A new class of central compact schemes with spectral-like resolution I: Linear schemes

Xuliang Liu[1], Shuhai Zhang[2], Hanxin Zhang[3] and Chi-Wang Shu[4]

Abstract

In this paper, we design a new class of central compact schemes based on the cell-centered compact schemes of Lele [S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16-42]. These schemes equate a weighted sum of the nodal derivatives of a smooth function to a weighted sum of the function on both the grid points (cell boundaries) and the cell-centers. In our approach, instead of using a compact interpolation to compute the values on cell-centers, the physical values on these half grid points are stored as independent variables and updated using the same scheme as the physical values on the grid points. This approach increases the memory requirement but not the computational costs. Through systematic Fourier analysis and numerical tests, we observe that the schemes have excellent properties of high order, high resolution and low dissipation. It is an ideal class of schemes for the simulation of multi-scale problems such as aeroacoustics and turbulence.

Key Words: Compact scheme, high resolution, direct numerical simulation, computational aeroacoustics.

[1]State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang, Sichuan 621000, China. E-mail: xlliu@foxmail.com.

[2]State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang, Sichuan 621000, China. E-mail: shuhai_zhang@163.com. Research supported by the Chinese National Natural Science Foundation grants 11172317, 91016001 and 973 program 2009CB724104.

[3]China Aerodynamics Research and Development Center, Mianyang, Sichuan 621000, China. Research supported by the Chinese National Natural Science Foundation grant 91016001.

[4]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu. Research partially supported by NASA grant NNX12AJ62A and NSF grant DMS-1112700.

1 Introduction

Direct numerical simulation (DNS) and large eddy simulation (LES) are two important methods to reveal the mechanism of multi-scale problems such as turbulence and aeroacoustics. Multi-scale problems possess a wide range of space and time scales. DNS for multi-scale problems requires that the numerical grid should be fine enough to resolve the structure of smallest scales. However, due to the limitation of computational resources, most DNS studies have been carried out with marginal grid resolution. For example, the grid size is chosen to be several times larger than the Kolmogorov length scale. Therefore, the energy spectrum is very flat in the resolved wavenumber range. In aeroacoustics, the flows that generate noises are nonlinear, unsteady and usually turbulent. Besides the common problems in DNS of turbulence, there are computational issues that are unique to aeroacoustics [30]. First, the aerodynamic noise is broadband and the spectrum is fairly wide. Second, the amplitudes of the physical variables of the aerodynamic noise are far smaller than those of the mean flow. For example, the radiated pressure fluctuation is less than 10^{-4} of the ambient pressure in turbo-jets with a noise level of 114dB, which is the "terrifyingly loud" noise, at the sideline point of certification [5]. Third, the distance from the noise source to the location of interest in aeroacoustic problems is quite long. To ensure that the computed solution is uniformly accurate over such a long propagation distance, the numerical scheme should have minimal numerical dispersion, dissipation, and anisotropy. The computation of these multi-scale problems requires that the numerical scheme has a good wave resolution, high order accuracy and low dissipation. For supersonic problems, the numerical scheme should also have the ability to capture shock waves. Of course, an obvious choice is spectral methods as they are uniformly accurate at all wave numbers. However, spectral methods are restricted to relatively simple computational domains. Also, for non-periodic boundary conditions, spectral methods typically require collocation at specific non-uniform points, which are not always convenient in simulations and could cause stiffness problems. Finite difference schemes are more flexible, however they usually do not have ideally small

numerical dispersion and dissipation errors. A possible compromise is the class of compact schemes [7, 14, 20, 23], which can be used on uniform meshes and usually have much smaller numerical dispersion and dissipation errors than finite difference schemes of the same order of accuracy on the same mesh.

The most influential compact schemes for derivatives, interpolation and filtering were proposed by Lele [15]. Through systematic Fourier analysis, it is shown that these compact schemes have spectral-like resolution for short waves. Through coupling the second derivatives, Mahesh [19] developed a family of compact schemes with good spectral-like resolution. Shukla and Zhong [29] developed a compact scheme for non-uniform meshes. Upwind compact schemes were also developed [4, 8, 35] for solving nonlinear hyperbolic problems. To improve the resolution of compact schemes, Lele [15] and Lui and Lele [18] optimized the coefficients of the compact schemes to improve the resolution by imposing the modified wave number of the finite difference scheme to be equal to the exact wave number at some specific control points. Tam and Webb [31] proposed the dispersion relation preserving (DRP) finite difference schemes. The resolution is improved significantly by minimizing the integrated errors over the range of the wave number of interest. The idea of DRP was extensively applied to construct compact schemes with better resolution [36, 3, 1, 13, 16] for spatial derivatives and Runge-Kutta schemes [11] for time derivatives.

The key issue of numerical schemes for DNS and LES is the resolution for short waves. In practice, cell-centered compact schemes have superiority to the collocated compact schemes. For example, the wave resolution of cell-centered compact scheme proposed by Lele [15] is much better than the resolution of collocated compact schemes. Nagarajan et al. [24] and Boersma [2] developed staggered compact schemes. Numerical tests indicate that their methods are quite robust. Beside this, Nagarajan et al. [24] thought that the staggered compact schemes have less aliasing error. However, the staggered compact schemes contain the cell-centered values, which should be obtained through interpolation from the values on grid nodes (cell boundaries). Lele [15] proposed a compact interpolation to compute

the values on the half-grid. Zhang et al. [34] developed a weighted interpolation method based on the WENO idea [17, 12] and formed a shock capturing weighted compact scheme. However, the application of interpolation introduces the transfer error, which can reduce the resolution.

In this paper, we propose a new idea to design the compact scheme based on the cell-centered compact scheme of Lele [15]. Instead of using only the values on cell centers, both the values of cell centers and grid nodes are used on the right hand side of compact schemes. The function values on both grid nodes and cell centers are directly computed with the same scheme instead of using interpolation to obtain values on cell centers. This approach increases the memory requirement, however it does not increase the computational cost, since the compact interpolation to compute the values on the half-grid is replaced by the compact formula to compute the spatial derivative (and the updating residue) at these half-grid points, with comparable computational cost. Both the accuracy order and the wave resolution property are improved significantly. Numerical tests show that this is an ideal scheme for the direct numerical simulation for multi-scale problems.

This paper is organized as follows. Section 2 presents the method to design our central compact scheme. Section 3 contains a Fourier analysis to systemically analyze the wave resolution of our designed schemes. Section 4 presents appropriate boundary closures and the implementation of a compact filter. The accuracy tests are performed in Section 5. In Section 6, our schemes are applied to compute Euler and Navier–Stokes equations. Numerical results are shown to demonstrate the good performance of the schemes. Finally concluding remarks are made in Section 7.

2 Central compact schemes

In this section, we present the methodology to design central compact schemes (CCS). We start our work from the cell-centered compact scheme proposed by Lele [15]. Then we extend this scheme to a class of higher order schemes with good spectral resolution. We consider

numerical approximations to the solution of the conservation law

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

A semidiscrete finite difference can be represented as:

$$\left(\frac{\partial u}{\partial t}\right)_i = -f'_i \tag{2.2}$$

where f'_i is the approximation to the spatial derivative $\frac{\partial f(u)}{\partial x}$ at the grid node x_i.

2.1 Lele's compact scheme

Lele [15] proposed two kinds of central compact schemes. One is a linear cell-centered compact scheme (CCCS) given by

$$\begin{aligned} &\beta f'_{j-2} + \alpha f'_{j-1} + f'_j + \alpha f'_{j+1} + \beta f'_{j+2} \\ &\quad = a\frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x} + b\frac{f_{j+\frac{3}{2}} - f_{j-\frac{3}{2}}}{3\Delta x} + c\frac{f_{j+\frac{5}{2}} - f_{j-\frac{5}{2}}}{5\Delta x} \end{aligned} \tag{2.3}$$

The other is a cell-node compact scheme (CNCS) given by

$$\begin{aligned} &\beta f'_{j-2} + \alpha f'_{j-1} + f'_j + \alpha f'_{j+1} + \beta f'_{j+2} \\ &\quad = a\frac{f_{j+1} - f_{j-1}}{2\Delta x} + b\frac{f_{j+2} - f_{j-2}}{4\Delta x} + c\frac{f_{j+3} - f_{j-3}}{6\Delta x}. \end{aligned} \tag{2.4}$$

The left hand sides of equations (2.3) and (2.4) contain the spatial derivatives f'_i at the grid nodes, while the right hand side of equation (2.3) contains the cell-centered values $f_{i+\frac{1}{2}}$ at the center $x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$ of a cell $x \in [x_i, x_{i+1}]$. The right hand side of equation (2.4), on the other hand, only contains the function values f_i at the grid node x_i. The stencil involved in the cell-centered compact scheme (2.3) is shown in Figure 2.1. The constraints on the coefficients α, β, a, b and c corresponding to different orders of accuracy can be derived by matching the Taylor series coefficients and these have been listed in Lele [15]. Lele [15] showed that the resolution of the cell-centered compact scheme (CCCS) is much better than the cell-node compact scheme (CNCS).

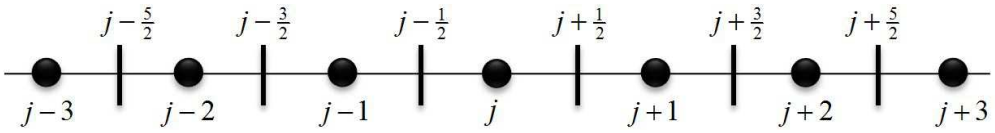


Figure 2.1: The stencil of cell-center and cell-node compact schemes

2.2 A new class of central compact schemes

In Lele's cell-centered compact schemes given by equation (2.3), the stencil contains both the grid point and half grid points $\{j-\frac{5}{2}, j-2, j-\frac{3}{2}, j-1, j-\frac{1}{2}, j, j+\frac{1}{2}, j+1, j+\frac{3}{2}, j+2, j+\frac{5}{2}\}$. However, only the values at the cell-centers $\{j-\frac{5}{2}, j-\frac{3}{2}, j-\frac{1}{2}, j+\frac{1}{2}, j+\frac{3}{2}, j+\frac{5}{2}\}$ are used to calculate the derivatives at the cell-nodes $\{j-2, j-1, j, j+1, j+2\}$. If the values at both the cell-nodes $\{j-2, j-1, j+1, j+2\}$ and the cell-centers $\{j-\frac{5}{2}, j-\frac{3}{2}, j-\frac{1}{2}, j+\frac{1}{2}, j+\frac{3}{2}, j+\frac{5}{2}\}$ are used, one could get a compact scheme with higher order accuracy and better resolution. Based on this idea, we design a class of central compact schemes (CCS) given by the following formula:

$$\begin{aligned} &\beta f'_{j-2} + \alpha f'_{j-1} + f'_j + \alpha f'_{j+1} + \beta f'_{j+2} \\ &\quad = a\frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x} + b\frac{f_{j+1} - f_{j-1}}{2\Delta x} + c\frac{f_{j+\frac{3}{2}} - f_{j-\frac{3}{2}}}{3\Delta x} \\ &\quad + d\frac{f_{j+2} - f_{j-2}}{4\Delta x} + e\frac{f_{j+\frac{5}{2}} - f_{j-\frac{5}{2}}}{5\Delta x} \end{aligned} \tag{2.5}$$

We note that the cell-node compact schemes (CNCS) given by equation (2.4) and cell-centered compact schemes (CCCS) (2.3) of Lele [15] are both special cases of this class of central compact schemes (CCS).

These schemes contain the values on the cell-centers, which are unknown. There are two methods to compute these unknowns. First, the physical values on cell-centers can be interpolated from the physical values of cell nodes. In fact, a high order compact interpolation was proposed by Lele [15], which has the following form:

$$\beta\hat{f}_{i-\frac{3}{2}} + \alpha\hat{f}_{i-\frac{1}{2}} + \hat{f}_{i+\frac{1}{2}} + \alpha\hat{f}_{i+\frac{3}{2}} + \beta\hat{f}_{i+\frac{5}{2}} = \frac{c}{2}(f_{i+3} + f_{i-2}) + \frac{b}{2}(f_{i+2} + f_{i-1}) + \frac{a}{2}(f_{i+1} + f_i) \tag{2.6}$$

Table 2.1 lists the coefficient constraints corresponding to different orders of accuracy for (2.6) [15]. We can use it to approximate the values on mid-cell points. Hereafter, we use

Table 2.1: Coefficients of the transfer function

order	parameters	truncation error
4^{th}	$a=\frac{1}{8}(9+10\alpha-14\beta+16c)$ $b=\frac{1}{8}(-1+6\alpha+30\beta-24c)$	$\frac{1}{128}(3-10\alpha+70\beta-128c)\triangle x^4 f^{(4)}$
6^{th}	$a=\frac{1}{64}(75+70\alpha-42\beta)$ $b=\frac{1}{128}(-25+126\alpha+270\beta)$ $c=\frac{1}{128}(3-10\alpha+70\beta)$	$\frac{1}{1024}(5-14\alpha+42\beta)\triangle x^6 f^{(6)}$
8^{th}	$\beta=\frac{1}{42}(14\alpha-5)$ $a=\frac{1}{8}(10+7\alpha)$ $b=\frac{1}{112}(-50+189\alpha)$ $c=\frac{1}{48}(-2+5\alpha)$	$\frac{1}{28672}(10-21\alpha)\triangle x^8 f^{(8)}$
10^{th}	$\beta=\frac{5}{126},\ \alpha=\frac{10}{21}$ $a=\frac{5}{3}, b=\frac{5}{14}$ $c=\frac{1}{126}$	$\frac{1}{258048}\triangle x^{10} f^{(10)}$

CCS-CI to represent the central compact scheme combined with the compact interpolation and CCCS-CI to represent the cell-centered compact scheme combined with the compact interpolation. However, the compact interpolation can introduce transfer errors, which will significantly reduce the resolution for high wave numbers. To overcome this drawback, we store the values at the cell centers as independent computational variables and use the same scheme for computing the updating values on cell nodes to compute the updating values on cell centers, by simply shifting the indices in (2.5) by 1/2. The resulting formula is

$$\begin{aligned}&\beta f'_{j-\frac{5}{2}}+\alpha f'_{j-\frac{3}{2}}+f'_{j-\frac{1}{2}}+\alpha f'_{j+\frac{1}{2}}+\beta f'_{j+\frac{3}{2}}\\ &\quad=a\frac{f_j-f_{j-1}}{\Delta x}+b\frac{f_{j+\frac{1}{2}}-f_{j-\frac{3}{2}}}{2\Delta x}+c\frac{f_{j+1}-f_{j-2}}{3\Delta x}\\ &\quad+d\frac{f_{j+\frac{3}{2}}-f_{j-\frac{5}{2}}}{4\Delta x}+e\frac{f_{j+2}-f_{j-3}}{5\Delta x}\end{aligned}\tag{2.7}$$

Noticed that, this change brings increased memory requirement for storing function values at cell centers, but it does not increase the computational cost, since the compact interpolation (2.6) is replaced by the compact updating (2.7) of comparable cost.

The relationships among the coefficients a, b, c, d, e and α, β in equation (2.5) or (2.7) are derived by matching the Taylor series coefficients of various orders. Schemes of order ranging from second to fourteenth can be obtained by solving the resulting set of linear

equations. The relationships of the coefficients for different orders are listed below:

Second order:

$$1+2\alpha+2\beta=\frac{2a}{1}\frac{(\frac{1}{2})^1}{1!}+\frac{2b}{2}\frac{(\frac{2}{2})^1}{1!}+\frac{2c}{3}\frac{(\frac{3}{2})^1}{1!}+\frac{2d}{4}\frac{(\frac{4}{2})^1}{1!}+\frac{2e}{5}\frac{(\frac{5}{2})^1}{1!} \tag{2.8}$$

Fourth order:

$$2\alpha\frac{1}{2!}+2\beta\frac{2^2}{2!}=\frac{2a}{1}\frac{(\frac{1}{2})^3}{3!}+\frac{2b}{2}\frac{(\frac{2}{2})^3}{3!}+\frac{2c}{3}\frac{(\frac{3}{2})^3}{3!}+\frac{2d}{4}\frac{(\frac{4}{2})^3}{3!}+\frac{2e}{5}\frac{(\frac{5}{2})^3}{3!} \tag{2.9}$$

Sixth order:

$$2\alpha\frac{1}{4!}+2\beta\frac{2^4}{4!}=\frac{2a}{1}\frac{(\frac{1}{2})^5}{5!}+\frac{2b}{2}\frac{(\frac{2}{2})^5}{5!}+\frac{2c}{3}\frac{(\frac{3}{2})^5}{5!}+\frac{2d}{4}\frac{(\frac{4}{2})^5}{5!}+\frac{2e}{5}\frac{(\frac{5}{2})^5}{5!} \tag{2.10}$$

Eighth order:

$$2\alpha\frac{1}{6!}+2\beta\frac{2^6}{6!}=\frac{2a}{1}\frac{(\frac{1}{2})^7}{7!}+\frac{2b}{2}\frac{(\frac{2}{2})^7}{7!}+\frac{2c}{3}\frac{(\frac{3}{2})^7}{7!}+\frac{2d}{4}\frac{(\frac{4}{2})^7}{7!}+\frac{2e}{5}\frac{(\frac{5}{2})^7}{7!} \tag{2.11}$$

Tenth order:

$$2\alpha\frac{1}{8!}+2\beta\frac{2^8}{8!}=\frac{2a}{1}\frac{(\frac{1}{2})^9}{9!}+\frac{2b}{2}\frac{(\frac{2}{2})^9}{9!}+\frac{2c}{3}\frac{(\frac{3}{2})^9}{9!}+\frac{2d}{4}\frac{(\frac{4}{2})^9}{9!}+\frac{2e}{5}\frac{(\frac{5}{2})^9}{9!} \tag{2.12}$$

Twelfth order:

$$2\alpha\frac{1}{10!}+2\beta\frac{2^{10}}{10!}=\frac{2a}{1}\frac{(\frac{1}{2})^{11}}{11!}+\frac{2b}{2}\frac{(\frac{2}{2})^{11}}{11!}+\frac{2c}{3}\frac{(\frac{3}{2})^{11}}{11!}+\frac{2d}{4}\frac{(\frac{4}{2})^{11}}{11!}+\frac{2e}{5}\frac{(\frac{5}{2})^{11}}{11!} \tag{2.13}$$

Fourteenth order:

$$2\alpha\frac{1}{12!}+2\beta\frac{2^{12}}{12!}=\frac{2a}{1}\frac{(\frac{1}{2})^{13}}{13!}+\frac{2b}{2}\frac{(\frac{2}{2})^{13}}{13!}+\frac{2c}{3}\frac{(\frac{3}{2})^{13}}{13!}+\frac{2d}{4}\frac{(\frac{4}{2})^{13}}{13!}+\frac{2e}{5}\frac{(\frac{5}{2})^{13}}{13!} \tag{2.14}$$

Solving the equations (2.8)-(2.14), one can get the coefficients of CCS. If the schemes are restricted to $\alpha=0$, $\beta=0$, a family of explicit CCS are obtained. If the schemes are restricted to $\alpha\neq 0$, $\beta=0$, a variety of tridiagonal CCS are obtained. If $\alpha\neq 0$ and $\beta\neq 0$, pentadiagonal CCS are generated. The three types of schemes are denoted by CCS-E, CCS-T and CCS-P respectively.

2.2.1 Explicit CCS schemes

The explicit schemes are generated by $\alpha = 0$, $\beta = 0$.

The fourth-order explicit schemes contain three free parameters. The coefficients are

$$a = \frac{1}{3}(4 + 5c + 12d + 21e), \quad b = -\frac{1}{3} - \frac{8c}{3} - 5d - 8e$$

The sixth-order explicit schemes have two free parameters. The coefficients are

$$a = \frac{1}{2}(3 - 7d - 28e), \quad b = -\frac{3}{5} + 7d + \frac{128e}{5}, \quad c = \frac{1}{10} - \frac{9d}{2} - \frac{63e}{5}$$

The eighth-order explicit schemes have one free parameter. The relationships of the coefficients are

$$a = \frac{2}{5}(4 + 21e), \quad b = -\frac{4}{5} - \frac{96e}{5}, \quad c = \frac{8}{35} + \frac{81e}{5}, \quad d = -\frac{1}{35} - \frac{32e}{5}$$

The coefficients of tenth order explicit scheme is uniquely defined as

$$a = \frac{5}{3}, \quad b = -\frac{20}{21}, \quad c = \frac{5}{14}, \quad d = -\frac{5}{63}, \quad e = \frac{1}{126}$$

We will denote these schemes by appending their formal order of accuracy to that of the schemes. For example, the explicit scheme described below is represented by CCS-E6

$$a = \frac{3}{2}, \quad b = -\frac{3}{5}, \quad c = \frac{1}{10}, \quad d = 0, \quad e = 0 \tag{2.15}$$

2.2.2 Tridiagonal CCS schemes

A family of tridiagonal systems are obtained, if the schemes are restricted to $\alpha \neq 0$, $\beta = 0$.

The fourth-order tridiagonal schemes have four free parameters. They are

$$a = \frac{1}{11}(12 - 8b - 3c + 4d + 13e), \quad \alpha = \frac{1}{22} + \frac{3b}{22} + \frac{4c}{11} + \frac{15d}{22} + \frac{12e}{11}$$

The sixth order tridiagonal schemes contain three free parameters which are given by

$$a = \frac{1}{9}(16 - 25c - 144d - 441e), \quad b = -\frac{17}{18} + \frac{31c}{9} + \frac{45d}{2} + 69e, \quad \alpha = -\frac{1}{12} + \frac{5c}{6} + \frac{15d}{4} + \frac{21e}{2}$$

The eighth order tridiagonal schemes have two free parameters. The relationships of the coefficients are

$$a = 2(1 + 7d + 49e), \quad b = -\frac{61}{50} - \frac{147d}{10} - \frac{2832e}{25},$$
$$c = -\frac{2}{25} - \frac{54d}{5} - \frac{1323e}{25}, \quad \alpha = -\frac{3}{20} - \frac{21d}{4} - \frac{168e}{5}$$

The tenth order tridiagonal schemes have one free parameter. The relationships of the coefficients are

$$a = -\frac{2}{15}(-16 + 441e), \quad b = -\frac{34}{25} + \frac{1284e}{25}, \quad c = -\frac{32}{175} + \frac{1701e}{25},$$
$$d = \frac{1}{105} - \frac{56e}{5}, \quad \alpha = -\frac{1}{5} + \frac{126e}{5}$$

The coefficients of the twelfth order tridiagonal scheme are uniquely defined by

$$a = \frac{20}{9}, \quad b = -\frac{634}{441}, \quad c = -\frac{2}{7}, \quad d = \frac{5}{189}, \quad e = -\frac{2}{1323}, \quad \alpha = -\frac{5}{21}$$

We will again denote these schemes by appending their formal order of accuracy to that of the schemes. For example, the tridiagonal scheme described below is represented by CCS-T8

$$a = 2, \quad b = -\frac{61}{50}, \quad c = -\frac{2}{25}, \quad \alpha = -\frac{3}{20}, \quad d = 0, \quad e = 0, \quad \beta = 0 \tag{2.16}$$

Another tridiagonal scheme CCS-T6 is

$$a = \frac{16}{9}, \quad b = -\frac{17}{18}, \quad \alpha = -\frac{1}{12}, \quad c = 0, \quad d = 0, \quad e = 0, \quad \beta = 0 \tag{2.17}$$

2.2.3 Pentadiagonal CCS schemes

Pentadiagonal schemes are obtained with $\alpha \neq 0$ and $\beta \neq 0$.

The sixth order pentadiagonal schemes contain four free parameters. The relationships of the coefficients are given by

$$a = \frac{1}{863}(960 - 608b - 303c - 128d - 335e),$$
$$\alpha = \frac{154}{2589} + \frac{261b}{1726} + \frac{809c}{2589} + \frac{300d}{863} + \frac{57e}{863},$$
$$\beta = -\frac{17}{5178} - \frac{3b}{863} + \frac{31c}{2589} + \frac{135d}{1726} + \frac{207e}{863}$$

The eighth order pentadiagonal schemes contain three free parameters. The relationships of the coefficients are given by

$$a = \frac{6400 - 5725c - 13824d + 33075e}{3429}, \quad b = -\frac{3670}{3429} + \frac{25669c}{13716} + \frac{700d}{127} - \frac{7235e}{508},$$
$$\alpha = -\frac{13}{127} + \frac{605c}{1016} + \frac{150d}{127} - \frac{2121e}{1016}, \quad \beta = \frac{1}{2286} + \frac{25c}{4572} + \frac{15d}{254} + \frac{147e}{508}$$

The tenth order pentadiagonal schemes contain two free parameters. The relationships of the coefficients are given by

$$a = \frac{2(4336 + 4704d - 68061e)}{4107},$$
$$b = -\frac{138098}{102675} - \frac{2156d}{1369} + \frac{1154116e}{34225}, \quad c = -\frac{5024}{34225} - \frac{5184d}{1369} + \frac{877149e}{34225},$$
$$\alpha = -\frac{1299}{6845} - \frac{1470d}{1369} + \frac{90174e}{6845}, \quad \beta = -\frac{1}{2738} + \frac{105d}{2738} + \frac{588e}{1369}$$

The twelfth order pentadiagonal schemes contain two free parameters. The relationships of the coefficients are given by

$$a = \frac{2}{225}(256 + 3969e), \quad b = -\frac{328}{225} - \frac{333e}{25}, \quad c = -\frac{512}{1225} - \frac{2187e}{25},$$
$$d = \frac{1579}{22050} + \frac{2987e}{100}, \quad \alpha = -\frac{4}{15} - \frac{189e}{10}, \quad \beta = \frac{1}{420} + \frac{63e}{40}$$

The coefficients of the fourteenth order pentadiagonal scheme are uniquely defined by

$$a = \frac{64}{27}, \quad b = -\frac{1976}{1323}, \quad c = -\frac{32}{49}, \quad d = \frac{3617}{23814}, \quad e = \frac{32}{11907}, \quad \alpha = -\frac{20}{63}, \quad \beta = \frac{5}{756}$$

The coefficients for the three different types of schemes are presented in Table 2.2.

The sixth-order tridiagonal (CCS-T6) and eighth-order tridiagonal(CCS-T8) schemes are found to be efficient and economical comparing with the pentadiagonal schemes, because they require only a tridiagonal matrix solver that is easy and fast for actual computation. In other words, these schemes seem to have the best combinations of the resolution characteristics, order of accuracy, and efficiency. They appear to be very effective compact schemes.

2.3 Time advancement

After the spatial derivative is discretized by the compact scheme (2.5)-(2.7), we obtain a system of initial value problems of ordinary differential equations (ODEs),

$$\frac{dU}{dt} = L(U) \tag{2.18}$$

Table 2.2: The coefficients of the three different types of schemes

Scheme	a	b	c	d	e	α	β	Order
CCS-E4	$\frac{4}{3}$	$-\frac{1}{3}$	0	0	0	0	0	4
CCS-E6	$\frac{3}{2}$	$-\frac{3}{5}$	$\frac{1}{10}$	0	0	0	0	6
CCS-E8	$\frac{8}{5}$	$-\frac{4}{5}$	$\frac{8}{35}$	$-\frac{1}{35}$	0	0	0	8
CCS-E10	$\frac{5}{3}$	$-\frac{20}{21}$	$\frac{5}{14}$	$-\frac{5}{63}$	$\frac{1}{126}$	0	0	10
CCS-T4	$\frac{12}{11}$	0	0	0	0	$\frac{1}{22}$	0	4
CNCS-T4	0	$\frac{3}{2}$	0	0	0	$\frac{1}{4}$	0	4
CCS-T6	$\frac{16}{9}$	$-\frac{17}{18}$	0	0	0	$-\frac{1}{12}$	0	6
CNCS-T6	0	$\frac{14}{9}$	0	$\frac{1}{9}$	0	$\frac{1}{3}$	0	6
CCCS-T6	$\frac{63}{62}$	0	$\frac{17}{62}$	0	0	$\frac{9}{62}$	0	6
CCS-T8	2	$-\frac{61}{50}$	$-\frac{2}{25}$	0	0	$-\frac{3}{20}$	0	8
CCCS-T8	$\frac{2675}{2832}$	0	$\frac{925}{1888}$	0	$-\frac{61}{5664}$	$\frac{25}{118}$	0	8
CCS-T10	$\frac{32}{15}$	$-\frac{34}{25}$	$-\frac{32}{175}$	$\frac{1}{105}$	0	$-\frac{1}{5}$	0	10
CCS-T12	$\frac{20}{9}$	$-\frac{634}{441}$	$-\frac{2}{7}$	$\frac{5}{189}$	$-\frac{2}{1323}$	$-\frac{5}{21}$	0	12
CCS-P6	$\frac{960}{863}$	0	0	0	0	$\frac{154}{2589}$	$-\frac{17}{5178}$	6
CNCS-P6	0	$\frac{30}{19}$	0	0	0	$\frac{17}{57}$	$-\frac{1}{114}$	6
CCS-P8	$\frac{6400}{3429}$	$-\frac{3670}{3429}$	0	0	0	$-\frac{13}{127}$	$\frac{1}{2286}$	8
CNCS-P8	0	$\frac{40}{27}$	0	$\frac{25}{54}$	0	$\frac{4}{9}$	$\frac{1}{36}$	8
CCCS-P8	$\frac{23400}{25669}$	0	$\frac{14680}{25669}$	0	0	$\frac{6114}{25669}$	$\frac{183}{51338}$	8
CCS-P10	$\frac{8672}{4107}$	$-\frac{138098}{102675}$	$-\frac{5024}{34225}$	0	0	$-\frac{1299}{6845}$	$-\frac{1}{2738}$	10
CCCS-P10	$\frac{683425}{865587}$	0	$\frac{505175}{577058}$	0	$\frac{69049}{1731174}$	$\frac{96850}{288529}$	$\frac{9675}{577058}$	10
CCS-P12	$\frac{512}{225}$	$-\frac{328}{225}$	$-\frac{512}{1225}$	$\frac{1579}{22050}$	0	$-\frac{4}{15}$	$\frac{1}{420}$	12
CCS-P14	$\frac{64}{27}$	$-\frac{1976}{1323}$	$-\frac{32}{49}$	$\frac{3617}{23814}$	$\frac{32}{11907}$	$-\frac{20}{63}$	$\frac{5}{756}$	14

where the operator $L(U)$ is an approximation to the spatial derivative in the partial differential equations (PDEs). This set of ODEs can be discretized by a third order TVD Runge–Kutta method [27, 28, 9], which is given as follows:

$$\begin{aligned} U^{(1)} &= U^n + \Delta t L(U^n) \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)}) \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}) \end{aligned} \tag{2.19}$$

Higher order versions of such time discretizations can of course also be used.

Updating the physical variables for a given time dependent equation contains two steps. First, the spatial derivative is computed by the numerical scheme of equations (2.5) and (2.7) for cell nodes and cell centers respectively. Then, both spatial derivatives of cell nodes and cell centers are substituted into equation (2.19) to update the physical variables on cell nodes and cell centers. The process can be schematically shown by Figure 2.2 for the two-dimensional case. The nodes are marked by filled circles, and the cell centers are marked by circles and crosses. The circles represent the cell centers of one direction, and the crosses represent the cell centers in both directions.

3 Fourier analysis of the errors

In this section, we analyze the dispersion and dissipation characteristics of CCS using Fourier analysis, and study the performance in terms of points per wavelength (PPW).

Fourier transformation and its inverse transformation have the following formulae:

$$\tilde{f}(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(x)e^{-ikx}dx \tag{3.1}$$

$$f(x) = \int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx}dk \tag{3.2}$$

where $i = \sqrt{-1}$, k is the wavenumber, and $\tilde{f}(k)$ represents the Fourier transformed function of $f(x)$. Fourier transformation is a common tool to analyze a finite difference scheme. Taking the above Fourier transformation to equation (2.5) and using Euler's formula, the

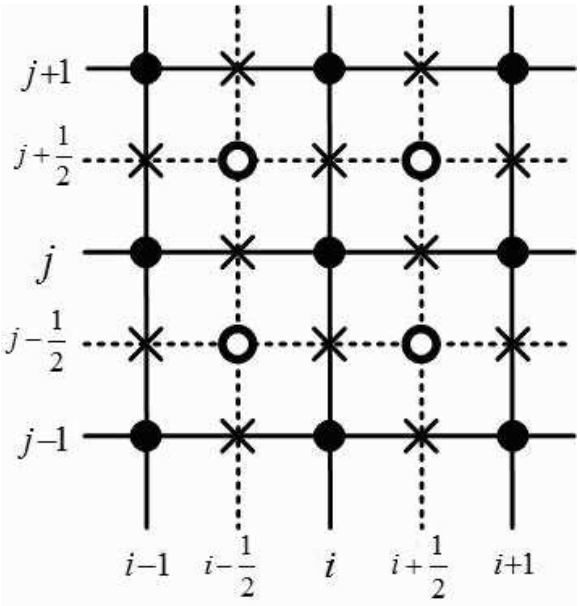


Figure 2.2: The schematic diagram for updating the values at cell nodes and cell centers of the two-dimensional case.

modified wavenumber of CCS can be obtained. It is:

$$w' = 2\frac{a\sin(\frac{w}{2}) + \frac{b\sin(w)}{2} + \frac{c\sin(\frac{3w}{2})}{3} + \frac{d\sin(2w)}{4} + \frac{e\sin(\frac{5w}{2})}{5}}{2\beta\cos(2w) + 2\alpha\cos(w) + 1} \tag{3.3}$$

where $w = k\Delta x$ is a scaled wavenumber, and $w' = k'\Delta x$ is a scaled modified wavenumber.

Figure 3.1 shows the modified wavenumber of CCS and the comparison with those of CCCS and CNCS. It is clear that the resolutions of CCS are much better than those of CCCS and CNCS. The difference between the modified wavenumber of CCS and the exact wavenumber is very small. Especially for pentadiagonal type schemes, we can not distinguish the modified wavenumber of CCS with the exact wavenumber on the graph. Therefore, these schemes have spectral-like resolution. Because CCS is symmetric, it has no dissipation error. On the other hand, the accuracy of the compact interpolation influences seriously the resolution of the central compact schemes. The accuracy order of the compact interpolation should be two order higher than the central compact scheme in order to minimize this influence.

Lele [15] defined the resolving efficiency of a scheme as $e = \frac{w_f}{\pi}$, where w_f is the shortest

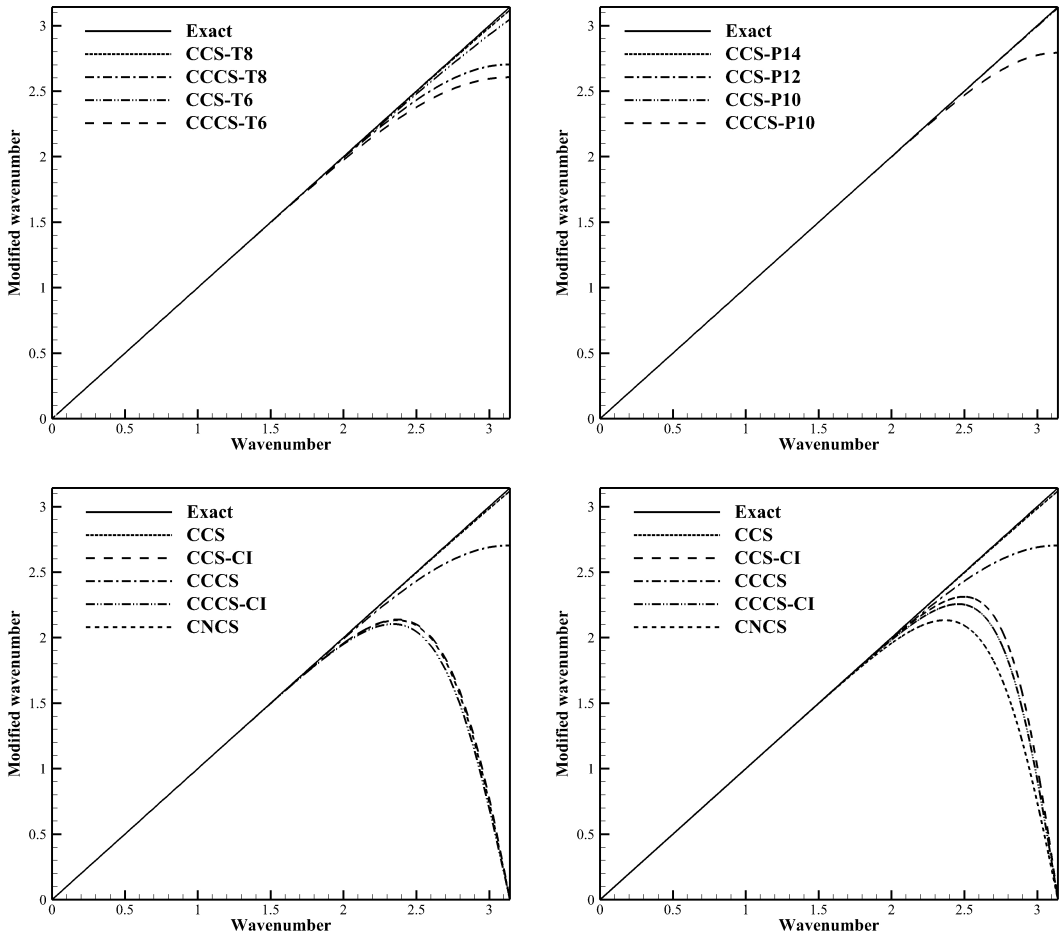


Figure 3.1: Modified wavenumber of CCS and comparison with CCCS and CNCS. Top left: sixth and eighth order tridiagonal schemes; Top right: tenth, twelfth and fourteenth order pentadiagonal schemes; Bottom left: CCS combined with eighth order compact interpolation; Bottom right: CCS combined with tenth order compact interpolation.

Table 3.1: The shortest well-resolved wave w_f, resolving efficiency e and PPW for $\epsilon = 0.01$. $CI8$ represents eighth order compact interpolation, $CI10$ represents tenth order compact interpolation

Scheme	w_f	e	PPW
CCS	3.118	0.992	2.015
CCS-CI8	1.822	0.580	3.448
CCS-CI10	2.100	0.668	2.992
CCCS	2.182	0.695	2.880
CCCS-CI8	1.783	0.567	3.525
CCCS-CI10	1.996	0.635	3.148
CNCS	1.816	0.578	3.460

well-resolved wave, which depends on the specific error tolerance defined by:

$$\left|1 - \frac{w'}{w}\right| \leq \varepsilon$$

When the comparison is implemented among different schemes, the error tolerance should be fixed. The ''resolution'' of spatial discretization is usually represented by the minimum points-per-wavelength (PPW) , which is needed to resolve the wave. Here, the PPW will be computed by PPW= $\frac{2\pi}{w_f}$ [11]. Table 3.1 and Table 3.2 contain w_f, e and PPW and their comparisons among different schemes with the error tolerance $\epsilon = 0.01$ and 0.001 respectively. Again, we find that CCS provides the best value of PPW, which is 2.015 and 2.657 for $\epsilon = 0.01$ and 0.001 respectively. They are the smallest values among all the schemes compared in the tables. The accuracy of compact interpolation influences seriously the features of numerical schemes including the resolving efficiency, the shortest waves and PPW. Notice that CCS does involve double-memory per grid point, hence PPW should be understood in the sense of "cells per wave" similar to that for Hermite schemes and discontinuous Galerkin schemes.

To test the performance of PPW, we compute a sine wave dominated by an advection scalar equation over a distance of 10 times its wave length. The numerical grids in a wavelength equals to the PPW listed in Table 3.2. Figure 3.2 contains the numerical results by CCS, CCS-CI, CCCS and CNCS. They are quite satisfactory on eye viewing.

Table 3.2: The shortest well-resolved wave w_f, resolving efficiency e and PPW for $\epsilon = 0.001$. $CI8$ represents eighth order compact interpolation, $CI10$ represents tenth order compact interpolation

Scheme	w_f	e	PPW
CCS	2.365	0.753	2.657
CCS-CI8	1.407	0.448	4.465
CCS-CI10	1.731	0.551	3.630
CCCS	1.662	0.529	3.780
CCCS-CI8	1.373	0.437	4.578
CCCS-CI10	1.583	0.504	3.968
CNCS	1.400	0.445	4.490

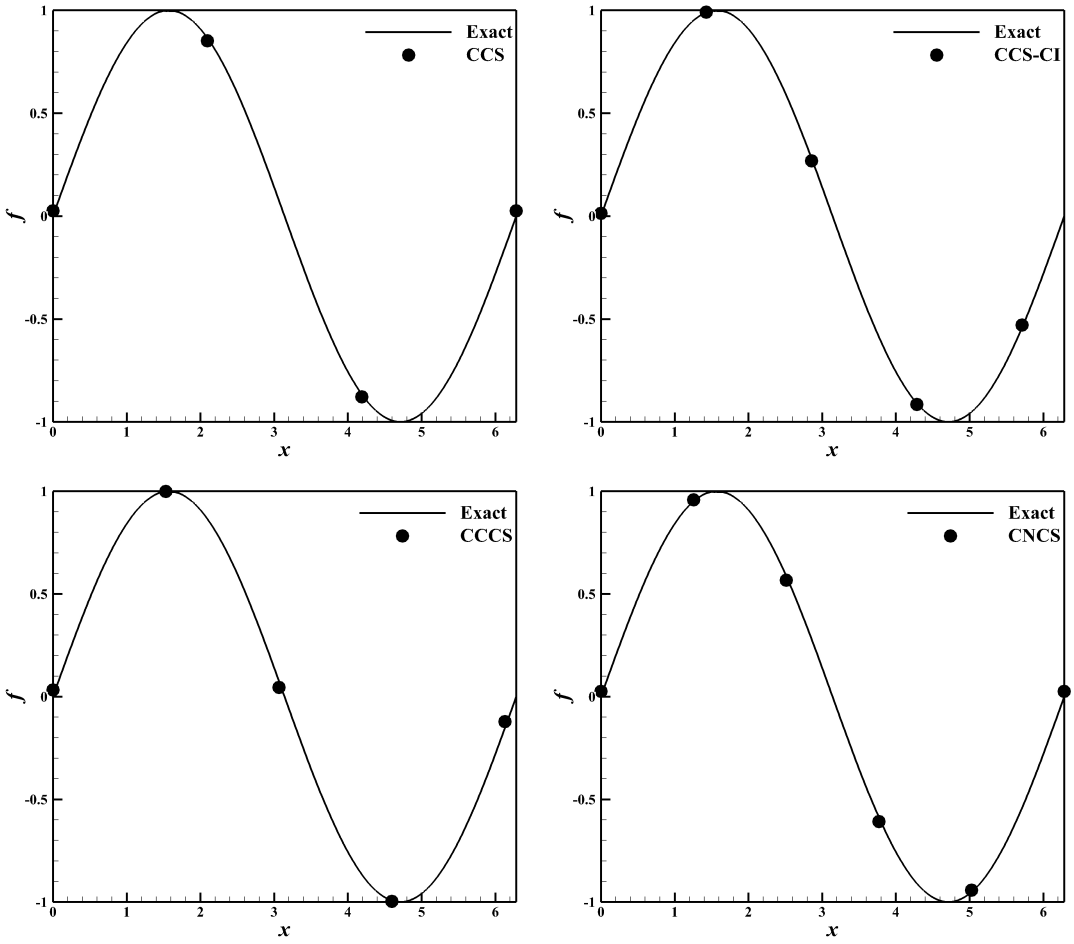


Figure 3.2: The performance of the points per wave of CCS (top left), CCS-CI (top right), CCCS (bottom left) and CNCS (bottom right).

4 Boundary closure and a filtering scheme

There are two important issues that should be considered for the linear central compact schemes. One is a stable boundary closure that preserves high order global formal accuracy. Another is adequate numerical dissipation to damp the unresolvable high-frequency modes (short waves) by the difference discretization.

4.1 Boundary scheme

Many physical problems for computation are non-periodic. Hence, the boundary conditions are also non-periodic in the computational domain for these problems, and the boundary schemes are needed to compute the physical values near the boundaries.

In this paper, we use the same formula to compute the first derivatives on the boundaries as for the inner points. The formula is

$$\alpha f'_{j-1} + f'_j + \alpha f'_{j+1} = a\frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x} + b\frac{f_{j+1} - f_{j-1}}{2\Delta x} + c\frac{f_{j+\frac{3}{2}} - f_{j-\frac{3}{2}}}{3\Delta x}$$

The relationships between the coefficients are

$$a = \frac{9 - 20\alpha}{6}, \quad b = \frac{-9 + 62\alpha}{15}, \quad c = \frac{1 + 12\alpha}{10}$$

The coefficients are different for the points near the boundary and for the inner points. At the inner points, we choose $\alpha = -\frac{3}{20}$. At the boundary point, we choose $\alpha = 0$. At the points near the boundary, we choose $\alpha = -\frac{1}{12}$. The physical values on the ghost points, such as $f_{-\frac{3}{2}}$, f_{-1} and $f_{-\frac{1}{2}}$, are computed by extrapolation of suitable orders of accuracy.

4.2 Filtering Scheme

Like other central schemes, CCS is nondissipative and are therefore subject to numerical instabilities for nonlinear problems due to the growth of high-frequency modes. High order filtering can be adopted to remove the high-frequency instabilities.

The implicit filter that we used has the following form

$$\beta\hat{f}_{i-2} + \alpha\hat{f}_{i-1} + \hat{f}_i + \alpha\hat{f}_{i+1} + \beta\hat{f}_{i+2} = \sum_{n=0}^{N} \frac{a_n}{2}(f_{i-n} + f_{i+n}) \tag{4.1}$$

Table 4.1: Coefficients for the filter formula

Scheme	a_0	a_1	a_2	a_3	a_4	Order
F2	$\frac{1}{2}+\alpha$	$\frac{1}{2}+\alpha$	0	0	0	2
F4	$\frac{5}{8}+\frac{3}{4}\alpha$	$\frac{1}{2}+\alpha$	$-\frac{1}{8}+\frac{1}{4}\alpha$	0	0	4
F6	$\frac{11}{16}+\frac{5}{8}\alpha$	$\frac{15}{32}+\frac{17}{16}\alpha$	$\frac{-3}{16}+\frac{3}{8}\alpha$	$\frac{1}{32}-\frac{1}{16}\alpha$	0	6
F8	$\frac{93+70\alpha}{128}$	$\frac{7+18\alpha}{16}$	$\frac{-7+14\alpha}{32}$	$\frac{1}{16}-\frac{1}{8}\alpha$	$\frac{-1}{128}+\frac{1}{64}\alpha$	8

The problem is most naturally formulated in terms of the transfer function associated with equation (4.1),

$$T(w)=\frac{\sum_{n=0}^{N} a_n \cos(nw)}{1+2\alpha\cos(w)+2\beta\cos(2w)}$$

The coefficients are derived with Taylor and Fourier series analysis and are presented in Table 4.1.

The filter is typically chosen to be at least two orders of accuracy higher than the difference scheme [32]. Accordingly, we choose F8 ($\alpha = 0.45$) at the inner points, and F6 ($\alpha = 0$) at the boundary points.

5 Numerical accuracy tests

In this section, we test the accuracy of CCS. To save space, we just list the numerical result for CCS-T8. In our computation, we have adjusted the time step to $\Delta t = \Delta x^{\frac{8}{3}}$ so that the error of the time discretization will not dominate.

We solve the following linear scalar equation with periodic boundary conditions:

$$\begin{aligned} u_t + u_x &= 0, -1 \le x \le 1 \\ u(x, t=0) &= u_0(x) \end{aligned} \tag{5.1}$$

We test three different initial conditions: $u_0(x) = \sin(\pi x)$, $u_0(x) = \sin(\pi x - \sin(\pi x)/\pi)$, and $u_0(x) = \sin^4(\pi x)$. No filtering is used for these tests.

In Table 5.1, Table 5.2 and Table 5.3, the L_1 and L_∞ errors and numerical orders of accuracy are given for CCS-T8. We can observe that the designed order of accuracy is achieved in all cases.

Table 5.1: L_1 and L_∞ errors and numerical accuracy orders of CCS-T8 on $u_t + u_x = 0$ with $u_0(x) = \sin(\pi x)$. N is the total number of grid points in a uniform mesh. $t = 1$.

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	0.642E-05	–	0.992E-05	–
20	0.257E-07	7.97	0.406E-07	7.93
40	0.101E-09	7.99	0.159E-09	8.00
80	0.394E-12	8.00	0.619E-12	8.00
160	0.154E-14	8.00	0.242E-14	8.00

Table 5.2: L_1 and L_∞ errors and numerical accuracy orders of CCS-T8 on $u_t + u_x = 0$ with $u_0(x) = \sin(\pi x - \sin(\pi x)/\pi)$. N is the total number of grid points in a uniform mesh. $t = 2$.

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	0.424E-04	–	0.119E-03	–
20	0.196E-06	7.76	0.685E-06	7.45
40	0.881E-09	7.80	0.307E-08	7.80
80	0.357E-11	7.95	0.126E-10	7.93
160	0.141E-13	7.99	0.496E-13	7.98

Table 5.3: L_1 and L_∞ errors and numerical accuracy orders of CCS-T8 on $u_t + u_x = 0$ with $u_0(x) = \sin^4(\pi x)$. N is the total number of grid points in a uniform mesh. $t = 10$.

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	0.149E-01	–	0.242E-01	–
20	0.987E-04	7.24	0.156E-03	7.28
40	0.483E-06	7.67	0.760E-06	7.68
80	0.205E-08	7.88	0.327E-08	7.86
160	0.818E-11	7.97	0.130E-10	7.97

6 Numerical experiments

In this section, we apply CCS-T8 as an example of CCS to simulate Euler and Navier-Stokes equations. For non-periodic problems, the sixth order boundary scheme given in Section 4 is used. The filtering F8 with the boundary adjustment F6 as described in Section 4.2 is used.

6.1 Governing equations

We consider the three-dimensional compressible nonlinear Navier–Stokes equations written in the conservation form:

$$\frac{\partial Q}{\partial t} + \frac{\partial \left(F_i - F_v\right)}{\partial x} + \frac{\partial \left(G_i - G_v\right)}{\partial y} + \frac{\partial \left(H_i - H_v\right)}{\partial z} = 0 \tag{6.1}$$

where Q is the vector of conserved variables

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}$$

F_i, G_i, H_i, F_v, G_v and H_v are the inviscid and viscous flux vectors in the x, y and z directions respectively, with the form:

$$F_i = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix}, \qquad F_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x \end{bmatrix}$$

$$G_i = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{bmatrix}, \qquad G_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y \end{bmatrix}$$

$$H_i = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{bmatrix}, \qquad H_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z \end{bmatrix}$$

where E is the total energy which has the form:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2) \tag{6.2}$$

The viscous stress terms τ_{ij} are written as

$$\begin{cases} \tau_{xx} = 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}) \\ \tau_{yy} = 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}) \\ \tau_{zz} = 2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}) \end{cases} , \quad \begin{cases} \tau_{xy} = \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) = \tau_{yx} \\ \tau_{yz} = \mu(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}) = \tau_{zy} \\ \tau_{zx} = \mu(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}) = \tau_{xz} \end{cases}$$

where q_x, q_y and q_z are the heat transfer ratios in x, y and z directions respectively, given by

$$\begin{cases} q_x = \frac{\mu}{(\gamma-1)M^2\,\mathrm{Pr}}\frac{\partial T}{\partial x} \\ q_y = \frac{\mu}{(\gamma-1)M^2\,\mathrm{Pr}}\frac{\partial T}{\partial y} \\ q_z = \frac{\mu}{(\gamma-1)M^2\,\mathrm{Pr}}\frac{\partial T}{\partial z} \end{cases}$$

Pr is the Prandtl number, which is chosen as $Pr = 0.72$. γ is the specific heats ratio, which is chosen as $\gamma = 1.4$.

In the above relations, ρ is the density, u, v and w are the velocity components in x, y and z directions respectively, p is the pressure, μ is the coefficient of viscosity and T is the temperature.

The primitive variables are nondimensionalized by their reference values,

$$\begin{array}{llll} x = \frac{\bar{x}}{L_{ref}}, & y = \frac{\bar{y}}{L_{ref}}, & z = \frac{\bar{z}}{L_{ref}}, & t = \frac{V_{ref}\bar{t}}{L_{ref}} \\ u = \frac{\bar{u}}{V_{ref}}, & v = \frac{\bar{v}}{V_{ref}}, & w = \frac{\bar{w}}{V_{ref}}, & \rho = \frac{\bar{\rho}}{\rho_{ref}} \\ p = \frac{\bar{p}}{\rho_{ref}V_{ref}^2}, & T = \frac{\bar{T}}{T_{ref}}, & \mu = \frac{\bar{\mu}}{\mu_{ref}} & \\ \mathrm{Re} = \frac{\rho_{ref}V_{ref}L_{ref}}{\mu_{ref}}, & M = \frac{V_{ref}}{\sqrt{\gamma RT_{ref}}} & & \end{array}$$

The equation of state is

$$p = \frac{\rho T}{\gamma M^2} \tag{6.3}$$

The Sutherland's formula is

$$\mu = T^{\frac{3}{2}}\frac{1 + C}{T + C}, \qquad C = \frac{110.4K}{T_{ref}} \tag{6.4}$$

6.2 Numerical experiments

6.2.1 Two-dimensional linearized Euler equations

The two-dimensional linearized Euler equation with a uniform mean flow in the generalized coordinates and in conservative form is

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

where

$$U = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}, \quad E = \begin{pmatrix} M_x\rho + u \\ M_x u + p \\ M_x v \\ M_x p + u \end{pmatrix}, \quad F = \begin{pmatrix} M_y\rho + v \\ M_y u \\ M_y v + p \\ M_y p + v \end{pmatrix}$$

Here, ρ, u, v and p are the density and two components of velocity and pressure respectively. M_x and M_y are the Mach numbers of the mean flow in the x and y directions. We test a benchmark of Computational Aeroacoustics (CAA) [10] in which $M_x = 0.5$ and $M_y = 0$. The initial value of the physical parameters is the same as that in [10]:

$$p = \exp\left[-(\ln 2)\left(\frac{x^2+y^2}{9}\right)\right]$$

$$\rho = \exp\left[-(\ln 2)\left(\frac{x^2+y^2}{9}\right)\right] + 0.1\exp\left[-(\ln 2)\left(\frac{(x-67)^2+y^2}{25}\right)\right]$$

$$u = 0.04y\exp\left[-(\ln 2)\left(\frac{(x-67)^2+y^2}{25}\right)\right]$$

$$v = -0.04\,(x-67)\exp\left[-(\ln 2)\left(\frac{(x-67)^2+y^2}{25}\right)\right]$$

The computational domain is $[-100, 100] \times [-100, 100]$ with sponge zones applied for the $[-200, 200] \times [-200, 200]$ around the boundary, which is shown in Figure 6.1. Sponge zones are used to absorb and minimize reflections from the computational boundaries. In the sponge zones, the flow satisfies the following equation:

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = S = -\sigma^3(U - U_{ref})$$

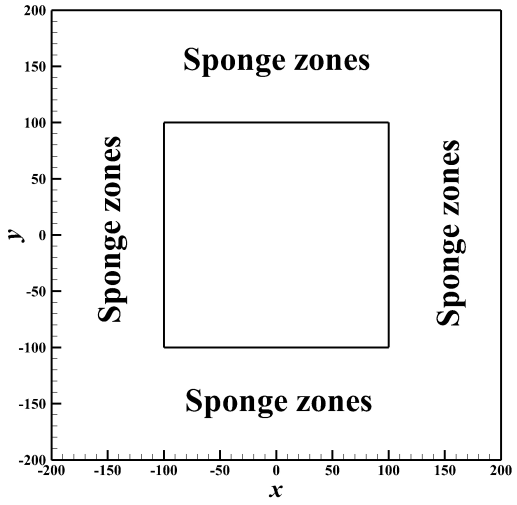


Figure 6.1: The computational domain with the sponge zones.

where

$$U_{ref} = 0$$

$$\sigma(x,y) = \begin{cases} 0\ ,\ when\ \ |x| \le 100\ \ and\ \ |y| \le 100 \\ \left|\frac{x-x_{in}}{x_{in}-x_{out}}\right|\ ,\ when\ \ |x| > 100\ \ or\ \ |y| > 100\ ,\ \ if\ \ |x-x_{out}| < |y-y_{out}| \\ \left|\frac{y-y_{in}}{y_{in}-y_{out}}\right|,\ when\ \ |x| > 100\ \ or\ \ |y| > 100\ ,\ \ if\ \ |y-y_{out}| < |x-x_{out}| \end{cases}$$

Here, (x_{in}, y_{in}) and (x_{out}, y_{out}) denote the inner boundary and outer boundary points which are closest to the point (x, y). We take a 400×400 equally spaced mesh and perform the simulation until $t = 600$.

Figures 6.2 contains the time evolution of density contours. No reflection is observed at the boundaries. Figures 6.3 contains the distributions of density along $y = 0$ at typical times and their comparison with the exact solution. No noticeable difference is observed between the numerical results and the exact solution.

6.2.2 Two-dimensional Euler equations

Our second numerical example is a two-dimensional advection of an isentropic vortex. The initial value of the physical parameters is the same as [26], which is given as follows:

velocity:

$$u = 1 + \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (5 - y)$$

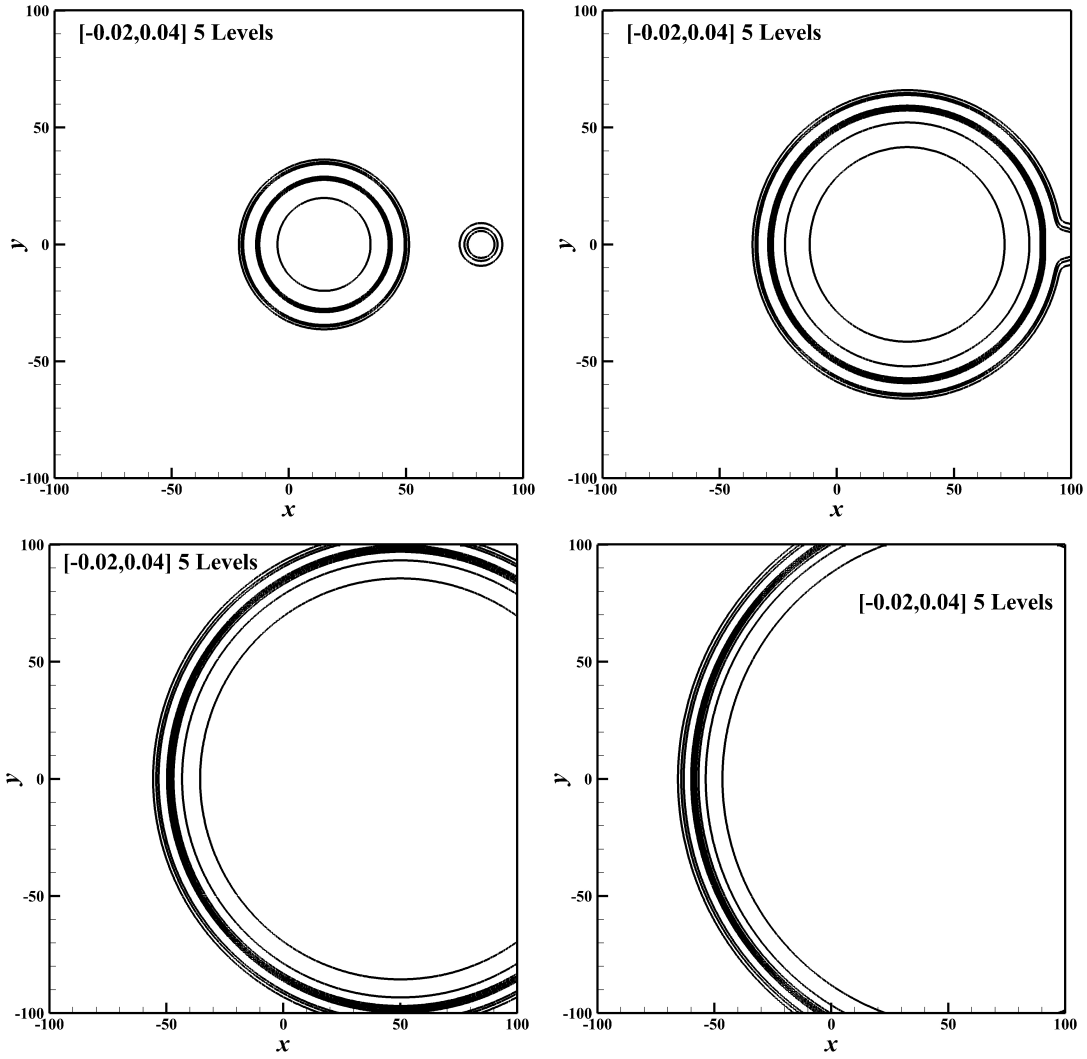


Figure 6.2: Contours of density at typical times of the benchmark of CAA. Top left: $t = 30$; Top right: $t = 60$; Bottom left: $t = 100$; Bottom right: $t = 120$.

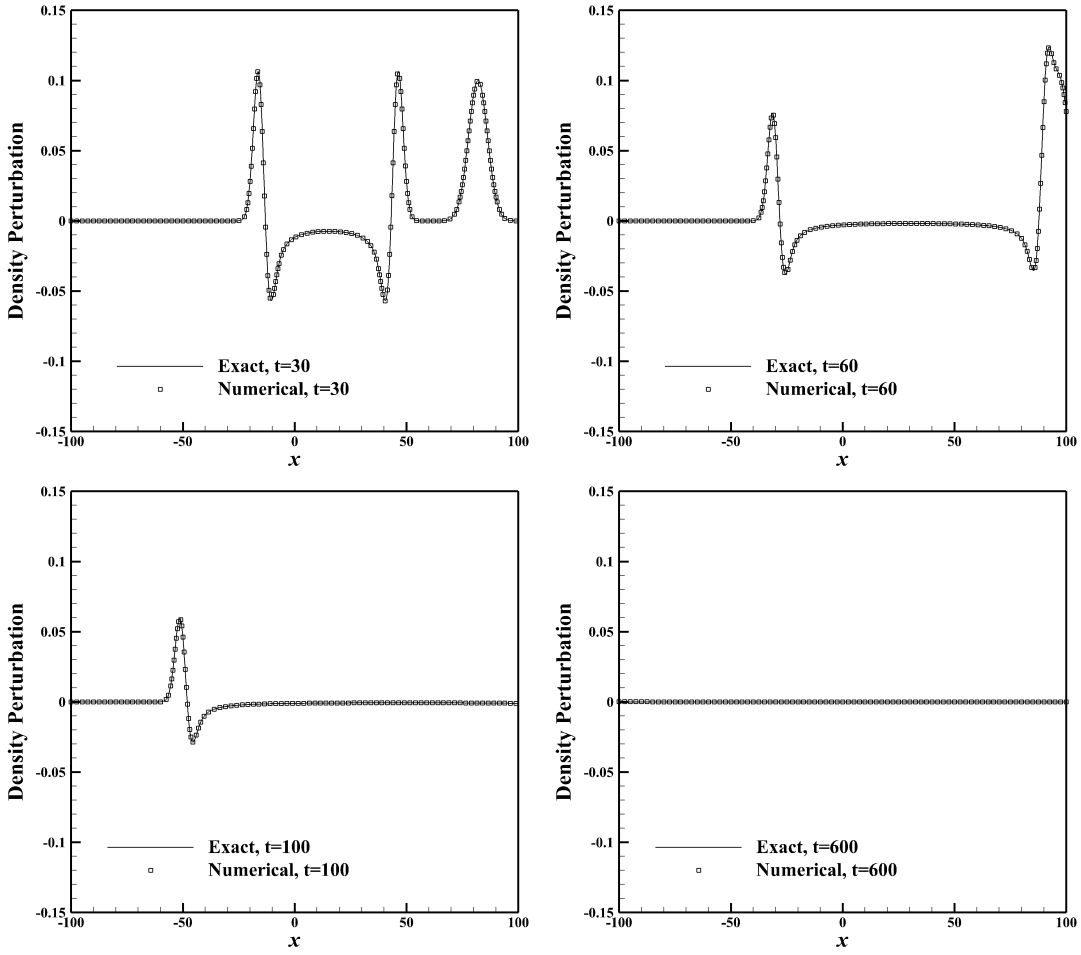


Figure 6.3: The distribution of density along $y = 0$ and the comparison with the exact solution at typical times of the benchmark of CAA. Top left: $t = 30$; Top right: $t = 60$; Bottom left: $t = 100$; Bottom right: $t = 120$.

$$v = 1 + \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (x - 5)$$

temperature:

$$T = 1 - \frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}$$

entropy:

$$S = 1$$

The relationships of the physical quantities are:

$$T = \frac{p}{\rho}, \qquad S = \frac{p}{\rho^\gamma}$$

where $r^2 = (x-5)^2 + (y-5)^2$. The initial vortex center is at the point of $(5, 5)$. ε is the strength of the vortex, which is taken as $\varepsilon = 5$. The computational domain is taken as $[0, 10] \times [0, 10]$. Periodical boundary conditions are used in all boundaries. We take a 80×80 equally spaced mesh and perform the simulation until $t = 200$.

Figures 6.4 contains the distributions of density along $x = 5$ at typical times $t = 50$ and 200. We can observe that the numerical results by CCS agree very well with the exact solution for all times, which means that the accuracy and resolution are high, and the dissipation is low.

6.2.3 Two-dimensional Navier–Stokes equations

Two Gaussian vortices are initially separated by a distance of $2R$ and the swirling flow associated with each vortex (when considered separately) achieves a maximum Mach number $M_0 = \frac{U_0}{c_\infty} = 0.56$, at a radius r_0 from the center of each vortex core. Figure 6.5 is a sketch of the flow with definitions of the relevant parameters.

The initial value of the physical parameters is the same as that in [6, 22]:

circulation:

$$\Gamma_0 = \frac{2\pi U_0 r_0}{\beta}$$

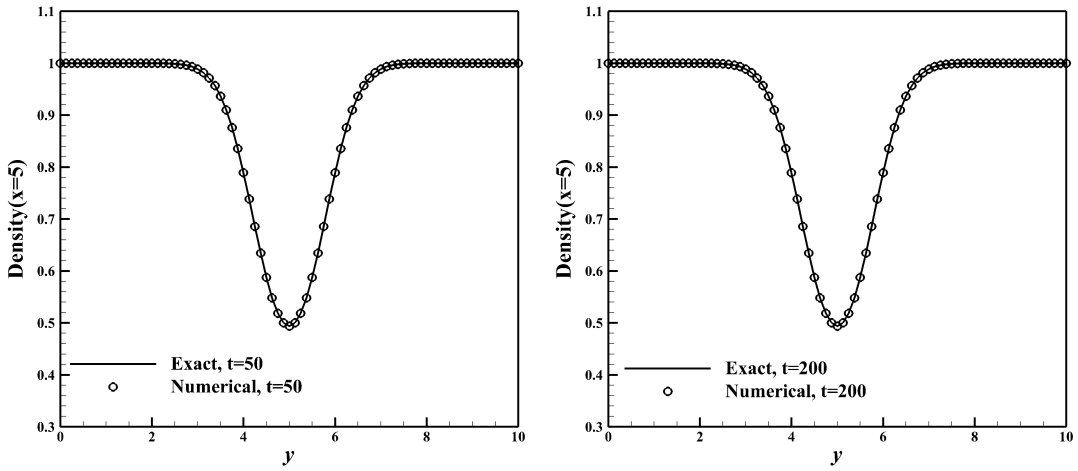


Figure 6.4: The distribution of density along $x = 5$ for the advection of two-dimensional isentropic vortex at typical times $t = 50$ (Left) and $t = 200$ (right).

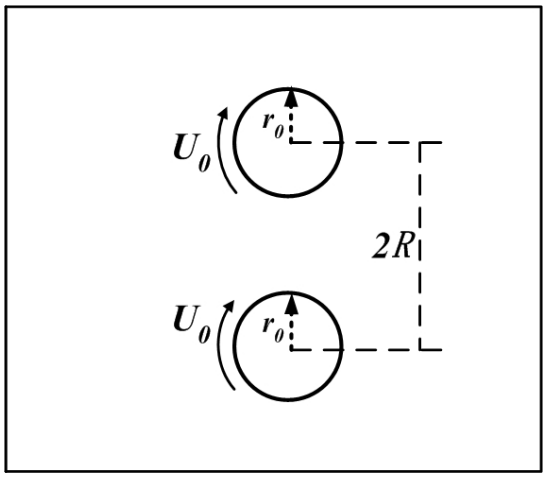


Figure 6.5: Schematic diagram of two Gaussian vortices which will merge.

vorticity:

$$\omega = \frac{\alpha\Gamma_0}{\pi {r_0}^2} e^{-\alpha(\frac{r}{r_0})^2}$$

velocity:

$$V = \frac{\Gamma_0}{2\pi r}(1 - e^{-\alpha(\frac{r}{r_0})^2})$$

where r_0 is the reference length scale, ρ_∞ is the reference density, c_∞ is the reference velocity, $\rho_\infty {c_\infty}^2$ is the reference pressure, $\alpha = 1.256431$, $\beta = 0.7153318035699323$, $\mathrm{Re} = \frac{\Gamma_0}{\nu} = 7500$, $\frac{r_0}{R} = 0.15$.

For a homentropic solenoidal flow, the initial pressure and density are obtained by solving the following Poisson equation:

$$\frac{\partial^2 (p/\rho)}{\partial x^2} + \frac{\partial^2 (p/\rho)}{\partial y^2} = -\frac{\gamma - 1}{\gamma}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right]$$

and the isentropic condition $\frac{p}{\rho^\gamma} = constant$.

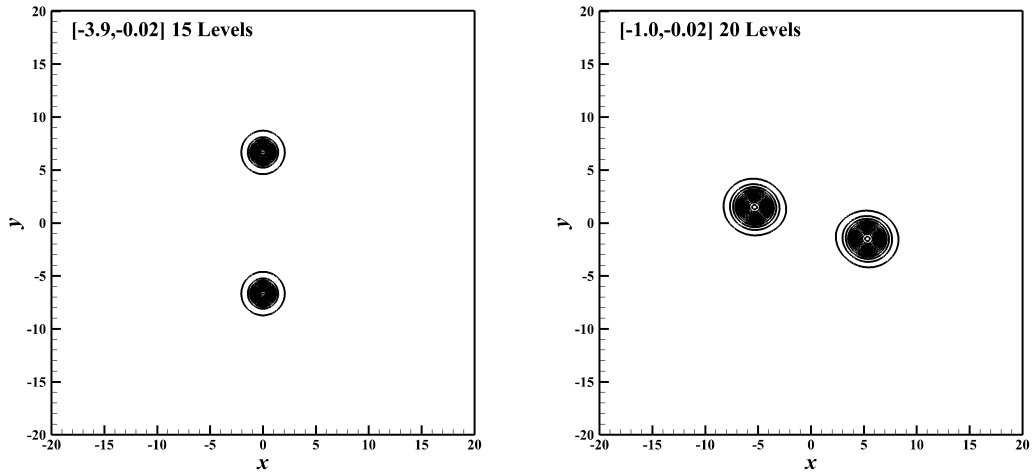
In the far field, $\rho = 1$, $p = \frac{1}{\gamma}$, $u = v = 0$.

The computational domain is $[-20, 20] \times [-20, 20]$. We take a 200×200 equally spaced mesh and perform the simulation until $t = 3000$. Figure 6.6 contains the time evolution of the vorticity. As can be seen from Figure 6.6, the coupling between the two Gaussian vortices makes the two vortices rotating around each other for a very long time. During this period, the distance between the two vortices is almost constant. Before the merging, the coupling effect compresses the vortices to elliptical shapes and makes them move closer to each other. Suddenly, the merging happens, which results in a singular circular vortex.

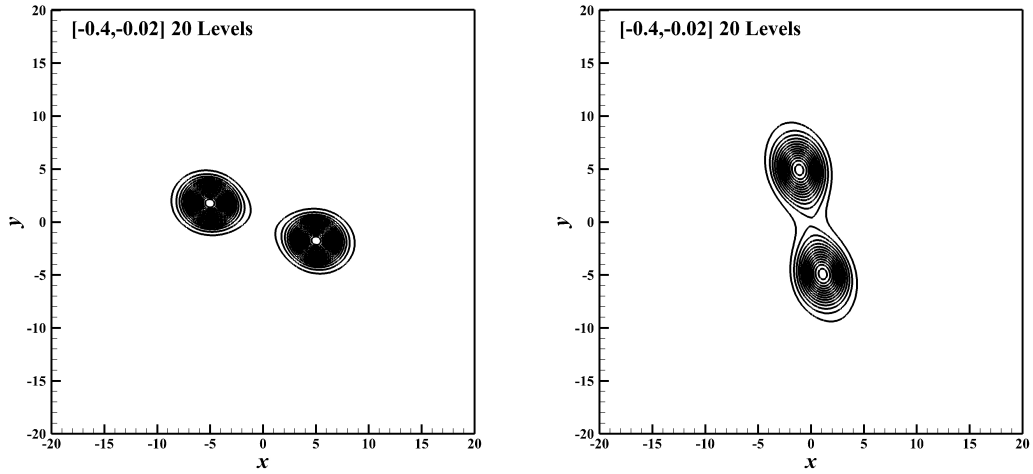
For the above numerical tests, we tested two numerical methods, CCS and CCS-CI. For the given numerical grids, we can not distinguish the numerical results by different methods. Hence, we have just presented the numerical results of CCS.

6.2.4 Three-dimensional Navier–Stokes equations

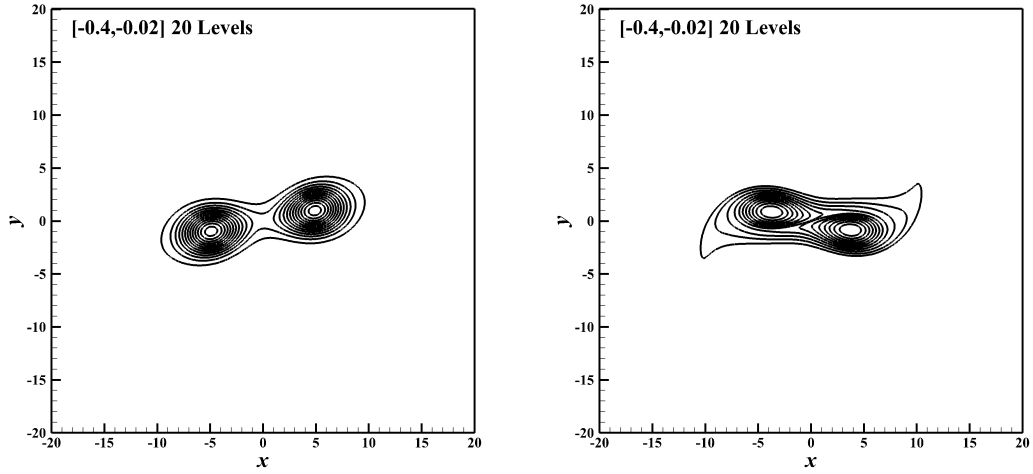
Turbulence has been a great challenge for computational fluid dynamics. In this section, we directly simulate three-dimensional decaying isotropic turbulence [25, 21, 33] to test the efficiency of CCS.



(a) $t = 0$ (b) $t = 500$

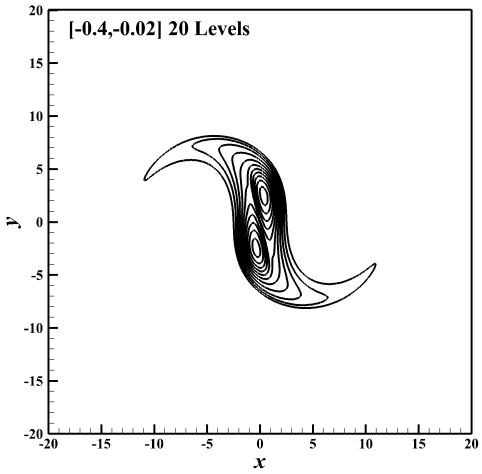


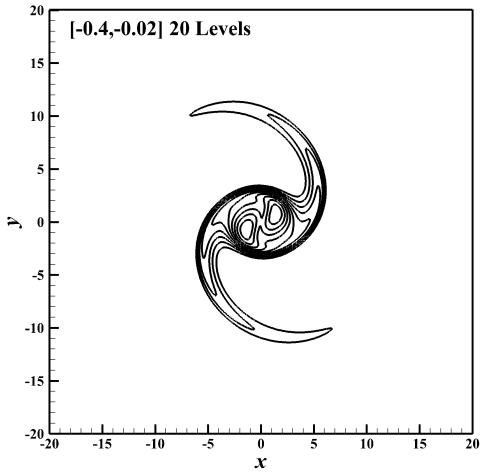
(c)$t = 1000$ (d) $t = 1500$



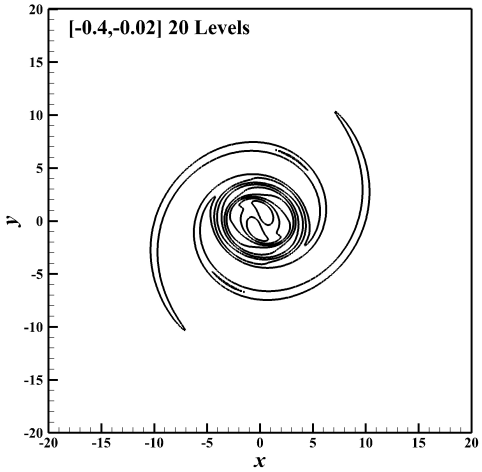
(e) $t = 1600$ (f) $t = 1800$

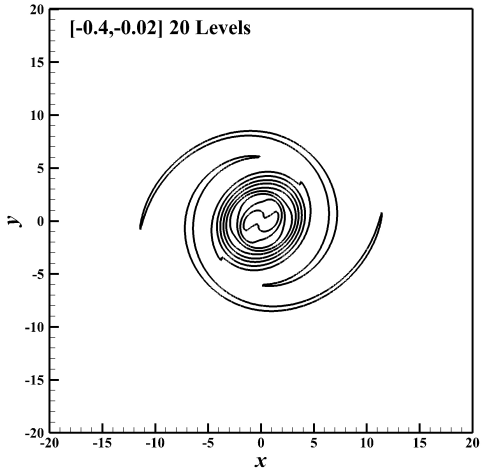
Figure 6.6: The evolution of the vorticity field in the vortex merging of two co-rotating Gaussian vortices.



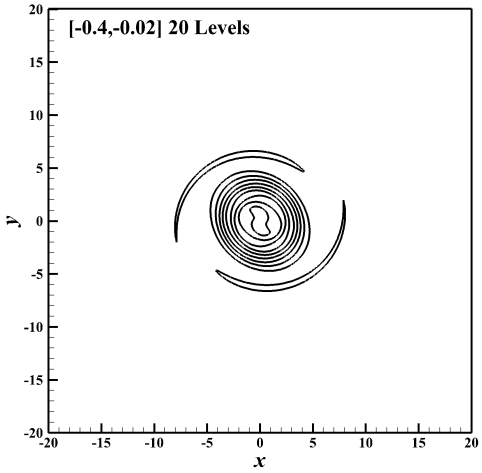


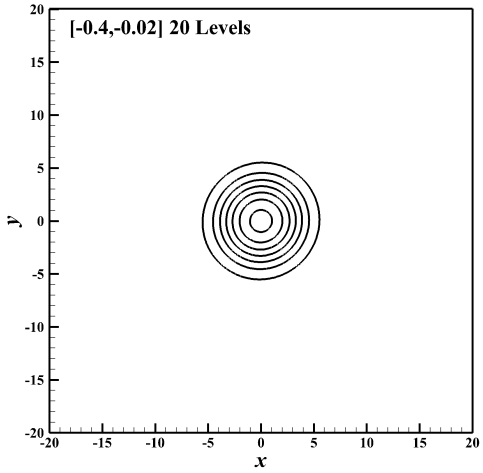
(g) $t = 1850$ (h) $t = 1900$





(i)$t = 2000$ (l) $t = 2050$





(m) $t = 2100$ (n) $t = 3000$

Figure 6.6: Continued.

We start with a specified spectrum for the initial velocity field which is divergence free, given by [25, 21, 33]:

$$E(k) = Ak^4 \exp\left(-2k^2/k_p^2\right). \tag{6.5}$$

The normalized temperature and density are simply initialized to one at all spatial points. The dimensionless parameters are $Re = 519$ and $M = 0.308$, yielding the initial turbulent Mach number M_t to be 0.3 and the initial Taylor microscale Reynolds number Re_λ to be 72. For decaying compressible turbulence, both M_t and Re_λ decrease as time evolves. Therefore, there is essentially no shock wave in this flow.

The computational domain is $[0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$. Periodic boundary conditions are used at all boundaries. The grid convergence is studied and compared with CNCS, CCCS and CCCS-CI.

Figure 6.7 contains the three-dimensional isosurfaces of vorticity $|\omega| = 20$ colored with pressure obtained by CCS with the grid density of $40 \times 40 \times 40$. The turbulent structure is clear. Figure 6.8 shows the grid convergence of the temporal evolution of the turbulent kinetic energy obtained by CCS, CCS-CI, CCCS, CCCS-CI and CNCS. The grid converged results agree very well with the numerical result of Samtaney et al. [25], which is obtained by the tenth order CNCS with a grid density of $128 \times 128 \times 128$. From this figure, we find that CCS needs $40 \times 40 \times 40$ grid density to reach grid converged solution, while the smallest grid density to obtain the grid converged results are $80 \times 80 \times 80$, $64 \times 64 \times 64$, $80 \times 80 \times 80$ and $80 \times 80 \times 80$ for CCS-CI, CCCS, CCCS-CI and CNCS respectively. Again, we find that the resolution of CCS is much better than those of CNCS and CCCS. It is an ideal numerical scheme for direct numerical simulation of turbulence. In addition, we test the grid convergence of sixth order explicit scheme CCS-E6, it needs $64 \times 64 \times 64$ grid density to reach grid converged solution, which will be a good choice for parallel computation.

In Table 6.1 and Table 6.2, the CPU time per time step with the grid density $64 \times 64 \times 64$ and the total CPU time for the converged grid of each method are listed respectively. We can observe that the CPU time per step of our current scheme is approximately eight times

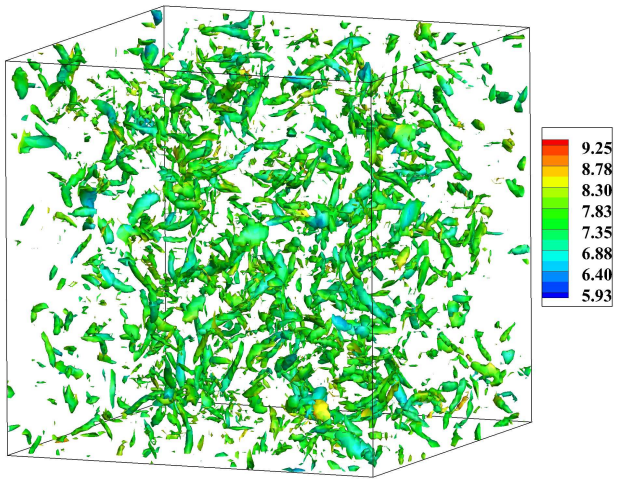


Figure 6.7: The isosurfaces of vorticity $|\omega| = 20$ colored with pressure of three-dimensional decaying isotropic turbulence.

Table 6.1: CPU time per step for 3D decaying isotropic turbulence with grid density $64 \times 64 \times 64$.

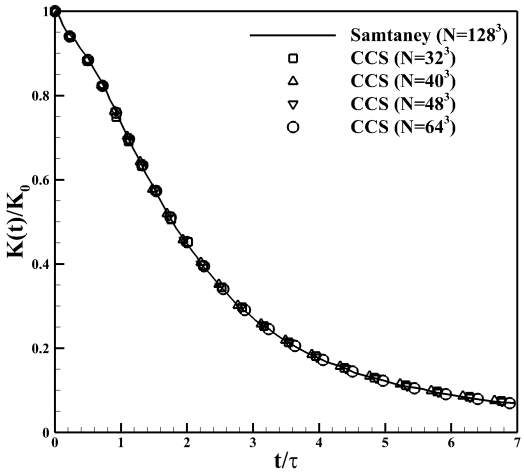
Schemes	CCS	CCS-E6	CNCS	CCCS-CI	CCS-CI	CCCS
CPU time(seconds) per step	21.6654	10.2046	2.8962	4.5003	4.5323	21.7254

more than that of Lele's cell node compact scheme. It approximately equal to that of Lele's cell center compact scheme. However, the total CPU time for the converged grid of our scheme is the smallest among all these schemes.

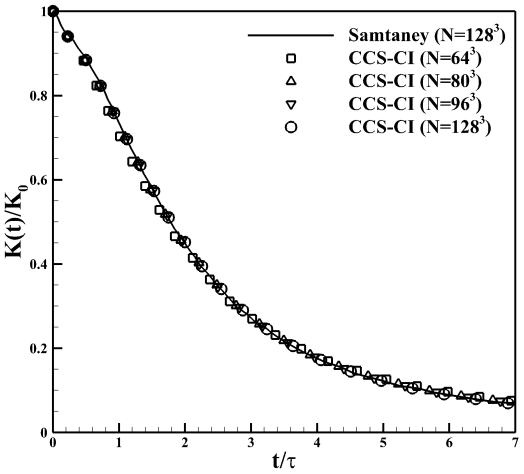
7 Concluding remarks

In this paper, we design a new family of linear compact schemes, named central compact schemes, for the spatial derivatives in the Navier–Stokes equations based on the cell-centered compact scheme proposed by Lele [15].

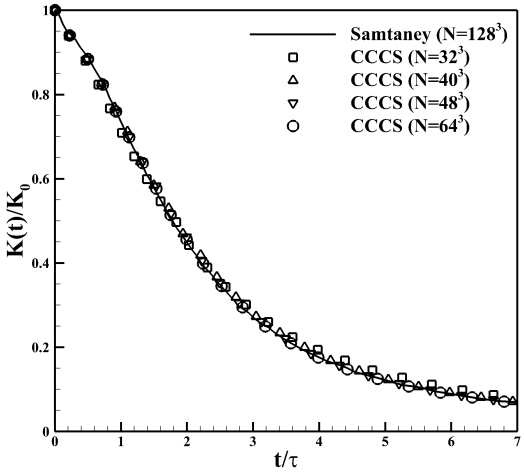
Compared to other linear compact schemes, cell-centered compact schemes have nice spectral-like resolution, and they are good methods for the computation of multi-scale problems. However, previous cell-centered compact schemes have two drawbacks. First, not all



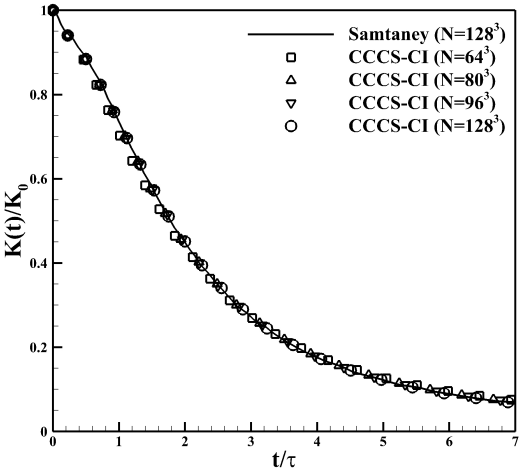
(a) CCS: converged grid at $40 \times 40 \times 40$



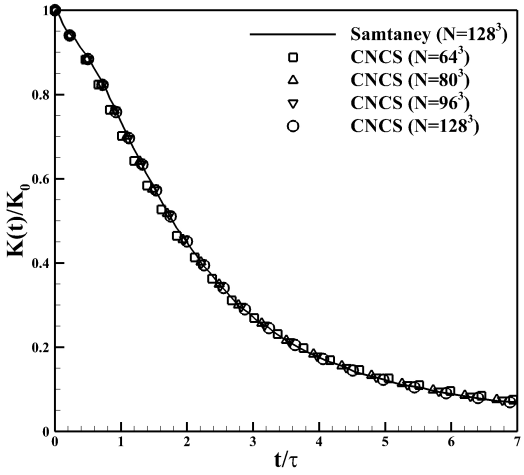
(b) CCS-CI: converged grid at $80 \times 80 \times 80$



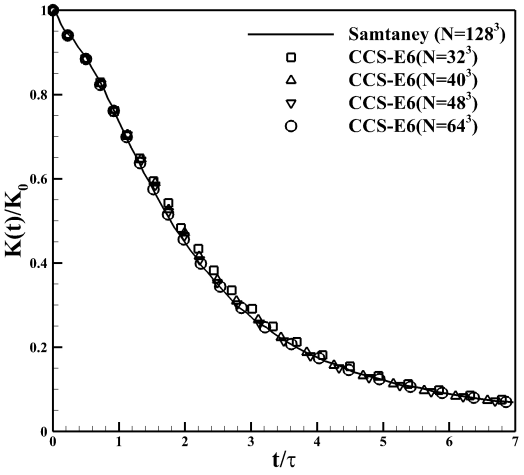
(c) CCCS: converged grid at $64 \times 64 \times 64 \times 64$



(d) CCCS-CI: converged grid at $80 \times 80 \times 80$



(e) CNCS: converged grid at $80 \times 80 \times 80$



(f)CCS-E6:converged grid at $64 \times 64 \times 64 \times 64$

Figure 6.8: Grid convergence of the temporal evolution of the turbulent kinetic energy with different schemes. The explicit schemes is sixth order. The implicit compact scheme is eighth order. The interpolation is tenth order.

Table 6.2: Total CPU time for the converged grid for 3D decaying isotropic turbulence.

Schemes	Converged grid	CPU time (hours)
CCS	40^3	5.04789
CCS-E6	64^3	8.20825
CNCS	80^3	5.64777
CCCS-CI	80^3	9.05911
CCS-CI	80^3	9.18892
CCCS	64^3	21.3563

physical values on the stencil are used, which results in the numerical scheme not reaching its maximum accuracy order. Second, the physical values on the cell-center points are computed by a compact interpolation. The use of an interpolation will introduce transfer error, which results in significant loss of resolution for high wavenumbers.

The central compact scheme designed in this paper overcomes the drawbacks mentioned above. First, all physical values on the stencil are used. The schemes could reach the maximum accuracy order. Second, the physical values on the cell-centers are stored as independent variables and computed by the same scheme as that for the grid point. This approach increases the memory requirement but not the computational costs. The accuracy of the scheme is improved and the resolution is preserved.

Numerous tests including a benchmark of computational aeroacoustics, two-dimensional isentropic vortex, vortex merging and three dimensional decaying turbulence are implemented by solving two-dimensional or three-dimensional Euler or Navier-Stokes equations. A systematic comparison with previous compact schemes, including cell node compact schemes and the cell-centered compact scheme, is made. The comparison shows the superiority of the central compact scheme over the previous compact schemes in accuracy and resolution. It appears to be an ideal numerical method for the computation of multi-scale problems such as turbulence and aeroacoustics.

In future work, we will study boundary closures in more detail for stability and accuracy. We will also explore WENO techniques to improve the robustness of the scheme for shock

waves without sacrificing its accuracy and resolution for smooth structures significantly.

References

[1] J. Berland, C. Bogey, O. Marsden, C. Bailly (2007) High-order, low dispersive and low dissipative explicit schemes for multiple-scale and boundary problems. Journal of Computational Physics 224, 637-662.

[2] B.J. Boersma (2005) A staggered compact finite difference formulation for the compressible Navier–Stokes equations. Journal of Computational Physics 208, 675-690.

[3] C. Bogey, C. Bailly (2004) A family of low dispersive and low dissipative explicit schemes for flow and noise computations. Journal of Computational Physics 194, 194-214.

[4] B. Cockburn, C.-W. Shu (1994) Nonlinearly stable compact schemes for shock calculations. SIAM Journal on Numerical Analysis 31, 607-627.

[5] T. Colonius, S.K. Lele (2004) Computational aeroacoustics: progress on nonlinear problems of sound generation. Progress in Aerospace Sciences 40, 345-416.

[6] T. Colonius, S.K. Lele, P. Moin (1994) The scattering of sound waves by a vortex: numerical simulations and analytical solutions. Journal of Fluid Mechanics 260, 271-298.

[7] G.S. Constantinescu, S.K. Lele (2001) Large eddy simulation of a near sonic turbulent jet and its radiated noise. AIAA Paper 2001-0376.

[8] D. Fu, Y. Ma, H. Liu (1993) Upwind compact schemes and applications, Proceedings 5th Symp. on Comput. Fluid Dyn., Vol.1, Japan Soc. of Comput. Fluid Dyn.

[9] S. Gottlieb, C.-W. Shu (1998) Total variation diminishing Runge-Kutta schemes. Mathematics of Computation 67, 73-85.

[10] J.C. Hardin, J.R. Ristorcelli, C.K.W. Tam, ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics (CAA). Hampton, Virginia, October 1995, NASA Conference Publication 3300.

[11] F.Q. Hu, M.Y. Hussaini, J.L. Manthey (1996) Low-dissipation and low-dispersion Runge–Kutta schemes for computational acoustics. Journal of Computational Physics 124, 177-191.

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

[13] J.W. Kim (2007) Optimised boundary compact finite difference schemes for computational aeroacoustics. Journal of Computational Physics 225, 995-1019.

[14] S. Lee, S.K. Lele, P. Moin (1997) Interaction of isotropic turbulence with shock waves: effect of shock strength. Journal of Fluid Mechanics 340, 225-247.

[15] S.K. Lele (1992) Compact finite difference schemes with spectral-like resolution. Journal of Computational Physics 103, 16-42.

[16] Z. Liu, Q. Huang, Z. Zhao, J. Yuan (2008) Optimized compact finite difference schemes with high accuracy and maximum resolution. International Journal of Aeroacoustics 7, 123-146.

[17] X.-D. Liu, S. Osher, T. Chan (1994) Weighted essentially non-oscillatory schemes. Journal of Computational Physics 115, 200-212.

[18] C. Lui, S. K. Lele (2001) Direct numerical simulation of spatially developing, compressible turbulent mixing layers, AIAA Paper 2001-0291.

[19] K. Mahesh (1998) A family of high order finite difference schemes with good spectral resolution. Journal of Computational Physics 145, 332-358.

[20] K. Mahesh, S.K. Lele, P. Moin (1997) The influence of entropy fluctuations on the interaction of turbulence with a shock wave. Journal of Fluid Mechanics 334, 353-379.

[21] M.P. Martin, E.M. Taylor, M. Wu, V.G. Weirs (2006) A bandwidth-optimized WENO scheme for the direct numerical simulation of compressible turbulence. Journal of Computational Physics 220, 270-289.

[22] B.E. Mitchell, S.K. Lele, P. Moin (1995) Direct computation of the sound from a compressible co-rotating vortex pair. Journal of Fluid Mechanics 285, 181-202.

[23] P. Moin, K. Squires, W. Cabot, S. Lee (1991) A dynamic subgridscale model for compressible turbulence and scalar transport. Physics of Fluids A 3, 2746.

[24] S. Nagarajan, S.K. Lele, J.H. Ferziger (2003) A robust high-order compact method for large eddy simulation. Journal of Computational Physics 191, 392-419.

[25] R. Samtaney, D.I. Pullin, B. Kosovic (2001) Direct numerical simulation of decaying compressible turbulence and shocklet statistics. Physics of Fluids 5, 1415-1430.

[26] C.-W. Shu (1998) Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, pp.325-432.

[27] C.-W. Shu, S. Osher (1988) Efficient implementation of essentially non-oscillatory shock capturing schemes. Journal of Computational Physics 77, 439-471.

[28] C.-W. Shu, S. Osher (1989) Efficient implementation of essentially non-oscillatory shock-capturing schemes II. Journal of Computational Physics 83, 32-78.

[29] R.K. Shukla, X. Zhong (2005) Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation. Journal of Computational Physics 204, 404-429.

[30] C.K.W. Tam (1995) Computational aeroacoustics: Issues and methods. AIAA Journal 33, 1788-1796.

[31] C.K.W. Tam, J.C. Webb (1993) Dispersion-relation-preserving finite difference schemes for computational acoustics. Journal of Computational Physics 107, 262-281.

[32] M.R. Visbal, D.V. Gaitonde (2002) On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. Journal of Computational Physics 181, 155-185.

[33] J. Wang, L.P. Wang, Z. Xiao, Y. Shi, S. Chen (2010) A hybrid numerical simulation of isotropic compressible turbulence. Journal of Computational Physics 229, 5257-5279.

[34] S. Zhang, S. Jiang, C.-W. Shu (2008) Development of nonlinear weighted compact schemes with increasingly higher order accuracy. Journal of Computational Physics 227, 7294-7321.

[35] X. Zhong (1998) High-Order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition. Journal of Computational Physics 144, 662-709.

[36] M. Zhuang, R.F. Chen (1998) Optimized upwind dispersion-relation-preserving finite difference scheme for computational aeroacoustics. AIAA Journal 36, 2146-2148.