. Mattor, T. J. Williams, and D. W. Hewett, *Algorithm for solving tridiagonal matrix problems in parallel*, Parallel Computing, 21 (1995), pp. 1769–1782.
[35] U. Meier, *A parallel partition method for solving banded systems of linear equations*, Parallel Computing, 2 (1985), pp. 33–43.
[36] S. Pirozzoli, *Conservative hybrid compact-WENO schemes for shock-turbulence interaction*, Journal of Computational Physics, 178 (2002), pp. 81–117.
[37] E. Polizzi and A. H. Sameh, *A parallel hybrid banded system solver: the SPIKE algorithm*, Parallel Computing, 32 (2006), pp. 177–194. Parallel Matrix Algorithms and Applications (PMAA04).
[38] ———, *SPIKE: A parallel environment for solving banded linear systems*, Computers & Fluids, 36 (2007), pp. 113–120. Challenges and Advances in Flow Simulation and Modeling.
[39] A. Povitsky and P. J. Morris, *A higher-order compact method in space and time based on parallel implementation of the Thomas algorithm*, Journal of Computational Physics, 161 (2000), pp. 182–203.
[40] Y.-X. Ren, M. Liu, and H. Zhang, *A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws*, Journal of Computational Physics, 192 (2003), pp. 365–386.
[41] R. S. Rogallo, *Numerical experiments in homogeneous turbulence*, Tech. Report NASA-TM-81315, NASA Ames Research Center, Moffett Field, CA, September 1981.
[42] Y. Saad, *Iterative Methods for Sparse Linear Systems: Second Edition*, Society for Industrial and Applied Mathematics, 2003.
[43] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Tech. Report NASA CR-97-206253 ICASE Report No. 97-65, Institute for Computer Applications in Science and Engineering, November 1997.
[44] C.-W. Shu, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM Review, 51 (2009), pp. 82–126.
[45] H. S. Stone, *An efficient parallel algorithm for the solution of a tridiagonal linear system of equations*, Journal of the ACM, 20 (1973), pp. 27–38.
[46] X.-H. Sun and S. Moitra, *A fast parallel tridiagonal algorithm for a class of CFD applications*, Tech. Report 3585, National Aeronautics and Space Administration, NASA Langley Research Center, Hampton, VA, August 1996.
[47] H. H. Wang, *A parallel method for tridiagonal equations*, ACM Transactions on Mathematical Software, 7 (1981), pp. 170–183.
[48] Z. Wang and G. P. Huang, *An essentially nonoscillatory high-order Padé-type (ENO-Padé) scheme*, Journal of Computational Physics, 177 (2002), pp. 37–58.
[49] P. Xie and C. Liu, *Weighted compact and non-compact scheme for shock tube and shock entropy interaction*, in 45th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2014/03/20 2007.
[50] S. Zhang, S. Jiang, and C.-W. Shu, *Development of nonlinear weighted compact schemes with increasingly higher order accuracy*, Journal of Computational Physics, 227 (2008), pp. 7294–7321.